THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 January 1983 Number 1

Copyright © 1983 American Telephone and Telegraph Company. Printed in U.S.A.

# InGaAsP LEDs for 1.3-μm Optical Transmission

By H. TEMKIN, C. L. ZIPFEL, M. A. DIGIUSEPPE, A. K. CHIN, V. G. KERAMIDAS, and R. H. SAUL

(Manuscript received May 26, 1982)

*High-radiance InGaAsP double heterostructure light-emitting diodes (LEDs) that are fast and reliable have been developed for 1.3-μm optical transmission applications. These devices are prepared by liquid phase epitaxy on InP substrates. Improved growth procedures result in large-area wafers (up to 5.4 $cm^2$) with excellent uniformity and high device yields. Optical power of ~120 μW can be launched into 0.29-NA, 62.5-μm core fiber. On the basis of accelerated reliability tests the median life of these InGaAsP LEDs is expected to exceed 4 × $10^6$ h at 70°C and current densities up to 40 kA/$cm^2$. The projected failure rate owing to output power degradation (−1 dB) is below 1 FIT. These devices are expected to be useful in lightwave systems operating at data rates ≤140 Mb/s.*

## I. INTRODUCTION

The present generation of optical transmission systems employs GaAlAs laser emitters operating at 0.8 to 0.9 μm.[1] In this wavelength range fiber attenuation (about 4 dB/km) and especially chromatic dispersion limit the usefulness of light-emitting diodes (LEDs) to short data links of few kilometers. By shifting the LED emission wavelength close to 1.3 μm two properties of the fused silica fiber permit much longer distances and higher data rates. First, the fiber loss is greatly reduced at longer wavelength,[2] with cabled fiber loss below 1 dB/km possible. Second, pulse spreading owing to material dispersion is reduced as $d^2n/d\lambda^2$ of the refractive index approaches zero at 1.27 to 1.30 μm.[3] Thus, 1.3-μm LED-based transmission systems with high

data rates and repeater distances in excess of 10 km are feasible.[4] Indeed, transmission experiments have demonstrated repeaterless LED-based operation over a distance of 24 km at the DS3 rate (44.7 Mb/s) and 7 km at rates as high as DS4 (274 Mb/s).[5] Such LED systems are attractive alternatives to laser systems since light-emitting diodes operate over a wide temperature range with unequalled reliability and their drive electronics is very simple.

These advantages of longer wavelength operation can be realized for devices fabricated out of the quaternary alloy $In_{1-x}Ga_xAs_yP_{1-y}$.[6] These alloys can be grown exactly lattice-matched to InP substrates (for $y/x \approx 2.2$) with compositions ranging from 1.1 $\mu$m to 1.65 $\mu$m.[7] These quaternary LEDs are efficient, fast, and very reliable.[8–11] In addition, very sensitive InGaAsP detectors have been demonstrated,[12] offering a possibility of a common material technology for both emitting and receiving devices.

In this paper we describe the development of state of the art InGaAsP LEDs for optical transmission applications. The paper is divided into six sections. Following the introduction, Section II describes material preparation procedures leading to double-heterostructure wafers with a large area and high quality. Device fabrication is discussed in Section III, with particular attention devoted to large-scale fabrication techniques. In these two sections a description of general procedures is followed by more detailed descriptions of some of the more challenging problems. LED performance in terms of power launched into fibers, speed, and temperature characteristics is discussed in Section IV. Section V describes accelerated aging tests and reliability estimates. These results are discussed and summarized in Section VI.

## II. MATERIAL PREPARATION

Double heterostructure wafers, from 1 $cm^2$ to 5.4 $cm^2$ in area, are grown in a multiple-well, graphite boat using a two-phase growth technique.[7] InP substrates are polished to an optically smooth finish with a ~1-percent bromine-methanol solution. All the melts used are first equilibrated at 675°C for 16 hours in a $H_2$ ambient and then quenched. These prehomogenized melts, containing excess InP and Sn dopant where applicable, are then placed in the liquid phase epitaxy (LPE) boat and again equilibrated at 675°C for an additional hour, as schematically shown in Fig. 1. Zinc is added prior to this final equilibration step. During this warm-up and melt homogenization cycle the thermal decomposition of the substrate is minimized by either a cover piece or phosphorus vapor.[13]

Prior to the growth of the first epitaxial InP layer at 650°C, the substrate is subjected to a shallow In etch, which removes about 5 $\mu$m

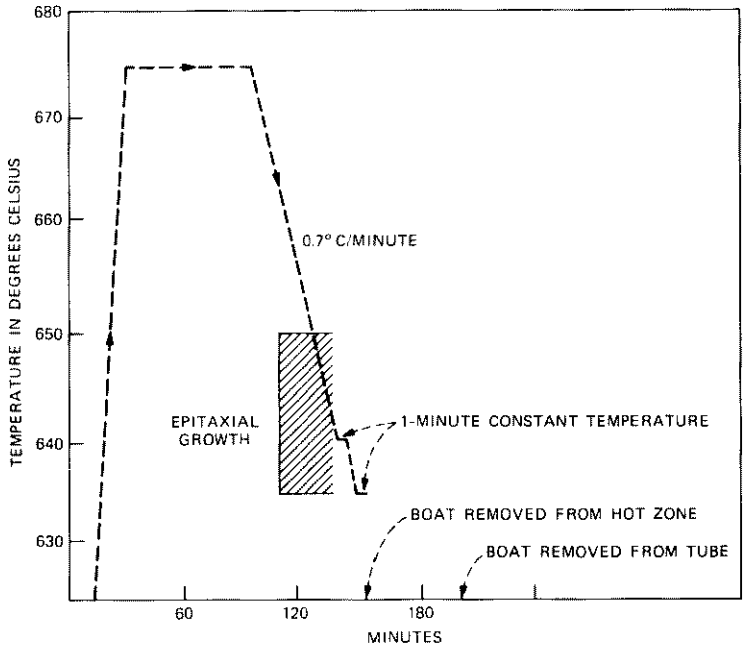



Fig. 1—Schematic diagram of melt homogenization and LPE growth cycle.

of the thermally degraded material. Following the growth of the Sn-doped InP buffer layer at a cooling rate of 0.7°C/min, a nominally undoped quaternary layer is grown isothermally at 640°C. Next, a Zn-doped InP confining layer is grown (again at 0.7°C/min), followed by a Zn-doped quaternary, or the InGaAs ternary, contact layer grown isothermally at 630°C. The InP buffer and confining layers are approximately 5 and 2 $\mu$m thick, respectively. The thickness of the InGaAsP active layer is varied between 0.5 and 1.4 $\mu$m, while the contact layer thickness is kept at ~0.5 $\mu$m.

Since the defects present in the substrate or epitaxial layer are reproduced in the next epitaxial layer,[14] substrate quality is of great importance. InP substrates used in this work have been cut in the ⟨100⟩ direction from the ⟨111⟩ pulled, twin-free, boules grown by the Liquid Encapsulated Czochralski (LEC) technique.[15] The substrates are doped $n$-type with either Sn or S up to the doping level of $\sim 5 \times 10^{18}$ $cm^{-3}$. The sulfur-doped material is preferred for reasons of superior macroscopic perfection and higher resistance to thermal damage.[16] The dislocation density in our S-doped wafers, as established by etch pitting and TCL,[17] is typically below $5 \times 10^3$ $cm^{-3}$, as much as an order of magnitude less than in Sn-doped wafers. This reduction has been attributed to lattice hardening.[18] Although the presence of threading

dislocations has been shown to greatly degrade reliability of GaAlAs LEDs,[19] we show in Section V that dislocation density has no significant effect on quaternary reliability. The influence of dislocations on device performance remains to be investigated.

The thermal damage to InP substrate prior to growth and the resulting damage to iso-epitaxial layers are by now well recognized.[20] The damage owing to loss of phosphorus ranges from the large dissociation pits to increased density of phosphorus vacancy-impurity complexes in InP layers,[13] and results in reduced device performance and yield. A number of techniques have been developed to control and eliminate the thermal decomposition, all attempting to provide phosphorus overpressure in the vicinity of the substrate. Commonly used are cover-piece protection,[8] In-Sn-P solution,[21] addition of $PH_3$,[22,23] or elemental phosphorus at elevated temperature to the growth ambient.[13] Deep, ~25-$\mu$m, In meltback also has been used to remove all the degraded material. We have evaluated these protection measures by means of spatially resolved photoluminescence and demonstrated that thermal degradation is present even on morphologically perfect surfaces.[20] A photoluminescence image of a thermally degraded substrate is shown in Fig. 2a. The damage manifests itself as spots of low luminescence efficiency. Iso-epitaxial layers grown on such a substrate show similar damaged regions and an increased dislocation density. This damage depends on the substrate dopant, and is greatly reduced, as indicated by photoluminescence (PL) intensity measurements, in S-doped InP, as compared to Sn-doped wafers. Elimination of thermal damage is achieved through a shallow, ~5-$\mu$m deep, In meltback of a

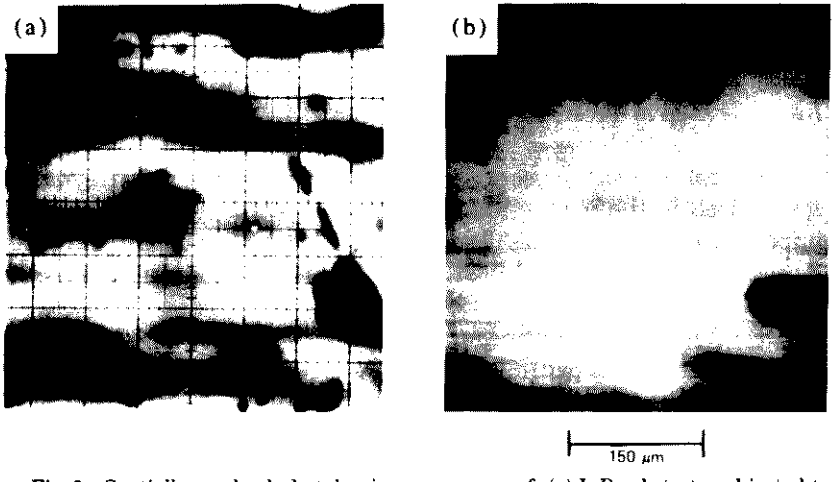



Fig. 2—Spatially resolved photoluminescence map of: (a) InP substrate subjected to a high temperature treatment, and (b) InP buffer layer grown on a well-protected substrate.

previously protected substrate, as shown by the uniform and bright PL scan of the InP buffer layer in Fig. 2b.

The second to grow is the the active layer of $In_{0.7}Ga_{0.3}As_{0.64}P_{0.36}$, a composition corresponding to the 1.3-$\mu$m emission wavelength. Particular attention is paid to the reproducible growth of layers with that composition, precisely lattice-matched to the InP substrate. In particular, a close lattice match, $\Delta a/a$ smaller than $6 \times 10^{-4}$, is required to prevent formation of misfit dislocations.[24] These act as nonradiative recombination centers and thus reduce luminescence efficiency. A transmission cathodoluminescence (TCL) image of misfit dislocations forming a characteristic criss-cross pattern on [100] oriented layer is shown in Fig. 3. This InGaAsP layer had a lattice mismatch of $1 \times 10^{-3}$.

In our experience, epitaxial layer morphology is not affected by lattice mismatch; layers of 1.3-$\mu$m InGaAsP with $\Delta a/a \approx 2.4 \times 10^{-3}$ have been grown with excellent morphology. Thus, it is necessary to monitor lattice match with X-ray diffractometry. The lattice constant can be well-controlled by precise weighing of the Ga and As components (up to $\pm 25$ $\mu$grams in a 10-gm melt) and careful reproduction of the growth temperature. Phosphorus loss in the equilibration cycle is compensated by the presence of excess InP. Accurate temperature control during the active layer growth results in reduced compositional

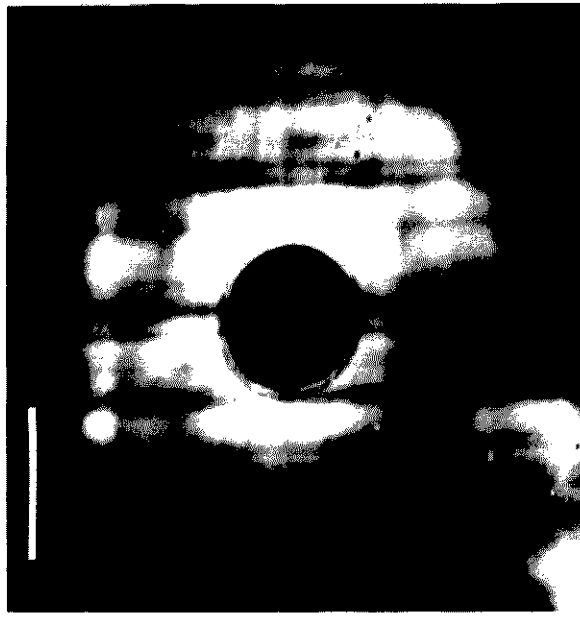


Fig. 3—Transmission cathodoluminescence (TCL) image of ⟨100⟩ misfit dislocations in a wafer with $\Delta a/a = 0.10$ percent.

grading. Compositional grading is especially pronounced for alloy compositions close to 1.3 $\mu$m.[25] The degree of compositional uniformity, obtained on a fairly routine basis, is demonstrated in a double X-ray spectrometer scan shown in Fig. 4. Double X-ray technique is used to fully resolve X-ray reflection width. The stronger peak originates in the InP buffer layer and the substrate, a much weaker and broader one, is due to the quaternary layer. The width of the InGaAsP reflection in this layer, ~60s of arc, is attributed to compositional grading in the direction of growth, i.e., along the thickness of the epitaxial layer. Larger grading results in broadening of the LED emission line width. In comparison, the width of the InP reflection is ~10s of arc, indicating the absence of any broadening because of thermal damage in the substrate.

The most common defects present in the epitaxial layers of InP and InGaAsP are inclusions of In. These originate at either meltback or wipe-off stages of the growth cycle and greatly reduce device yield and

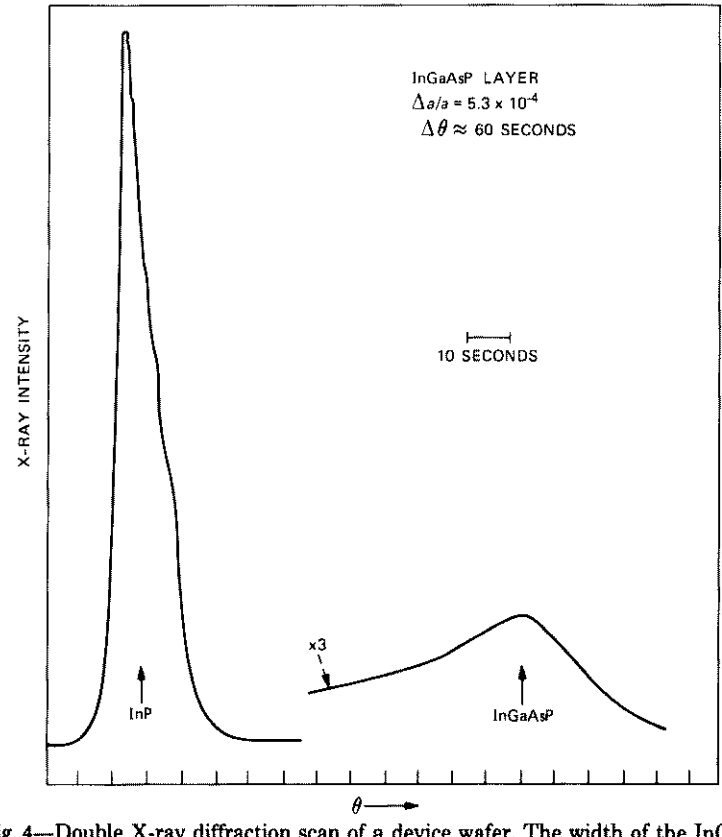



Fig. 4—Double X-ray diffraction scan of a device wafer. The width of the InGaAsP peak reveals a low degree of compositional grading.

reliability. A dramatic example of the wipe-off-related problem is shown in Fig. 5a. The InGaAsP active layer imaged in this Nomarski micrograph has been grown on a buffer layer that was not completely wiped off. The In inclusions left at the buffer surface locally prevent InGaAsP growth and produce holes in the active layer. This problem can be controlled by careful graphite boat design and maintenance of stringent wipe-off tolerances. Similarly, it is important to avoid melt contamination with particulate matter, such as carbon particles generated during prolonged equilibration baking. Figure 5b shows a more typical Nomarski image of the active layer. No defects were present throughout the entire 2.7-$cm^2$ wafer, resulting in excellent LED performance and uniformity. The horizontal lines visible in this image are growth lamellae characteristic of LPE-grown layers.

The most often used *p*-type dopant for the InP and InGaAsP layers is Zn. This dopant suffers from two well-known problems: high distribution coefficient and very high vapor pressure.[26] These can result in the contamination of the neighboring melts with concomitant junction misplacement into the buffer layer. This type of contamination is readily controlled by covering the melt wells and reducing the amount of Zn added. The undoped active layer is in fact grown as *n*-type material, and formation of the *p-n* junction is obtained by diffusion of Zn from the *p*-InP confining layer, as shown schematically in Fig. 6. To keep the *p-n* junction within the active layer, the amount of Zn in the InP confining layer must be kept low. However, to assure adequate electrical contact to the *p*-InP layer its doping level must be as high as

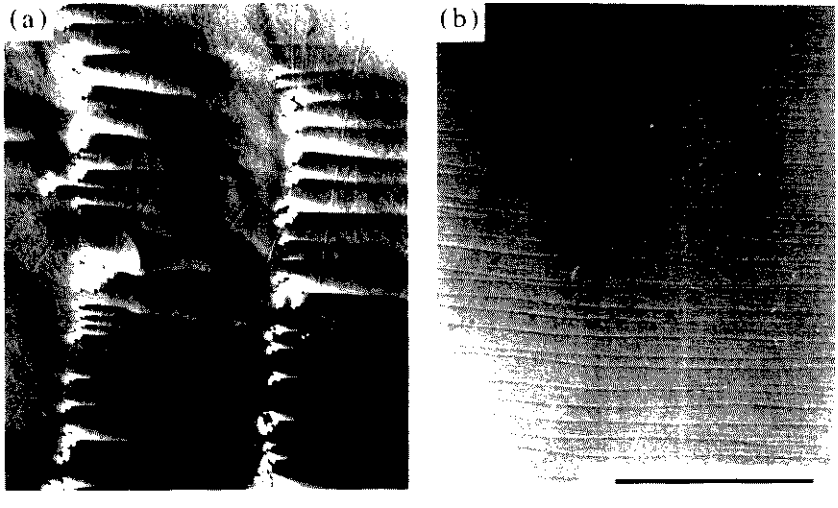



Fig. 5—A Nomarski micrograph (black bar = 100 μm) of the exposed InGaAsP active layer showing: (a) In inclusions owing to an incomplete wipe-off, and (b) inclusion-free surface of properly wiped-off layer.

possible. These conflicting requirements can be reconciled by reducing the doping level in the confining layer to $5 \times 10^{17}$ cm$^{-3}$ and adding a highly doped, $p \sim 2 \times 10^{19}$ cm$^{-3}$, InGaAsP contact layer. Thus, diffusion out of the contact and confining layers is used to form the $p$-$n$ junction, the position of which can be well-controlled within the active layer. The influence of the junction position on LED performance is discussed in Section IV. In addition, contact resistance to the InGaAsP layer is found to be lower than that possible to the InP layer, as discussed in the LED fabrication section.

## III. DEVICE FABRICATION

To obtain high-power, reliable LEDs, a great deal of attention must be devoted to processing procedures. A sequence of processing steps developed for the large-scale manufacture of the 1.3-$\mu$m InGaAsP LED is described in this section, with particular attention given to stress reduction and yield improvement. A typical processing flow chart is shown in Fig. 7. Device fabrication starts with the thinning of the LPE wafer from ~400 to approximately 100 $\mu$m. This is done by chemical-mechanical polishing with a bromine-methanol solution of less than 1

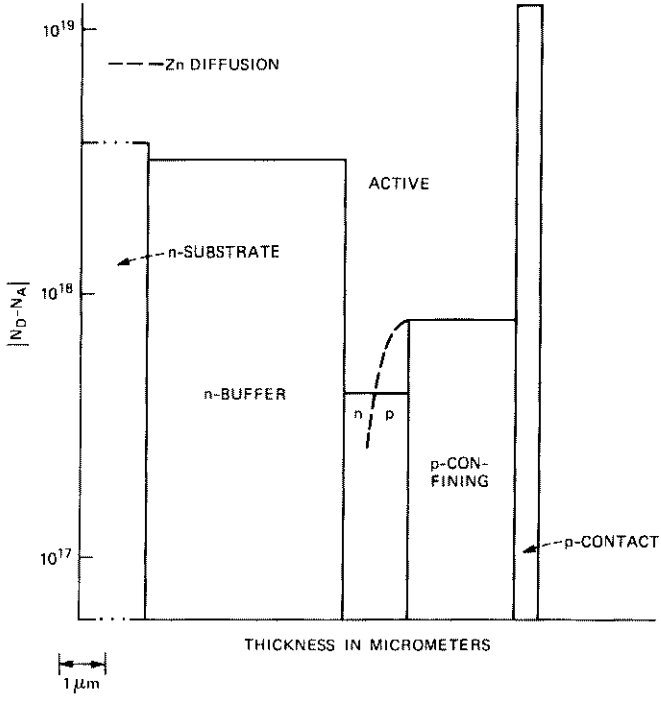



Fig. 6—A schematic diagram of the wafer structure indicating doping types and the Zn diffusion front.

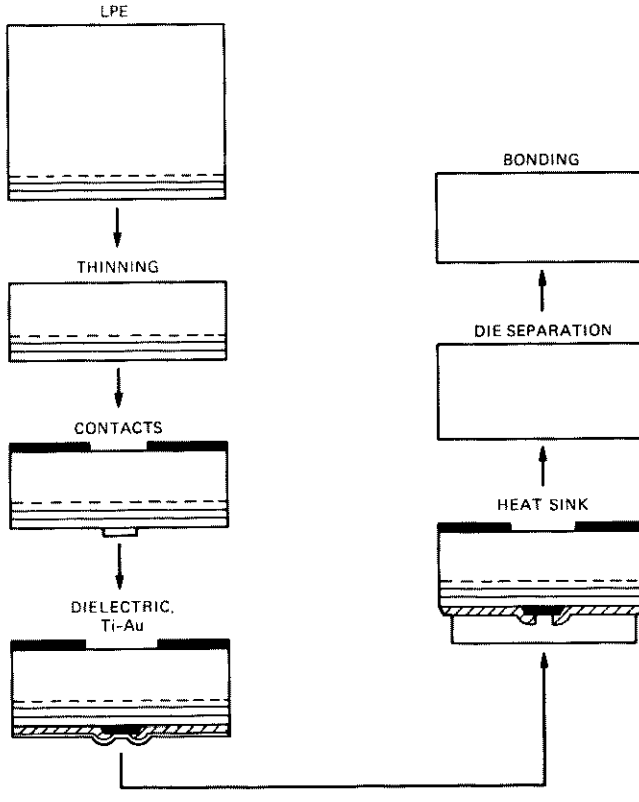



Fig. 7—Schematic processing diagram for a dielectric-isolated LED.

percent. Preliminary mechanical grinding is avoided to prevent formation of deep scratches and to minimize stresses. The finished substrate side should be mirror-like and free of polishing scratches since these affect light emission. After thinning, contacts on both sides are evaporated through a matched pair of shadow masks. Prior to contact deposition, careful cleaning with standard solvents followed by a 1-minute etch in 1:1 $HF/H_2O$ solution are performed. The last etch is important for removal of surface oxides to ensure contact adherence and reproducibility of contact resistance. All the contacts are deposited by *e*-gun evaporation in vacuum of better than $10^{-6}$ torr. The *p*-contact metallization consists of an 800Å-thick layer of 1 wt-percent Be in Au alloy, deposited from an alloy source of the same composition, followed by a 2100Å-thick layer of Au. The BeAu alloy is prepared in advance and maintains its composition owing to nearly congruent evaporation of these two metals. The *n*-contact consists of a "sandwich" of 2000Å of Au, 500Å of Sn, and another overlay of gold at least 10 kÅ thick. The metallizations are alloyed in a $N_2$-$H_2$ flow and a fixed alloying time

of 8 minutes at 420°C is used. The order of metal layers in the *n*-contact is essential for low contact resistance and good bondability. When the first layer of Au is omitted and Sn is deposited directly on the semiconductor, contact morphology and adherence degrade owing to the well-known Sn-Au interdiffusion.[27] In contrast, the layered metallization results in a smooth contact of excellent uniformity and bondability.

Following ohmic contact deposition a dielectric film is placed on the *p*-surface. This film of plasma-deposited $SiO_2$ is used to electrically isolate the exposed surface of the *p*-layer and restrict the current flow to the *p*-contact area. The $SiO_2$ film should be sufficiently thick to prevent pinhole formation and yet as thin as possible to minimize the stress on the semiconductor. A dramatic example of dielectric stress-induced ⟨110⟩ dark-line defects in the *p*-InP confining layer is shown in a TCL image of Fig. 8. While these DLDs do not propagate into the active layer, LEDs in which they have been induced degrade extremely fast. This problem does not arise when the $SiO_2$ layer thickness is reduced to ~1000Å and the deposition conditions are adjusted to minimize stress. Next, the $SiO_2$ layer above the contact is photolithographically removed from an area about half the diameter of the contact. For instance, for the 25-μm diameter *p*-contact, a dielectric opening of ~12 μm is used. The small opening results in the metal contact cushioning the stress of the dielectric edge. The $SiO_2$ and the photoresist are stripped from this area in an $O_2$ plasma. It is important to assure cleanliness of the dielectric openings to obtain stable, low-forward-voltage devices.

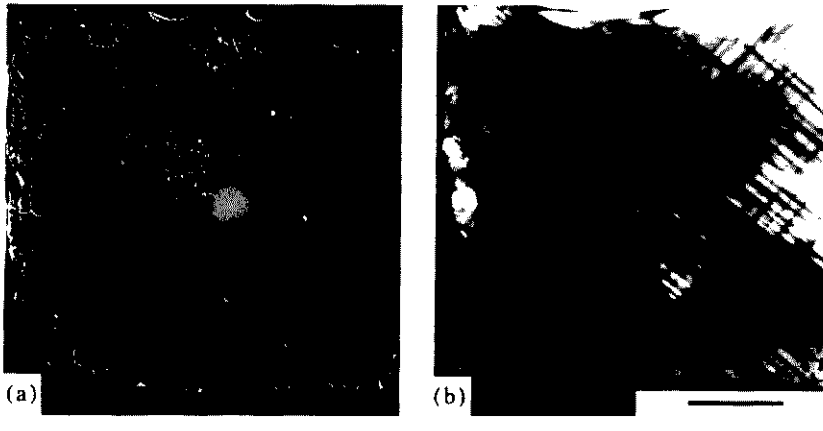



Fig. 8—Dark line defects induced by the dielectric stress in the *p*-InP confining layer: (a) Secondary electron image of the *p*-side of a rapidly degraded LED. The contact area is visible in the center. (b) TCL image showing ⟨110⟩ DLDs throughout the entire device.

A layer of Ti (1000Å) and a Au overlay (3000Å) are subsequently deposited over the full surface of the *p*-side, again by *e*-gun evaporation, and a 20-$\mu$m thick Au heatsink is electroplated. A Ti layer is used for good adhesion between $SiO_2$ and the heatsink. The heatsinks are formed in a photolithographically defined 20-mil-square pad pattern centered around the *p*-contact. The discontinuous Au film reduces wafer stress and makes diamond saw dicing easier. The "streets" in the heatsink pattern, similar to the *n*-type metallization squares, are aligned in the ⟨100⟩ direction. Misalignment tends to cause cleaving along ⟨110⟩ direction during the dicing operation and results in chip breakage. The dicing damage is removed by a ~30s etch in Br-methanol solution, which completes the processing sequence.

A recently described simplified processing sequence in which the dielectric layer and the subsequent photolithography are eliminated is also being used.[29] In this process, schematically shown in Fig. 9, current confinement is achieved through the formation of a Schottky barrier. The ohmic contacts are deposited and alloyed in the usual manner. Afterwards the entire *p*-surface, including the *p*-contact, is covered with a metal layer forming a highly resistive or non-ohmic contact (Schottky barrier) to the exposed semiconductor. This metal layer consists of 1000Å of Ti and 3000Å of Au. The initial layer of Ti adheres well to the semiconductor and forms the Schottky barrier. The processing sequence is again completed by electroplating heatsinks and dicing.

It is well known that the LED-fiber coupling efficiency improves as

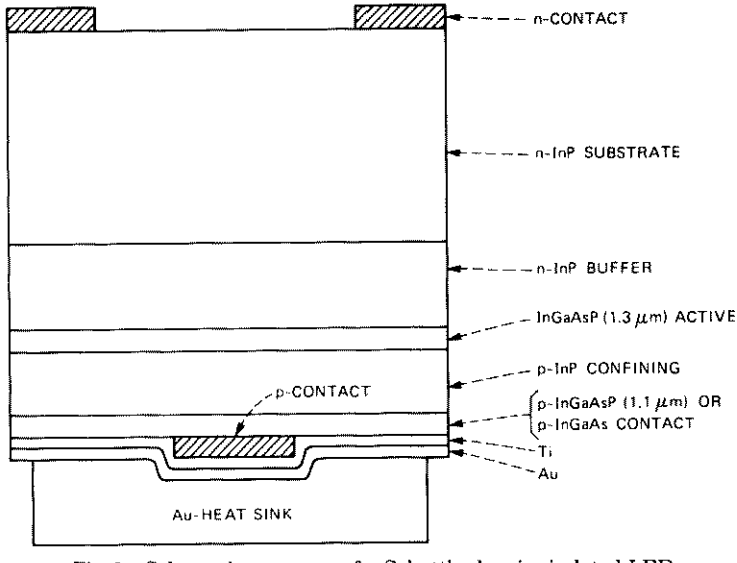



Fig. 9—Schematic structure of a Schottky-barrier isolated LED.

the light spot is made smaller than the fiber diameter. Similarly, the launched power can be increased by increasing the drive current. The resulting high-current density imposes stringent limits on the resistivity of the $p$-metallization. The resistance of the $p$-contact is strongly dependent on the material doping level and composition of the contact layer. In particular, contact resistance to both InP and InGaAsP layers decreases rapidly with increased doping level.[29] However, the low saturation limit for Zn incorporation in epitaxial InP ($\sim 4 \times 10^{18}$ cm$^{-3}$) limits the specific contact resistance to $r_c \sim 7.5 \times 10^{-5} \Omega$ cm$^2$ (see Ref. 30). This value of the specific contact resistance is sufficiently low only for $p$-contacts $\gtrsim 50$ $\mu$m in diameter. Furthermore, an unusual metallurgical reaction between Au and InP results in a spreading and enlargement of the $p$-contact upon metallization alloying, making formation of very small contacts difficult.[29] In contrast, the solubility of Zn in InGaAsP layers saturates at a doping level of $\sim 2 \times 10^{19}$ cm$^{-3}$, resulting in a specific contact resistance of $r_c \sim 2 \times 10^{-5} \Omega$ for the alloy composition corresponding to the 1.2-$\mu$m bandgap.

This composition of the contact layer is typically chosen to avoid absorption of the 1.3-$\mu$m light and allow back-reflection from the $p$-contact. However, the contact resistance continues to decrease as the composition of the InGaAsP layer is changed toward the ternary (1.65-$\mu$m) InGaAs alloy.[31] The results of specific contact resistance measurements as a function of alloy composition are shown in Fig. 10. The lowest $r_c \sim 7 \times 10^{-6} \Omega$ cm$^2$ was obtained on the InGaAs layer. This value is about a factor of two lower than the $r_c$ measured for the commonly used 1.2-$\mu$m InGaAsP and results in 25-$\mu$m-diameter $p$-

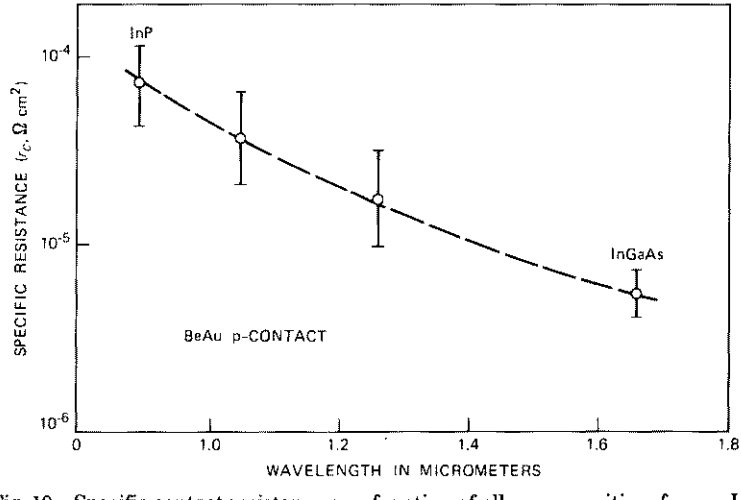



Fig. 10—Specific contact resistance as a function of alloy composition, from $p$-InP to $p$-InGaAs.

contact LEDs with a series resistance of less than 2Ω. LEDs incorporating such a ternary contact layer offer stable operation at very high current density without any appreciable reduction in the launched power, indicating that back-reflected light is not collected by the small acceptance angle of the fiber.

## IV. LED PERFORMANCE

A number of parameters should be considered in determining LED performance in a transmission application. For data rates below ~100 Mb/s, increasing launched power increases repeater distance (at higher data rates modal dispersion begins to be important). The launched power decreases with increasing temperature and this temperature dependence also affects the repeater spacing. The injection level, on which the optical power depends, and the operating temperature may strongly influence reliability considerations, as discussed in the next paragraph. Finally, modulation bandwidth of the device which determines the maximum data rate, and the optical power are related and can be traded for each other by proper adjustments of the doping density and active layer thickness.

The light emitted into the air is plotted as a function of bias current in Fig. 11a at the three ambient temperatures of 0°, 25°, and 70°C. The *L-I* curves are clearly sublinear above 100 mA. At room temperature, a 50-percent increase in the forward current, up to 150 mA, increases the light output by ~20 percent or 1 dB. It should be remembered, however, that for 1.3-μm operation a 1-dB improvement in power output could correspond to a ~1-km increase in repeater distance. Such high current densities, up to 30 kA/cm$^2$ at 150 mA bias for a 25-μm-diameter *p*-contact device, are possible only for devices with a very low series resistance, $r_s \leq 3\Omega$, as discussed in the preceding section. At 25°C nearly 3 mW of optical power are emitted by these high-radiance devices. A higher power output into the air can be obtained for LEDs with larger *p*-contact areas, where the ohmic heating is reduced. However, small light-emitting areas are necessary for the efficient coupling of light into optical fibers.

The temperature dependence of the light output between −40°C and +70°C is plotted in more detail in Fig. 11b. The light-output/temperature characteristic follows the empirical relationship $L = L_0 \exp(-T/T_1)$, where the characteristic temperature $T_1$ of our devices ranges between 180 and 220 degrees and is independent of current.[32] Similarly, the functional form of $L(I)$ is independent of temperature. Thus, a temperature rise of 50°C above room ambient results in a power output decline of approximately 1 dB. While this temperature sensitivity is somewhat greater than that of comparable GaAlAs LEDs,

where $T_1 \approx 300$–$350°C$ has been observed, it is sufficiently small for LED operation without need for cooling devices.

Powers launched into fibers, as a function of fiber numerical aperture (NA) and core radius, are shown in Fig. 12. Two standard fibers, 0.23-NA, 50-$\mu$m core and 0.29-NA, 62.5-$\mu$m core are indicated by arrows. Closed circles denote butt-coupled powers; open circles and triangles refer to power launched using a lensing scheme in which a small sphere of high-index glass ($\sim$110 $\mu$m in diameter, $n \approx 1.9$) is mounted on the cleaved fiber end. All the devices have a 25-$\mu$m $p$-contact, active layer

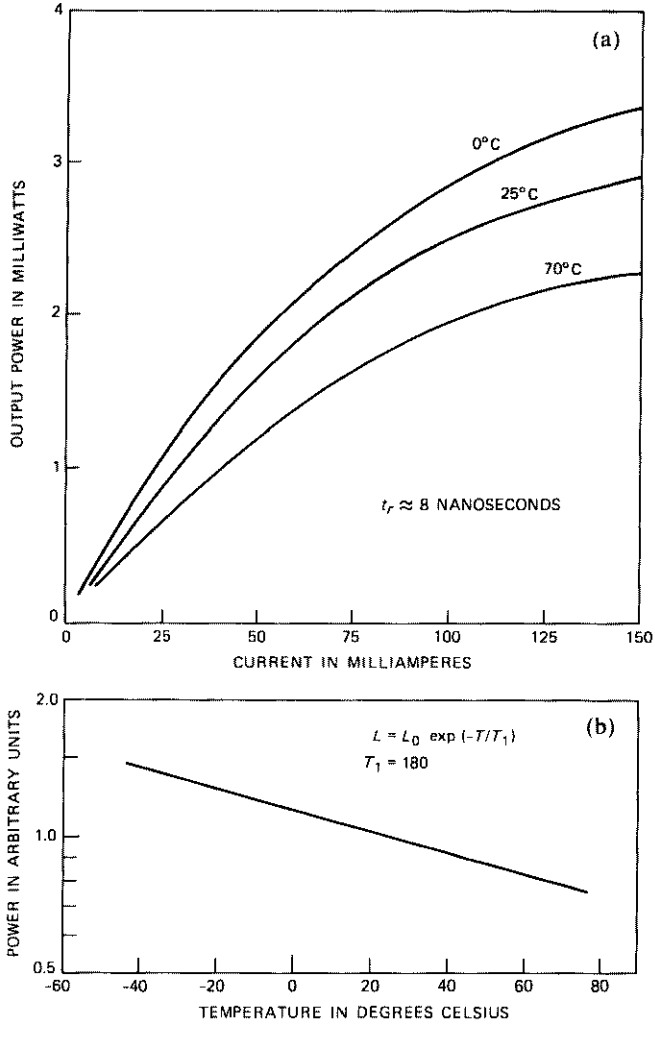



Fig. 11—(a) Light-current relation for a 25-$\mu$m diameter $p$-contact LED at three different ambient temperatures. (b) Temperature dependence of the light output at a constant current.

thickness of less than 0.7 $\mu$m, and a 10- to 90-percent risetimes of $\lesssim$4 ns. Because of current spreading the light spots are 30 to 35 $\mu$m in diameter, somewhat larger than the $p$-contact. No differences in current spreading or power launched have been found between dielectric and Schottky-isolated LEDs.

In the first approximation, the power coupled by an LED with a light-spot diameter $d_s$ to a graded index fiber with a core $d_f$, when $d_s \leq d_f$, is given by

$$P = 1/2(\mathrm{NA})^2 P_0 \text{ butt coupled,} \qquad \text{and}$$

$$P = 1/2(\mathrm{NA})^2 M^2 P_0 \text{ lens coupled,}$$

where NA is the numerical aperture of the fiber, $P_0$ is the power radiated from the LED surface, and $M$ is the magnification of the LED

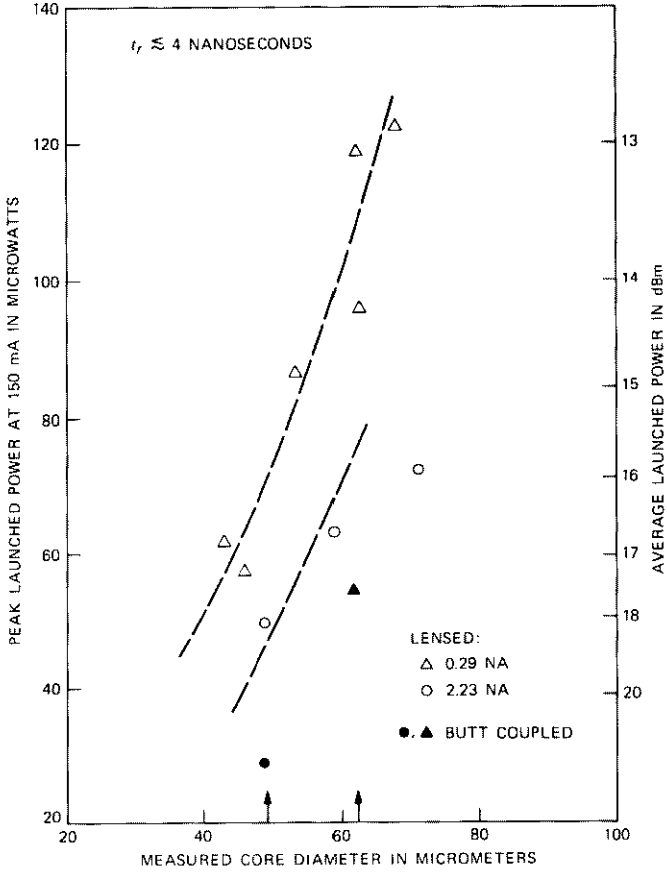



Fig. 12—Optical power launched into fiber as a function of NA and core radius. Two common fibers (0.23 NA, 50-$\mu$m core; 0.29 NA, 62.5-$\mu$m core) are indicated by arrows.

spot size by the lens. The latter has a maximum value when $M = d_f/d_s$, i.e., when the image just fills the fiber core. For a lensed system the coupled power, $P \approx (\mathrm{NA} \cdot d_f)^2$, as indicated by the dashed lines of Fig. 12. The data points and the dashed lines are in reasonably good agreement. The highest powers (peak) coupled by these fast LEDs were 50 and 120 $\mu$W into 50-$\mu$m and 62-$\mu$m core fibers, respectively. These powers are clearly a very strong function of the $d_f/d_s$ ratio, the numerical aperture, and the lensing scheme used. Addition of an antireflection coating results in as much as a 20-percent increase in the power launched.[33] A coupling experiment with a 100-$\mu$m core 0.25-NA step index fiber achieved 410 $\mu$W of optical power. Such fibers, however, are not useful for optical communications.

The 1.3-$\mu$m LEDs discussed in Fig. 12 are very fast. Using a square-current pulse of 100 mA and a dc offset of 6 mA, risetimes of less than 4 ns are measured. The fall times are typically 1 to 2 ns longer. The fastest devices show risetimes close to 2 ns. As indicated by electron beam induced current (EBIC) scans, these thin active-layer (0.5 to 0.7 $\mu$m) devices have a $p$-$n$ junction at the interface between the InGaAsP active layer and the $n$-InP buffer layer. The active layer is thus converted to $p$-type by Zn out diffusing from the confining $p$-InP layer. Hall measurements on InGaAsP layers grown under conditions simulating active-layer growth indicate doping level of $p \gtrsim 7 \times 10^{17}$ cm$^{-3}$. These LEDs have been successfully used in transmission experiments at 90-Mb/s rate.[34] On the basis of these risetime data, operation at 140 Mb/s is feasible without any speed-up circuitry.

The interplay of power and speed is well understood in GaAlAs LEDs.[35,36] The active layer of those devices can be doped directly with Ge and the resulting $p$-type doping level is well controlled. Thus, it is possible to change the doping level from $p \sim 1 \times 10^{17}$ cm$^{-3}$ to $p \sim 1 \times 10^{19}$ cm$^{-3}$ and the corresponding risetimes decrease from 25 to 4 ns. At the same time the power output is inversely proportional to the risetime, resulting in a constant power-bandwidth product in good agreement with theoretical models.[36] The results of a bandwidth-power study of 1.3-$\mu$m InGaAsP LEDs are summarized in Fig. 13. The 10- to 90-percent risetime (ns) is plotted on a log-log scale versus power launched into the larger fiber (62.5-$\mu$m core, 0.29 NA). Each data point represents an average of 10 devices from a single wafer. For the purpose of this study, the active-layer thickness is increased to 1.4 $\mu$m. It should be remembered that the $p$-$n$ junction is formed by Zn diffusion from the confining layer and increased active-layer thickness permits better control of the diffusion front depth. The risetime correlates well with the amount of Zn added to the confining layer. The heavily Zn-doped devices have risetimes on the order of 4 to 6 ns, and the $p$-$n$ junction is shifted all the way to the $n$-InP buffer interface.

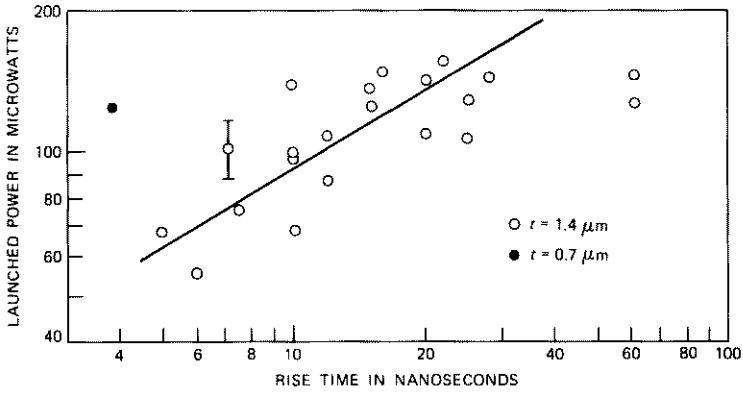



Fig. 13—Power versus risetime relationship for LEDs with a 1.4-μm thick active layer. Full dot demonstrates performance of 0.7-μm active-layer device.

As the amount of Zn is reduced, risetimes increase, up to 30 ns and output power increases by a factor of 2 to 3. However, in contrast to GaAlAs devices, the *p-n* junction displacement is gradually reduced, resulting in an *n*-type active layer. High powers of these LEDs demonstrate efficient hole injection into the InGaAsP layer. A simple model of a displaced junction gives a decay rate[37]

$$R = \frac{1}{\tau} = \omega \frac{1}{\tau_p} + (1 - \omega) \frac{1}{\tau_n},$$

where under high injection conditions

$$\omega = \left(1 + \frac{t_p}{t_n}\right)^{-1},$$

and $\tau_{n,p}$ refers to electron/hole lifetimes, respectively, $t_p$ is the thickness of the active-layer region converted to *p*-type, and $t_n = t_{\text{active}} - t_p$. Thus, the decay rate is equal to the volume-weighted average of both minority carrier lifetimes. In our devices $\tau_n \approx 4$ ns and $\tau_p \geq 30$ ns. This finding is in good agreement with a previous observation[9] of increased power output for increased $t_n/t_p$. However, the *n*-active layer LEDs are slower and may not be useful for high-data-rate applications.

The best power result on thin (0.7 μm), active-layer LEDs is shown for comparison as a full circle in Fig. 13. The slower devices launch as much as 150 μW into 62-μm core fiber and up to 177 μW with an antireflection coating. The relatively high power and high speed of thin active-layer LEDs are interpreted in terms of higher injected carrier density and the resulting bimolecular recombination, which shortens the recombination lifetime.[38] It should, however, be noted that the 1.4-μm active-layer preparation is more reproducible and device results more uniform.

## V. RELIABILITY

To estimate the reliability of LEDs for lightwave systems, life tests are performed under forward bias at elevated temperatures. The purpose is (*i*) to identify the failure modes relevant at operating temperatures, (*ii*) to find and eliminate the infant failure population, (*iii*) to determine the time, current, and temperature dependence of the degradation in light output to allow accurate extrapolations to end-of-life, and (*iv*) to determine the failure distribution so that failure rates can be calculated that reflect both the median life and the spread in the data. It is useful to begin by contrasting the failure modes in GaAlAs LEDs with what is known for quaternary LEDs.

The reliability of double heterostructure GaAs/GaAlAs has been studied extensively.[39,40] It is well known that ⟨100⟩ dark line defects (DLDs) form readily at grown-in dislocations in these devices. Their rate of growth is proportional to the square of current density ($J^2$), and independent of temperature. Depending upon the dislocation density in the epitaxial material, a certain fraction of devices (active area multiplied by the dislocation density) will fail catastrophically owing to DLD formation and must be screened out in a burn-in of about 100h. Under forward bias aging at 6 kA/cm$^2$ and $T_J$ up to 250°C, the surviving devices remain free of DLDs but change gradually in light output owing to two competing processes with different time dependences and different activation energies.[40] These processes can be separated, and the degradation process is found to have an activation energy of 0.65 eV. This degradation is assumed to be due to the introduction of nonradiative defects at the junction. The projected mean-time-to-failure (MTTF) at 70°C is $4 \times 10^6$h. A notable feature of GaAlAs LED degradation is the small spread in times-to-failure for samples of diodes.[40] The failure distribution can be described as lognormal with $\sigma \cong 0.5$. This small $\sigma$ coupled with the long MTTF gives very low estimated failure rates (e.g., a maximum of 10 FITs over a 20-year service life at $T_J = 70$°C).

In contrast, quaternary LEDs do not form ⟨100⟩ DLDs at threading dislocations, permitting much larger, ~$10^4$/cm$^2$, dislocation densities in InP substrates. Diodes with several dislocations in the active area have been aged at 10 kA/cm$^2$ for $2 \times 10^4$h without the appearance of DLDs.[41,42] The only dislocations that have been observed in electroluminescence after aging are ⟨110⟩ misfit lines in some diodes where the lattice mismatch between the buffer and quaternary layers is greater than 0.06 percent. In some cases the appearance of ⟨110⟩ lines accompanies a rapid drop in light output. The only infant failures observed in lattice-matched InGaAsP LEDs are due to instabilities in *I-Vs* or in light output. These instabilities result from shunt paths across the junction caused by In inclusions. A burn-in of 100h at 150

mA is done to screen out unstable devices. However, the burn-in yield for good, i.e., inclusion-free, wafers is almost 100 percent.

We report here the results of two series of accelerated aging experiments on dielectric isolated devices. In the first series, diodes labeled FC291 were fabricated from a wafer grown early in the quaternary LED development program. These were three-layer structures, i.e., without a contact layer, with 50-$\mu$m diameter contacts (the active-layer thickness was 0.7 $\mu$m). The epitaxial material contained a number of In inclusions up to $\sim$10 $\mu$m in diameter, as discussed in Section II, and would be considered poor by present standards. The elapsed time for this series of experiments is now $1.5 \times 10^4$h. The second series of experiments is on wafer-labelled LP137, with an active-layer thickness of 1.4 $\mu$m. These are four-layer structures with 25-$\mu$m contacts, so that the current density $J$ for a given device current is about four times that in FC291. These diodes were free of inclusions as shown by their electroluminescence patterns. The elapsed time for this series is $4 \times 10^3$h.

In FC291-type wafers, two modes of degradation have been identified: ⟨110⟩ DLD formation above 140°C and dark spot defect (DSD) formation at all temperatures. In the first mode, some diodes form ⟨110⟩ DLDs and drop rapidly in light output, even without bias. The DLDs originate at microscopic inclusion-like defects and are confined to the InP buffer layer. These ⟨110⟩ DLDs are similar to the dielectric stress-induced defects discussed in Section III. It is concluded that these DLDs are caused by thermal stress and that they grow by dislocation glide on the {111} slip planes. Because 140°C is apparently a threshold temperature for this effect, it will not be a relevant failure mode at operating temperatures.[43] This type of DLD has not been observed in recent wafers.

In analyzing the data for FC291, we eliminate all diodes that have ⟨110⟩ DLDs. In the remaining diodes, the light output falls linearly in time as shown in Fig. 14 and DSDs appear and grow. Because these LEDs do not fail abruptly, but rather drop gradually in light output, the definition of end-of-life should consider the overall system objective.[44] Since the repeater span is generally power-limited, it is desirable to allow as little power margin for end-of-life as possible, within system reliability requirements. For purposes of illustration, we define end-of-life as $-1$ dB change in output power. Figure 15 shows a typical failure distribution. It demonstrates that the failures are lognormally distributed with $\sigma = 0.6$. This $\sigma$ is similar to that of GaAlAs devices and demonstrates excellent sample homogeneity. Figure 16 is an Arrhenius plot for median life under the various aging conditions. The activation energy is 0.85 eV. This activation energy is significantly larger than that obtained for the degradation of GaAlAs LEDs.

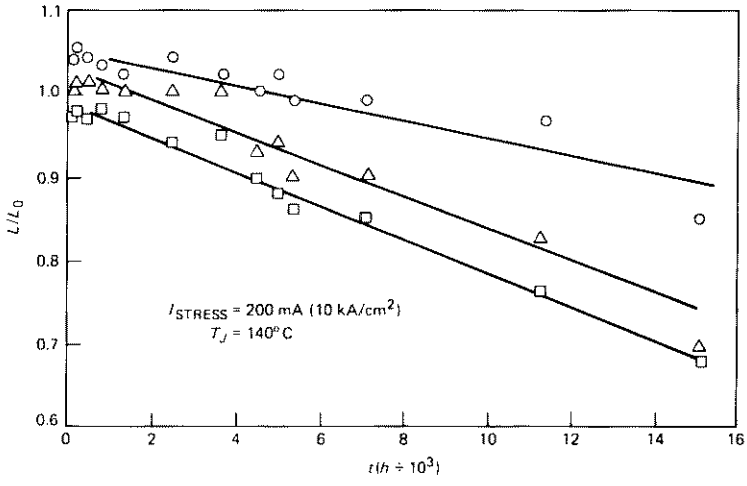



Fig. 14—Accelerated aging tests: light output versus time as a function of temperature for three typical diodes.

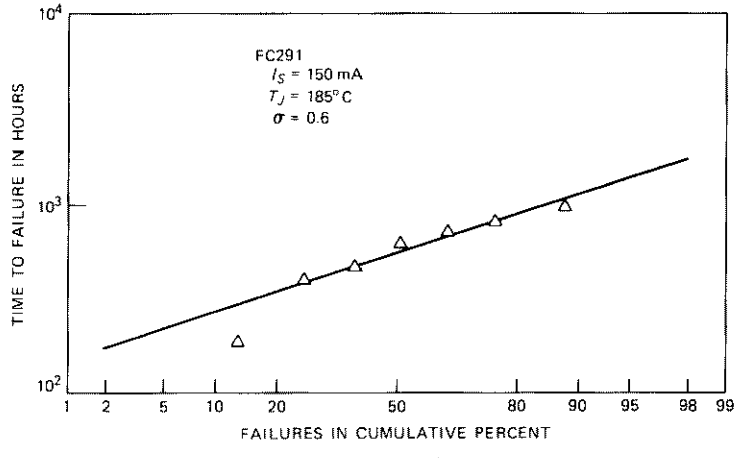



Fig. 15—Log-normal failure distribution typical of InGaAsP LED aging results.

Figure 16 also shows the results of experiments on wafer LP137. These diodes drop in light output linearly in time, but do *not* form DSDs. This is important, since DSD formation was believed to be a main degradation mechanism for InGaAsP LEDs. There is very little difference between median life (ML) for 20 kA/cm$^2$ and 40 kA/cm$^2$, suggesting that the failure mechanism (uniform degradation in light output) is not a strong function of current. Although the current density is four times higher in LP137 than in FC291, the ML has gone up by a factor of four. This probably reflects the improvement in the quality of the material.

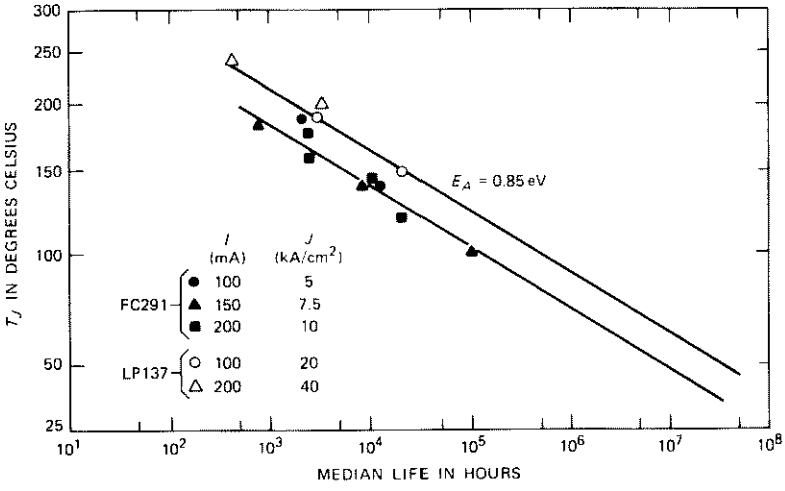



Fig. 16—Arrhenius plot of median life under various aging conditions. The activation energy is 0.85 eV.

The maximum operating temperature for these LEDs in most applications is 70°C. The 40 kA/cm$^2$ data projects median life of $4 \times 10^6$h at this temperature. For a service life of 20 years, using $\sigma = 0.6$, the projected failure rate is <1 FIT.[45] At 25°C median life of $3 \times 10^8$h is predicted. Higher reliability results if greater degradation is permitted. The system designer must trade off initial system performance for reliability objectives.

These reliability results are in good agreement with previous studies.[42,46] Thus, median life well in excess of $10^6$ hours has been predicted at 25°C and current densities below 20 kA/cm$^2$. Similarly, the activation energy of 0.85 eV is consistent, within experimental uncertainties, with the 1.0 eV obtained previously. However, absence of DSDs in the high-quality wafer LP137 suggests that this degradation mechanism may be extrinsic. Further progress in material preparation and processing techniques should result in even better reliability of InGaAsP LEDs.

## VI. SUMMARY

We have described the 1.3-$\mu$m InGaAsP LED developed for application in optical fiber transmission. This includes improvements and scale-up in the epitaxial growth procedures, careful design of the processing sequence, and optimization of the device structure. The resulting LED has a number of very attractive features. Its excellent optical powers allow repeater distances in the 10- to 20-km range for 62-$\mu$m fibers, depending on the cable loss. These long repeaterless spans can be realized at data rates of up to 90 Mb/s. This is accom-

plished with a relatively simple drive circuitry, as compared to lasers. In addition, the InGaAsP LED exhibits excellent thermal stability. As the ambient temperature is increased from 25°C to 70°C, the power output decreases by only ~1 dB. Finally, even at 70°C the median life exceeds $4 \times 10^6$h and the resulting failure rate is expected to be below 1 FIT.

## VII. ACKNOWLEDGMENTS

We thank T. P. Lee and M. A. Pollack for their generosity with materials and good advice, W. A. Bonner for all the excellent InP substrates, and S. Mahajan for defect characterization and many useful discussions. Many thanks to J. A. Lourenco, D. D. Roccasecca, I. Camlibel, R. H. Frahm, and G. Minneci for the superb technical assistance in crystal growth, processing, and device characterization; to F. Ermanis for uncounted SEM pictures; and to R. Caruso for X-ray diffraction measurements. Above all we would like to acknowledge A. A. Bergh, whose support has made this work possible.

## REFERENCES

1. A. A. Bergh, J. A. Copeland, and R. W. Dixon, "Optical Sources for Fiber Transmission Systems," IEEE Proc., *68,* No. 10 (October 1980), pp. 1240–47.
2. P. C. Schultz, Third Int. Conf. on Integrated Optics and Optical Fiber Comm., Technical Digest, April 1981, San Francisco; and D. Charlton and P. C. Schultz, "Progress in Optical Waveguide Processes, 1980," Electro-Optical System Design, December 1980, pp. 23–9.
3. J. W. Fleming, "Material Dispersion in Lightguide Glasses," Electron. Lett., *14,* No. 11 (May 1978), pp. 326–8.
4. J. Conradi, F. P. Kapron, and J. C. Dyment, "Fiber-Optical Transmission Between 0.8 and 1.4 $\mu$m," IEEE Trans. Electron. Dev., *ED-25,* No. 2 (February 1978), pp. 180–93.
5. D. Gloge, A. Albanese, C. A. Burrus, E. L. Chinnock, J. A. Copeland, A. G. Dentai, T. P. Lee, Tingye Li, and K. Ogawa, "High-Speed Digital Lightwave Communications Using LEDs and PIN Photodiodes at 1.3 $\mu$m," B.S.T.J., *59,* No. 8 (October 1980), pp. 1365–82.
6. G. A. Antypas, R. L. Moon, L. W. James, J. Edgecumbe, and R. L. Bell , "III-V Quaternary Alloys," Inst. Phys. Conf. Series. 1972 Symp. on GaAs, pp. 48–54.
7. M. A. Pollack, R. E. Nahory, J. C. DeWinter, and A. A. Ballman, "Liquid phase epitaxial $In_{1-x}Ga_xAs_yP_{1-y}$ lattice matched to (100) InP over the complete waveguide range $0.92 < \lambda < 1.65$ $\mu$m," Appl. Phys. Lett., *33,* No. 4 (August 1978), pp. 314–16.
8. P. D. Wright, Y. G. Chai, and G. A. Antypas, "InGaPAs-InP Double-Heterojunction High-Radiance LED's," IEEE Trans. Electron. Dev. *ED-26,* No. 8 (August 1975), pp. 1220.
9. O. Wada, S. Yamakoshi, M. Abe, Y. Nishitani, and T. Sakurai, "High Radiance InGaAsP/InP Lensed LED's for Optical Communication Systems at 1.2–1.3 $\mu$m," IEEE J. Quant. Electr., *QE-17,* No. 2 (February 1981), pp. 174–8.
10. R. C. Goodfellow, A. C. Carter, I. Griffith, and R. R. Bradley, "GaInAsP/InP, High-Radiance, 1.05–1.3 $\mu$m Wavelength LED's with Efficient Lens Coupling to Small Numerical Aperture Silica Optical Fibers," IEEE Trans. Electron. Dev., *ED-26,* No. 8 (August 1979), pp. 1215–20.
11. A. G. Dentai, T. P. Lee, and C. A. Burrus, "Small-Area High Radiance C.W. InGaAsP LEDs Emitting at 1.2 to 1.3 $\mu$m," Electr. Lett., *13* (1977) pp. 484–5.
12. R. F. Leheny, R. E. Nahory, and M. A. Pollack, "$In_{0.53}Ga_{0.47}As$ PIN Photodiodes for Long-Wavelength Fiber Optic Systems," Electr. Lett., *15* (1979), pp. 713–15.
13. V. G. Keramidas, H. Temkin, and W. A. Bonner, "Growth of InP and InGaAsP

layers by liquid phase epitaxy under phosphorus overpressure," Appl. Phys. Lett., *40* (1982), p. 731.
14. S. Mahajan, V. G. Keramidas, A. K. Chin, W. A. Bonner, and A. A. Ballman, "Perfection of homoepitaxial layers grown on (001) InP substrates," Appl. Phys. Lett., *38*, No. 4 (February 1981), pp. 255–8.
15. W. A. Bonner, "Reproducible Preparation of Twin-Free InP Crystals Using the LEC Technique," Mat. Res. Bull., *15* (1980), pp. 63–72.
16. M. A. DiGiuseppe, H. Temkin, and W. A. Bonner, "Large Area LPE Growth of InGaAsP/InP double heterostructures on InP preserved in a phosphorus ambient," J. Cryst. Growth, to be published.
17. A. K. Chin, H. Temkin, and S. Mahajan, "Transmission Cathodoluminescence," B.S.T..J., *60*, No. 9 (November 1981), pp. 2187–226.
18. Y. Seki, H. Watanabe, and J. Matsui, "Impurity effect on grown-in dislocation density of InP and GaAs crystals," J. Appl. Phys., *49* (February 1978), pp. 822–8.
19. A. K. Chin, V. G. Keramidas, W. D. Johnston, Jr., S. Mahajan, and D. D. Roccasecca, "Evaluation of defects and degradation in GaAs-GaAlAs wafer using transmission cathodoluminescence," J. Appl. Phys., *51*, No. 2 (February 1980), pp. 978–83.
20. H. Temkin, V. G. Keramidas, and S. Mahajan, "Thermal Decomposition of InP and Its Influence on Iso-Epitaxy," J. Electrochem. Soc., *128*, No. 5 (May 1981), pp. 1088–91.
21. G. A. Antypas, "Prevention of InP surface decomposition in liquid phase epitaxial growth," Appl. Phys. Lett., *37*, No. 1 (July 1980), pp. 64–5.
22. M. G. Astles, F. G. H. Smith, and E. W. Williams, "Indium Phosphide, Epitaxial Growth," J. Electrochem. Soc., *120* (1973), pp. 1750–7.
23. W. Y. Lum and A. R. Clawson. "Thermal Degradation of InP and Its Control in LPE Growth," J. Appl. Phys., *50*, No. 8 (August 1979), pp. 5296–301.
24. K. Nakajima, S. Yamazaki, S. Komiya, and K. Akita, "Misfit dislocation-free InGaAsP/InP heterostructure wafers grown by liquid phase epitaxy," J. Appl. Phys., *52*, No. 7 (July 1981), pp. 4575–82.
25. H. Temkin, V. G. Keramidas, M. A. Pollack, and W. R. Wagner, "Temperature dependence of photoluminescence of *n*-InGaAsP," J. Appl. Phys., *52*, No. 3 (March 1981), pp. 1574–8.
26. O. Wada, A. Majerfeld, and P. N. Robson, "Control of Zn-doping for growth of InP *pn* junction by liquid phase epitaxy," J. Electrochem. Soc., *127*, No. 10 (October 1980), pp. 2278–84.
27. S. Nakahara, R. J. McCoy, L. Buene, and J. M. Vandenberg, "Room temperature interdiffusion studies of Au/Sn thin film complexes," Thin Solid Films, *84* (1981), pp. 185–6.
28. A. K. Chin, C. L. Zipfel, B. V. Dutt, M. A. DiGiuseppe, K. B. Bauers, and D. D. Roccasecca, "New Restricted Contact LEDs Using a Schottky Barrier," Japn. J. Appl. Phys., *20*, No. 8 (August 1981), pp. 1487–91.
29. V. G. Keramidas, H. Temkin, and S. Mahajan, "Ohmic contacts to InP and InGaAsP," Inst. Phys. Conf. Ser., *56* (1981), pp. 293–9.
30. H. Temkin, R. J. McCoy, V. G. Keramidas, and W. A. Bonner, "Ohmic contacts to *p*-type InP using Be-Au metallization," Appl. Phys. Lett., *36*, No. 6 (March 1980), pp. 444–6.
31. H. Temkin, A. K. Chin, and M. A. DiGiuseppe, "$In_{0.53}Ga_{0.47}As$ contact layer for 1.3-$\mu$m light emitting diodes," Electron. Lett., *17*, No. 19 (September 1981), pp. 703–5.
32. H. Temkin, A. K. Chin, M. A. DiGiuseppe, and V. G. Keramidas, "Light-current characteristics of InGaAsP of light-emitting diodes," Appl. Phys. Lett., *39*, No. 5 (September 1981), pp. 405–7.
33. R. H. Burton, unpublished work.
34. G. T. Daryanani, unpublished work.
35. V. G. Keramidas, A. K. Chin, C. L. Zipfel, D. D. Roccasecca, and F. Ermanis, unpublished work.
36. T. P. Lee, and A. G. Dentai, "Power and modulation bandwidth of GaAs-AlGaAs high-radiance LED's for optical communication systems," IEEE J. Quant. Electr., *QE-14*, No. 3 (March 1978), pp. 150–9.
37. H. Temkin, W. B. Joyce, A. K. Chin, M. A. DiGiuseppe, and F. Ermanis, "Effect of p-n function position on the performance of InGaAsP light emitting diodes", Appl. Phys. Lett., *41*, No. 8 (October 1982), pp. 745–7.
38. J. S. Blakemore, *Semiconductor Statistics*, New York: Pergamon Press, 1962.
39. S. Yamakoshi, T. Sugahora, O. Hasegawa, Y. Toyama, and H. Takanashi, "Growth mechanism of (100) dark-line defects in high radiance GaAlAs LEDs," IEDM Proc., Washington, D. C. (1978), pp. 642–5.
40. C. L. Zipfel, R. H. Saul, A. K. Chin, and V. G. Keramidas, "Competing processes in

long-term accelerated aging of DH GaAlAs LEDs," J. Appl. Phys., *53,* No. 3, Part 1 (March 1982), pp. 1781–6.
41. S. Yamakoshi, M. Abe, S. Komiya, and Y. Toyama. "Degradation of high radiance InGaAsP/InP LEDs at 1.2–1.3 μm wavelength," IEDM Proc., Washington, D. C. (1979), pp. 122–5.
42. S. Yamakoshi, M. Abe, O. Wada, S. Komiya, and T. Sakurai, "Reliability of high radiance InGaAsP/InP LEDs operating in the 1.2–1.3 μm wavelength," IEEE J. Quant. Electr., *EQ-17,* No. 2 (February 1981), pp. 167–73.
43. H. Temkin, C. L. Zipfel, and V. G. Keramidas, "High-temperature degradation of InGaAsP/InP light emitting diodes," J. Appl. Phys., *52,* No. 8 (August 1981), pp. 5377–80.
44. R. H. Saul and C. L. Zipfel, unpublished work.
45. A. S. Jordan, "A comprehensive review of the lognormal failure distribution with application to LED reliability," *Microelectronics and Reliability, 18,* No. 3, Oxford: Pergamon Press, 1978, pp. 267–79.
46. R. Yeats, Y. G. Chai, T. D. Gibbs, and G. A. Antypas, "Performance Characteristics and Extended Lifetime Data for InGaAsP/InP LED's," IEEE Electr. Dev. Lett., *EDL-2,* No. 9 (September 1981), pp. 234–6.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Analog Scramblers for Speech Based on Sequential Permutations in Time and Frequency

By N. S. JAYANT, R. V. COX, B. J. McDERMOTT, and A. M. QUINN

(Manuscript received July 20, 1982)

*Permutation of speech segments is frequently utilized in scramblers for analog speech privacy. This paper discusses a* "sequential" *permutation procedure that has better segment-separation properties than the well-known procedure of* "block" *permutation, where contiguous segments are arranged in blocks of appropriate size, and permuted within such blocks. It further proposes the application of the sequential procedure to a novel technique for simultaneous permutations in time and frequency. The paper also presents results of a subjective experiment where we measured residual speech intelligibility at the output of scramblers using permutations in time [time segment permutation (TSP)], or permutations in time* and *frequency [time-frequency segment permutation (TFSP)]. The experiment included examples of block TSP, sequential TSP, and sequential TFSP. We measured spoken-digit-intelligibility as a function of the communication delay introduced by the scrambling operation. We found that even with a delay of 512 ms, the residual intelligibility in a TSP scrambler is no lower than about 50 percent; however, a sequential TFSP scrambler can realize an average digit intelligibility in the order of 20 percent with a delay of 256 ms. A companion paper discusses the implementation of the sequential TFSP scrambler, and the quality of descrambled speech in the context of real-channel operation.*

## I. INTRODUCTION

Permutation of speech segments that are about 10 to 30 ms long is a bandwidth-preserving operation[1] that is frequently utilized in the design of scramblers for analog speech privacy.[2,3] The procedure,

known as Time Segment Permutation (TSP), lends itself to fairly robust real-channel operation[2] and efficient microprocessor implementation.[4] However, the reduction in speech intelligibility by this method is very small if the communication delay introduced by the scrambler and descrambler is constrained to be no longer than, say, 256 to 512 ms. This rather well-known deficiency was quantitatively demonstrated in a recent article[1] that discussed a segment scrambler, which will be referred to as "block" TSP in this paper (see Section 2.2). In this scrambler, contiguous speech segments are arranged in blocks of appropriate size, and permuted *within* such blocks. The entire block with permuted segments is then transmitted as scrambled speech, before proceeding to the next block. In this paper, we discuss another segment permutation procedure to be called "sequential" TSP (see Section 2.3). In this case, permutations are not constrained to be within blocks, and transmissions of scrambled speech are not constrained to be on a block-by-block basis. We will compare the two procedures in terms of how well they separate segments that are initially adjacent in the unscrambled speech. We will show that the sequential TSP has segment-separation properties that are much better than those of block TSP, but that, unfortunately, this is not accompanied by substantial gains in the residual intelligibility in scrambled speech, except for large values of communication delay (see Section 3.3).

To realize substantial reductions of intelligibility, it is imperative to use so-called two-dimensional approaches to scrambling.[1,3] One example of two-dimensional scrambling is the combination of block TSP and frequency inversion.[1] However, frequency inversion is a very straightforward, simple and time-invariant operation with only one possible input-output mapping, or "key." It therefore has no cryptanalytical strength. An important purpose of this paper is to propose and evaluate a two-dimensional procedure that offers a residual intelligibility very similar to that of block TSP plus frequency inversion, and a cryptanalytical strength that is much higher than in that method. This new procedure (Section IV) will be called Time-Frequency Segment Permutation (TFSP). In particular, we will be discussing a sequential version of this procedure, "sequential" TFSP.

A companion paper[5] discusses the implementation of the sequential TFSP scrambler, and the quality of descrambled speech in the context of a real-channel operation.

## II. ONE-DIMENSIONAL SCRAMBLERS: BLOCK TSP AND SEQUENTIAL TSP

This section describes a sequential approach to segment-permutation and shows that it has much better segment separation properties

than a non-sequential, or block, procedure such as that discussed in Ref. 1. The block and sequential approaches to be discussed below are sometimes referred to as "*hopping window*" and "*sliding window*" approaches.[6]

### 2.1 Temporal distance d

The purpose of permutation scrambling is to reduce intelligibility by altering the normal time order of speech segments. The greater the separation in scrambled speech between normally adjacent segments, the lower the intelligibility is expected to be. Conversely, adjacent samples in the scrambled speech should be well separated in normal speech. An important result of this paper is that average segment separation and intelligibility are generally, if not always, monotonically related. It is useful, therefore, to define the following objective measure of effectiveness for permutation scramblers as the *temporal distance between a pair of segments in normal speech that appear as adjacent segments in scrambled speech.* For simplicity, we will henceforth refer to this non-zero, positive parameter merely as the "temporal distance $d$." By definition,

$$\begin{aligned} d &= 1 \quad \text{for adjacent segments in unpermuted speech} \\ d &\geq 1 \quad \text{for adjacent segments in permuted speech.} \end{aligned} \tag{1}$$

Illustrations of temporal distance appear in Fig. 1. The scramblers used in this figure will be defined in the next two sections.

Briefly, block TSP is a procedure where an entire block of segments in scrambler memory is transmitted before proceeding to the next block. Segment selection involves a number of candidates that decreases from the initial number of segments in the block to 1, as a given block is processed. In sequential TSP, each stage of segment

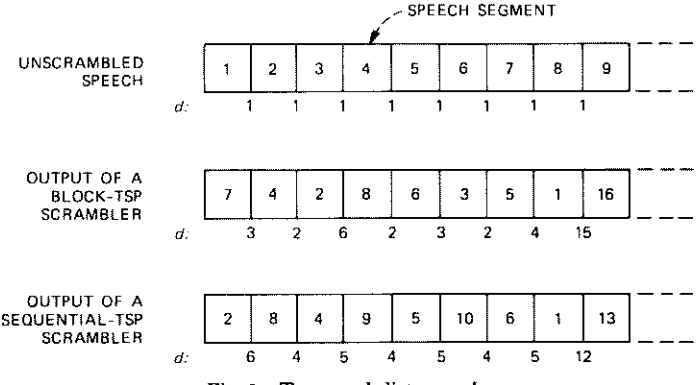



Fig. 1—Temporal distance $d$.

selection involves a constant number of candidates equal to the maximum number of segments that can be stored in scrambler memory.

### 2.2 Block TSP

Figure 2(a) defines the block approach to segment scrambling. In this approach, the scrambler memory consists of a block of $b'$ contiguous segments; one can identify in the scrambler output a corresponding block whose member segments are the same as the segments that comprise the input block. A succeeding block (block 2 in Fig. 2a), also comprising $b'$ segments, enters the scrambler memory after all the $b'$ segments of the preceding block (block 1 in Fig. 2a) have been processed and transmitted. Table Ia depicts a realization of the random processes in block TSP for the example of $b' = 8$ (Fig. 1). Shown are the successive contents of scrambler memory, the sequence of transmitted segments, and the temporal distance $d$ between adjacent segments in the scrambler output.

Note that transmitted segment $s$ and temporal distance $d$ are both random variables. In a practical implementation, the variable $s$ will be pseudorandom so that the intended receiver can invert the scrambler operation. What is significant is that the maximum value of $d$ in the table is 15. This is indeed a global maximum for a block TSP scrambler with memory $b' = 8$. This maximum separation is attained when the random permutation is such that the *last* segment of block $n$ [segment 16 from block $n = 2$ in Table Ia immediately follows the *first* segment of block $n - 1$ (segment 1 from block $n = 1$ in the table)]; and in general,

$$\max(d) = 2b' - 1 \text{ (in segments)}. \tag{2}$$

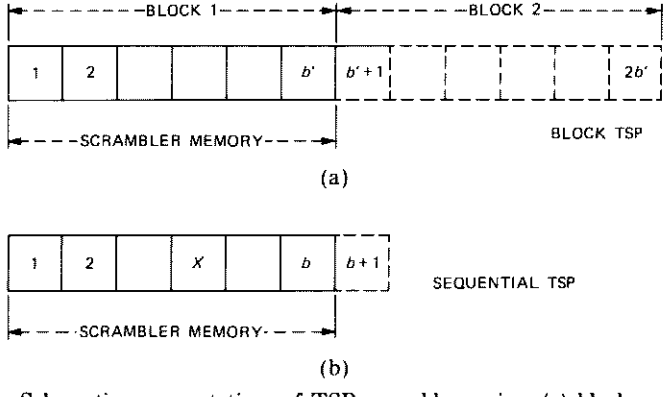



Fig. 2—Schematic representations of TSP scramblers using: (a) block and (b) sequential modes. In (a), an entire block of segments is transmitted before proceeding to the next block. Segment selection involves a number of candidates that decreases from $b'$ to 1, as a given block is processed. In the sequential case (b), each stage of segment selection involves a constant number of candidates equal to the scrambler memory $b$.

Table I—Illustration of two TSP algorithms where **m** is the contents of scrambler memory, $s$ is the transmitted segment, and $d$ is the temporal distance to previous transmitted segment

| (a) Block TSP ($b' = 8$) | | | | | | | | | | (b) Sequential TSP ($b = 8, t = 16$) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **m** | | | | | | | | $s$ | $d$ | **m** | | | | | | | | $s$ | $d$ |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 7 | — | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 2 | — |
| 1 | 2 | 3 | 4 | 5 | 6 | 8 | | 4 | 3 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 8 | 6 |
| 1 | 2 | 3 | 5 | 6 | 8 | | | 2 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 9 | 10 | 4 | 4 |
| 1 | 3 | 5 | 6 | 8 | | | | 8 | 6 | 1 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 9 | 5 |
| 1 | 3 | 5 | 6 | | | | | 6 | 2 | 1 | 3 | 5 | 6 | 7 | 10 | 11 | 12 | 5 | 4 |
| 1 | 3 | 5 | | | | | | 3 | 3 | 1 | 3 | 6 | 7 | 10 | 11 | 12 | 13 | 10 | 5 |
| 1 | 5 | | | | | | | 5 | 2 | 1 | 3 | 6 | 7 | 11 | 12 | 13 | 14 | 6 | 4 |
| 1 | | | | | | | | 1 | 4 | 1 | 3 | 7 | 11 | 12 | 13 | 14 | 15 | 1 | 5 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 16 | 15 | 3 | 7 | 11 | 12 | 13 | 14 | 15 | 16 | 13 | 12 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | | 12 | 4 | 3 | 7 | 11 | 12 | 14 | 15 | 16 | 17 | 3 | 10 |
| 9 | 10 | 11 | 13 | 14 | 15 | | | 10 | 2 | 7 | 11 | 12 | 14 | 15 | 16 | 17 | 18 | 18 | 15 |
| 9 | 11 | 13 | 14 | 15 | | | | 9 | 1 | 7 | 11 | 12 | 14 | 15 | 16 | 17 | 19 | 14 | 4 |
| . | . | . | | | | | | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | | | | | | . | . | . | . | . | . | . | . | . | . | . | . |

The communication delay $C$ in block TSP scrambling is the sum of two delays: ($i$) a delay of $b' - 1$ at the transmitter (the additional time for completion of the block subsequent to the arrival of segment 1), followed by ($ii$) an additional delay of $b'$ at the receiver (the maximum time for which the descrambler may have to wait before it has access to the permuted segment 1). As a result,

$$C = 2b' - 1 \text{ (in segments)}. \tag{3}$$

### *2.3 Sequential TSP*

Figure 2(b) defines a sequential approach to scrambling. In this approach, processing proceeds one segment at a time rather than one block at a time. The permutation process is constrained by two parameters: the scrambler memory $b$, and the maximum time $t$ (in multiples of segment duration) that a segment is allowed to stay in scrambler memory. When the first segment [such as $x$ in Fig. 2(b)] is pseudorandomly selected for transmission and released, the contents of the scrambler memory to the right of $x$ are shifted to the left by one unit, and the last position is *immediately* occupied by segment $b' + 1$. The next transmission is based on another random selection that, unlike in the block approach of Fig. 2(a), can involve the newest segment $b' + 1$, which was not a member of the original block. The above process continues on a segment-by-segment basis, with one constraint mentioned earlier: if a segment is retained in scrambler memory for $(t - 1)$ units of time, it is unconditionally released for transmission at time $t$, even if this means that the dictates of the random segment selector algorithm should be overridden. An important property of the sequential scrambler is that every segment has an

equal probability of spending the maximum allowed time in scrambler memory. To permit a meaningful comparison with block scrambling, we shall concentrate on the special case of

$$t = 2b \tag{4}$$

in most of the ensuing discussion. With the above specific design for $t$, the total encoding delay will be the same for both block and sequential scramblers for a given scrambler memory. This will be clear from subsequent discussion [see eq. (6)]. The design in (4) is also known to maximize the total number of unique permutations for a given delay and block length.[6] Results with a sequential TFSP scrambler indicate that (4) is also an optimal design from the viewpoint of residual intelligibility (see Section V).

Table Ib depicts a realization of the random process in sequential TSP for the example of $b = 8$ and $t = 16$ (Fig. 1). Shown once again are the successive contents of scrambler memory, the sequence of transmitted segments, and the temporal distance $d$ between adjacent segments in the scrambler output. Note in this example that segment 1 indeed stays in scrambler memory for the maximum of 16 time units. It stays in the extreme left-hand slot of scrambler memory for $t - b = 8$ time units; subsequent segments that succeed in reaching the extreme left slot tend to have maximum allowed staying times less than $t - b$ in that slot. Note also that, as in Table Ia, the temporal distance $d$ is a random variable with a maximum value of 15, equal to the maximum in block TSP. In fact this is a global maximum for the sequential design $b = t/2 = 8$; and in general, as in (2),

$$\max(d) = t - 1 = 2b - 1 \text{ (in segments)}. \tag{5}$$

In block TSP, the maximum separation (2) can be realized only in output segment pairs that involve the first and last segments of adjacent blocks (adjacent output segments 1 and 16 in Table Ia). In sequential TSP the maximum separation (4) can be realized in more general instances (for example, with adjacent output segments 3 and 18 in Table Ib).

The communication delay in sequential TSP is given by

$$C = t - 1 = 2b - 1 \text{ (in segments)}. \tag{6}$$

This is a delay inherent in the scrambling parameter $t$. There is no additional delay at the descrambler because of the absence of a block operation. After an initial waiting time of $t - 1$ segments, the descrambler has a guaranteed access to every consecutive segment needed to reconstitute the original input signal.

Table II provides a summary comparison of block and sequential approaches. The parameter $\min(d)$ in the last row will be explained

Table II—Summary comparison of block and sequential TSP

| | Block TSP | Sequential TSP (special case, $t = 2b$) | Sequential TSP (general case, $t > b$) |
|---|---|---|---|
| Scrambler memory | $b'$ | $b$ | $b$ |
| Total communication delay $C_t = C + 1$ | $2b'$ | $2b$ | $t$ |
| Maximum temporal distance $max(d)$ between adjacent output segments | $2b' - 1$ | $2b - 1$ | $t - 1$ |
| Minimum temporal distance $\min(d)$ that can be specified between adjacent output segments | 1 if $b' < 8$<br>2 if $b' \geq 8$ | $[(b + 1)/2]$<br>$[x]$: greatest integer $< x$ | |

presently. The total communication delay $C_t$ includes a delay of 1 unit that is inherent in buffer read-in and read-out, and hence equals $C + 1$ segments. If the duration of a time segment is $B$ ms, the delay in ms equals $C_tB$. In all of this paper, $B = 16$ ms.

### 2.4 min(d) and $\bar{d}$

The maximum temporal distance between adjacent output segments has been discussed above and shown to be a function of scrambler memory size. This section will provide further characterization of the random variable $d$, in particular, the minimum value of $d$ that can be specified a priori, the probability density function of $d$, and its average value $\bar{d}$.

An important property of sequential TSP—one that is not shared by block TSP—is that the selection of a segment for transmission always involves a constant number of candidate segments; this number is equal to the scrambler memory parameter $b$. A consequence of this property is that it is possible to specify in general a minimum distance $\min(d) > 1$ in the output of the sequential scrambler. The random number generator that dictates output segment selection is simply resampled repeatedly, if needed, until the output has a distance of at least $\min(d)$ from its predecessor in the output sequence. The intended descrambler is assumed to know the "key," or the random number sequence used by the scrambler. The descrambler does not need to know the value of $\min(d)$ implied by that key.

The maximum value of $\min(d)$ that will ensure a legitimate scrambler output depends both on scrambler memory $b$ and the contents thereof. A globally safe value (one that will not cause algorithm "hanging," as explained below) is

$$\min(d) = [(b + 1)/2], \tag{7}$$

where $[x]$ denotes the greatest integer less than $x$. The only occasion when (7) will have to be violated in scrambler operation is when a

segment has spent its full life span $t$ in scrambler memory, and it has to be released unconditionally without regard to its temporal distance from the most recent output.

Table III illustrates how the scrambling algorithm "hangs" when a $\min(d)$ greater than the value in (7) is specified a priori. In this example, $b = 8$, $t = 16$, $\min(d) = 6$, a value that exceeds the limit of 4 suggested by the formula (7).

As indicated in the last row of Table II, the $\min(d)$ values that can be specified a priori are much smaller in block TSP. For example, if $b' = 8$, after a possible output sequence of [3 5 7 4 6 8 2], segment 1 can never be transmitted (immediately after segment 2) unless $\min(d) = 1$. Even if the $\min(d)$ requirement is overridden in the case of the last transmitted segment of a block, the greatest value of $\min(d)$ that will not result in a "hanging" of the scrambler operation is a very slowly increasing function $b$. For example with $b = 8$, the $\min(d)$ value that can be specified a priori is no greater than 2. This should be compared with the value of 4 for sequential TSP with $b = 8$.

In both block and sequential TSP schemes, a priori insistence on a $\min(d)$ value greater than unity has attendant penalties in the cryptanalytical strength of the scrambler.[6] This is due of course to the fact that with $\min(d) > 1$, fewer random segment permutations are legal, in comparison with the totally random situation that obtains with $\min(d) = 1$.

An interesting property of a sequential scrambler with $\min(d) = 1$ and the constraint $t = 2b$ is that the fraction of segments that spends the maximum allowable time in scrambler memory is very nearly 20 percent for values of $b > 4$. An analytical demonstration of this property appears in the appendix.

Figure 3 shows histograms of the random variable $d$ in block TSP with $b' = 8$ and sequential TSP with $b = t/2 = 8$, and two values of $\min(d)$ in each case. The results are from a simulation involving a total of 132 segments (about 4 seconds of speech, with 32 ms segments). Note that, in general, the probability of $d$-values *less than* $\min(d)$ is very small. The probability is non-zero, however, because of occasional situations where a segment has spent the maximum lifespan of $t = 2b$

Table III—Illustration of an unrealizable $min(d) = 6$ in sequential TSP with $b = t/2 = 8$

| Contents of Scrambler Memory m | | | | | | | | Transmitted Segment $s$ | Temporal Distance $d$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 2 | — |
| 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 8 | 6 |
| 1 | 3 | 4 | 5 | 6 | 7 | 9 | 10 | 1 | 7 |
| 3 | 4 | 5 | 6 | 7 | 9 | 10 | 11 | 7 | 6 |
| 3 | 4 | 5 | 6 | 9 | 10 | 11 | 12 | | ≥6 |

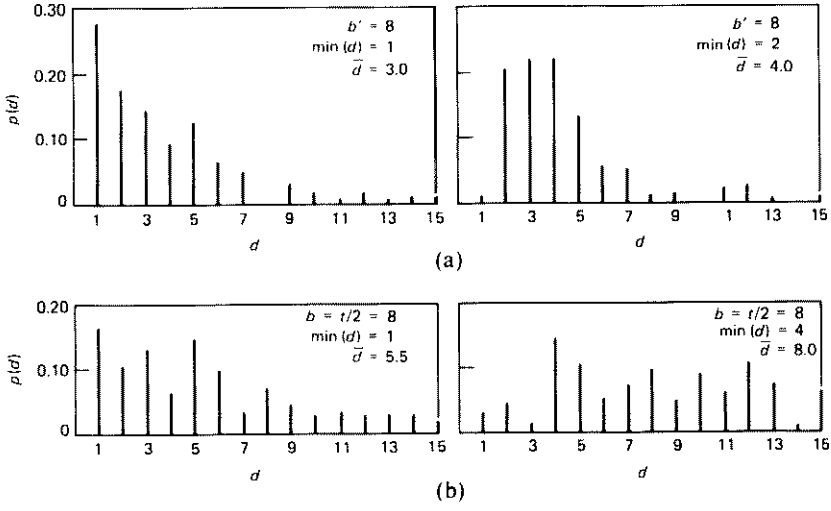



Fig. 3—Histograms of temporal distance $d$ in (a) block TSP ($b' = 8$) and (b) sequential TSP ($b = 8$, $t = 16$). The non-zero probability of $d < \min(d)$ is related to violations of $\min(d)$ because of unconditional releases of segments that have spent a full life-span = 16 in scrambler memory.

times in scrambler memory, and it becomes necessary to release it unconditionally without regard to the resulting value of $d$. Figure 3 also notes respective values of average separation $\bar{d}$. Note that this depends on scrambler type as well as on min $d$. It increases with $\min(d)$ for both types of scramblers, and it is greater for sequential TSP than for block TSP, for the case of $\min(d) = 1$.

Figure 4 compares $\bar{d}$ values for block and sequential TSP scramblers as a function of scrambler memory. Results for the sequential system correspond to the special case of $t = 2b$. Note that $\bar{d} \sim b$ in this case, one-half of $\max(d) = 2b$. The faster increase of $\bar{d}$ in sequential TSP is a result of the greater values of $\min(d)$ that can be specified in this system. Recall from Tables I and II that $\max(d)$ is the same for block and sequential systems for a given value of scrambler memory.

## III. RESIDUAL INTELLIGIBILITY IN BLOCK AND SEQUENTIAL TSP SCRAMBLERS

The residual digit intelligibility in a block TSP system has been the subject of a recent comprehensive article.[1] The emphasis in this section is on the digit intelligibility performance of sequential TSP, as evaluated in formal listening tests.

### *3.1 Test conditions*

The duration of speech segments was $B = 16$ ms for all schemes. This is a design that provides a useful compromise between the

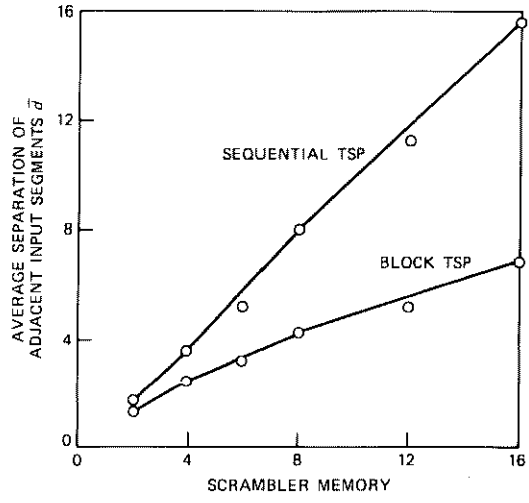



Fig. 4—Average temporal distance $\bar{d}$ as a function of scrambler memory in (*i*) block TSP and (*ii*) sequential TSP with $t = 2b$. Minimum specified distances are the min($d$) values in Table II.

conflicting requirements of encoding delay and bandwidth expansion.[1] Total communication delay ($C_t$) and minimal separation min($d$) were variables in the test. The three delay conditions that were tested were 256, 384, and 512 ms. The min($d$) values that were combined with each delay condition were the smallest and largest values that could be successfully assigned for each type of scrambler. Of course, the smallest value of min($d$) in all cases is 1, which would, at most, allow adjacent segments to be in reverse order when transmitted. However, the maximum value that min($d$) can assume depends upon the type of scrambler and the length of the memory buffer, as shown in Table II. The three delay conditions at each of two min($d$) values generated six test conditions for each type of scrambler, block and sequential.

One other condition was also included with the sequential scrambler using a delay of 512 ms. This included a $\mu$-law logarithmic compression at the scrambler output, with $\mu = 100$.

Before making the test tapes, we listened to recordings of four-digit numbers (as described in Section 3.2 below) as processed by each of the test conditions. Two sets of recordings were employed, one from a male talker and one from a female talker. The recorded speech bandwidth was 200 to 3200 Hz in each case. The consensus of opinion in this pilot listening session was that the intelligibility was higher for the recordings of the female talker, presumably because her speaking rate was slower. The average duration of the four-digit utterances was 2.34s for the female talker, as compared to 1.99s for the male talker. Recordings from both talkers were included for the final test.

The speech samples were 50 four-digit numbers such as 3860, spoken as: *three-eight-six-zero*. Each talker recorded a different list of fifty numbers. The list of numbers was balanced so that within every set of ten numbers, each of the ten digits occurred four times, and each occurrence of a given digit was at a different position in a number. Although the same list of 50 numbers was used for all the tests with a given talker, each subset of ten numbers was presented in a different random order in each test.

The use of digits rather than continuous speech as test inputs follows the procedure in previous tests.[1] Conversational speech has redundancies that make the task of an analog scrambler more difficult; sentence intelligibility, for example, would be higher than word intelligibility as a result of these redundancies. There are no such redundancies in the digit strings in our tests, these strings being sequences of randomly chosen digits. But still, our experience with analog scramblers indicates that digit intelligibility, measured as discussed, is a fairly critical test of scrambler performance. The fact that there is a limited stimulus vocabulary (of ten) makes the task of the analog scrambler quite difficult, perhaps more so than in the case of a continuous speech input, which has a much larger, albeit redundant vocabulary. In the context of scrambled speech,[3] as well as in the context of speech corrupted by additive noise,[7] there is clear evidence that digit intelligibility scores tend to be much higher than word intelligibility scores.

### 3.2 Test procedure

The subjects were employees of Bell Laboratories at Murray Hill, New Jersey. They were not formally trained listeners of scrambled speech. Each scrambler scheme was judged by 18 subjects, although each subject judged only six schemes, chosen to represent the range of expected intelligibility. The subjects listened to the recordings through earphones while seated in a sound-treated booth. They were told to listen to each number and write the four digits they heard on their answer sheets. They were also told that some of the numbers would be difficult to understand and, if they were uncertain, they were to write their best guess rather than to leave blanks.

### 3.3 Results

For each scrambler type, the mean and standard deviation of the percent-correct identification of the digits was computed for each digit position of the four digit numbers. These values confirmed two observations that we had made in the pilot test mentioned earlier.

The mean intelligibility scores confirmed our observation about the effect of the slower speaking rate (by the female talker) on TSP performance. The effect was most apparent in the scores for the two

voices with the sequential scrambler and $C_t = 512$ ms. While the mean intelligibility scores for the two middle digits of this condition were only 66 and 61 percent with the male talker, the same scores were 84 and 91 percent for the female talker.

In the preliminary listening session, we had also noticed that the first and last digits of the four-digit numbers seemed to be easier to identify than the two in the middle. To see whether this was also true in the test data, the percent-correct identification was computed for each digit position. The scores illustrated quite clearly that the digit position affects the residual intelligibility. At each time delay, the percent correct for the third-digit position has the lowest value and the fourth digit position the highest value. Digit positions one and four have less context than positions two and three in the sense that digit position one has no left neighbor and digit position four has no right neighbor. The consistently lower scores of the third digit position are very likely due to the strong influence of context. Indeed, the scores at this digit position are probably a better indication of the level of intelligibility that could be expected in continuous speech where pauses are less frequent.

Intelligibility scores showed that there is no general advantage in using $\min(d)$ values greater than one in sequential TSP. The result is surprising in view of the effect of $\min(d)$ on the average temporal separation $\bar{d}$ (see Fig. 3). One possible explanation of the result is the presence of some kind of a threshold effect in the perception of temporally scrambled speech.

The upper right-hand corner of Fig. 5 shows residual intelligibility in sequential TSP scrambling as a function of communication delay, for $\min(d) = 1$, and for input conditions most favorable to the scrambler—digit position three and male talker. The lower edge of the cross-hatched region refers to the same most favorable case, while the upper edge refers to the average, over both male and female talkers, and over all four digit positions.

Intelligibility scores for block TSP were significantly different from those of sequential TSP only in the case of a communication delay equal to 512 ms. The dashed-line block TSP characteristic in Fig. 5 refers to the most favorable case of male speaker and digit position three, and to the (only available) example of $\min(d) = 2$.

The results of Fig. 5 indicate that there is no intelligibility advantage in sequential TSP as compared with block TSP at low values of delay $C_t$. When $C_t$ is increased to 512 ms, the sequential approach produces a significant reduction of about 10 percent over the block approach. The condition involving $\mu$-law compression of speech reduces the residual intelligibility even more, as shown by the point marked $\mu = 100$. The refinement of $\mu$-law compression is simple to implement,

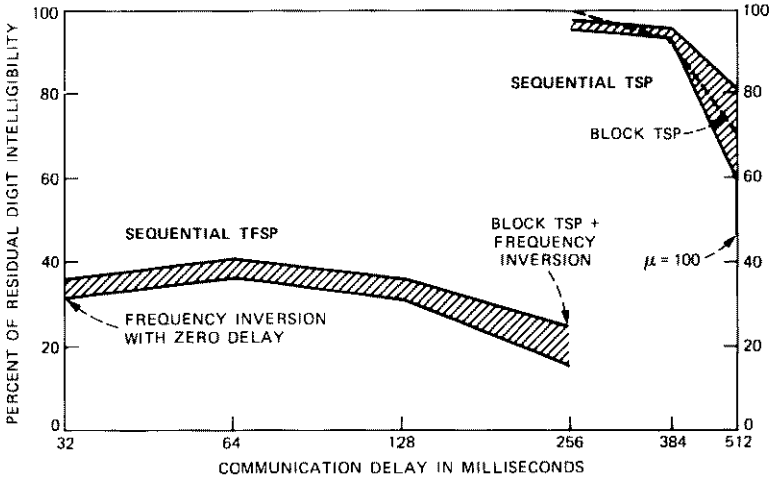



Fig. 5—Mean intelligibility of digits as a function of delay for sequential TSP sequential TFSP scramblers. Upper curves refer to averages over all four digit positions and over both male and female speakers. Lower curves refer to scores for the third digit position for the male speaker. The dashed lines refer to block TSP, digit position three, and male speaker.

although a penalty paid for reduced intelligibility in this case is increased bandwidth expansion and real-channel sensitivity, as compared with the conventional case of $\mu = 0$.

### *3.4 Discussion*

It is interesting that sequential TSP, in spite of its better segment separation properties, provides an insignificant reduction of intelligibility as compared with block TSP, at low values of delay $C_t$. However, this result can be reconciled with the objective results in Fig. 4, which show that the increase of average speech segment separation owing to sequential scrambling is an increasing function of scrambler memory. In the "one-dimensional" procedure of time segment permutation, the only way of providing greater scrambling memory is by increasing delay and, apparently, the objective separation gain becomes perceptually significant only at $C_t$ values in the order of 512 ms.

In the task of identifying one of ten possible digits, the lowest meaningful intelligibility score is 10 percent, corresponding to purely random guessing. This lower bound is particularly meaningful if listeners do not use complex cues and indeed perform decision tasks with ten alternatives. Clearly, none of the TSP scramblers in this study approaches a score in the order of 10 percent. This reinforces our earlier stand[1] that time permutation is best used in conjunction with frequency manipulations such as frequency inversion or frequency band permutations to provide practical and useful values of residual

intelligibility. The much lower residual intelligibility of TSP with frequency inversion is illustrated by the 25-percent result in Fig. 5, for the case of block TSP and $C_t$ = 256 ms, a condition tested in earlier work.[1] In that study, frequency inversion alone provided a residual intelligibility of 30 percent. Unfortunately, however, frequency-inverted speech has identifiable characteristics that can be learned,[8] and frequency inversion is also very easy to undo. The next section describes another "two-dimensional" procedure for analog scrambling; it employs sequential permutations of a time-frequency speech matrix using sub-band partitions of 16 ms time segments. This two-dimensional procedure also realizes a residual intelligibility on the order of 15 to 25 percent. In addition, it has better cryptanalytical properties than TSP with frequency inversion.

## IV. PERMUTATIONS WITH A TIME-FREQUENCY MATRIX: TIME-FREQUENCY SEGMENT PERMUTATION (TFSP)

In this section, we propose a scrambler that provides a simultaneous and fully two-dimensional manipulation of both time and frequency information in speech. The permutations of time-frequency segments are based on the sequential permutation approach described in Section 2.3.

The basic principle of the proposed scrambler can be explained with reference to Fig. 6. The ($f \times b$) matrix depicts a total of $fb$ time-frequency segments. These belong to $b$ contiguous time segments of speech, each of which is split into $f$ contiguous frequency sub-bands or segments. The scrambler memory is considered to be equal to the product $fb$. For subsequent discussions, the contents of this memory can be considered to be a one-dimensional array of $fb$ time-frequency segments. A random number algorithm picks one of these $fb$ segments for transmission. When this $p$-th segment is transmitted, the contents of all $q$-th cells ($q > p$) are promoted by one position in the memory, and the $fb$-th cell is filled by an incoming ($fb + 1$)-th time-frequency

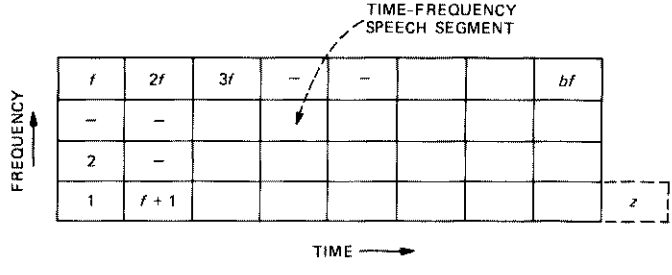



Fig. 6—Sequential permutations of a time-frequency speech matrix: sequential TFSP. Time-frequency segment $z$ enters scrambler memory as soon as $p$-th segment in memory ($1 < p \leq bf$) is randomly selected and transmitted.

segment. This random scrambling procedure is repeated with two constraints previously described in the discussion of sequential TSP (see Section II):

(*i*) No time-frequency segment stays in the scrambler memory matrix for a number of stages greater than $t_f = 2bf$. This corresponds to the special case $t = 2b$ in the one-dimensional case [Section 2.3; eq. (4) and Table II]. It implies a total communication delay of $C_t = 2fb$ segments. However, since $f$ successive frequency segments are combined to produce one time segment, the total communication delay is independent of $f$, and given as in the one-dimensional case, by

$$C_t \text{ (in ms)} = 2bB,$$

where $B$ is the duration (in ms) of a time segment. As in Sections II and III, $B = 16$ ms.

(*ii*) Contiguous time-frequency segments in scrambler output can be arranged to have a separation which, with a high probability, exceeds $\min(d_f)$; $1 < min(d_f) \leq fb/2$. The parameter $\min(d_f)$ provides a trade-off between average segment separation in scrambler output, which is an increasing function of $\min(d_f)$, and total number of possible permutations, which is a decreasing function of $\min(d_f)$.

Every set of $f$ successive outputs from the scrambler matrix is reconstituted into a pseudo-speech time segment for transmission. One-dimensional sequential TSP is the special case of $f = 1$ in Fig. 6.

## V. RESIDUAL INTELLIGIBILITY IN A SEQUENTIAL TFSP SCRAMBLER

Residual intelligibility tests were conducted following the same general procedures used for the block and sequential TSP tests (see Sections 3.1 and 3.2). The same original recordings of four-digit numbers spoken by a male and female talker were used as speech inputs. Eighteen subjects listened to the recorded digits as processed by each test scheme.

### *5.1 Test conditions*

A total of seven conditions were tested, with the simple design of $\min(d_f) = 1$ in each case. The first four conditions correspond to the $t_f = 2bf$ design, with $f = 4$ and $b = 1$, 2, 4, and 8. Corresponding memory sizes are 4, 8, 16, and 32 time-frequency segments. Corresponding communication delays are $2bB = 32b = 32$, 64, 128, and 256 ms. The other three conditions tested used $b = 8$ also, but values of $t_f \neq 2bf$; specifically, $t_f = 1.5bf$, $3bf$, and $4bf$. In other words, the memory size is fixed at 32 segments for these three cases, but the maximum age in memory varies from 1.5 to 4 times the memory length. The communication delays corresponding to these three conditions are 192, 384, and 512 ms.

### 5.2 Results

The mean and standard deviation of the percent-correct identification was computed for each digit position of each test condition. In general, the subject variability was not large. The average standard deviation was about 6 percent for the male voice and slightly higher, 9 percent, for the female voice. However, there was one subject whose scores were consistently very high, 1.5 to 2.25 standard deviations above the mean. (On one test he correctly identified 72 percent of the digits in the third position while the average for the remaining subjects was 37 percent.) When questioned, he claimed that he did not have any special strategy. However, a closer examination of his data showed a pattern that was evident to some extent in the scores of other better-than-average listeners. The data suggest that these listeners would focus their attention on one of the four digit positions, even when the pauses between digits were difficult to detect.

Since the four digit numbers were balanced and presented in groups of ten, their scores at each digit position for each group could be compared. For instance, for one type of scrambler, the unusual subject mentioned above had scores of 0.20, 0.40, 0.10, and 0.60 for the four digit positions of one group of ten numbers. For the next group of ten numbers, his scores were 0.60, 0.50, 0.30, and 0.10, suggesting that he had shifted his attention to the first two digit positions while listening to the second group of ten numbers. Because of the difference owing to the speaking rate of the two talkers, the mean intelligibility (across subjects and digits) was compared for the two talkers. Even though the differences in the average scores were small—on the order of 5 to 8 percent—the scores for the female voice were consistently higher and, in all cases but one, the difference was statistically significant.

In general, the scores for different digit positions did not display the context effect mentioned for one-dimensional TSP. On the whole, about 90 to 95 percent of the subjects' scores were not significantly different for different digit positions. The remaining 5 to 10 percent of the subjects, who had significantly different scores owing to digit position, were generally the listeners whose overall scores were higher than the average. Although the effects of talker and digit position were not as strong as in the TSP experiment, to be consistent with those results, the scores of the third digit position with the male voice were again evaluated separately, and considered as the closest approximation to scores with continuous speech. These values are indicated by the lower edge of the cross-hatched TFSP region in the lower part of Fig. 5. The upper edge of this region refers, once again, to averages over both male and female talkers, and over all four digit positions.

Four observations are worth noting in the TFSP characteristic of Fig. 5: (*i*) the intelligibility with the smallest communication delay (32

ms) is close to that in frequency inversion (which has a zero communication delay); (*ii*) the intelligibility with a delay of 256 ms is close to that in block TSP (with the same delay) plus frequency inversion; (*iii*) there is a significant drop in intelligibility when the delay exceeds 128 ms (corresponding to a 4 × 4 time-frequency matrix in Fig. 6), and, finally, (*iv*) the intelligibility with a 256-ms delay is significantly higher than the expected lower bound of 10 percent; the characteristic, however, shows a tendency to drop further at delays greater than 256 ms, and with the suggested design of $t_f = 2bf$.

The differences among the scores with the design $t_f = 2bf$ and delays less than 256 ms are not statistically significant, but these scores are all significantly different from the score of 15 percent at a delay of 256 ms. For the case of $b = 8$, the scores at the other three values of $t_f$ ($1.5bf$, $3bf$, and $4bf$) were also not significantly different from each other, but they were all significantly higher than the score when $t_f = 2bf$ (the 256-ms point in Fig. 5). This result suggests that the maximum staying time of $2bf$ may represent an optimal design that minimizes identification. This result is very interesting because the last two of the three $t_f$ conditions above involve communication delays that are 50 and 100 percent greater than the delay in the $t_f = 2bf$ design.

A significant property of all the analog scramblers in this paper is that intelligibility is digit-dependent. This is shown by the illustrative confusion matrices of Table IV. As seen from the diagonal terms in these matrices, digits six and five are the most difficult to scramble. The very high residual intelligibilities for these digits expose what may be an inherent limitation of analog scramblers, at least those based on permutations, as opposed to digital scramblers that transform any given input to an output that sounds like white noise. The fact that the output of the analog scramblers discussed is not the white-noise type is well illustrated by the spectrograms of Fig. 7. Because of the residual structure in these spectrograms, they can be used as the starting point for non-real-time descrambling by a trained eavesdropper. This is especially the case with the TSP spectrogram of Fig. 7b.

The matrices in Table IV also show that the male speaker was easier to scramble than the female speaker. As stated earlier, we feel that this is due to the slower speaking of the female speaker.

### *5.3 Interpretation of digit-intelligibility scores*

An important consideration in interpreting the results of the tests described in this paper is the use of spoken digits as speech input. The lower bound of 10 percent is particularly meaningful if the information available to the listener is limited only to the possibility of the ten digits. Actually, the subjects are trying to recognize phonemes and the phonemes of the spoken words for each digit are not a balanced sample

Table IV—Digit confusion matrices in a TFSP scrambler with communication delay = 256 ms (results are averages over 18 listeners)

| | | LISTENERS' RESPONSE | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| | 0 | *0.30* | 0.08 | 0.05 | 0.05 | 0.10 | 0.09 | 0.05 | 0.11 | 0.05 | 0.11 | |
| S | 1 | 0.09 | *0.17* | 0.05 | 0.05 | 0.06 | 0.10 | 0.03 | 0.10 | 0.07 | 0.27 | |
| C | 2 | 0.14 | 0.05 | *0.18* | 0.19 | 0.02 | 0.07 | 0.18 | 0.07 | 0.05 | 0.05 | |
| R I | 3 | 0.14 | 0.08 | 0.15 | *0.14* | 0.11 | 0.07 | 0.15 | 0.07 | 0.05 | 0.03 | |
| A N | 4 | 0.08 | 0.12 | 0.04 | 0.05 | *0.23* | 0.18 | 0.10 | 0.06 | 0.08 | 0.05 | FEMALE |
| M P | 5 | 0.03 | 0.06 | 0.02 | 0.04 | 0.07 | *0.57* | 0.03 | 0.06 | 0.07 | 0.07 | SPEAKER |
| B U | 6 | 0.03 | 0.05 | 0.03 | 0.04 | 0.08 | 0.09 | *0.53* | 0.09 | 0.04 | 0.03 | |
| L T | 7 | 0.06 | 0.11 | 0.03 | 0.07 | 0.07 | 0.15 | 0.10 | *0.28* | 0.07 | 0.07 | |
| E | 8 | 0.06 | 0.13 | 0.07 | 0.06 | 0.06 | 0.14 | 0.15 | 0.07 | *0.23* | 0.03 | |
| R | 9 | 0.08 | 0.10 | 0.04 | 0.06 | 0.04 | 0.22 | 0.02 | 0.06 | 0.06 | *0.32* | |

| | | LISTENERS' RESPONSE | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| | 0 | *0.24* | 0.12 | 0.08 | 0.05 | 0.08 | 0.11 | 0.09 | 0.08 | 0.09 | 0.06 | |
| S | 1 | 0.08 | *0.17* | 0.06 | 0.08 | 0.09 | 0.19 | 0.06 | 0.09 | 0.09 | 0.09 | |
| C | 2 | 0.22 | 0.10 | *0.13* | 0.08 | 0.10 | 0.05 | 0.18 | 0.08 | 0.04 | 0.03 | |
| R I | 3 | 0.14 | 0.11 | 0.09 | *0.20* | 0.10 | 0.06 | 0.10 | 0.10 | 0.04 | 0.06 | |
| A N | 4 | 0.08 | 0.13 | 0.07 | 0.07 | *0.18* | 0.15 | 0.11 | 0.10 | 0.03 | 0.07 | MALE |
| M P | 5 | 0.09 | 0.11 | 0.02 | 0.03 | 0.06 | *0.40* | 0.04 | 0.11 | 0.08 | 0.08 | SPEAKER |
| B U | 6 | 0.11 | 0.07 | 0.09 | 0.09 | 0.07 | 0.05 | *0.35* | 0.08 | 0.06 | 0.04 | |
| L T | 7 | 0.15 | 0.12 | 0.06 | 0.06 | 0.08 | 0.07 | 0.16 | *0.16* | 0.09 | 0.05 | |
| E | 8 | 0.09 | 0.13 | 0.09 | 0.07 | 0.09 | 0.09 | 0.12 | 0.12 | *0.13* | 0.08 | |
| R | 9 | 0.18 | 0.11 | 0.04 | 0.09 | 0.09 | 0.12 | 0.08 | 0.10 | 0.10 | *0.08* | |

of the English language. For instance, eight is the only spoken digit with an initial vowel, seven and zero are the only words with two syllables, and five of the ten spoken digits have unvoiced fricatives as the initial phoneme (three, four, five, six, seven). Thus, if an astute listener recognized the first phoneme as an unvoiced fricative, then the probability of being correct by guessing is 20 percent rather than 10 percent. If, in addition, the word were recognized as having only one syllable (eliminating seven), the probability would be 25 percent.

In the first experiment of this series (Section III), the unusually high scores for the spoken digit "six" were observed, but a more detailed analysis was not done. In the analysis of the TFSP data, the scores for each of the spoken digits were computed and compared as shown in Fig. 8. The mean intelligibility of each spoken digit (with digit position disregarded) is indicated for both the male ($M$) and female talker ($F$), for scramblers with $t_f = 2fb$ and $b = 1$, 2, 4, and 8. Labels 1, 2, 4, and 8 refer to these values of $b$; they also indicate respective communication delays of 32, 64, 128, and 256 ms. The generally higher scores for the female talker are apparent. More important, these plots indicate that some of the linguistic cues were affecting the listeners' judgments. The scores for the two-syllable words, zero and seven, are elevated and the scores for five and six are extremely high. The range of scores indicate

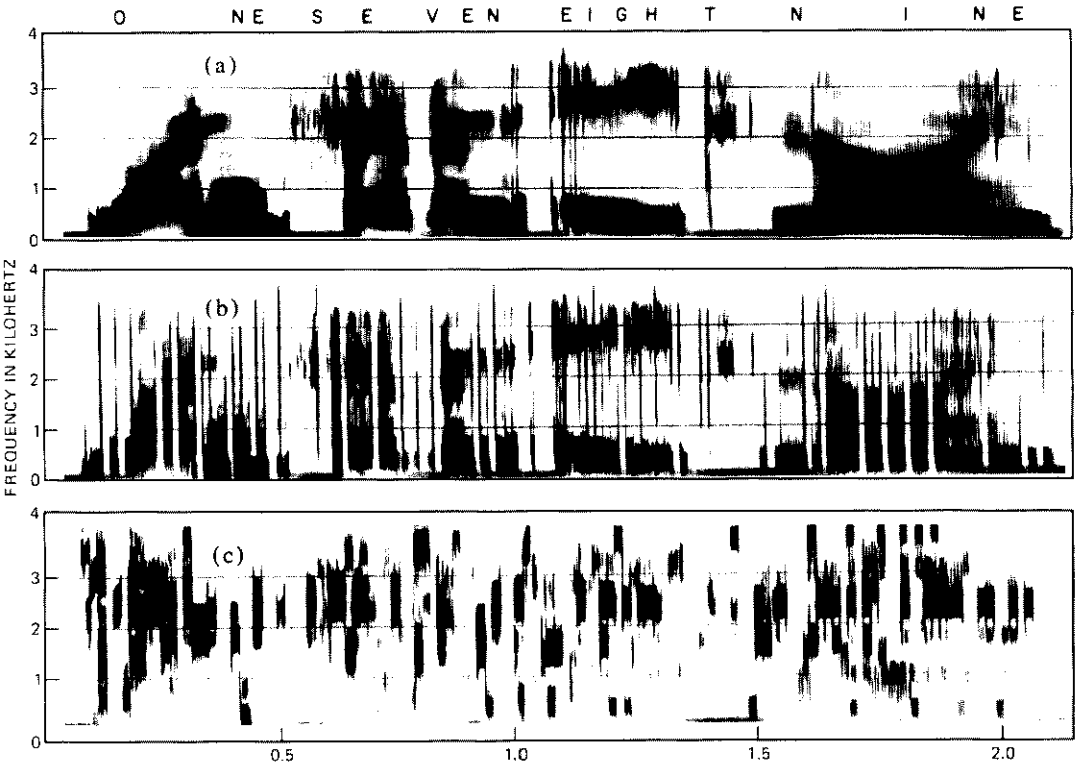



Fig. 7—Speech spectrograms: (*i*) unscrambled speech (female speaker, digit sequence "1789"), (*ii*) output of sequential TSP scrambler with communication delay of 128 ms, and (*iii*) output of sequential TFSP scrambler with the same communication delay of 128 ms.

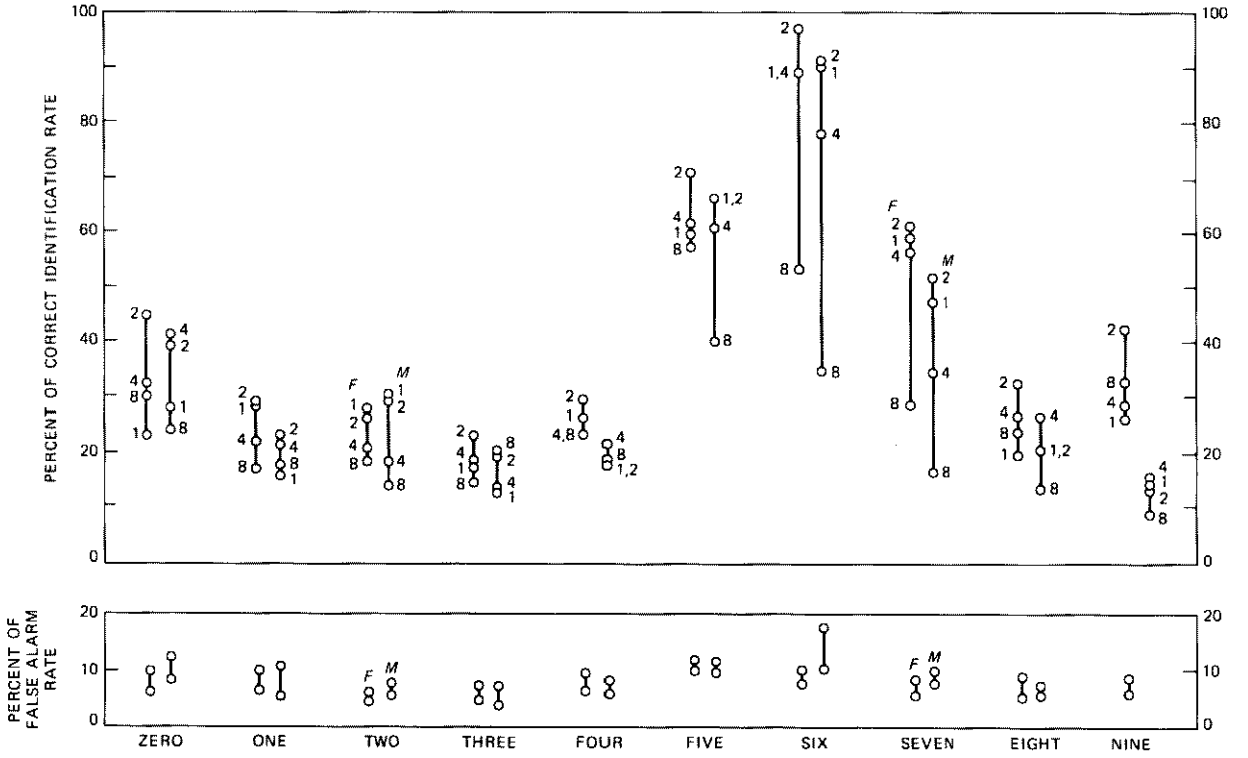



Fig. 8—Correct identification and false alarm rate scores as a function of spoken digit. For each digit, there are two sets of scores, one for female speaker (left vertical bar) and one for male speaker (right vertical bar). These are distinguished by labels *F* and *M* for the digit examples TWO and SEVEN. Each set of scores has four values, corresponding to four values of total communication delay. These four values are identified by numbers that represent multiples of 32 ms. Numbers 1, 2, 4, and 8 therefore indicate delays of 32, 64, 128, and 256 ms.

that the reduced intelligibility with a delay of 256 ms is largely the result of disproportionately lowering the identification of "five," "six," and "seven."

The lower plot in Fig. 8 shows the false alarm rate, i.e., the percent of time that a digit was incorrectly identified as the one labelled in the figure. For simplicity, these plots show only the maximum and minimum values of false alarm rates, as a function of $b$, rather than values for all four values of ($b$), as in the upper plot of the figure. If subjects were really guessing among the ten possible digits when they were uncertain, these values should be about 10 percent. The somewhat higher false alarm rate for the digit six indicates that it was probably being used as a default response by some of the subjects.

The observations of this section reinforce our earlier stand that the intelligibility of connected speech where the words are less predictable may, indeed, be much lower than the scores shown for the spoken digits in Fig. 5.

In Fig. 8, the condition of 256-ms delay (labeled 8 in the figure) represents the least intelligibility in only 14 out of 20 conditions shown, and the condition of 32-ms delay (labeled 1 in the figure) represents the greatest intelligibility in only 3 out of 20 conditions. A good example of monotonic behavior is the digit "two." Good examples of non-monotonic behavior are the female utterances of "zero" and "nine." In the latter two examples, the intelligibility difference between delays of 32 and 64 ms (points labeled 1 and 2) were in fact tested to be statistically significant. We do not have an adequate explanation of why an increase of delay from 32 to 64 ms (and in some cases, to 128 and 256 ms) should actually *increase* residual intelligibility, a feature that is counter to the general trends of the average scores in Fig. 5. (Recall that in these averages, the differences between scores at 32, 64, and 128 ms were stated to be statistically insignificant). The phenomenon of significant intelligibility increases owing to delay seems however to be peculiar to digits with sustained sounds such as the *o* in zero and the *n* in nine. In these cases, certain increases of communication delay, or equivalently, certain increases of staying time in memory, may increase the probability that at least one of many fragments of the long sustained sound gets outputted in an "interference-free" context such as an interdigit silence, causing an increase of intelligibility. On the average, however, separation of intra-digit fragments *increases* as a function of delay, and this causes a *decrease* of intelligibility. This was indeed noted in the average scores of Fig. 5.

## REFERENCES

1. N. S. Jayant, B. J. McDermott, S. W. Christensen, and A. M. S. Quinn, "A Comparison of Four Methods for Analog Speech Privacy," IEEE Trans. Commun., *COM-29* (January 1981), pp. 18–23.

2. R. C. French, "Speech scrambling and synchronization," Philips Res. Rep., No. 9 (1973), pp. 1–115.
3. E. R. Brunner, "Efficient scrambling techniques for speech signals," Proc. Int. Conf. Commun., Seattle, WA, June 1980, pp. 16.1.1–6.
4. S. Udalov, "Microprocessor-based techniques for narrow-band scrambling," in Proc. Int. Conf. Commun., Seattle, WA, June 1980, pp. 16.4.1–5.
5. R. V. Cox and J. M. Tribolet, "Analog Voice Privacy Systems Using TFSP Scrambling: Full Duplex and Half Duplex," B.S.T.J., this issue.
6. S. T. Hong and W. Kuebler, "An Analysis of Time Segment Permutation Methods in Analog Voice Privacy Systems", Proc. 1981 Carnahan Conference on Crime Countermeasures, Univ. of Kentucky, Lexington, KY, May 1981.
7. G. A. Miller, G. A. Heise, and W. Lichten, "The intelligibility of speech as a function of the context of test materials," J. Exp. Psychol., *41*, (1951), pp. 329–35.
8. B. Blesser, "Speech perception under conditions of spectral transformation: I. Phonetic characteristics," J. Speech and Hearing, *15*, (1972), pp. 5–41.

## APPENDIX

Let the probability of "forced-exit" (i.e., the probability that a segment spends the maximum allowable time in scrambler memory) be $P$. The probability that a segment exits as a result of random selection is therefore $1 - P$. With a buffer size $b$, and a uniform pdf for random segment selections in buffer positions 1 to $b$, the probability of unforced exit from any given buffer stage is therefore $(1 - P)/b$, and the corresponding probability of non-exit is $P_n = [1 - (1 - P)/b]$. With the $t = 2b$ design, the probability of non-exit for all possible ages $s$, $s = 1, 2, \cdots, 2b$ of a given segment is $(P_n)^{2b}$. By definition, this should equal the forced-exit probability $P$. Therefore,

$$P = P_n^{2b} = \left(1 - \frac{1 - P}{b}\right)^{2b}. \tag{8}$$

Taking logarithms and using $\ln(1 - x) \sim -x$ for $x \ll 1$,

$$\ln P = 2(P - 1); \qquad P \sim 0.203$$

for large $b$. Numerical solution of (8) shows that $P$ is extremely close to the above asymptotic value of 20 percent for values of $b > 4$. The value at $b = 4$ is about 0.210.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Analog Voice Privacy Systems Using TFSP Scrambling: Full Duplex and Half Duplex

By R. V. COX and J. M. TRIBOLET *

(Manuscript received July 20, 1982)

*In this paper we present possible full-duplex and half-duplex analog voice privacy systems that have been simulated over real channels. Previous papers have been concerned primarily with the issues of the strength of a system (i.e., unintelligibility to the casual eavesdropper and relative cryptanalytical strength for the sophisticated eavesdropper) and the amount of delay of a system. Well-known but not addressed have been the problems of decoding the scrambled signal in a real-channel environment. At the heart of the encryption systems proposed here is the sequential time and frequency segment permutation structure proposed by Jayant and Cox. This structure relies on digital processing to divide the signal into sub-bands and then to permute these bands in both time and frequency simultaneously to synthesize the scrambled analog signal. In discussing the decoding we address the issues of compensating for the properties of the channel, re-sampling the analog signal, and establishing and maintaining synchronization between the de-scrambler and the scrambler.*

## I. INTRODUCTION

In this paper we discuss a possible analog voice encryption scheme. Although the channel signal is analog, all of the signal processing is done digitally. This gives us a flexibility that analog processing cannot give, and thereby allows us to do simultaneous time and frequency permutation. It also allows us to perform synchronization and equalization at the receiver. With the advent of the Digital Signal Processor (DSP),[1] digital processing becomes economically competitive as well as feasible.

* This work was performed while Mr Tribolet was on sabbatical leave from the Instituto Superior Tecnico, Lisbon, Portugal.

The systems that we will discuss in this paper are a full-duplex system and a half-duplex system. We have assumed either system would be used over all standard telephone channels with a bandwidth of 200 to 3500 Hz. Because of the delays involved in the signal processing, echoes could become a problem, so we chose to look at solutions that would avoid the echo problem. Alternatively, echoes might be cancelled using an appropriate algorithm, but this problem was not addressed in this study. One such solution to the echo problem is to choose a half-duplex system. Alternatively, for a full-duplex system we have divided the telephone band into three segments: 200 to 500 Hz is used for signalling and equalization by both users, 500 to 2000 Hz is allocated to the first user, and 2000 to 3500 Hz is allocated to the second user. In this way faulty echo cancellation need not impair the quality of the speech, since the two speakers use disjoint bands. In a half-duplex system only one user could speak at a time and would be allocated the entire channel except for the frequencies used for equalization.

In this study we have relied almost entirely on computer simulation of the algorithms involved. However, the algorithms were designed with future implementation on the Bell Laboratories DSP in mind.[1] The DSP is a powerful, single-chip, programmable processor that is especially suited for performing digital signal processing functions. It has an 800-ns machine cycle that is established by a 5-MHz clock. It contains provision for a 1024- × 16-bit read-only memory (ROM) for storage of the program, tables, and coefficients. A 128- × 20-bit random-access memory (RAM) is available for the storage of dynamic data and state variables. Our experience in working with the DSP has shown that the three constraints of RAM, ROM, and execution speed all come into play in the final design of the algorithm.

In Section II we give an overall description of the system. Subsequent sections deal with the major components of the system and the problems we addressed and solved for these components. Section VIII presents conclusions.

## II. THE OVERALL SYSTEM

Figures 1a and 1b are overall block diagrams for the proposed full-duplex system's transmitter and receiver. At the transmitter an analog speech signal $s(t)$ is first filtered and sampled at 8 kHz by a standard analog/digital (A/D) converter, such as the M7062 coder-decoder (CODEC). The resulting digital signal is denoted as $s(n)$. This signal is then fed to a filter bank that splits $s(n)$ into three frequency components: $s_1(n)$ is the 0- to 500-Hz band, $s_2(n)$ is the 500- to 1000-Hz band, and $s_3(n)$ is the 2000- to 2500-Hz band. Each of these signals has its sampling rate reduced from 8 kHz to 1 kHz as part of the filtering

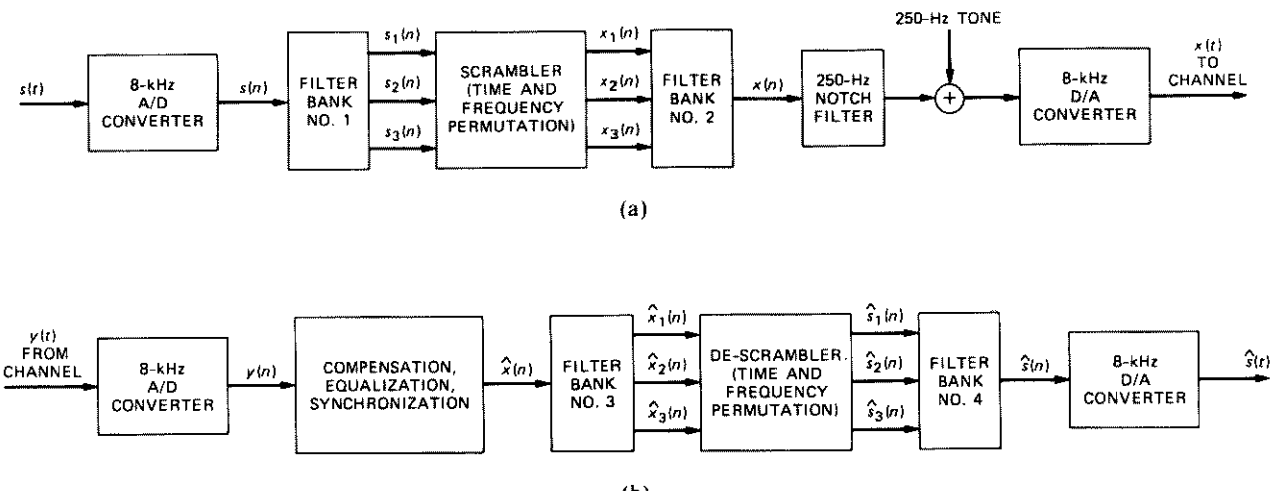



Fig. 1—Block diagram of the proposed full-duplex system. (a) Transmitter. (b) Receiver.

process. For the half-duplex system there are five bands, including these three and also the bands 1000 to 1500 and 1500 to 2000 Hz. The sub-band signals are then fed to a time-and-frequency block scrambler. We have been using a block size of 5 ms. The scrambler uses a pseudo-random key to output three or five blocks from its buffer. These output signals, $x_i(n)$, are fed to a second filter bank. The second filter bank is a synthesis filter bank using quadrature mirror filters. For the full-duplex system it combines the three signals into either a 500-Hz to 2-kHz band or a 2-kHz to 3.5-kHz band, and each user gets one of these bands. In the half-duplex system the five bands are combined into a 500- to 3000-Hz band. As mentioned previously, this division of the available bandwidth can be used to prevent echoes.

At start-up and as required subsequently an impulse is transmitted to equalize the channel at the receiver. A 250-Hz tone is also transmitted continuously to compensate for frequency shift. For this reason $x(n)$ is filtered by a 250-Hz notch filter before adding the 250-Hz tone. Then the synthesized signal is fed to a standard digital/analog (D/A) converter and transmitted as $x(t)$.

The received signal is denoted as $y(t)$ because a number of changes to $x(t)$ may have taken place. The job of the synchronization and equalization subsystem is to produce an output $\hat{x}(n)$ that is as close to $x(n)$ as possible. To produce $\hat{s}(t)$ with as little noise as possible, $\hat{x}(n)$ must have sample-to-sample integrity with $x(n)$. A small slippage of the sampling instants of the receiver relative to the transmitter will result in an error at the ends of the blocks, thus producing a frame-rate-type noise. Because the filters are long the time duration of the noise is much longer than the relative slippage. More details are given about this subsystem in the section on synchronization and equalization. Next, the signal $\hat{x}(n)$ is fed to the receiver analysis filter bank (filter bank #3). This filter bank uses the same quadrature mirror filter (QMF) structure as filter bank #2 (a synthesis filter bank). Because the same QMFs are used, $x_i(n)$ can be almost exactly recovered in the absence of noise and other distortion in the channel. Only QMFs have this property.[2] This is because they are bandwidth preserving. The scrambling process increases the bandwidth of the signal. The individual bands can only be recovered if the increased bandwidth is fully preserved. The de-scrambler delays the time-and-frequency-permuted blocks by an amount complementary to their delay in the transmitter. In this way all blocks experience the same total delay and can be output in the correct order—oldest blocks first. Filter bank #4 is the synthesis counterpart to filter bank #1. Presumably, the $\hat{s}_i$ have the same bandwidths as the $s_i$ and so QMFs are not necessary. However, in order that as much as possible of the 0- to 1000-Hz bandwidth be preserved in the full-duplex system, it is recommended that a QMF be

used to split the 0- to 1000-Hz band and subsequently to reconstruct it. The signal $\hat{s}(n)$ is then fed to a standard D/A converter to form the decrypted output signal $\hat{s}(t)$. For the full-duplex system $\hat{s}(t)$ is not a full-bandwidth signal. Its frequency content is limited to 0 to 1 kHz and 2 to 2.5 kHz. As a result it sounds somewhat muffled. Nevertheless, it seems quite intelligible.

In the following sections details will be presented on the major subsystems: the filter banks, the scrambler and de-scrambler algorithms, and the synchronization-equalization system.

## III. THE FILTER BANKS FOR THE FULL-DUPLEX SYSTEM

Considerable time and effort went into the design of the filter banks used here. Our primary constraint was to come up with a set of filter banks such that each filter bank would occupy only one DSP. At the same time the overall structure must have fairly sharp filters to preserve as much bandwidth as possible of the original speech, since this in turn preserves intelligibility. Also, aliasing between speakers had to be prevented.

Figure 2 shows the structure of filter bank #1. The signal $s(n)$ is first split into two bands (0 to 1 kHz and 2 to 3 kHz). A 95-tap finite impulse response (FIR) filter is used for each band. Since both filters use the same input data the memory required can be shared by both. That way only 95 RAM locations are used. The output of each filter is decimated by four since each filter accomplishes a 4:1 bandwidth reduction on the original signal. This also reduces the computation load on the processor. The output of the low-band filter is fed to a QMF pair to split the signal by another factor of two, producing $s_1$ and $s_2$. These two filters can also share RAM memory, thus using 16 additional locations. The high-band signal is low-pass filtered using the same QMF to produce $s_3$. This requires 16 more RAM locations in the DSP. In all we have used 127 DSP RAM locations. One way of

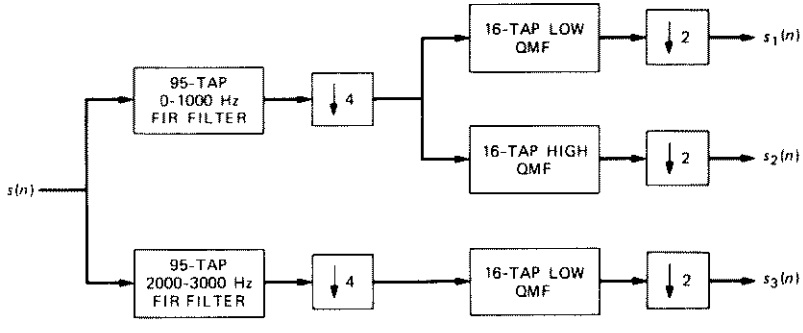



Fig. 2—Structure of filter bank #1 for the full-duplex system.

viewing this structure is that for every eight input samples there will be three output samples. Thus, there is ample computation time for the DSP.

Filter bank #4 uses the same filters. Its structure is shown in Fig. 3. Each QMF requires only eight RAM locations since the input signals are being interpolated 2:1. This accounts for 24 RAM locations. The 95 tap filters each require 24 RAM locations since they are interpolating 4:1. Thus, only 72 RAM locations are used. Larger filters might have been used but they do not add appreciably to the quality. The number of multiply-accumulates per output point is also 72, since each RAM location must be accessed once per output sample. This is also ample time for the DSP.

Figure 4 shows the structure of filter bank #2. The three outputs from the scrambler are first interpolated by 16-tap QMFs. This requires eight RAM locations for each filter, or 24 total. The two outputs from $x_1$ and $x_2$ are combined using a QMF pair before the next stage. The two outputs from the first stage are combined using a 32-tap QMF pair in the second stage. This produces a 4-kHz sampled signal with a bandwidth of 500 to 2000 Hz. Depending on whether the user is assigned to the upper or lower band, the next QMF is either high or low. The second stage required 32 RAM locations as does the third stage. This gives a total RAM requirement of 88 and also means 88 multiply accumulates are performed per output point. The 64-point QMF is a fairly sharp filter with a large rejection band. It is quite adequate to keep separate the encrypted speech of the two talkers. All of the QMFs were designed by Johnston.[3]

Filter bank #3 uses the same QMFs for analysis and its structure is shown in Fig. 5. The encrypted speech is first filtered by a 64-tap QMF. This is a high- or low-pass filter, depending on the band assigned. The output is decimated 2:1 to put the signal on the 0- to 2000-Hz band. A 32-tap QMF pair splits this band and then 16-tap QMF filters are used to do the final splitting. Stage 1 requires 64 RAM locations and stages

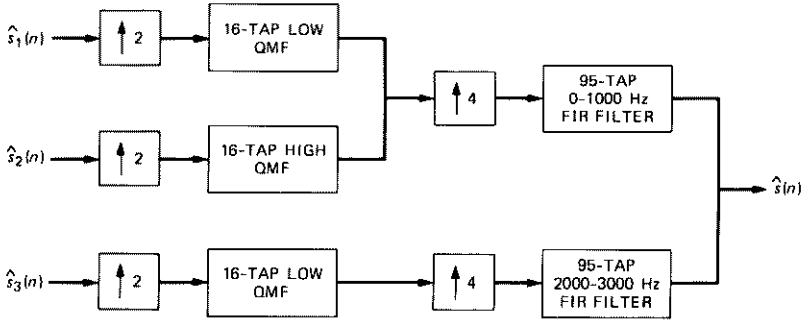



Fig. 3—Structure of filter bank #4 for the full-duplex system.

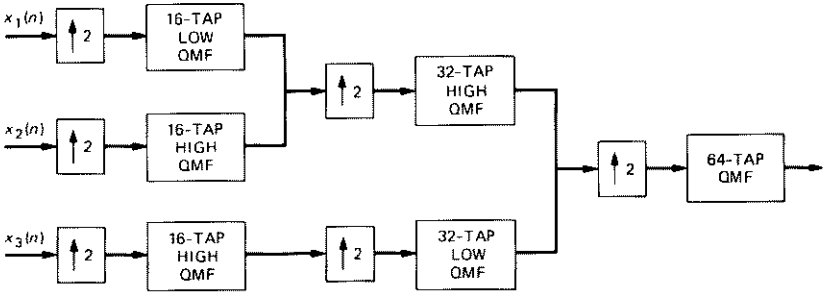



Fig. 4—Structure of filter bank #2 for the full-duplex system.

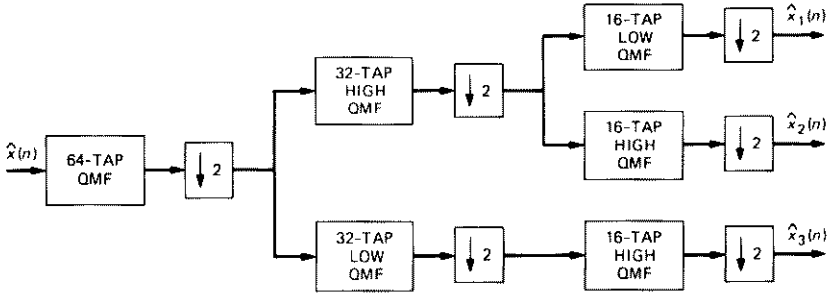



Fig. 5—Structure of filter bank #3 for the full-duplex system.

2 and 3 require 32 each, thus using all 128 RAM locations in a DSP. As in the other analysis filter, eight input samples produce three output samples, thus allowing adequate time for the computations.

The reason for using quadrature mirror filters in filter banks 2 and 3 is worth reiterating. Since we do segment permutation both within and between bands, the bandwidth of the signals $x_1$, $x_2$, and $x_3$ will be the full 500 Hz. If we chose a filter that did not preserve the full bandwidth, these signals would be altered. The result would sound like frame rate noise or "burble." By using the QMF structure the bands will be preserved to within 30 dB with no loss of bandwidth. Conversely, if we chose a full bandwidth filter other than a QMF, then when the bands were combined they would alias with each other. This also happens with a QMF, but when they are separated by the same QMF in filter bank #3 the aliasing is cancelled. This is not the case with an ordinary full-bandwidth filter. This is why the QMF structure is essential.

## IV. THE FILTER BANKS FOR THE HALF-DUPLEX SYSTEM

For the half-duplex system quadrature mirror filters were used in all the filter banks. The five bands of the input speech that are used are

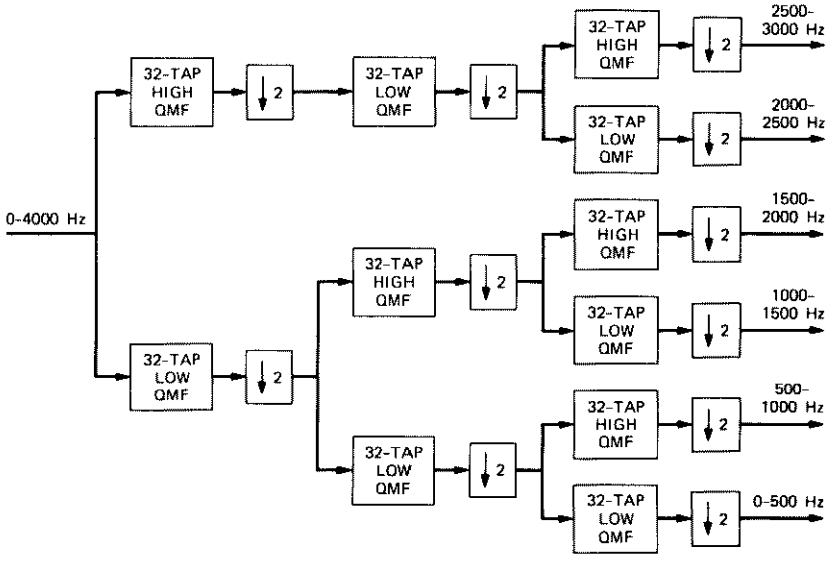



Fig. 6—Analysis filter bank for the half-duplex system.

0 to 500, 500 to 1000, 1000 to 1500, 1500 to 2000, and 2000 to 2500 Hz. A three-stage filter bank structure was devised as shown in Fig. 6. In the first split a 32-tap filter divides the 0- to 2000- and 2000- to 4000-Hz bands. In the next stage the same 32-tap filter was used to divide these two bands into four bands: 0 to 1000, 1000 to 2000, 2000 to 3000, and 3000 to 4000. The highest band was discarded. In the third stage each of these three bands is again split using the 32-tap QMF. After each stage of filtering 2:1 decimation takes place. Thus, each band is sampled at 1000 Hz. For the transmitter the 2500- to 3000-Hz band is discarded. However, for the receiver it is the 0- to 500-Hz band that is discarded.

The synthesis filter bank counterpart is shown in Fig. 7. For the transmitter nothing is put in the 0- to 500-Hz band, since it is used later for the phase-roll compensation. For the receiver nothing is put in the 2500- to 3000-Hz band.

These filter banks require more RAM memory than the full-duplex banks did. The analysis filter bank requires 192 memory locations while the synthesis filter bank requires 160 memory locations. No structure could be found that used less than 128 locations and also gave good performance.

## V. THE SCRAMBLER AND DE-SCRAMBLER

The scrambler and de-scrambler are based on Jayant's idea of a rolling code scrambler[4] with one new twist. Figure 8 shows a memory

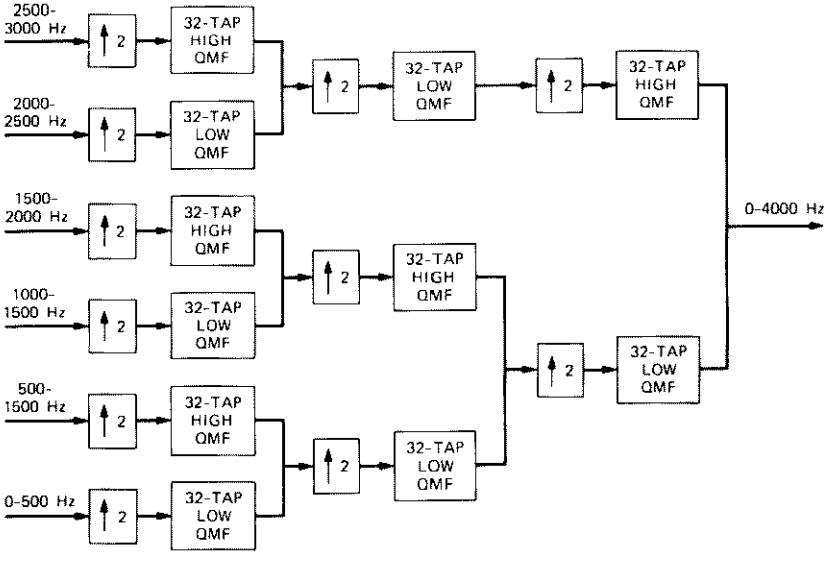



Fig. 7—Synthesis filter bank for the half-duplex system.

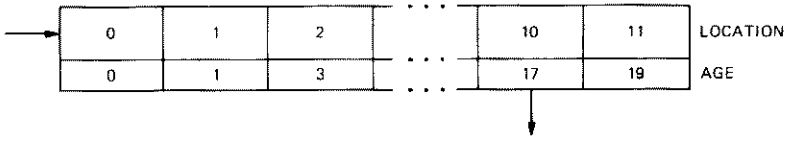



Fig. 8—Scrambler buffer.

buffer. The locations actually correspond to blocks of memory. The age is the age of the block in that location. At any instant data is always being entered into the leftmost block; hence its age is always zero. One of the blocks is also being emptied from the buffer. In this example it is block 10. The algorithm for choosing the block to output is derived as follows. First, the rightmost block (block 11, here) is checked to see if it has reached its maximum age. It if has, then it must be transmitted. Otherwise a random number generator is used to pick a random integer from 0 to 11, and that number determines the block chosen. Once a block is emptied, all blocks to its left are shifted right one location, thus freeing up block 0 for a new input.

The de-scrambler buffer shown in Fig. 9 is similar to the scrambler buffer. Here we see block 10 from Fig. 8 being entered into the buffer. Block 23 is always the oldest block in the buffer and so it is chosen for emptying. Other blocks in this buffer are kept in order of increasing age. So we see that the structure for the two buffers is identical. The method chosen for selecting the output block from the scrambler

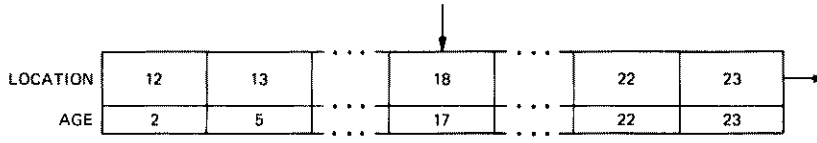



Fig. 9—De-scrambler buffer.

buffer is equivalent to determining where to place the input block in the de-scrambling buffer. Note that the total delay for all blocks will be the same. All of this is exactly the same as Jayant's ideas.

In any real system provision must be made for resynchronizing the de-scrambler if an error occurs at the receiver. This is the basis for modifying Jayant's ideas. The new idea presented here is to periodically reset the transmitter or scrambler buffer to its initial state. In its initial state the age of each block corresponds to its location, i.e., block 11 is only age 11, etc. Suppose we want the period between re-initialization to be $N$ and we have a buffer with $M$ blocks in it. At times 1 through $N - M$ we use Jayant's ideas as just described to select the output block. At time $N - M + 1$ we eliminate block 0 as a candidate for transmission (we still have $M - 1$ blocks to choose from). At time $N - M + 2$ we eliminate both blocks 0 and 1. We continue in this manner, restricting ourselves to a smaller set of blocks each time. At time $N - 2$ we can only choose block $M - 2$ or block $M - 1$ and at time $N - 1$ we can only choose block $M - 1$. At that point the buffer will have been re-initialized, i.e., at time $N$ the locations and ages of the blocks will be in correspondence, just as they were at time 1. At time $N$ we substitute a synchronization pulse for block $M - 1$. Note that the eavesdropper without the key will still be unable to tell what block is coming out at any given time, even though one can tell when the key ends by detecting the sync pulse.

In our proposed system each block represents a frequency band as well as a time. At any given instant three or five blocks are being entered rather than one. This easily can be accounted for without modifying the algorithm very much. At any time we have three or five output pointers instead of one. We just read the key once for each band. The 3- or 5-band structure can be used to our advantage in doing the re-initialization. Suppose we let $N$ and $M$ be multiples of three for the full-duplex case or five for the half-duplex case. The $N$-th entry in the key will correspond to the output for the highest band. At time $N$ we are supposed to transmit the known sync signal. Because of the way we constructed the key, the omitted data in block $M - 1$ will also be from the highest band and therefore its loss will cause less degradation to the speech than if it were from a lower band. At the receiver the reception of this known signal indicates that the scrambler buffer

has been re-initialized and so the de-scrambler buffer should also be re-initialized. While this procedure will allow a potential eavesdropper to know the period of the key, it will not give away the key or any more information than before. At the receiver when the known sync signal is received, zeros are substituted so that the sync signal is not heard as part of the decrypted signal. The momentary loss of bandwidth in the decrypted signal does not affect intelligiblity while the signaling "chirps" that we send are quite noticeable in the encrypted signal.

We note that for the buffer sizes shown in Figs. 8 and 9, here, namely 12 blocks with five samples per block, a single DSP is more than adequate for use as the scrambler buffer. A DSP would be sufficient for a buffer twice this size.

## VI. SYNCHRONIZATION AND EQUALIZATION

Our system uses the same signals for both synchronization and equalization and therefore we discuss them together. First we discuss what we mean by these terms. Then we address how the problems are solved.

As mentioned previously sample-to-sample integrity is essential if a good quality $\hat{s}(t)$ is to be recovered from $y(t)$. This means both that the input samples $\hat{x}(n)$ should be as close as possible to $x(n)$ and that the state of the de-scrambler should exactly match that of the scrambler. Thus, there are two types of synchronization to perform. We must synchronize the de-scrambler state with the scrambler state and match $\hat{x}(n)$ with $x(n)$.

The signaling chirp indicating the re-initialization of the transmitter is used to synchronize the state of the receiver with the transmitter. This chirp can be detected to one of two ways. One way is to use a matched filter to detect the chirp. The advantage of this method is that it can be used on the full-band, 8000-Hz sampled signal. The alternate method is to use the filter bank structure and apply a match filter to $\hat{x}_3$ or $\hat{x}_5$. This method will work provided that the decimation cycles of the filter bank structure are in the proper phase. In general the odds of randomly selecting the right phase are 7:1 against. For this reason we use the first method for start-up. After sync is established the second method can be used to check sync.

To match $\hat{x}(n)$ with $x(n)$ we need to do channel equalization so that the two sets of samples are as close as possible. In turn this means looking at all the possible channel degradations and being prepared to handle each one. Figure 10 shows some of the things that can happen to the signal $x(n)$. First the signal is passed through an anti-aliasing filter. Next it is modulated up by a frequency $f$ and passed through the channel. We can view the passage through the channel as more filtering

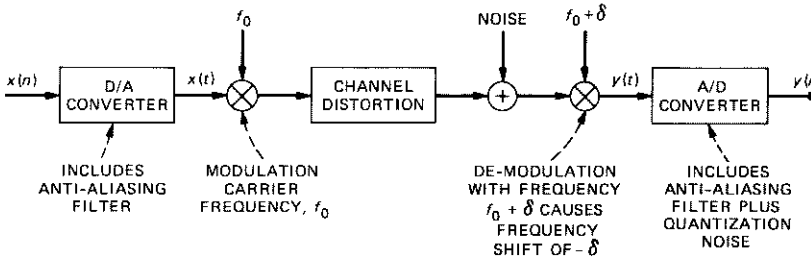



Fig. 10—Channel model.

followed by additive noise. The signal is then demodulated by a frequency $f + \delta$, which causes a frequency shift, or "phase roll," of $-\delta$. Finally, in the A/D process it is filtered once more and quantization noise is added. Fortunately, all of these degradations are independent and can be attacked separately.

We first argue that the noise will have the same effect as white noise of the same power, since the speech bands are constantly being permuted. Other than that the noise is essentially left unchanged, since we have no way to remove it.

We next attack the frequency shift and remove it. This frequency shift is also referred to as "phase roll" in the literature. A small shift in frequency of normal speech would probably not be detectable. However, for our work it is catastrophic. This is because the QMF filters cannot tolerate any frequency shift if they are to properly cancel out aliasing.

To compensate for the phase roll we propose that a 250-Hz sidetone be transmitted. This tone will be received with the same frequency shift as the rest of the signal. Figure 11 shows a novel phase-roll compensation scheme we have devised that works quite well. In Figure 11 the input signal branches immediately. The top branch goes first through a sharp bandpass filter (BPF #1), which passes only 240 to 260 Hz. The purpose is to immediately isolate the 250-Hz tone (which may have been slightly shifted). Next the tone is frequency inverted, making it approximately 3750 Hz. Then it is interpolated 2:1 using a second bandpass filter as the interpolation filter (BPF #2) with a center frequency of 3750 Hz. This signal is now modulated by a 250-Hz tone and then bandpass-filtered again (BPF #3). This produces a modulation signal of about 4000 Hz. If the original signal was shifted up $\delta$ Hz, the modulation signal will be 4000-$\delta$ Hz. In the second branch the remainder of the signal is interpolated 2:1 and then modulated up by 4000-$\delta$. It is then decimated and frequency inverted. In this way the received signal $s(n)$ is modulated by exactly the right amount to compensate for the phase roll. One might even say the phase roll is

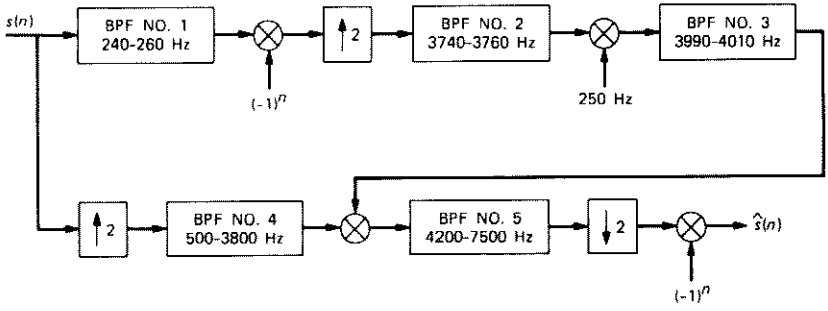



Fig. 11—Phase-roll compensation network.

made to compensate for itself. The interpolation filter in the second branch is actually a bandpass filter that removes the 250-Hz sinewave. Thus, the final output signal does not contain the 250-Hz sinewave.

The phase-roll compensation scheme was tested using computer simulations. The received signal was passed through a computer program that introduced an unknown amount of phase roll. The output from this program was then passed through the phase-roll compensation system. The compensation system completely cancelled the phase roll and its effects, but introduced a small amount of background noise. The noise appeared to be independent of the amount of phase roll in the signal. It was such a low-level noise that a listener needed to be in a quiet room to hear it. On this basis the compensation scheme was judged to work well.

Next, we can measure $h(n)$ by transmitting an impulse and measuring the response at the receiver. If noise is a problem, the measurement can be done several times. Given $h(n)$, $n = 0, 1, \cdots, N$ we form $g(n) = h(N - n)$, $n = 0, 1, \cdots, N$. If we convolve $h(n)$ with $g(n)$, the result will be symmetric sequence $f(n)$, $n = 0, 1, \cdots, 2N$. Because $f(n)$ is symmetric it is linear phase. This suggests that if we use $g(n)$ to filter all the data, $y(n)$, the output will be an integer-delayed version of the signal $x(n)$ with some amplitude modification in the different frequency bands. If we feed $f(n)$ into filter bank #3, we can find the gain to apply to the signals in each band to correct for the amplitude modification. In general the frequency response of $h(n)$ is fairly flat in amplitude for the frequency bands of interest. We have found that an impulse response length of 64 samples centered around the maximum value of the impulse response works well.

## VII. SIMULATION RESULTS TO DATE

Simulation results to date have been extremely encouraging. In the transmitter program some header information is first generated. This

consists of a sine wave that has one of two frequencies, 1000 Hz or 3000 Hz, and thereby determines whether the receiver will use the high channel or low channel. This sine wave is 32 ms long. After a 16-ms wait it transmits an impulse for measuring the channel's impulse response. After 16 ms more it transmits the sine wave again for 32 ms. It then begins the encryption process by sending the "chirp" sync pulse in $x_3$ (or $x_5$) to indicate that the buffer is in initialized status. Superimposed on the signal throughout is a 250-Hz sine wave for the phase-roll compensation. The output of this program is the digital file $x(n)$.

The first channel we used for transmission was just to output $x(n)$ to the computer's D/A and loop right back through the computer's A/D. This "channel" is noise free and has no phase roll but is not perfectly flat in frequency response. It was a good test of our synchronization and equalization algorithms. Our receiver program is able to automatically synchronize itself and do the adaptive equalization needed for this channel. We found that using 5-ms block lengths gave the best results. For block lengths this short, any frame-rate noise sounds more like quantization noise. For larger block lengths, such as 16 ms, the frame-rate noise sounds like burble. Since some frame-rate noise always creeps in, we chose the 5-ms block length as subjectively better.

The next channel we tried was the telephone system at Murray Hill. Using two data sets connected to the D/A and A/D of the computer we were able to loop through the Murray Hill switch. Again the results were good. We then tried introducing a phase roll to the received signal and using our phase-roll compensation system in the same way as described in the previous section. As reported above, this also worked well.

Thus far we have not encountered a noisy channel and so we were forced to simulate one, as we did for the phase roll. The result confirmed our prediction that the noise would sound the same as white noise added to clear speech.

Another advantage we have enjoyed thus far is that the A/D and D/A have the same clock. Therefore, there is no relative drift between them. Such a drift for real hardware would cause a problem if it were at all large. In practice, if the system is regularly resynchronized, the receiver clock can be tied to the resynchronization to compensate for any drift. When we resynchronize the buffers, the chirp could also be used to resynchronize the receiver clock with the transmitter clock.

## VIII. CONCLUSIONS

We have proposed a novel time-and-frequency scrambling system. To avoid the echo cancellation problem we have resorted to one of

two solutions. For a full-duplex system we used a narrow bandwidth, which reduced the quality of the speech but permitted simultaneous talking by both speakers. The narrow bandwidth may have also caused some small loss in intelligibility. For a half-duplex system this is not the case. We used a wider bandwidth system. One can even conceive of an automatic system in which each user receives half bandwidth when both speak but when one is silent the other receives the full bandwidth.

What makes our novel system feasible are the flexibility of digital signal processing and the equalization and synchronization schemes that were devised using digital processing. The digital processing also permits a full time-and-frequency permutation.

## REFERENCES


1. "Digital Signal Processor—A Programmable Integrated Circuit for Signal Processing," B.S.T.J., *60*, No. 7, Part 2 (September 1981).
2. D. Esteban and C. Galand, "Application of Quadrature Mirror Filters to Split Band Voice Coding Schemes," Proc. of 1977 ICASSP (May 1977), pp. 191–5.
3. J. D. Johnston, "A Filter Family Designed For Use in Quadrature Mirror Filter Banks," Proc. of 1980 ICASSP (April 1980), pp. 291–4.
4. N. S. Jayant, R. V. Cox, B. J. McDermott, and A. M. Quinn, "Analog Scramblers for Speech based on Sequential Permutations in Time and Frequency," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Maximum-Power and Amplitude-Equalizing Algorithms for Phase Control in Space Diversity Combining

By P. D. KARABINIS

(Manuscript received October 12, 1981)

*Space diversity receivers equipped with continuous combiners have, in recent years, found wide use in terrestrial microwave radio systems as a means of mitigating the effects of multipath fading. An analysis of two phase-control algorithms for space diversity combining, comparing the performance capabilities of a maximum-power and an amplitude-equalizing algorithm, is presented in this paper. The extended capabilities of the equalizing algorithm in mitigating linear and quadratic channel distortions are investigated through computer simulations. A reduction in digital radio outage time afforded by the equalizing algorithm (relative to that of the maximum-power algorithm) is approximated for over-water-paths where it is assumed that fading can be described in terms of a two-ray fading model. For the model chosen, we show that the additional outage reduction owing to the equalizing phase-control algorithm is highly dependent on the statistics of the multipath delay parameter, $\tau$, and can vary from unity to a factor of five for different probability-density functions of $\tau$.*

## I. INTRODUCTION AND SUMMARY

High-speed digital radio systems operating in the 4-, 6-, and 11-GHz common-carrier bands experience severe performance degradations during periods of multipath fading.[1-7] These degradations are mainly due to intersymbol interference resulting from dispersive channel characteristics (amplitude and group-delay distortions) that typically accompany multipath fading.

Space diversity reception with continuous combining of the received signals has proven very effective in mitigating the amplitude and group-delay distortions caused by multipath fading.[4,5] Traditionally, maximum combined-signal power has been used as the control crite-

rion for combining the received signals.[8] Although maximum-power combining does not react to inband dispersion directly, the method does provide a substantial statistical improvement in this respect. In general, however, the output signal of a maximum-power combiner during dispersive fading will contain, to some degree, linear and quadratic amplitude distortion, with the linear component usually being the predominant one.[4,8] Because of this, the use of an amplitude slope equalizer following a maximum-power combiner has proven very effective in reducing the multipath outage of high-speed digital radio signals to levels compatible with long-haul outage objectives.[4]

With the advent of higher-level digital-modulation formats, equalization requirements of microwave radio channels are becoming increasingly stringent.[9] To optimize the performance of space diversity combiners used in modern digital radio systems, alternate phase-control algorithms have recently been proposed that depart from the traditional maximum-power algorithm in that, instead of maximum combined-signal power, minimum combined-signal dispersion becomes the controlling criterion.[10,11] With these equalizing phase-control algorithms, space diversity combiners have been shown to provide reduced levels of signal dispersion during periods of multipath fading[7] (as compared with maximum-power combiners), thus further reducing the multipath-related outage of high-speed digital signals. One such equalizing algorithm is proposed, analyzed, and compared with a maximum-power algorithm in this paper.

In Section II, the conventional maximum-power combiner is reviewed first, with a particular implementation of the maximum-power phase-control algorithm described in detail. Then we show how the maximum-power algorithm can be converted into an equalizing algorithm by judiciously extracting a phase-control signal from selected portions of the combined-signal spectrum. The analysis reveals that the maximum-power algorithm is only a special case of a more general algorithm that can adaptively maximize the combined-signal power over any frequency interval in the channel. The particular equalizing phase-control algorithm described minimizes the power difference between two spectral samples derived symmetrically from the upper and lower portions of the combined-signal spectrum. To achieve this, the algorithm maximizes the power of the upper or lower spectral sample (as determined by a control criterion) and attempts to bring the two frequency subbands in power equilibrium. As a consequence, we show that the linear and quadratic amplitude distortions of the combined-signal spectrum are substantially reduced for many dispersive channel conditions.

Section III presents results of computer simulations illustrating the response of the maximum-power and equalizing combiners to disper-

sive channel conditions. An improvement factor of the equalizing combiner over the maximum-power combiner is calculated for several assumed probability-density functions of the parameters of a two-ray fading model. Since a two-ray fading model is more likely to represent fading accurately for microwave radio paths over water (where a strong reflective component is typically present during fading) our results too should be viewed in that context. For microwave radio paths over land, it has been shown by Rummler[12] that a three-ray model is required to accurately describe multipath fading. For the two-ray model chosen here, we show that the equalizing combiner improvement factor is strongly dependent on the assumed probability-density function of the multipath delay parameter $\tau$ and varies from approximately unity (no improvement) to a factor of five, for different probability-density functions of $\tau$ and expected values of $\tau$.

## II. MAXIMUM-POWER AND AMPLITUDE-EQUALIZING PHASE-CONTROL ALGORITHMS

A simplified block diagram of a space diversity receiver providing two signal inputs to a continuous combiner is shown in Fig. 1. Following a fixed delay adjustment of the diversity-signal path (to equalize the

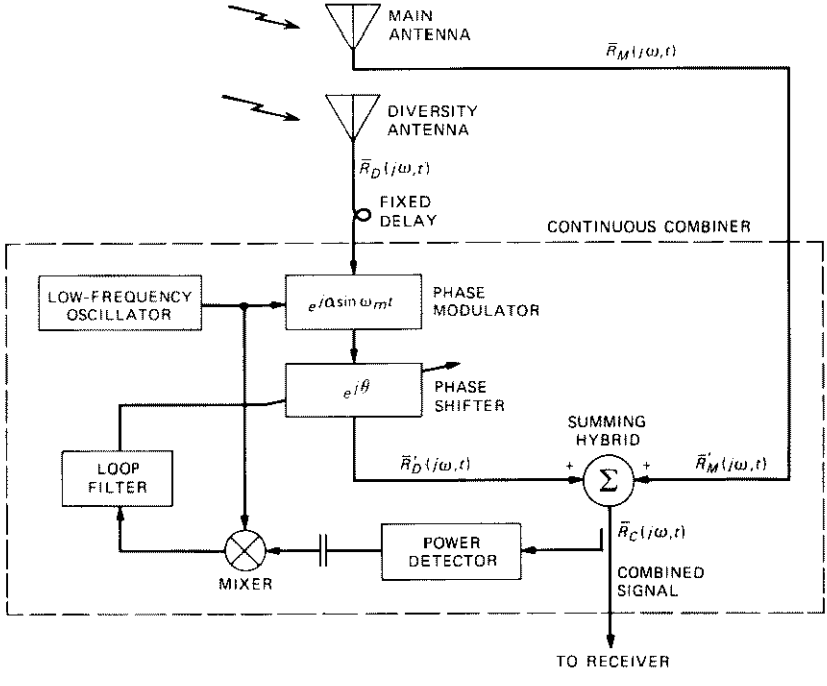



Fig. 1—Block diagram of a space diversity continuous combiner.

electrical path lengths leading to the combiner inputs), the phase of the diversity signal undergoes a dynamic adjustment so as to always maintain the proper relationship with respect to the main-antenna signal phase at the summing hybrid. For the purpose of generating a phase-control signal, the phase of the diversity antenna signal is perturbed sinusoidally, resulting in a periodic modulation of the combined-signal power.[13] The fundamental component of the combined-signal power modulation is detected and used in a feedback arrangement to control the phase-shifter value.[14] Depending on the phase-control algorithm used, the phase correction can be chosen to maximize the average combined-signal power or minimize the dispersion of the combined signal.

In the appendix, the combined-signal power modulation resulting from the sinusoidal phase modulation of the diversity antenna signal is calculated in terms of the spectral densities and phase misalignment of the antenna signals at the inputs of the summing hybrid. In Section 2.1 below we show how the fundamental component of the combined-signal power modulation can be used as the control signal in a simple feedback loop to form a maximum-power combining algorithm. Section 2.2 describes a method of converting the maximum-power combiner of Section 2.1 into an equalizing combiner by selectively extracting the combined-signal power modulation from two frequency bands symmetrically located to the left and right of the channel center frequency.

### *2.1 Maximum-power phase-control algorithm*

The average combined-signal power over the entire channel can be expressed, using eqs. (44) through (46) of the appendix, as:

$$p_0(t, \theta) = \frac{1}{2\pi}\int_{\omega_1}^{\omega_2} \delta^2(\omega, t)V^2(\omega, t)d\omega + \frac{1}{2\pi}\int_{\omega_1}^{\omega_2} V^2(\omega, t)d\omega + \frac{1}{\pi}J_0(\alpha)\int_{\omega_1}^{\omega_2} \delta(\omega, t)V^2(\omega, t)\cos\{\psi(\omega, t) + \theta\}d\omega, \quad (1)$$

where the frequency interval $\omega_1 \leq \omega \leq \omega_2$ is assumed to span the channel bandwidth, $\delta^2(\omega, t)V^2(\omega, t)$ and $V^2(\omega, t)$ denote the diversity and main antenna power-spectral densities, respectively, $\psi(\omega, t)$ is the phase of the diversity antenna spectrum relative to the phase of the main antenna spectrum before any phase correction, $\theta$ represents the frequency-independent phase correction introduced in the diversity antenna spectrum by the phase shifter (see Fig. 1), and $J_0(\alpha)$ denotes a Bessel function of the first kind whose argument is the magnitude of the sinusoidal phase modulation introduced in the diversity signal. The explicit dependence of the variables appearing in eq. (1) on time

accounts for slow variations of these parameters caused by changes in the transmission medium. However, these variations are assumed quasi-static with respect to the transmission rate of the microwave radio channel.

To find $\theta_m$, the value of $\theta$ that maximizes $p_0(t, \theta)$, we set the first partial derivative of $p_0(t, \theta)$ to zero and require the second partial derivative to be negative. That is,

$$\left.\frac{\partial p_0(t, \theta)}{\partial \theta}\right|_{\theta=\theta_m} = -\frac{1}{\pi} J_0(\alpha) \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \sin\{\psi(\omega, t) + \theta_m\} d\omega = 0 \quad (2)$$

and

$$\left.\frac{\partial^2 p_0(t, \theta)}{\partial \theta^2}\right|_{\theta=\theta_m} = -\frac{1}{\sqrt{2}\pi} J_0(\alpha) \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \cos\{\psi(\omega, t) + \theta_m\} d\omega < 0. \quad (3)$$

Having established the conditions for maximum-power combining, we will now show that the phase-control loop of Fig. 2 finds a value of $\theta$ that satisfies these conditions.

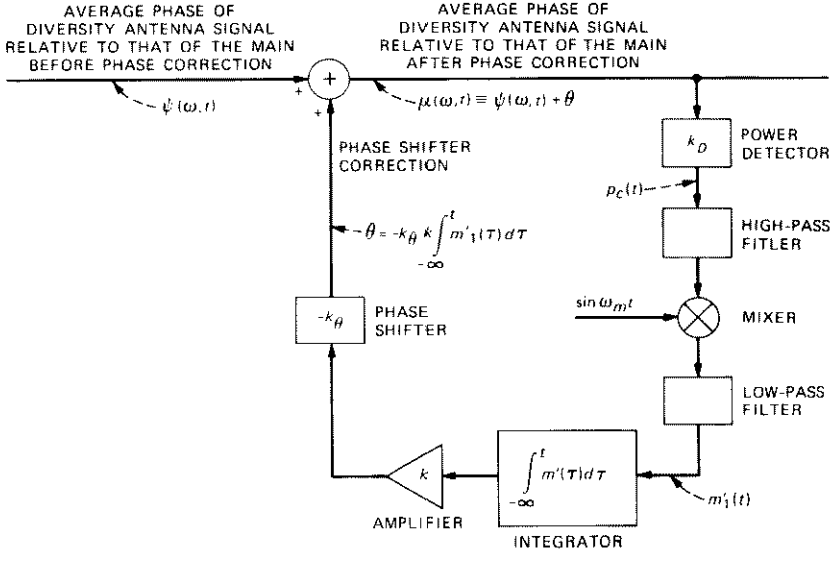



Fig. 2—Block diagram of maximum-power combiner phase-control loop.

From the assumption of a slowly varying transmission medium, the fundamental component of the combined-signal power modulation, $m_1'(t)$, [derived in the appendix, eq. (55)] will be a narrowband process. Choosing the corner frequencies of the high-pass and low-pass filters of Fig. 2 well below and above the highest spectral component of $m_1'(t)$, respectively, the function $m_1'(t)$ will be unaffected by these filters.* Therefore, from an examination of the phase-control loop of Fig. 2 we can write:

$$\theta = -k_\theta k \int_{-\infty}^{t} m_1'(\tau)d\tau, \tag{4}$$

where $k$ and $-k_\theta$ denote component gains as shown in Fig. 2. The rate of change of $\theta$ with respect to time can be expressed from eq. (4) and eqs. (55) and (49) of the appendix as

$$\begin{aligned}\frac{d\theta}{dt} &= -k_\theta k m_1'(t) \\ &= -4J_1(\alpha)k_\theta k \frac{\displaystyle\int_{\omega_1}^{\omega_2} \delta(\omega, t)V^2(\omega, t)\sin\{\psi(\omega, t) + \theta\}d\omega}{\displaystyle\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t)d\omega}.\end{aligned} \tag{5}$$

Observe from eq. (5) that a nonchanging value of $\theta$ can exist if and only if

$$m_1'(t) = \frac{4J_1(\alpha)\displaystyle\int_{\omega_1}^{\omega_2} \delta(\omega, t)V^2(\omega, t)\sin\{\psi(\omega, t) + \theta\}d\omega}{\displaystyle\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t)d\omega} = 0, \tag{6}$$

which implies that

$$\int_{\omega_1}^{\omega_2} \delta(\omega, t)V^2(\omega, t)\sin\{\psi(\omega, t) + \theta\}d\omega = 0. \tag{7}$$

Therefore, if $m_1'(t) \neq 0$ the phase shifter correction, $\theta$, will continue to change until a value satisfying eq. (7) is found. In satisfying eq. (7), eq. (2) and, therefore, the first condition for maximum-power combining is simultaneously satisfied. It remains to be shown that this value of

* It is also assumed that the modulation frequency $\omega_m$ is large compared with the bandwidth of $m_1'(t)$. For example, if the bandwidth of $m_1'(t)$ is limited to 1 Hz, then we choose $f_m = (\omega_m/2\pi) \geq 10$ Hz.

$\theta$ also satisfies the second condition for maximum-power combining as expressed by inequality (3).

Assume that a value of $\theta$, $\theta = \theta_0$, satisfying eq. (7) is found by the phase-control loop. That is, assume an initial condition of $\theta = \theta_0 \pm \Delta\theta$, $\Delta\theta > 0$, and a final condition of $\theta = \theta_0$. With $\theta = \theta_0 \pm \Delta\theta$ we have

$$\int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \sin\{\psi(\omega, t) + \theta_0 \pm \Delta\theta\} d\omega \neq 0. \tag{8}$$

Defining,

$$\xi \equiv \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \sin\{\psi(\omega, t) + \theta_0 \pm \Delta\theta\} d\omega$$

$$= \frac{\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t) d\omega}{4J_1(\alpha)} m_1'(t), \tag{9}$$

$\xi$ may be expressed as

$$\xi = \cos(\pm\Delta\theta) \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \sin\{\psi(\omega, t) + \theta_0\} d\omega$$

$$+ \sin(\pm\Delta\theta) \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \cos\{\psi(\omega, t) + \theta_0\} d\omega. \tag{10}$$

Since by assumption, the first term of eq. (10) is zero we are left with

$$\xi = \sin(\pm\Delta\theta) \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \cos\{\psi(\omega, t) + \theta_0\} d\omega$$

$$= \frac{\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t) d\omega}{4J_1(\alpha)} m_1'(t). \tag{11}$$

If the assumed $\theta$ increment is taken positive ($+\Delta\theta$) the integral of eq. (11), $\int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \cos\{\psi(\omega, t) + \theta_0\} d\omega$, must be positive to make $m_1'(t)$ positive. A positive $m_1'(t)$ will generate a negative $d\theta/dt$ [see eq. (5)] as is required to force $\theta = \theta_0 + \Delta\theta$ to $\theta = \theta_0$, as per assumption. Otherwise, $\theta = \theta_0 + \Delta\theta$ will migrate away from $\theta = \theta_0$, contradicting the initial assumption. Similarly, if we assume a negative $\theta$ increment ($-\Delta\theta$) the integral $\int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \cos\{\psi(\omega, t) + \theta_0\} d\omega$ must again be positive in order for a positive $d\theta/dt$ to be generated as is required to force $\theta = \theta_0 - \Delta\theta$ to $\theta = \theta_0$.

Having shown that $d\theta/dt = 0$ if and only if $m_1'(t) = 0$ (which requires

that $\int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \sin\{\psi(\omega, t) + \theta_0\} d\omega = 0$), in conjunction with $d\theta/dt = 0$ if and only if $\int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \cos\{\psi(\omega, t) + \theta_0\} d\omega > 0$, we have shown that the phase-control loop of Fig. 2 satisfies both conditions for maximum-power combining.

### 2.2 Amplitude-equalizing phase-control algorithm

In the previous section we showed that the phase-control loop of Fig. 2 finds a phase shifter value that forces the fundamental component of the combined-signal power modulation, $m'_1(t)$, to zero over the frequency interval $\omega_1 \leq \omega \leq \omega_2$, and in doing so, also maximizes the average combined-signal power over the same frequency interval. Therefore, when $\omega_1 \leq \omega \leq \omega_2$ spans the entire channel bandwidth, as is the case in Fig. 2 (where the combined-signal spectrum is allowed to be seen in its entirety by the power detector), the combined-signal power will be maximized over the entire channel. However, if a narrowband filter is inserted before the power detector of Fig. 2, allowing only a fraction of the combined-signal spectrum to be seen by the power detector, the combined-signal power will be maximized over the frequency band seen by the power detector and not necessarily over the entire channel bandwidth, since the phase difference between the received spectra will, in general, be frequency dependent during periods of multipath fading.

In this section we will examine a more general version of the phase-control loop of Fig. 2, where the signal at the power-detector input is a filtered version of the combined-signal spectrum. Specifically, we will show that an equalizing phase-control algorithm, which minimizes the amplitude slope across the combined-signal spectrum, can be derived from the phase-control loop of Fig. 2 by requiring that the power detector input signal be a weighted sum of spectral components derived symmetrically from the left and right portions of the combined-signal spectrum.

Consider Fig. 3 where two narrowband portions of the combined-signal spectrum, each occupying a bandwidth $2\,\Delta\omega$ and centered at $\omega_l$ and $\omega_r$, respectively, are extracted by the two narrowband filters. The center frequencies of the narrowband filters, $\omega_l$ and $\omega_r$, are equally spaced from the center of the channel and are located to the left and right of the channel center-frequency, respectively. From the output signal of each narrowband filter, a power detector generates a voltage proportional to the combined-signal power found within the corresponding frequency interval, $\omega_l - \Delta\omega \leq \omega \leq \omega_l + \Delta\omega$ and $\omega_r - \Delta\omega \leq \omega \leq \omega_r + \Delta\omega$. Denoting the power detector output voltages by $p_l(t)$ and $p_r(t)$, corresponding to frequency intervals $\omega_l - \Delta\omega \leq \omega \leq \omega_l + \Delta\omega$ and $\omega_r - \Delta\omega \leq \omega \leq \omega_r + \Delta\omega$, respectively, the development in the appendix can be used to express $p_l(t)$ and $p_r(t)$ as

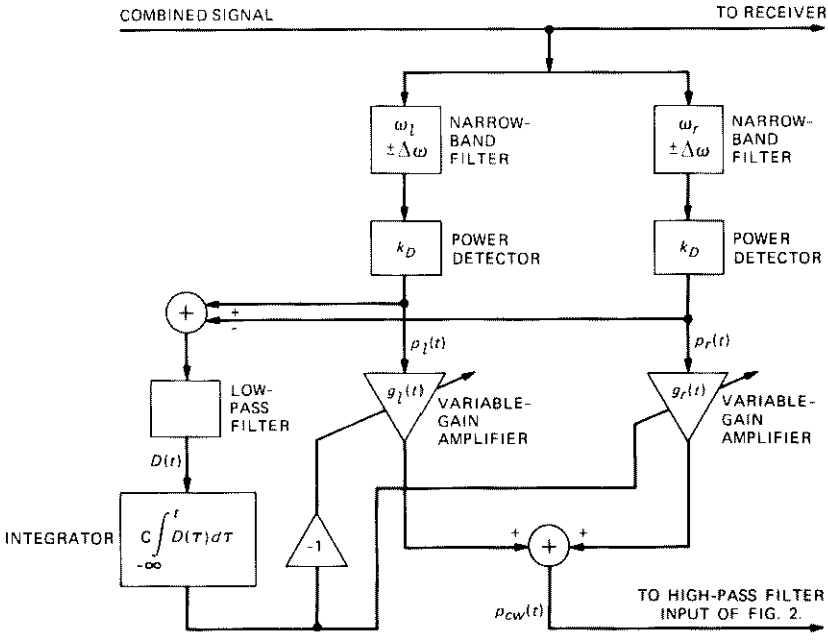



Fig. 3—Power detector portion of equalizing-combiner phase-control loop.

$$p_l(t) = k_D p_{0l}(t)[1 - m'_{1l}(t)\sin \omega_m t + m'_{2l}(t)\cos 2\omega_m t - \cdots] \quad (12)$$

and

$$p_r(t) = k_D p_{0r}(t)[1 - m'_{1r}(t)\sin \omega_m t + m'_{2r}(t)\cos 2\omega_m t - \cdots], \quad (13)$$

where $k_D$ is a proportionality constant associated with each power detector (two identical power detectors are assumed) and with all other variables, $p_{0l}(t)$, $p_{0r}(t)$, $m'_{1l}(t)$, $m'_{1r}(t)$, $\cdots$, conforming to eqs. (52) through (54) of the appendix.

The power detector outputs, $p_l(t)$ and $p_r(t)$, are weighted by variable-gain amplifiers and then summed. Thus, a voltage, $p_{cw}(t)$, representing a weighted combined-signal power over the channel is derived (see Fig. 3), and can be expressed as

$$p_{cw}(t) = g_l(t)p_l(t) + g_r(t)p_r(t). \quad (14)$$

The weighting gains, $g_l(t)$ and $g_r(t)$, are determined by the integral of a difference function, $D(t)$, which is proportional to the average power difference between the frequency bands $\omega_l - \Delta\omega \le \omega \le \omega_l + \Delta\omega$ and $\omega_r - \Delta\omega_r \le \omega \le \omega_r + \Delta\omega$. That is, we let

$$D(t) \equiv k_D[p_{0l}(t) - p_{0r}(t)], \quad (15)$$

$$g_l(t) \equiv -k_g C \int_{-\infty}^{t} D(\tau)d\tau, \tag{16}$$

and

$$g_r(t) \equiv k_g C \int_{-\infty}^{t} D(\tau)d\tau, \tag{17}$$

where $k_g$ is a proportionality constant associated with the variable-gain amplifiers and C is an integration constant. In addition, we require that the gains $g_l(t)$ and $g_r(t)$ be bounded

$$\left.\begin{aligned} 0 \le g_l(t) \le g_{\max} \\ 0 \le g_r(t) \le g_{\max} \end{aligned}\right\} \text{for all } t. \tag{18}$$

It can now be seen from eqs. (15) through (17) that if a positive amplitude slope exists across the combined-signal spectrum, $p_{0r}(t) > p_{0l}(t)$, $D(t)$ will be negative causing $g_l(t)$ to increase [or remain fixed at $g_{\max}$ if, initially, $g_l(t) = g_{\max}$] and $g_r(t)$ to decrease [or remain fixed at zero if, initially, $g_r(t) = 0$]. That is,

$$\begin{aligned} &\text{if } p_{0r}(t) > p_{0l}(t) \\ &\text{(positive amplitude slope)} \end{aligned} \qquad \text{then:} \quad \begin{cases} \dfrac{dg_l(t)}{dt} \ge 0 \\ \dfrac{dg_r(t)}{dt} \le 0. \end{cases} \tag{19}$$

Similarly, if a negative slope exists across the combined-signal spectrum, $p_{0r}(t) < p_{0l}(t)$, $D(t)$ will be positive causing $g_l(t)$ to decrease (or remain fixed at zero) and $g_r(t)$ to increase (or remain fixed at $g_{\max}$). Therefore:

$$\begin{aligned} &\text{if } p_{0r}(t) < p_{0l}(t) \\ &\text{(negative amplitude slope)} \end{aligned} \qquad \text{then:} \quad \begin{cases} \dfrac{dg_l(t)}{dt} \le 0 \\ \dfrac{dg_r(t)}{dt} \ge 0. \end{cases} \tag{20}$$

As the integration constant C [appearing in eqs. (16) and (17)] increases, the response of $p_{cw}(t)$ to a changing amplitude slope becomes faster. In the limit, as C approaches infinity, $p_{cw}(t)$ can assume only two values: either $g_{\max} p_l(t)$ or $g_{\max} p_r(t)$ corresponding to $D(t) < 0$ and $D(t) > 0$, respectively. That is, $p_{cw}(t)$ will instantaneously reflect the power-detector voltage corresponding to the weaker side of the combined-signal spectrum. Thus, with C very large, substituting the block diagram of Fig. 3 in place of the power detector of Fig. 2 results in a

phase-control loop that will always maximize the power of the weaker portion of the combined-signal spectrum and, therefore, minimize the amplitude slope across the combined-signal spectrum.

In the absence of dispersive channel conditions, the amplitude slope of the combined-signal spectrum will be zero. During such periods, random noise effects will dictate the value of $D(t)$. This, however, is of no consequence since in the absence of channel dispersion the phase difference between the received spectral densities is frequency independent and, therefore, maximizing the combined-signal power over any frequency interval within the channel necessarily maximizes the combined-signal power over the entire channel. Consequently, during dispersion-free periods, the equalizing phase-control algorithm described above becomes indistinguishable from the maximum-power phase-control algorithm.

In the following section, results of computer simulations, illustrating the relative performance capabilities of the maximum-power and amplitude-equalizing phase-control algorithms during dispersive channel conditions, are presented.

## III. RELATIVE PERFORMANCE OF MAXIMUM-POWER AND AMPLITUDE-EQUALIZING PHASE-CONTROL ALGORITHMS

The algorithms described above have been stressed by computer simulations using a two-ray fading model. The fade parameters of the model, relative amplitude and delay, are treated as independent random variables and are assigned physical significance. That is, the model adopted here differs from that of Rummler's[12] where the delay parameter $\tau$ is assigned mathematical significance only and treated as a constant. Following the approach of Jakes[3] and Greenstein and Prabhu[15] we assume an exponential form for the probability-density function of the delay parameter $\tau$. For comparison purposes we also examine other forms of the probability-density function of $\tau$, including a Gaussian probability-density function.[16] The probability-density function of the relative amplitude of the delayed wavefront is assumed constant for deep fades.[3,15,16]

To facilitate the simulation of the space diversity arrangement of Fig. 1, we adopt a ray-optical view of wave propagation,[17–19] and assume a circular trajectory for the delayed wavefront. These assumptions enable one to calculate the fade on the diversity antenna given the fade on the main antenna. However, for a given fade-notch location on the main antenna, the fade-notch location on the diversity antenna will in practice be distributed over a wide frequency range since there are many physical $\tau$'s that can generate a particular fade-notch location on the main antenna. Each one of these $\tau$'s will generate a fade notch on the diversity antenna at a different frequency (because of the

geometry of the space diversity arrangement), thus giving the appearance of independent fading between the two antennas.

The receiving antennas are assumed identical and spaced vertically by 30 feet. The channel bandwidth is set at 30 MHz, centered at an RF frequency of 6 GHz, and propagation is assumed over a hop length of 26.4 miles. Finally, all fades seen by the two antennas are assumed, for simplicity, to be of the minimum-phase type.

The space diversity channel model resulting from the above assumptions provides a tractable means of generating a set of stressing simulated fade conditions. The results generated by these simulations are to be viewed as first-order approximations only, until such time as other analytical methods are developed allowing for alternate, perhaps more accurate, channel descriptions.

### *3.1 Amplitude responses of maximum-power and amplitude-equalizing combiners during dispersive fading*

The transfer function of the multipath medium as seen by the main and diversity antennas is expressed as:

$$\bar{H}_M(j\omega, t) = 1 + b(t)e^{-j\omega\tau(t)} \tag{21}$$

and

$$\bar{H}_D(j\omega, t) = 1 + b(t)e^{-j\omega\{\tau(t)+\tau'(t)\}}, \tag{22}$$

where the subscripts $M$ and $D$ refer to the main and diversity antennas, respectively. Assuming a circular trajectory for the delayed wavefront, $\tau'(t)$ is approximated by:

$$\tau'(t) \approx h\left\{\frac{6\tau(t)}{cd}\right\}^{1/2}, \tag{23}$$

where $h$, $d$, and $c$ denote antenna separation, hop length, and speed-of-light in air, respectively. Having calculated the state of the channel from eqs. (21) through (23), for given $b(t)$ and $\tau(t)$ values, the computer program then computes $V(\omega, t)$, $\delta(\omega, t)$, and $\psi(\omega, t)$, as defined by eqs. (37) through (39) of the appendix. The response of the phase-control loop of Fig. 2 is then evaluated in both the maximum-power and equalizing phase-control modes. In the equalizing mode, the narrowband filters (see Fig. 3) are assumed ideal, with $\omega_l$, $\omega_r$, and $\Delta\omega$ set at $\omega_c - B/4$, $\omega_c + B/4$, and $B/4$, respectively, with $B/2\pi$ denoting the channel bandwidth (in Hz) and $\omega_c$ the channel center frequency. With these filter settings, the equalizing-combiner algorithm performs maximum-power combining over the entire upper or lower half of the channel.

The computer simulations were repeated with other combinations of $\Delta\omega$, $\omega_l$, and $\omega_r$ to parametrically investigate the effect of these filter

settings on the equalizing-combiner response. For some particular dispersive channel conditions, the equalizing-combiner response varied for different $\Delta\omega$, $\omega_l$, and $\omega_r$ values. On an overall basis, however, the improvement afforded by the equalizing combiner was found insensitive to these filter parameters as long as $\omega_l$ and $\omega_r$ were chosen reasonably close to the channel edges.

For each simulated channel condition, two phase shifter corrections, $\theta_M(t)$ and $\theta_E(t)$, corresponding to the maximum-power and equalizing phase-control loops, respectively, were calculated. Then, effective transfer functions were formed, relating the maximum-power and equalizing-combiner output signals to the transmitted signal. The effective transfer functions generated by the maximum-power and equalizing combiners are expressed by

$$\bar{E}_M(j\omega, t) \equiv \frac{1}{\sqrt{2}}[\bar{H}_M(j\omega, t) + \bar{H}_D(j\omega, t)e^{j\theta_M(t)}] \tag{24}$$

and

$$\bar{E}_E(j\omega, t) \equiv \frac{1}{\sqrt{2}}[\bar{H}_M(j\omega, t) + \bar{H}_D(j\omega, t)e^{j\theta_E(t)}], \tag{25}$$

respectively, where the factor $1/\sqrt{2}$ is required to satisfy power conservation constraints at the output of the summing hybrid of Fig. 1.

Representative combiner responses, $\bar{E}_M(j\omega)$ and $\bar{E}_E(j\omega)$, are shown in Figs. 4 and 5. In Fig. 4, both transfer functions, $\bar{H}_M(j\omega)$ and $\bar{H}_D(j\omega)$, as seen by the main and diversity antennas respectively, contain a 40-dB fade notch within the channel bandwidth. The response of the maximum-power combiner, as shown, contains a substantial amount of linear and quadratic amplitude distortion across the channel. Its group-delay response (not shown) is also severely distorted. Unlike the maximum-power combiner, the equalizing-combiner amplitude response is flat (to within 1 dB) across the entire channel. The group-delay response of the equalizing combiner (not shown) is also flat across the channel.

Figure 5 shows a channel condition where only one antenna sees a transfer function with an inband notch. As with the situation of Fig. 4, here too, the maximum-power combiner suffers from linear and quadratic amplitude distortions. Note again the almost ideally flat response of the equalizing combiner. Figures 4 and 5 are typical of most simulated channel conditions where one or both antennas experience fade notches well within the channel bandwidth. There are channel conditions, however, where both antenna signals are affected by fade notches that fall outside the channel bandwidth, both below or above the channel-center frequency. These channel conditions produce a predominantly linear amplitude distortion in the channel, accompa-

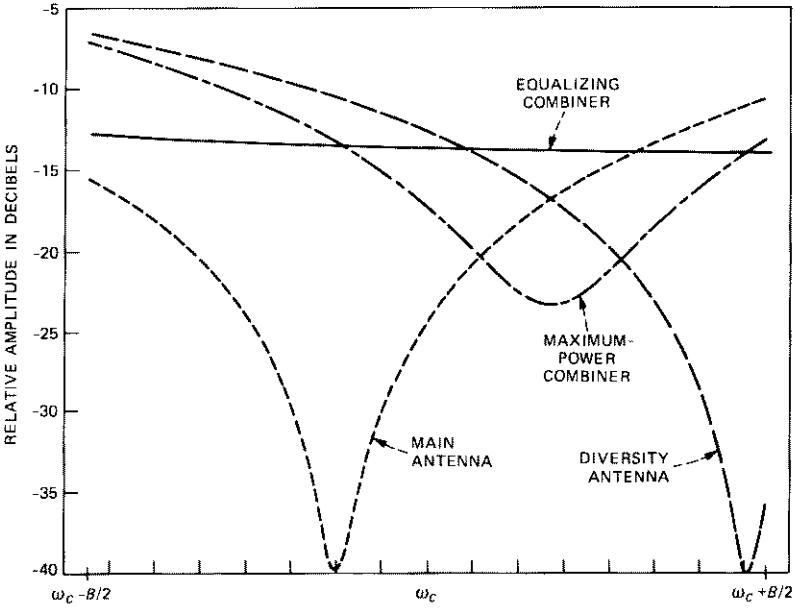



Fig. 4—Combiner response to a dispersive channel condition generated by $b = 0.99$ and $\tau = 2.4183$ ns. The channel bandwidth $= B/2\pi = 30$ MHz.

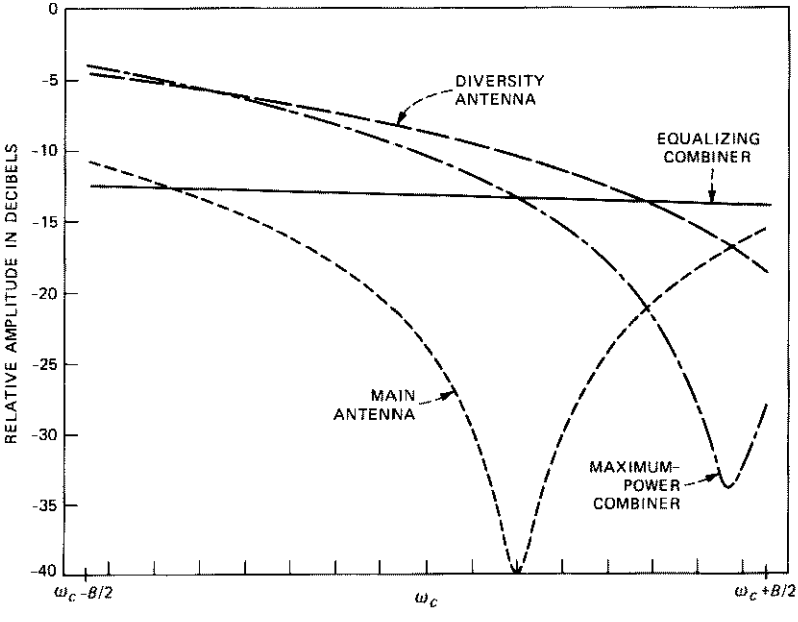



Fig. 5—Combiner response to a dispersive channel condition generated by $b = 0.99$ and $\tau = 2.4151$ ns. The channel bandwidth $= B/2\pi = 30$ MHz.

nied by negligible phase dispersion, since the phase dispersion produced by a fade is concentrated in the vicinity of the fade notch. In response to such a channel condition, the equalizing-combiner output signal can be only marginally better than that of the maximum-power combiner since the inband phase dispersion, required by the equalizing combiner to achieve amplitude equalization, does not exist. Unfortunately, the channel condition described above has a higher probability of occurrence than those depicted in Figs. 4 and 5 (where at least one antenna signal contains a fade notch within the channel) since the channel bandwidth is typically a small fraction of the period $1/\tau$, or $1/(\tau + \tau')$, with which the fade notches repeat. As a result, the reduction in outage time afforded by the equalizing combiner, relative to that of the maximum-power combiner, is limited to relatively small values, as we will see in the following section.

### *3.2 Relative improvement factor of amplitude-equalizing combiner*

In this section, a method of calculating regions on the $\lambda - \tau$ plane ($\lambda \equiv 1 - b$, denoting the fade depth at the fade notch) corresponding to system outage is described. Several outage regions corresponding to the maximum-power and equalizing combiners are calculated. Integration over these regions (having assumed some joint probability-density function for the variables $\lambda$, $\tau$) allows us to compare the performance of the two combiner algorithms and associate a relative improvement factor with the equalizing-combiner algorithm. The relative-improvement factor of the equalizing combiner is shown to be highly dependent on the assumed probability-density function associated with the multipath delay $\tau$, and on the expected value of the multipath delay, $\tau_0$. The relative improvement factor of the equalizing combiner is calculated and plotted as a function of $\tau_0$ for two assumed probability-density functions of $\tau$.

The computer simulation described in the previous section was extended to statistically evaluate the relative performance of the maximum-power and amplitude-equalizing combiners with respect to dispersive channel conditions of at least 6-dB peak-to-peak amplitude distortion in the channel. The threshold value of 6-dB peak-to-peak amplitude distortion was chosen for simplicity, since the accumulated time at this, or greater dispersive levels, has been found strongly correlated with the total outage time of high-speed digital signals.[2,8]

Starting from a value of $\tau$ that produces a fade notch at the center of the main antenna spectrum, as determined by

$$\tau = \frac{2n + 1}{2f_c}, f_c = \frac{\omega_c}{2\pi}, n = 0, 1, 2, \cdots,$$

values of $\lambda$ satisfying the outage condition at the maximum-power

and/or equalizing-combiner outputs are determined. Then, by perturbing each initial value of $\tau$, the fade notch is moved by a small amount (2-MHz increment) and $b$ is varied again until all $\lambda$ values satisfying the outage condition at the new setting are found. Perturbations in $\tau$ continue until a frequency band of $\pm 100$ MHz around the band center has been examined. Following this procedure, the first five outage regions, corresponding to $n = 13$ through 17, were evaluated for the maximum-power and amplitude-equalizing combiners. No outage regions were found for $n = 0$ through 12 and $n = 18$ through 25. The effect of outage regions corresponding to values of $n$ greater than 25 were neglected in this study. The first five outage regions that were evaluated are shown in Figs. 6 through 10. On each figure, the value of $n$ and the value of $\tau$ generating the fade notch at the center of the

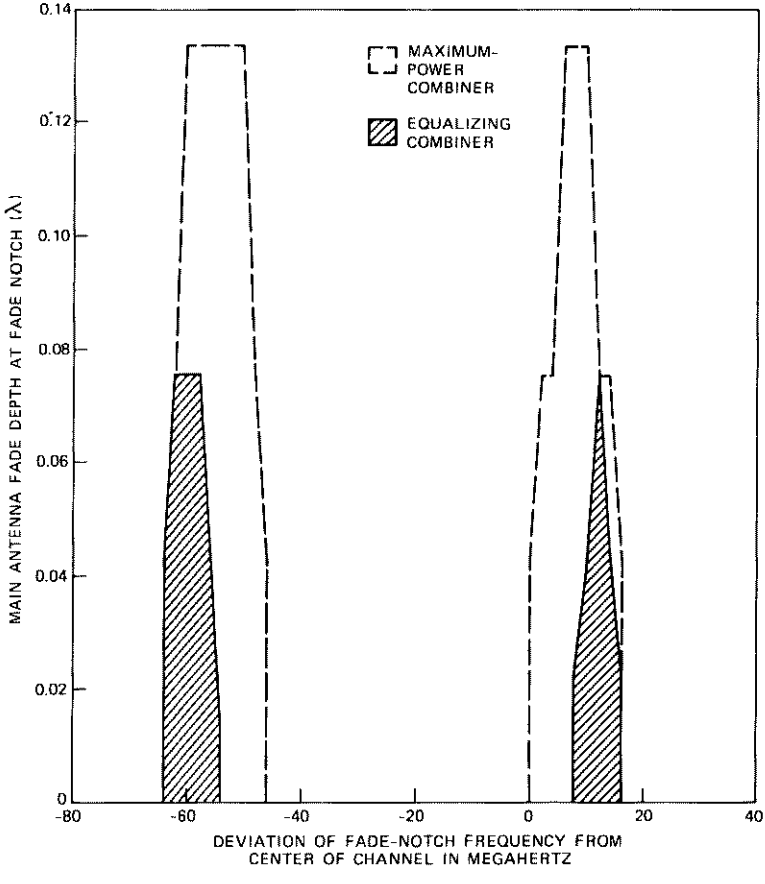



Fig. 6—Maximum-power and equalizing-combiner outage regions for $n = 13$ and $\tau = 2.25$ ns.

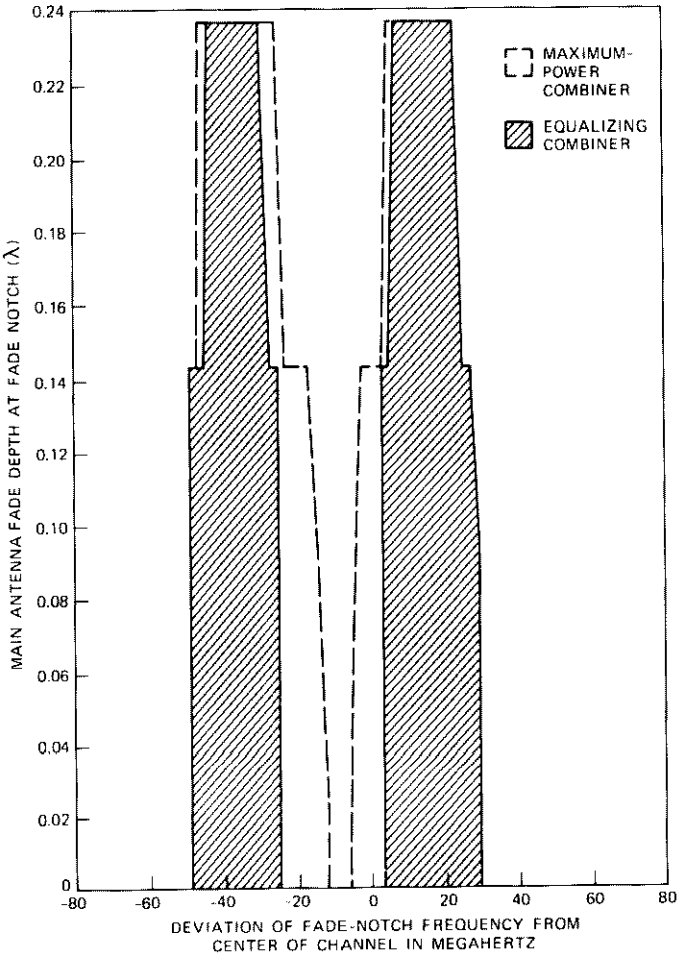



Fig. 7—Maximum-power and equalizing-combiner outage regions for $n = 14$ and $\tau = 2.4167$ ns.

main antenna spectrum are shown. The curves show fade-notch depth as a function of fade-notch location for the main antenna, which results in system outage as defined by the 6-dB amplitude distortion on the combiner outputs. Note that in every case the equalizing-combiner outage region (the shaded area) is totally contained within the maximum-power combiner outage region, which is bounded by the dashed curve. From this simple observation we conclude that the equalizing combiner is outperforming the maximum-power combiner in maintaining reduced levels of channel distortion. The performance improvements, however, are not large. As we can see in Figs. 7, 8, and 9, which contain the larger maximum-power combiner outage regions, the

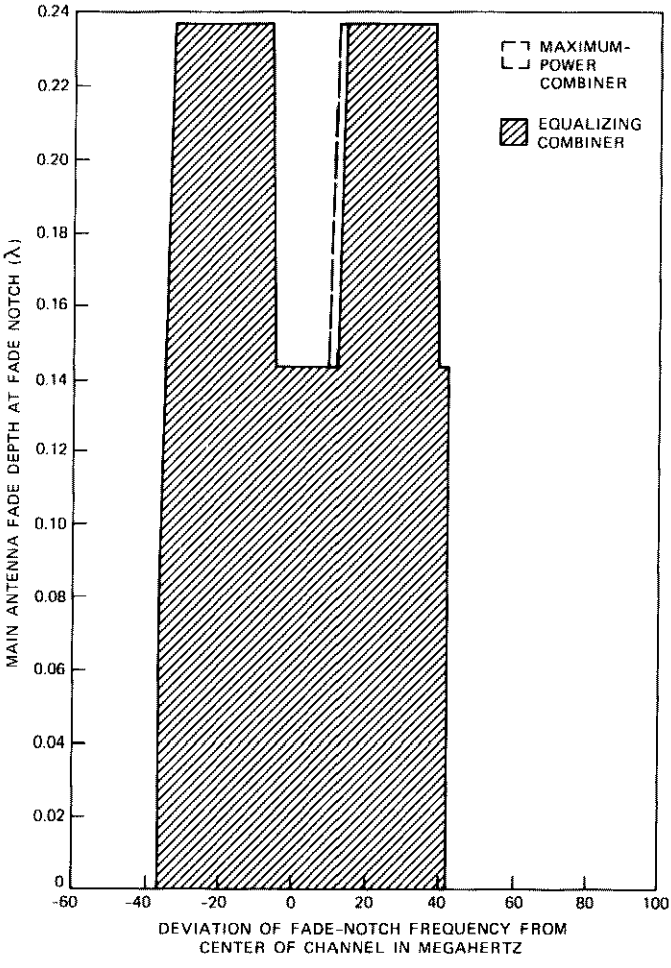



Fig. 8—Maximum-power and equalizing-combiner outage regions for $n = 15$ and $\tau = 2.5833$ ns.

equalizing-combiner outage regions are comparable to those of the maximum-power combiner. Specifically, Fig. 8 shows almost identical outage regions for the two combiners. Close examination of the channel conditions responsible for the outage regions of Fig. 8 reveals very closely correlated fading between the two antenna spectra (i.e., almost simultaneous frequency selective fading). On all other figures, the equalizing combiner outage regions are due to channel conditions that contain little or no inband phase distortion (i.e., fade notches very close to, or completely outside, the channel limits).

To calculate an outage probability for the maximum-power and

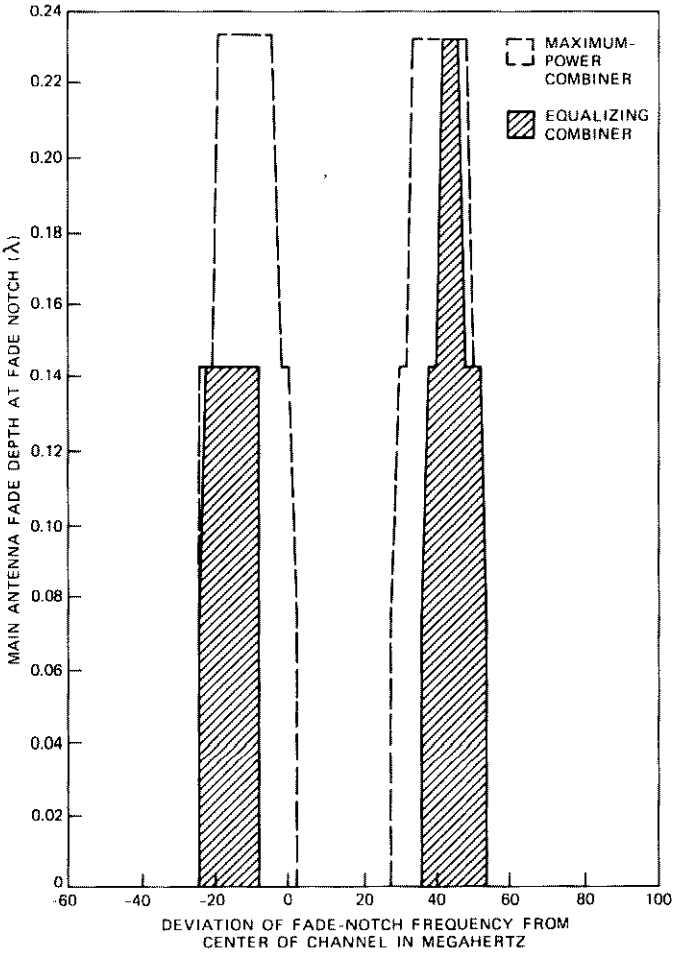



Fig. 9—Maximum-power and equalizing-combiner outage regions for $n = 16$ and $\tau = 2.75$ ns.

equalizing combiners, their corresponding outage regions must be weighted by the joint probability-density function $p_{\tau\lambda}(\tau, \lambda)$, followed by integration over the appropriate limits and summation of all weighted contributions from all outage regions. Accordingly, the probability of outage of the maximum-power combiner can be expressed as:

$$P_M = \sum_{n=13}^{17} \int_{\tau_M^-(n)}^{\tau_M^+(n)} \int_0^{\lambda_M(n,\tau)} p_{\tau\lambda}(\tau, \lambda) d\lambda d\tau, \tag{26}$$

where the subscript $M$ stands for maximum-power combiner, the limits

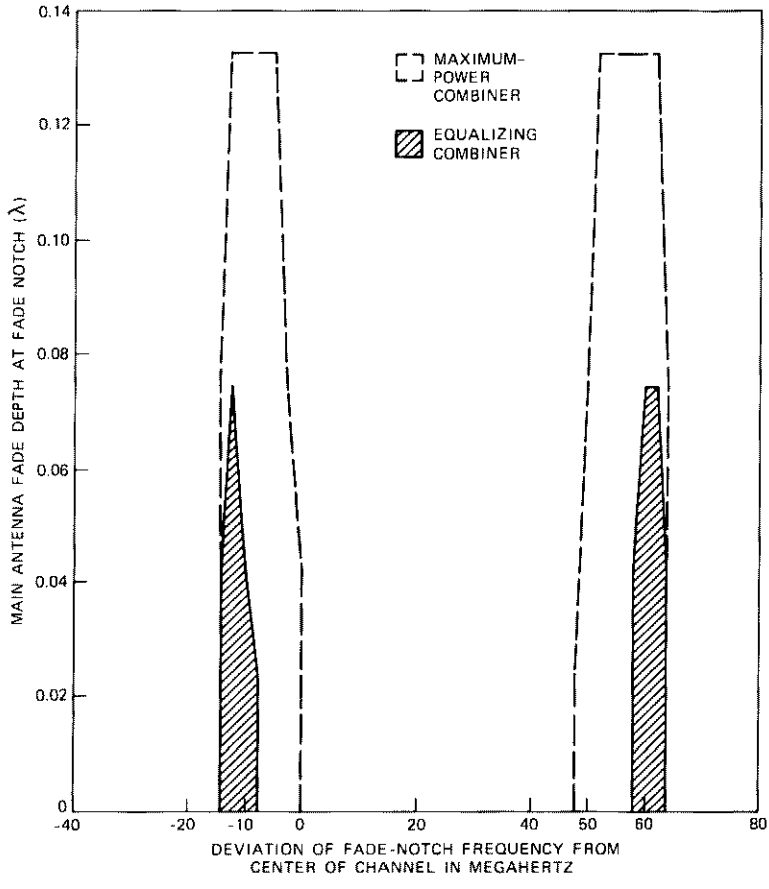



Fig. 10—Maximum-power and equalizing-combiner outage regions for $n = 17$ and $\tau = 2.9167$ ns.

$\tau_M^-(n)$ and $\tau_M^+(n)$ define the range of variation of $\tau$ (for a particular value of $n$) inside which the outage region lies, and $\lambda_M(n, \tau)$ denotes the minimum fade-depth resulting in outage as a function of $\tau$, for a specific value of $n$. Similarly, the probability of outage of the equalizing combiner can be evaluated from:

$$P_E = \sum_{n=13}^{17} \int_{\tau_E^-(n)}^{\tau_E^+(n)} \int_0^{\lambda_E(n,\tau)} p_{\tau,\lambda}(\tau, \lambda) d\lambda d\tau. \tag{27}$$

Using eqs. (26) and (27) we define an equalizing-combiner improvement factor, $I$, over the maximum-power combiner,

$$I \equiv \frac{P_M}{P_E}. \tag{28}$$

To simplify the calculations in evaluating eq. (28) we assume statistical independence between the fade depth $\lambda$ and delay $\tau$, allowing the joint probability-density function to be written as

$$p_{\tau,\lambda}(\tau, \lambda) = p_\tau(\tau)p_\lambda(\lambda). \tag{29}$$

Two probability-density functions of $\tau$ are considered.[3,15,16] The exponentially distributed

$$p_\tau^{(e)}(\tau) = \frac{1}{\tau_0} e^{-\frac{\tau}{\tau_0}}, \qquad \tau \geq 0, \tag{30}$$

and the Gaussian distributed

$$p_\tau^{(G)}(\tau) = \frac{2}{\pi\tau_0} e^{-\frac{\tau^2}{\pi\tau_0^2}}, \qquad \tau \geq 0, \tag{31}$$

where in both cases $\tau_0$ denotes the expected value of $\tau$. The probability-density function of $\lambda$ is assumed of the form[3,15]

$$p_\lambda(\lambda) = \frac{\rho}{1 - e^{-\rho}} e^{-\rho\lambda}, \qquad 0 \leq \lambda \leq 1, \tag{32}$$

with $\rho = 10$.

The equalizing-combiner improvement factor, $I$, is evaluated for the two joint probability-density functions, $p_\lambda(\lambda)p_\tau^{(e)}(\tau)$ and $p_\lambda(\lambda)p_\tau^{(G)}(\tau)$, as a function of the expected delay difference $\tau_0$. The results are shown on Fig. 11. Note the strong dependence of $I$ on $\tau_0$ for the curve corresponding to $p_\tau(\tau) = p_\tau^{(G)}(\tau)$. This is due to the Guassian nature of $p_\tau^{(G)}(\tau)$ which, for small values of $\tau_0$, provides smaller weighting factors for the large outage regions (of Fig. 7, 8, and 9) than does $p_\tau^{(e)}(\tau)$. Also,

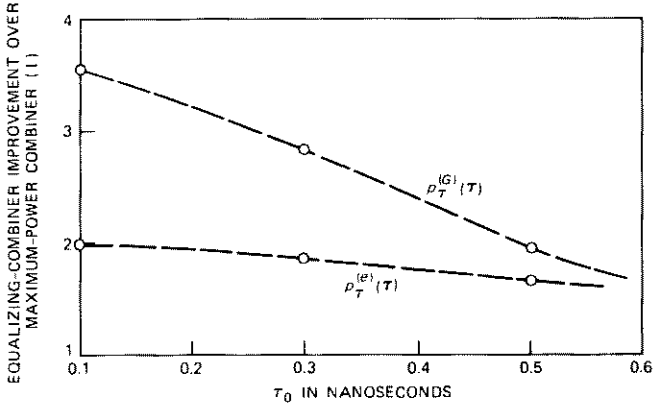



Fig. 11—Equalizing-combiner improvement factor as a function of the expected value of $\tau$.

note the change in improvement factor, for a given value of $\tau_0$, as a function of the two assumed probability-density functions of $\tau$. This is to be expected since the ratio of the outage areas of the maximum-power combiner to those of the equalizing combiner varies considerably for Figs. 6 through 10. Thus, the particular weighting given to the areas of each figure, as determined by the assumed probability-density function of $\tau$, will control the outcome of $I$. To illustrate this point further, we consider a delay difference $\tau$, which is uniformly distributed between $\tau = 2.8$ and 3.1 ns, and zero outside this range. That is, we assume:

$$p_\tau(\tau) \equiv p_\tau^{(u)}(\tau) = \begin{cases} \text{constant}, & \text{for } 2.8 \le \tau \le 3.1 \text{ ns}, \\ 0, & \text{otherwise}. \end{cases} \tag{33}$$

With this probability-density function we calculate an equalizing-combiner improvement factor of five relative to that of the maximum-power combiner, since only the outage region of Fig. 10 contributes (equalizing-combiner improvement factors of this magnitude have recently been observed on a reflective microwave radio path over water[7]). By a similar argument, if $\tau$ is assumed uniformly distributed between $\tau = 2.4$ and 2.7 ns, with zero probability of assuming a value outside this range, the equalizing-combiner improvement factor reduces to approximately unity.

From the above considerations it becomes apparent that the delay-difference statistics used in the two-ray model will play a major role in determining the equalizing-combiner performance relative to that of the maximum-power combiner. Regardless of the delay-difference statistics, however, our analysis predicts only moderate outage reductions to be gained from converting a maximum-power combiner to an equalizing combiner. This suggests that an amplitude-slope equalizer may still be required following the equalizing combiner to bring the outage time of high-speed digital signals to levels below long-haul outage objectives. Nevertheless, the equalizing combiner's ability to mitigate quadratic as well as linear amplitude distortions, during highly dispersive channel conditions, will contribute to enhancing the reliability and robustness of high-speed digital radio systems.

## IV. ACKNOWLEDGMENTS

I am indebted to L. A. Palazzo for the great deal of support he provided in generating the computer simulations and to T. L. Osborne for his constant encouragement. I am also grateful to W. D. Rummler for his many useful suggestions.

## REFERENCES

1. A. J. Giger and W. T. Barnett, "Effects of Multipath Propagation on Digital Radio," Paper 46.7, Int. Conf. on Commun., June 14–18, 1981, Denver, Colorado.

2. C. W. Lundgren and W. D. Rummler, "Digital Radio Outage Due to Selective Fading—Observation vs. Prediction from Laboratory Simulation," B.S.T.J., *58*, No. 5 (May-June 1979), pp. 1073–100.
3. W. C. Jakes, Jr., "An Approximate Method to Estimate an Upper Bound on the Effect of Multipath Delay Distortion on Digital Transmission," IEEE Trans. on Commun., *COM-27*, No. 1 (January 1979), pp. 76–81.
4. C. W. Anderson, S. G. Barber, and R. N. Patel, "The Effect of Selective Fading on Digital Radio," IEEE Trans. on Commun., *COM-27*, No. 12 (December 1979), pp. 1870–5.
5. W. T. Barnett, "Multipath Fading Effects on Digital Radio," IEEE Trans. on Commun., *COM-27*, No. 12 (December 1979), pp. 1842–8.
6. S. Komaki, I. Horikawa, K. Morita, and Y. Okamoto, "Characteristics of a High Capacity 16 QAM Digital Radio System in Multipath Fading," IEEE Trans. on Commun., *COM-27*, No. 12 (December 1979), pp. 1854–61.
7. T. Murase, K. Morita, and S. Komaki, "200 Mb/s 16-QAM Digital Radio System with New Countermeasure Techniques for Multipath Fading," Paper 46.1, Int. Conf. on Commun., June 14–18, 1981, Denver, Colorado.
8. Y. Y. Wang, "Simulation and Measured Performance of a Space Diversity Combiner for 6 GHz Digital Radio," IEEE Trans. on Commun., *COM-27*, No. 12 (December 1979), pp. 1896–907.
9. P. Hartmann and B. Bynum, "Design Considerations for an Extended Range Adaptive Equalizer," Paper 46.5, Int. Conf. on Commun., June 14–18, 1981, Denver, Colorado.
10. W. T. Barnett, C. W. Lundgren, Jr., W. D. Rummler, and Y. Y. Wang, "Equalizing Signal Combiner," U. S. Patent 4,261,056, applied for July 16, 1979, issued April 7, 1981.
11. S. Komaki, Y. Okamoto, and K. Tajima, "Performance of 16-QAM Digital Radio System Using New Space Diversity," Paper 52.2, Int. Conf. on Commun., June 8–12, 1980, Seattle, Washington.
12. W. D. Rummler, "A New Selective Fading Model: Application to Propagation Data," B.S.T.J., *58*, No. 5 (May-June 1979), pp. 1037–71.
13. H. Miedema, unpublished work.
14. K. L. Seastrand, Jr., "Space Diversity Receiver with Combined Step and Continuous Pase Control," U. S. Patent 4,160,952, applied for May 12, 1978, issued July 10, 1979.
15. L. J. Greenstein and V. K. Prabhu, "Analysis of Multipath Outage with Applications to 90 Mbit/s PSK Systems at 6 and 11 GHz," IEEE Trans. on Commun., *COM-27*, No. 1 (January 1979), pp. 68–75.
16. M. Emshwiller, "Characterization of the Performance of PSK Digital Radio Transmission in the Presence of Multipath Fading," Int. Conf. on Commun., ICC 1978.
17. L. W. Pickering and J. K. DeRosa, "Refractive Multipath Model for Line-of-Sight Microwave Relay Links," IEEE Trans. on Commun., *COM-27*, No. 8 (August 1979), pp. 1174–82.
18. O. Sasaki and T. Akiyama, "Multipath Delay Characteristics on Line-of-Sight Microwave Radio System," IEEE Trans. on Commun., *COM-27*, No. 12 (December 1979), pp. 1876–86.
19. M. Ramadan, "Availability Prediction of 8 PSK Digital Microwave Systems During Multipath Propagation," IEEE Trans. on Commun., *COM-27*, No. 12 (December 1979), pp. 1862–9.

## APPENDIX

### *Modulation of Combined-Signal Power*

By modulating the phase of one antenna signal with a low-frequency sinsuoid, the combined-signal power over some arbitrary frequency interval, $\omega_1 \leq \omega \leq \omega_2$ (spanning all or some portion of the channel) will be modulated periodically. In this appendix, we expand the periodic modulation of the combined-signal power in a Fourier series and show that, in the absence of channel dispersion, the fundamental component of the expansion is proportional to the sine of the phase difference between the antenna signals at the inputs of the summing hybrid (see

Fig. 1). During dispersive channel conditions, we show that the fundamental component of the expansion is a weighted average of the sine of the phase difference between the antenna signals at the inputs of the summing hybrid. We thus show that the fundamental component of the combined-signal power modulation can be used as a measure of phase-lead or phase-lag of one antenna signal relative to the other.[13]

Let $\bar{R}_M(j\omega, t)$ and $\bar{R}_D(j\omega, t)$ denote the complex spectral densities received by the main and diversity antennas, respectively, in response to a transmitted signal $s(t)$, with complex spectral density $\bar{S}(j\omega)$. Denoting the transmission-medium transfer function by $\bar{H}_M(j\omega, t)$ and $\bar{H}_D(j\omega, t)$, as seen by the main and diversity antennas, respectively, we can write

$$\bar{R}_M(j\omega, t) = \bar{H}_M(j\omega, t)\bar{S}(j\omega)e^{-j\omega T_0} \tag{34}$$

and

$$\bar{R}_D(j\omega, t) = \bar{H}_D(j\omega, t)\bar{S}(j\omega)e^{-j\omega T_0}, \tag{35}$$

where the delay $T_0$ denotes the line-of-sight propagation delay of the radio link. The dependence of $\bar{H}_M$ and $\bar{H}_D$ on $t$ allows for variations in the transmission-medium transfer function during anomalous propagation conditions (i.e., during multipath fading). However, in writing eqs. (34) and (35) the transmission-medium transfer functions are assumed quasi-static with respect to the rate-of-change of $s(t)$. Defining,

$$\bar{S}(j\omega) \equiv S_0(\omega)e^{j\phi(\omega)}, \tag{36}$$

$$\bar{H}_M(j\omega, t) \equiv H(\omega, t)e^{j\eta(\omega,t)}, \tag{37}$$

$$\bar{H}_D(j\omega, t) \equiv \delta(\omega, t)H(\omega, t)e^{j(\eta(\omega,t)+\psi(\omega,t))}, \tag{38}$$

and

$$V(\omega, t) \equiv H(\omega, t)S_0(\omega), \tag{39}$$

eqs. (34) and (35) can be rewritten as

$$\bar{R}_M(j\omega, t) = V(\omega, t)e^{j[\phi(\omega)+\eta(\omega,t)-\omega T_0]} \tag{40}$$

and

$$\bar{R}_D(j\omega, t) = \delta(\omega, t)V(\omega, t)e^{j[\phi(\omega)+\eta(\omega,t)+\psi(\omega,t)-\omega T_0]}. \tag{41}$$

The quantities $H(\omega, t)$ and $\eta(\omega, t)$ appearing in eq. (37) represent the amplitude and phase transfer functions associated with the dynamic transmission medium. The parameter $\delta(\omega, t)$ of eq. (38) denotes the voltage ratio of the diversity to the main antenna signal as a function of frequency and time, and $\psi(\omega, t)$ describes the phase of the spectrum

received by the diversity antenna referenced to the phase of the spectrum received by the main antenna. Both parameters $\delta(\omega, t)$ and $\psi(\omega, t)$ must be frequency- as well as time-dependent since during anomalous propagation periods the spectral densities received by the two antennas will, in general, differ by a complex, frequency-dependent, time-varying factor.

Assuming ideal waveguide systems and ideal phase-shifting elements, the spectral densities at the inputs of the summing hybrid (Fig. 1) can be expressed as:

$$\bar{R}'_M(j\omega, t) = \bar{R}_M(j\omega, t)e^{-j\omega\tau_{WG}} \tag{42}$$

and

$$\bar{R}'_D(j\omega, t) = \bar{R}_D(j\omega, t)e^{j(\theta+\alpha\sin\omega_m t-\omega\tau_{WG})}, \tag{43}$$

where $\tau_{WG}$ denotes the waveguide propagation delay (assumed the same for both equalized waveguide systems), and $\theta + \alpha \sin \omega_m t$ represents the instantaneous frequency-independent phase shift introduced into the diversity signal by the phase modulator/phase shifter combination. It can be seen from eqs. (40) through (43) that at the inputs of the summing hybrid the instantaneous phase difference between the diversity antenna spectrum and the main antenna spectrum is $\psi(\omega, t) + \theta + \alpha \sin \omega_m t$. The combined signal spectrum can now be expressed as

$$\begin{aligned}\bar{R}_C(j\omega, t) \equiv V_C(\omega, t)e^{j\phi_C(\omega,t)} &= \frac{1}{\sqrt{2}}[\bar{R}'_M(j\omega, t) + \bar{R}'_D(j\omega, t)] \\ &= \frac{1}{\sqrt{2}}[1 + \delta(\omega, t)e^{j\{\psi(\omega,t)+\theta+\alpha\sin\omega_m t\}}] \\ &\quad \cdot V(\omega, t)e^{j\{\phi(\omega)+\eta(\omega,t)-\omega(T_0+\tau_{WG})\}}.\end{aligned} \tag{44}$$

Expanding the exponential function inside the square brackets in terms of sine and cosine functions, followed by an expansion of $\sin\{\alpha \sin \omega_m t\}$ and $\cos\{\alpha \sin \omega_m t\}$ in terms of Bessel functions, yields (after some algebraic manipulation),

$$\begin{aligned}&V_C^2(\omega, t) \\ &\quad = \frac{1}{2} V_{co}^2(\omega, t)[1 - m_1(\omega, t)\sin \omega_m t + m_2(\omega, t)\cos 2\omega_m t \cdots],\end{aligned} \tag{45}$$

where

$$V_{co}^2(\omega, t) \equiv \delta(\omega, t)V^2(\omega, t)[\delta(\omega, t) + \frac{1}{\delta(\omega, t)} + 2J_0(\alpha)\cos \mu(\omega, t)], \tag{46}$$

$$m_1(\omega, t) \equiv \frac{4J_1(\alpha)\sin \mu(\omega, t)}{\delta(\omega, t) + \dfrac{1}{\delta(\omega, t)} + 2J_0(\alpha)\cos \mu(\omega, t)}, \tag{47}$$

$$m_2(\omega, t) \equiv \frac{4J_2(\alpha)\cos \mu(\omega, t)}{\delta(\omega, t) + \dfrac{1}{\delta(\omega, t)} + 2J_0(\alpha)\cos \mu(\omega, t)}, \cdots, \tag{48}$$

$$\mu(\omega, t) \equiv \psi(\omega, t) + \theta, \tag{49}$$

and $J_0$, $J_1$, $J_2$ denote Bessel functions of the first kind. Note that the modulation component, $m_1(\omega, t)$, at the fundamental frequency of phase modulation, $\omega_m$, is proportional to the sine of the average phase difference $\mu(\omega, t)$. [The average phase difference is obtained by averaging the instantaneous phase difference, $\psi(\omega, t) + \theta + \alpha \sin \omega_m t$, over one period of $\alpha \sin \omega_m t$. Over this time interval the term $\psi(\omega, t) + \theta$ is assumed fixed.] Furthermore, since $\alpha$ (the magnitude of phase modulation) is typically kept small, $0 < \alpha \ll \pi/2$, $J_0(\alpha)$ will be positive and less than unity. Also, the denominator term of $m_1(\omega, t)$ will always be positive since $-2 < 2J_0(\alpha)\cos \mu(\omega, t) < 2$ for all $\mu(\omega, t)$, given that $0 < \alpha \ll \pi/2$, and $\delta(\omega, t) + 1/\delta(\omega, t) \geq 2$ for all $\delta(\omega, t)$, $0 \leq \delta(\omega, t_0) \leq \infty$. Therefore, the sign of $m_1(\omega, t)$ will indicate whether, at a particular frequency and point in time, the phase of the diversity spectrum leads or lags the phase of the main antenna spectrum.

The combined-signal power contained in the frequency interval $\omega_1 \leq \omega \leq \omega_2$ can be expressed as:

$$p_C(t) = \frac{1}{\pi} \int_{\omega_1}^{\omega_2} V_C^2(\omega, t) d\omega. \tag{50}$$

Substituting eq. (45) into eq. (50) and manipulating terms results in:

$$p_C(t) = p_0(t)[1 - m_1'(t)\sin \omega_m t + m_2'(t)\cos 2\omega_m t \cdots] \tag{51}$$

where

$$p_0(t) \equiv \frac{1}{2\pi} \int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t) d\omega, \tag{52}$$

$$m_1'(t) \equiv \frac{\displaystyle\int_{\omega_1}^{\omega_2} m_1(\omega, t) V_{co}^2(\omega, t) d\omega}{\displaystyle\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t) d\omega}, \tag{53}$$

$$m_2'(t) \equiv \frac{\int_{\omega_1}^{\omega_2} m_2(\omega, t) V_{co}^2(\omega, t) d\omega}{\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t) d\omega}, \quad \cdots . \tag{54}$$

Note that $m_1'(t)$ represents a weighted average of $m_1(\omega, t)$ over the interval $\omega_1 \leq \omega \leq \omega_2$, where the weighting function $V_{co}^2(\omega, t)$ is proportional to the average combined-signal power spectral density.

In the absence of channel dispersion (i.e., flat fading, or no fading at all) $\mu$ and $m_1$ become frequency independent, thus making $m_1'$ identically equal to $m_1$ [see eq. (53)]. Since $m_1$ is proportional to $\sin \mu$, the sign of $m_1'$ can be used to determine whether $\mu$ represents a positive or negative angle (phase lead or phase lag of the diversity signal with respect to the main antenna signal) during dispersion-free periods. When the channel is dispersive, however (i.e., during periods of frequency-selective fading), the phase difference between the two spectra becomes frequency-dependent and $m_1'$ will reflect a weighted average of $\sin \mu$ over the interval $\omega_1 \leq \omega \leq \omega_2$. This can be seen by substituting eqs. (46) and (47) into eq. (53). Doing so, we find

$$m_1'(t) = \frac{4J_1(\alpha) \int_{\omega_1}^{\omega_2} \delta(\omega, t) V^2(\omega, t) \sin \mu(\omega, t) d\omega}{\int_{\omega_1}^{\omega_2} V_{co}^2(\omega, t) d\omega}, \tag{55}$$

where the weighting function $\delta(\omega, t) V^2(\omega, t)$ is the magnitude of the product of the received spectral densities. In Section 2.1 we show that when $\omega_1 \leq \omega \leq \omega_2$ covers the entire channel bandwidth and $m_1'(t)$ is used in a feedback arrangement to control the phase-shifter value $\theta$, maximum-power combining of the received signals results for all channel conditions.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Stochastic Analysis of Mechanizing Transaction Data Bases

By J. A. MORRISON and W. W. YALE

(Manuscript received February 16, 1982)

*In this paper we consider the transfer of records from a manual file system (MFS) to a mechanized data base management system when the conversion takes place through two processes. In the going-forward process a transfer is made by the regular work force when an order is received to change or delete a record in the MFS. In addition, the transfer of records is carried out by a crash force, working at a fixed rate. We investigate the expected number of records remaining in the MFS at future times, and the expected number of records removed from the MFS during the corresponding periods by the going-forward work force and by the crash force. We also derive the expected time taken to go from a prescribed number of records in the MFS to a smaller number. We give simple approximations for all of these quantities. The results have been used elsewhere in the construction of an economic model used to estimate cost/benefits and labor force levels during the mechanization of transaction data bases. The numerical results presented graphically are typical of two Bell System applications.*

## I. INTRODUCTION

In this paper, we analyze the generic problem of implementing a system that requires manual effort for converting paper records in a manual file system (MFS) to mechanized computer records in a data base management system (DBMS). We can achieve conversion to a mechanized environment in a number of ways: by going-forward on an activity basis, by crash-conversion utilizing an augmented work force, or by some mix of the two dictated by available resources such as money, people, space, and time objectives.

A paper record is mechanized using the going-forward approach if the record is transferred from the MFS to the DBMS by regular record

maintenance employees when a change or deletion activity is performed upon that record. If a randomly chosen record is transferred by a separate temporary work force, the record is said to be transferred using the crash approach. The paper record could contain all of the information that is intended to be mechanized or an enhancement to an already existing mechanized record.

The arrival times of activities (change or deletion orders) for a particular record are assumed to form a Poisson process. We further assume that the distributions of the activity arrival times for every record are independent and identical. Hence, the sequence of times at which these change and deletion activities occur forms a nonstationary Poisson process. In addition, the transfer of records is carried out by a special task force with constant total mean rate, the amount of work required for a record to be transferred being exponentially distributed. Hence, there are two different stochastic processes that are concurrently operating against the MFS. Each record in the MFS is transferred if an activity occurs upon it, while at the same time each record in the MFS has a chance of randomly being selected by a crash person when he/she completes a conversion.

When an activity occurs against a record in the MFS, it will be assumed that a going-forward person will immediately attend to transferring and processing the record. Therefore, if substantially more than the average number of records concurrently experience activity, it will be assumed that management will borrow going-forward people from other areas or have the normal going-forward force work overtime. However, the time it takes for a crash-conversion person to perform the conversion is pertinent since the force is assumed to be fixed. In this case, the average number of records that the entire crash conversion force can convert in a specified period determines the mean of the stochastic process describing the crash effort.

In this paper we highlight the derivations of formulas that are used in the construction of a model[1] used to estimate cost/benefits and labor force levels during the mechanization of transaction data bases. In the next section, we investigate the expected number of records remaining in the MFS, removed by the going-forward people, and removed by the crash force people. In particular, we point out a simple approximation to the expected number of records remaining in the MFS, which is valid over a significant part of the range of interest. We derive the expected passage time from $l$ to $m$ records in the MFS ($l > m$) in Section III, and obtain a simple asymptotic approximation for the expected time at which the MFS becomes empty. Finally, in Section IV, we present some conclusions, and discuss the relevance of the results to Bell System applications.

## II. EXPECTED NUMBER OF RECORDS

We will now consider a stochastic model for the transfer of records from a MFS to a DBMS. We assume that there are $n$ crash force people, and that the conversion times of the records are independent and exponentially distributed with mean rate $\rho$. At time $\tau = 0-$ there are $S$ records in the MFS. At time $\tau = 0$ there are $n$ of these records removed from the MFS by the crash force. When the conversion of a record is completed, that record is instantaneously transferred to the DBMS, and another record is removed from the MFS by the crash person involved. A record is also removed from the MFS, by a going-forward person, when a change or deletion order is received; that record is transferred to the DBMS at some later time. We assume that the sequence of times at which change and deletion orders are received forms a nonstationary Poisson process with rate $i\mu$ where $i$ is the number of records in the MFS. (The mean deletion and change order rates are summed to get $\mu$, since it is a well-known fact that if two independent Poisson processes are affecting the records, then the composite of these two processes can be described by a Poisson process where the parameters are summed.) If a change or deletion order is received for a record that is in the process of being converted by a crash person, the change or deletion order is not executed until after the record has been transferred to the DBMS.

The number of records in the MFS at time $\tau = 0+$ is

$$M = S - n. \tag{1}$$

For convenience, we let

$$n\rho = \nu\mu. \tag{2}$$

This is the mean rate at which the crash force converts records. We also let $P_i(\tau)$ denote the probability that there are $i$ records in the MFS at time $\tau$. But, the number of records in the MFS is a pure death process,[2] with death rate

$$\mu_i = n\rho(1 - \delta_{i0}) + i\mu = \mu[\nu(1 - \delta_{i0}) + i], \qquad i = 0, \cdots, M, \tag{3}$$

where $\delta_{ij}$ denotes the Kronecker delta; $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ if $i \neq j$. Hence,

$$\frac{dP_i}{d\tau} = -\mu[\nu(1 - \delta_{i0}) + i]P_i + \mu(1 - \delta_{iM})(\nu + i + 1)P_{i+1}, \tag{4}$$

for $\tau > 0$ and $i = 0, \cdots, M$. The initial condition at time $\tau = 0+$ is

$$P_i(0+) = \delta_{iM}, \qquad i = 0, \cdots, M. \tag{5}$$

The expected number of records in the MFS at time $\tau$ is

$$N(\tau) = \sum_{i=1}^{M} iP_i(\tau). \tag{6}$$

It follows from (4) through (6) that

$$\frac{dN}{d\tau} + \mu N = -\nu\mu[1 - P_0(\tau)]; \qquad N(0+) = M. \tag{7}$$

It is straightforward to calculate the Laplace transforms of $P_i$, $i = 0, \cdots, M$, from (4) and (5), and thence that of $N$ from (7). By inverting the latter transform, Morrison[3] obtained an explicit expression for $N(\tau)$ in terms of an incomplete beta function.[4] Asymptotic approximations to $N(\tau)$, involving the complementary error function,[4] were derived when $M \gg 1$ and $\nu \gg 1$.

Since $0 \le P_0(\tau) \le 1$, it follows from (7) that

$$L(\tau) \equiv (M + \nu)e^{-\mu\tau} - \nu \le N(\tau) \le Me^{-\mu\tau}. \tag{8}$$

The lower bound is, of course, superfluous if $L(\tau) \le 0$. When $M \gg 1$ and $\nu \ll M$, these bounds yield an accurate approximation to $N(\tau)$ as long as $Me^{-\mu\tau} \gg \nu$. An upper bound for $N(\tau)$, which is valid only for $L(\tau) \ge 0$, and exact for $L(\tau) = 0$, was derived from the exact result.[3] It was shown that

$$0 \le N(\tau) - L(\tau) \le \frac{\Gamma(M + \nu)}{\Gamma(M)\Gamma(\nu)} e^{-\nu\mu\tau}(1 - e^{-\mu\tau})^M \quad \text{for} \quad L(\tau) \ge 0. \tag{9}$$

From (9) we deduce that

$$N(\tau) \approx L(\tau) \quad \text{for} \quad M \gg 1, \qquad L(\tau) \gg \min(\sqrt{M}, \sqrt{\nu}) \frac{\nu^\nu e^{-\nu}}{\sqrt{\nu}\Gamma(\nu)}, \tag{10}$$

where, from the properties of the gamma function,[4]

$$\frac{\nu^\nu e^{-\nu}}{\Gamma(\nu)} < \nu \quad \text{for} \quad \nu > 0; \qquad \frac{\nu^\nu e^{-\nu}}{\sqrt{\nu}\Gamma(\nu)} \sim \frac{1}{\sqrt{2\pi}} \quad \text{for} \quad \nu \gg 1. \tag{11}$$

Hence, (10) gives a larger range of validity for the approximation $N(\tau) \approx L(\tau)$ than does (8), and significantly so if $\nu \gg 1$, as it is in cases of interest. We remark that intuitively we expect $P_0(\tau)$, the probability that there are no records in the MFS at time $\tau$, to be extremely small for a significant time when $M \gg 1$, and in that time interval we deduce from (7) that $N(\tau) \approx L(\tau)$. The intuitive derivation, however, does not give the range of validity of the approximation.

Typical numerical results are illustrated in Figs. 1 and 2 for the case $S = 10^5$, $n = 10$, $\mu = 0.001608$ per day, and $\nu = 18657$. Only $N(\tau)$ and the lower bound $L(\tau)$ are shown in Fig. 1. The upper bound given by (9) is also shown in Fig. 2. This figure depicts the tail region, in which the expected number of records in the MFS is considerably fewer than

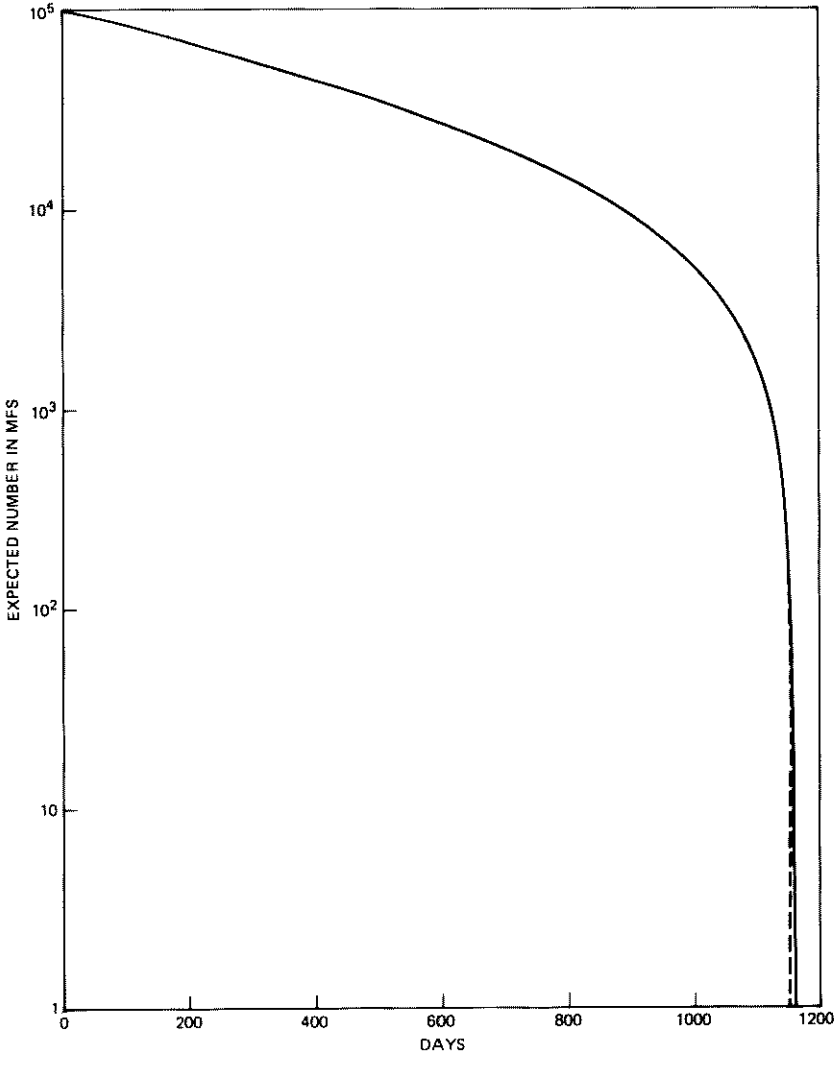



Fig. 1—Expected number of records in the MFS, and lower bound, as a function of time, for $S = 10^5$, $n = 10$, $\mu = 0.001608$ per day, and $\nu = 18657$.

the initial number. It is seen that $N(\tau)$ is still quite close to $L(\tau)$ for values of $L(\tau)$ just a few times greater than $\min(\sqrt{M}, \sqrt{\nu})/\sqrt{2\pi} \approx 54.5$.

Another quantity of interest is the expected number of records $F(\tau)$ removed from the MFS by the going-forward people in the time interval $(0, \tau]$. Let $P_{ik}(\tau)$ denote the probability that there are $i$ records in the MFS at time $\tau$, and that $k$ records have been removed from the MFS by the going-forward people in $(0, \tau]$. Then

$$F(\tau) = \sum_{i=0}^{M-1} \sum_{k=1}^{M-i} kP_{ik}(\tau). \tag{12}$$

But,

$$\frac{dP_{ik}}{d\tau} = -\mu[\nu(1 - \delta_{i0}) + i]P_{ik} + \mu\nu(1 - \delta_{k,M-i})P_{i+1,k} + \mu(1 - \delta_{k0})(i + 1)P_{i+1,k-1}, \tag{13}$$

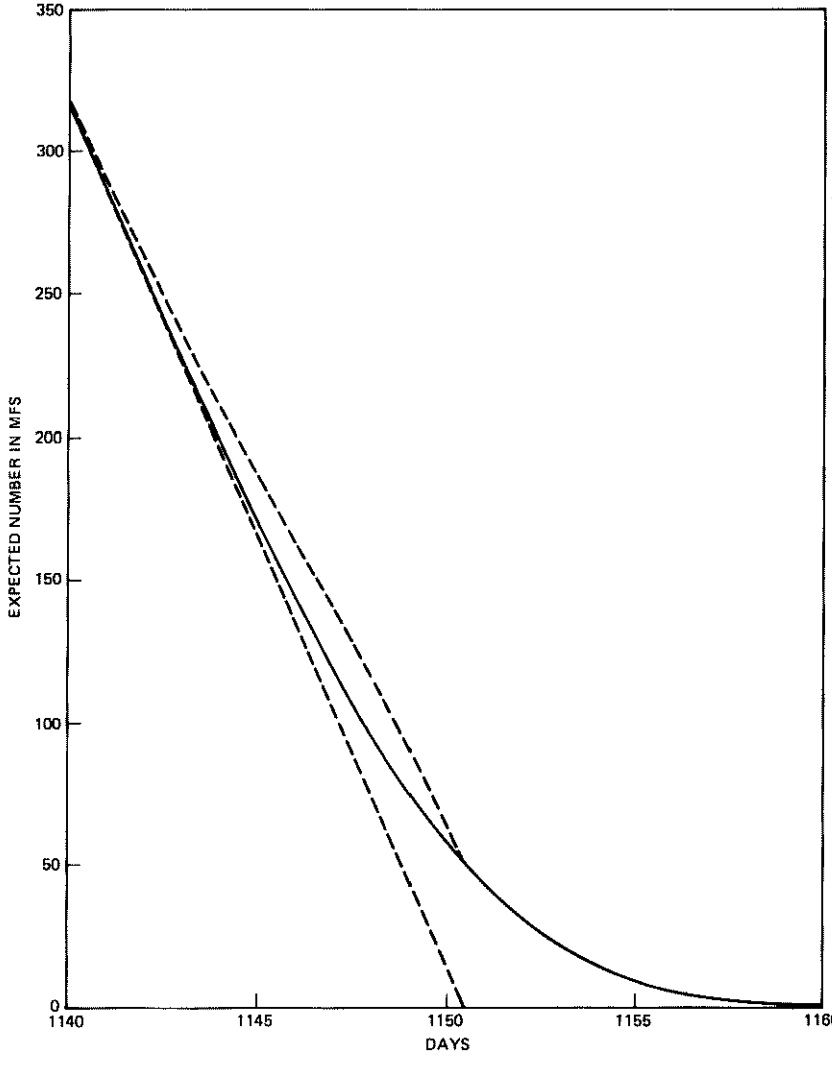



Fig. 2—Expected number of records in the MFS, and upper and lower bounds, as a function of time, for $S = 10^5$, $n = 10$, $\mu = 0.001608$ per day, and $\nu = 18657$.

for $\tau > 0$, $k = 0, \cdots, M - i$, and $i = 0, \cdots, M$. The initial condition at time $\tau = 0+$ is

$$P_{ik}(0+) = \delta_{iM}, \qquad k = 0, \cdots, M - i, \qquad \text{and} \qquad i = 0, \cdots, M. \tag{14}$$

We note that $i = M$ implies that $k = 0$. Also,

$$P_i(\tau) = \sum_{k=0}^{M-i} P_{ik}(\tau), \tag{15}$$

and if we sum (13) and (14) from $k = 0$ to $M - i$ we obtain (4) and (5).

It follows directly from (12), (13), and (15) that

$$\frac{dF}{d\tau} = \mu \sum_{i=1}^{M} iP_i(\tau) = \mu N(\tau). \tag{16}$$

This implies that the rate of change of the expected number of records removed from the MFS by the going-forward people is equal to the expected value at time $\tau$ of the rate at which change and deletion orders are received. If we let $C(\tau)$ denote the expected number of records removed from the MFS by the crash people in $(0, \tau]$, then

$$C(\tau) = M - N(\tau) - F(\tau). \tag{17}$$

From (7), (16), and (17), we obtain

$$\frac{dC}{d\tau} = \mu\nu[1 - P_0(\tau)] = \mu\nu \sum_{i=0}^{M} (1 - \delta_{i0})P_i(\tau), \tag{18}$$

which has an interpretation analogous to that of (16). We note that $dC/d\tau \approx \mu\nu$ in the region where $P_0(\tau)$ is small. Also, corresponding to the approximation $N(\tau) \approx L(\tau)$, where $L(\tau)$ is defined in (8), it follows from (16) that

$$F(\tau) \approx (M + \nu)(1 - e^{-\mu\tau}) - \mu\nu\tau, \tag{19}$$

since $F(0+) = 0$.

## III. EXPECTED PASSAGE TIMES

We next turn to the calculation of the expected passage time from $l$ to $m$ records in the MFS. Now,[5] for a pure death process, with death rate $\mu_i$, the passage time $\tau_{i,i-1}$ from state $i$ to state $i - 1$ is exponentially distributed with density $\mu_i \exp(-\mu_i\tau)$. Hence, in particular, the expected value of $\tau_{i,i-1}$ is

$$E\tau_{i,i-1} = \frac{1}{\mu_i}. \tag{20}$$

The passage time from $l$ to $m$ records in the MFS is

$$\tau_{l,m} = \sum_{i=m+1}^{l} \tau_{i,i-1}, \tag{21}$$

and the random variables in the sum are independent. From (3), (20), and (21), we obtain

$$\mu E\tau_{l,m} = \sum_{i=m+1}^{l} \frac{1}{(\nu + i)} = \psi(l + \nu + 1) - \psi(m + \nu + 1), \tag{22}$$

where $\psi$ denotes the logarithmic derivative of the gamma function.[4]

We will make use of the asymptotic result

$$\psi(x) = \log x + O\left(\frac{1}{x}\right) \qquad \text{for} \quad x \gg 1. \tag{23}$$

Hence, from (22), the expected time at which the MFS becomes empty is given by

$$\begin{aligned} \mu E\tau_{M,0} &= \psi(M + \nu + 1) - \psi(\nu + 1) \\ &\sim \log\left(\frac{M + \nu}{\nu}\right) \qquad \text{for} \quad \nu \gg 1. \end{aligned} \tag{24}$$

Also, the expected time starting from $l$ records until the MFS becomes empty is given by

$$\mu E\tau_{l,0} = \psi(l + \nu + 1) - \psi(\nu + 1) \sim \log\left(\frac{l + \nu}{\nu}\right) \qquad \text{for} \quad \nu \gg 1$$

$$\sim \frac{1}{\nu} \qquad \text{for} \quad \nu \gg 1, \qquad l \ll \nu. \tag{25}$$

Next, the expected time at which the number of records in the MFS first reaches $m$ is given by

$$\begin{aligned} \mu E\tau_{M,m} &= \psi(M + \nu + 1) - \psi(m + \nu + 1) \\ &\sim \log\left(\frac{M + \nu}{m + \nu}\right) \qquad \text{for} \quad \nu \gg 1. \end{aligned} \tag{26}$$

But, from (8) to (10), the time $\tau_m$ at which the expected number of records in the MFS is equal to $m$ satisfies

$$\mu\tau_m \sim \log\left(\frac{M + \nu}{m + \nu}\right)$$

$$\text{for} \quad M \gg 1, \qquad \nu \gg 1, \qquad m \gg \min(\sqrt{M}, \sqrt{\nu})/\sqrt{2\pi}. \tag{27}$$

Hence, under the restrictions in (27), we have $E\tau_{M,m} \sim \tau_m$.

We now consider the final stage of the conversion by the crash force, starting from the time $\tau_{M,0}$ when the MFS becomes empty. Each of the $n$ crash people is then busy with a record. We first assume that a person is removed from the crash force when he/she completes the conversion of the record on which he/she has been working. If $j$ is the number of records remaining to be converted by the crash force, we then have a pure death process with $\mu_j = j\rho, j = 0, \cdots, n$. If $\tau_c$ denotes

the time at which the conversion by the crash force is completed, then, with the help of (2),

$$\mu E(\tau_c - \tau_{M,0}) = \mu \sum_{j=1}^{n} \frac{1}{\mu_j} = \frac{n}{\nu} \sum_{j=1}^{n} \frac{1}{j} = \frac{n}{\nu} [\psi(n+1) - \psi(1)]. \quad (28)$$

At the other extreme, we assume that the entire crash force works jointly on the remaining records, until the conversion of the last record is completed. This will give a lower bound on the expected time to complete the conversion of the final $n$ records, since the crash force continues to work at its maximum rate. In this case the death rate is $\hat{\mu}_j = n\rho = \nu\mu$, $j = 1, \cdots, n$. If $\hat{\tau}_c$ denotes the corresponding time at which the conversion by the crash force is completed, then

$$\mu E(\hat{\tau}_c - \tau_{M,0}) = \mu \sum_{j=1}^{n} \frac{1}{\hat{\mu}_j} = \frac{n}{\nu}. \quad (29)$$

## IV. CONCLUSION

In this paper we presented formulas that are employed in an economic model[1] used to estimate cost/benefits and labor force levels associated with the mechanization of transaction data bases. Because the formulas can be computed very rapidly on any modern computer, the model can be used as an interactive managerial decision tool to perform what-if studies in real time. Numerical results presented graphically are typical of two applications, namely, (*i*) the conversion of manual records in a service order processing operation of a Bell Operating Company business office, and (*ii*) the mechanization of equipment inventory records during the implementation of Trunks Integrated Record Keeping System.[1] We do not presume that the stochastic model is appropriate for all possible data base conversions, but evidence from field trials indicates that it does adequately model a certain class of Bell System mechanization problems.

## V. ACKNOWLEDGMENTS

We would like to thank T. A. Bottomley for many helpful contributions concerning the definition of the problem, and J. B. Seery for writing the programs to obtain the numerical results presented in Figs. 1 and 2.

## REFERENCES

1. W. W. Yale, unpublished work.
2. L. Kleinrock, *Queueing Systems, Volume I: Theory*, New York: John Wiley, 1975.
3. J. A. Morrison, unpublished work.
4. W. Magnus, F. Oberhettinger, and R. P. Soni, *Formulas and Theorems for the Special Functions of Mathematical Physics*, New York: Springer-Verlag, 1966.
5. J. Keilson, *Markov Chain Models—Rarity and Exponentiality*, Applied Mathematical Sciences, *28*, New York: Springer-Verlag, 1979, p. 21.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Smoothing With Periodic Cubic Splines

By N. Y. GRAHAM

(Manuscript received May 7, 1982)

*In this paper we present a mathematical algorithm for constructing a smoothing cubic spline with periodic end conditions and a predetermined 'closeness of fit' to a given set of points in the plane. In addition to providing a mathematical tool for smoothing raw data in which the underlying function is known to be periodic, this algorithm has special significance in computer graphics, because the use of smoothing functions with periodic end conditions is essential for producing visually acceptable, smooth, closed curves. Sample plots are included to illustrate the power and flexibility of this algorithm.*

## I. INTRODUCTION

Although "natural" splines are used extensively and are quite appropriate for smoothing many types of data, they often produce less than satisfactory results when used to smooth data points that belong to a periodic function. The inappropriateness of using "natural" splines to approximate periodic data is especially evident in graphics applications. In particular, when the data points represent a closed curve, smoothing (parametrically) with "natural" splines will lead to unacceptable results because the "natural" end conditions will cause the curve either to close up with a noticeable discontinuity, or to not close up at all (see Fig. 1).

Existing methods for constructing smoothing splines with a predetermined closeness of fit all lead to splines with "natural" end conditions.[1,2] A method developed by Spath[3] produces a smoothing spline with periodic end conditions, but the closeness of fit cannot be determined in advance. In this paper we will describe a method for constructing a smoothing cubic spline that has periodic end conditions and that also satisfies a predetermined closeness of fit to a given set of data points.

This algorithm has potentially wide applicability, especially in the realm of interactive graphics. It makes possible the computer genera-

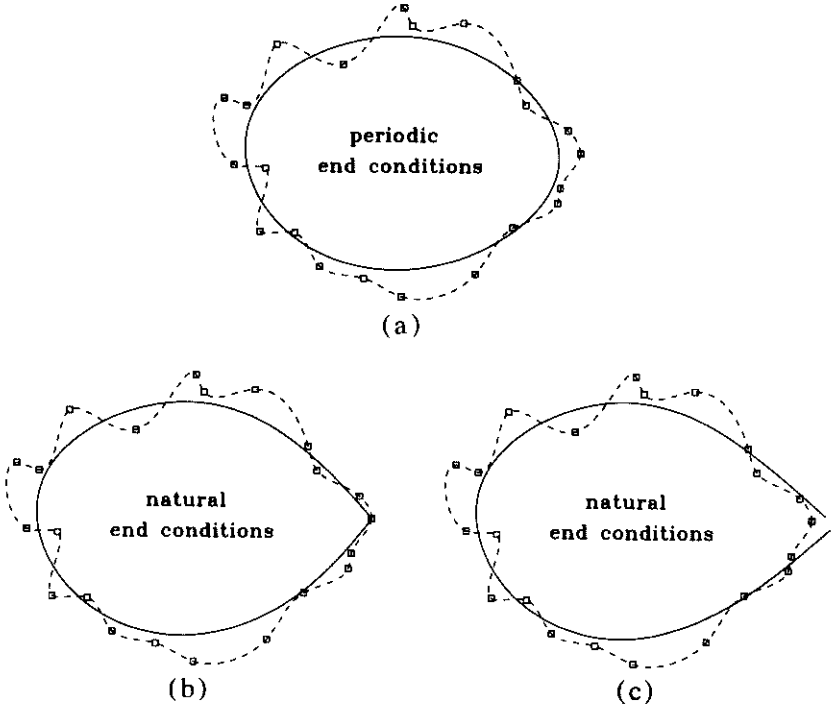



Fig. 1—Comparison of periodic versus natural spline smoothing. (a) Periodic cubic spline smoothing with uniform weights. (b) Natural cubic spline smoothing with small weights at end points. (c) Natural cubic spline smoothing with uniform weights.

tion of free-form, smooth, closed curves by merely specifying the approximate locations of as few as three distinct points. The shape of the curve can be controlled easily by moving one or more points, or by adjusting the weighting factors associated with some or all of the points.

An efficient program based on this new algorithm has been written and tested. Sample plots illustrating this method are included.

## II. TERMINOLOGY

Let $P_k = (x_k, y_k)$, $k = 1, n$, be $n$ points in the plane. A "cubic spline" on $[x_1, x_n]$ with knots at $x_1, \cdots, x_n$, is a function $f$ that coincides with a third-order polynomial $f_k$ on each sub-interval $[x_k, x_{k+1}]$, $k = 1, n - 1$, and such that $f$ is continuous and has continuous first and second derivatives over the entire interval $[x_1, x_n]$.

In other words, $f$ is a cubic spline on $[x_1, x_n]$ if, for each $k = 1, n - 1$ there exist real numbers $a_k$, $b_k$, $c_k$, $d_k$ (the "spline coefficients" of $f$) such that for every $x$ in $[x_k, x_{k+1}]$,

$$f(x) = f_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3. \quad (1)$$

Furthermore, the continuity of $f$, $f'$, and $f''$ on $[x_1, x_n]$ implies that at each interior knot $x_k$, $k = 2, n - 1$,

$$f_{k-1}(x_k) = f_k(x_k), \quad (2)$$

$$f'_{k-1}(x_k) = f'_k(x_k), \quad (3)$$

$$f''_{k-1}(x_k) = f''_k(x_k). \quad (4)$$

The cubic spline $f$ is said to be "periodic" if it satisfies the following additional conditions (known as "periodic end conditions"):

$$f(x_n) = f(x_1), \quad (5)$$

$$f'(x_n) = f'(x_1), \quad (6)$$

$$f''(x_n) = f''(x_1). \quad (7)$$

A "natural" cubic spline differs from a "periodic" cubic spline in that it satisfies the so-called "natural end conditions": $f''(x_1) = f''(x_n) = 0$.

## III. FORMAL STATEMENT OF THE PROBLEM

Let $P_k = (x_k, y_k)$, $k = 1, n$, be $n$ points in the plane, with $x_1 < x_2 < \cdots < x_n$. Let $w_k$, $k = 1, n$, be positive real numbers ("weighting factors") associated with $P_k$, $k = 1, n$, respectively. (Assume $y_n = y_1$, $w_n = w_1$.) Given an arbitrary constant, $M > 0$, the problem is to determine the set of $4(n - 1)$ coefficients of the periodic cubic spline $f$ on $[x_1, x_n]$ with knots at $x_1, \cdots, x_n$, such that

(*i*) $f$ has minimal "total curvature" $G(f) = \int_{x_1}^{x_n} f''(x)^2 dx$, and

(*ii*) $f$ satisfies the following weighted, distance-squared constraint, or "closeness of fit," with respect to the given points:

$$H(f) = \sum_{k=1}^{n} \left[ \frac{f(x_k) - y_k}{w_k} \right]^2 \leq M.$$

Note that the weighting factors give inverse importance to the points. Note also that $M$ controls the degree of smoothness, so that increasing the value of $M$ for a fixed set of weighting factors will lead to a smoother, or flatter, spline. Conversely, choosing a sufficiently small value of $M$ will lead to a spline that closely approximates an interpolating spline. In general, an appropriate choice for the value of $M$ will depend on the values chosen for the weighting factors. If, for example, the weighting factors are chosen so that $w_k$ is the standard deviation at $x_k$, then a suitable choice for $M$ would be some value between the confidence limits, $n - \sqrt{2n}$ and $n + \sqrt{2n}$.

## IV. GENERAL APPROACH TO THE SOLUTION

To minimize $G(f)$ subject to the constraint $H(f) \leq M$, we introduce an auxiliary variable $z$ and a non-negative Lagrange multiplier $p$ and minimize

$$F(f) = G(f) + p[H(f) + z^2 - M].$$

This is the approach used by Reinsch in Ref. 2 to solve the analogous problem for "natural" cubic splines. However, Reinsch minimizes $F$ over the class of all continuous functions with continuous first and second derivatives on $[x_1, x_n]$, which leads to a "natural" cubic spline as the solution. We will restrict the class of admissible functions in the minimization of $F$ to periodic cubic spline splines on $[x_1, x_n]$ with knots at $x_1, \cdots, x_n$, obtaining a direct solution to our problem.

## V. LINEAR SYSTEM RESULTING FROM CONTINUITY AND PERIODICITY CONDITIONS

Let $f$ be an arbitrary periodic cubic spline on $[x_1, x_n]$, with spline coefficients $a_k$, $b_k$, $c_k$, $d_k$, $k = 1, n-1$. Then, for $x$ in $[x_k, x_{k+1}]$, $f(x)$ is given by (1) above, and $f'(x)$ and $f''(x)$ are given below:

$$f'(x) = f_k'(x) = b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2, \tag{8}$$

$$f''(x) = f_k''(x) = 2c_k + 6d_k(x - x_k). \tag{9}$$

Expressing $f$, $f'$, and $f''$ explicitly in eqs. (1), (8), and (9), respectively, allows us to derive linear relationships among the spline coefficients. From the continuity of $f'$ at the interior knots of (3) and the periodicity of $f'$ on $[x_1, x_n]$ in (6), it follows that:

$$2c_k h_k = b_{k+1} - b_k - 3d_k h_k^2, \qquad \text{for} \quad k = 1, n-1, \tag{10}$$

where $h_k = x_{k+1} - x_k$ for $k = 1, n-1$, and $b_n$ denotes $b_1$.

From the continuity of $f$ at the interior knots of (2) and the periodicity of $f$ on $[x_1, x_n]$ in (5), the first-order coefficients may be expressed in terms of the constant, second-order, and third-order coefficients:

$$b_k = (a_{k+1} - a_k)/h_k - c_k h_k - d_k h_k^2, \qquad \text{for} \quad k = 1, n-1, \tag{11}$$

where $a_n$ denotes $a_1$.

From the continuity of $f''$ at the interior knots of (4) and the periodicity of $f''$ on $[x_1, x_n]$ in (7), the third-order coefficients may be expressed as a function of the second-order coefficients:

$$d_k = (c_{k+1} - c_k)/3h_k, \qquad \text{for} \quad k = 1, n-1, \tag{12}$$

where $c_n$ denotes $c_1$.

Using (11) and (12) to eliminate the $b_k$'s and $d_k$'s from (10) leads to a system of linear equations in the $a_k$'s and $c_k$'s, given in matrix notation as follows:

$$\mathbf{Sc} = 3\mathbf{Qa}, \tag{13}$$

where $\mathbf{S}$ and $\mathbf{Q}$ are symmetric, cyclic-tridiagonal matrices of order $n - 1$, and $\mathbf{c}$, $\mathbf{a}$ are the column vectors $(c_1, \cdots, c_{n-1})^T$, $(a_1, \cdots, a_{n-1})^T$, respectively.

The non-zero entries of $\mathbf{S}$ and $\mathbf{Q}$ are expressed in terms of the distances between successive knots:

$$\begin{aligned}
\mathbf{S}(k, k) &= 2(h_{k-1} + h_k) \qquad \text{for} \quad k = 1, \ n - 1 \\
\mathbf{S}(k, k+1) &= \mathbf{S}(k+1, k) = h_k \qquad \text{for} \quad k = 1, \ n - 2 \\
\mathbf{S}(1, n-1) &= \mathbf{S}(n-1, 1) = h_{n-1} \\
\mathbf{Q}(k, k) &= -1/h_{k-1} - 1/h_k \qquad \text{for} \quad k = 1, \ n - 1 \\
\mathbf{Q}(k, k+1) &= \mathbf{Q}(k+1, k) = 1/h_k \qquad \text{for} \quad k = 1, \ n - 2 \\
\mathbf{Q}(1, n-1) &= \mathbf{Q}(n-1, 1) = 1/h_{n-1},
\end{aligned}$$

where $h_0$ denotes $h_{n-1}$. By Gershgorin's Theorem[4] it can be shown that $\mathbf{S}$ is positive definite (and therefore non-singular), while $\mathbf{Q}$ is positive semi-definite and singular with rank $n - 2$.

## VI. LINEAR SYSTEM RESULTING FROM MINIMIZING $F$ WITH RESPECT TO THE CONSTANT COEFFICIENTS

Note that (13) is a system of $n - 1$ linear equations in $2(n - 1)$ unknowns: the constant coefficients and the second-order coefficients. We shall derive a second system of $n - 1$ linear equations in these unknowns. We proceed by first showing that $H$ and $G$ can be expressed as functions of the constant coefficients only.

From the spline representation of $f$ in (1), it follows immediately that

$$H = \sum_{k=1}^{n} \left[ \frac{f(x_k) - y_k}{w_k} \right]^2 = \sum_{k=1}^{n} \left( \frac{a_k - y_k}{w_k} \right)^2$$

is a function of $a_1, \cdots, a_n$. And since the periodicity of $f$ implies

$$a_n = f(x_n) = f(x_1) = a_1,$$

$H$ is a function of the constant spline coefficients $a_1, \cdots, a_{n-1}$.

From the explicit representation of $f''$ in (9), the "total curvature" $G$ can be expressed as

$$\begin{aligned}
G = \int_{x_1}^{x_n} f''(x)^2 dx &= \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} f''(x)^2 dx \\
&= \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} [2c_k + 6d_k(x - x_k)^2 dx.
\end{aligned}$$

Evaluating the integrals over each sub-interval directly and eliminating the $d_k$'s with (12) lead to the following:

$$G = \sum_{k=1}^{n-1} \frac{4}{3} h_k (c_k^2 + c_k c_{k+1} + c_{k+1}^2).$$

Rewriting in matrix notation and applying (13), we have:

$$G = (\tfrac{2}{3})\mathbf{c}^T\mathbf{S}\mathbf{c} = (\tfrac{2}{3})(3\mathbf{S}^{-1}\mathbf{Q}\mathbf{a})\mathbf{S}(3\mathbf{S}^{-1}\mathbf{Q}\mathbf{a}) = 6\mathbf{a}^T\mathbf{Q}\mathbf{S}^{-1}\mathbf{Q}\mathbf{a}.$$

Thus, $G$ also can be expressed in terms of the constant spline coefficients. Note that from this representation of $G$,

$$\frac{\partial G}{\partial a_k} = 12\mathbf{Q}_k\mathbf{S}^{-1}\mathbf{Q}\mathbf{a}$$

for $k = 1, n-1$, where $\mathbf{Q}_k$ is the $k$th row of $\mathbf{Q}$.

Since $G$ and $H$ are functions of $a_1, \cdots, a_{n-1}$, then $F$ is a function of the independent variables $a_1, \cdots, a_{n-1}, p$, and $z$. In order for $F$ to be minimized, the partial derivative of $F$ with respect to each of its independent variables must vanish. Thus, for each $k = 1, n-1$, differentiating $F$ with respect to $a_k$ yields:

$$\frac{\partial F}{\partial a_k} = \frac{\partial G}{\partial a_k} + p\frac{\partial H}{\partial a_k} = 12\mathbf{Q}_k\mathbf{S}^{-1}\mathbf{Q}\mathbf{a} + 2p\left(\frac{a_k - y_k}{w_k^2}\right) = 0.$$

Rewriting this set of $n-1$ linear equations in matrix notation and using (13) to replace $\mathbf{S}^{-1}\mathbf{Q}\mathbf{a}$ with $\mathbf{c}/3$ lead to:

$$4\mathbf{Q}\mathbf{c} + 2p\mathbf{W}^{-2}(\mathbf{a} - \mathbf{y}) = \mathbf{0}, \tag{14}$$

where $\mathbf{y}$ is the column vector $(y_1, \cdots, y_{n-1})^T$. Combining (13) and (14), we have the following linear system in $\mathbf{c}$:

$$(p\mathbf{S} + 6\mathbf{Q}\mathbf{W}^2\mathbf{Q})\mathbf{c} = 3p\mathbf{Q}\mathbf{y}. \tag{15}$$

The matrix $\mathbf{A}_p = (p\mathbf{S} + 6\mathbf{Q}\mathbf{W}^2\mathbf{Q})$ is symmetric, five-banded with three non-zero entries in the upper right and lower left corners. It can be shown that, for all positive values of $p$, $\mathbf{A}_p$ is positive definite. Thus, for each $p > 0$, (15) has a unique solution in $\mathbf{c}$.

Note that for each non-zero value of $p$, $\mathbf{a} = \mathbf{y} - (2/p)\mathbf{W}^2\mathbf{Q}\mathbf{c}$ from (14). Note also that from (12) $\mathbf{d}$ is uniquely determined by $\mathbf{c}$, and from (11) $\mathbf{b}$ is uniquely determined by $\mathbf{a}$, $\mathbf{c}$, and $\mathbf{d}$. Thus, to each positive value of $p$ corresponds a unique periodic cubic spline on $[x_1, x_n]$, whose coefficients are given in the vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$.

## VII. CONSEQUENCES OF MINIMIZING $F$ WITH RESPECT TO $p$ AND $z$

Minimizing $F = G + p(H + z^2 - M)$ with respect to the Lagrange multiplier $p$ leads to:

$$\frac{\partial F}{\partial p} = H + z^2 - M = 0,$$

which merely states that the distance constraint on $H$ (expressed as an equality in terms of the auxiliary variable $z$) must be satisfied when the minimal value of $F$ is attained.

On the other hand, minimizing $F$ with respect to $z$ yields:

$$\frac{\partial F}{\partial z} = 2pz = 0,$$

which implies that at least one of the two variables, $p$ or $z$, must be equal to 0 when the minimal value of $F$ is attained.

Note that if $p = 0$ when $F$ is minimized, then $F = G$. Since $G = (2/3)\mathbf{c}^T\mathbf{S}\mathbf{c}$ and $\mathbf{S}$ is positive definite, then $G$ is minimized when $\mathbf{c} = \mathbf{0}$. This in turn implies $\mathbf{d} = \mathbf{0}$, so that the second- and third-order coefficients vanish, resulting in a piecewise linear minimizing spline. The properties of being piecewise linear and having a continuous first derivative together imply that the minimizing spline is a straight line. Furthermore, periodicity of the spline implies that the straight line is in fact horizontal.

On the other hand, if $p > 0$ when $F$ is minimized, then $z = 0$, so that $H = M$. Since $\mathbf{a} = \mathbf{y} - (2/p)\mathbf{W}^2\mathbf{Q}\mathbf{c}$ from (14), and $\mathbf{c} = 3p\mathbf{A}_p^{-1}\mathbf{Q}\mathbf{y}$ from (15), $H$ can be expressed as a function of $p$. Thus, if minimization of $F$ occurs for a positive value of $p$, it remains to determine the value of $p$ for which $H(p) = M$.

## VIII. PROPERTIES OF $H$ AS A FUNCTION OF $p$

The following facts can be established: for all positive values of $p$, $H(p)$ is a continuous, convex function of $p$ with negative slope. Furthermore, as $p$ approaches zero from the right, $H(p)$ becomes arbitrarily large.

## IX. ALGORITHM FOR DETERMINING SPLINE COEFFICIENTS

We can now state the following algorithm for determining the minimizing spline. Compute the equation of the horizontal straight line with the least-squares fit to the given data points:

$$f(x) = \left(\sum_{k=1}^{n} \frac{y_k}{w_k^2}\right) \Big/ \left(\sum_{k=1}^{n} \frac{1}{w_k^2}\right).$$

Determine if this line satisfies the distance constraint on $H(f)$. If it does, we are done. If it does not satisfy the distance constraint, start with some positive value of $p$ and search for the value of $p$ for which $H(p) = M$, using a combination of Newton's method when moving to the right and a binary search when moving to the left (or any applicable

search). Insert this value of $p$ in (15), solve the linear system for $\mathbf{c}$, and compute the related values of $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{d}$ using (14), (11), and (12). The periodic cubic spline associated with this value of $p$ will be our solution.

## X. PRACTICAL CONSIDERATIONS IN SOLVING THE LINEAR SYSTEM

Since the matrix $\mathbf{A}_p$ is positive definite for each $p > 0$, it can be decomposed into the product of a lower triangular matrix $\mathbf{R}$ and its transpose, using the square-root (Cholesky's) method.[5] The linear system $\mathbf{A}_p\mathbf{c} = 3p\mathbf{Q}\mathbf{y}$ can then be solved efficiently in two steps, by applying forward substitution to the lower triangular system $\mathbf{R}\mathbf{v} = 3p\mathbf{Q}\mathbf{y}$, followed by backward substitution to the upper triangular system $\mathbf{R}^T\mathbf{c} = \mathbf{v}$.

Furthermore, since $\mathbf{A}_p$ is symmetric and five-banded with three non-zero entries in two corners, its decomposition $\mathbf{R}$ will consist of three non-zero bands (the main diagonal and the two diagonals below it) and two non-zero rows along the bottom, so that $\mathbf{R}$ can be stored in fewer than $5n$ locations. (The entries of $\mathbf{A}_p$ need not be stored; they may be computed as needed.)

The sparseness of the matrix $\mathbf{R}$ and its structure described above lead not only to its compact storage, but also to the linear time solution of the upper and lower triangular systems, and hence to the linear time solution of the system $\mathbf{A}_p\mathbf{c} = 3p\mathbf{Q}\mathbf{y}$, for each non-zero value of $p$.

It should be pointed out that, unless the number of data points to be smoothed is rather limited (approximately 30 or fewer), the straightforward application of Cholesky's method to decompose $\mathbf{A}_p$ will encounter underflow problems. This is due to the fact that, as the dimension of $\mathbf{A}_p$ increases, entries with exponentially decreasing magnitudes will appear in its decomposition $\mathbf{R}$. This difficulty can be circumvented by truncating sufficiently small values to zero, while still retaining single-precision accuracy in the solution of the triangular systems. (Truncation has the additional advantage of significantly reducing the number of arithmetic operations required in computing the entries of $\mathbf{R}$ when the number of data points is large.)

The program implementing this algorithm has been tested on an IBM-370 computer using single-precision arithmetic, and has successfully smoothed up to 250 points before encountering detectable round-off errors. On the average, the Newton and binary search converged after six to eight iterations, independent of the number of data points.

## XI. SAMPLE PLOTS

The data points in Figs. 1 and 2 were generated by adding random noise to an ellipse. The dotted curves represent cubic spline interpo-

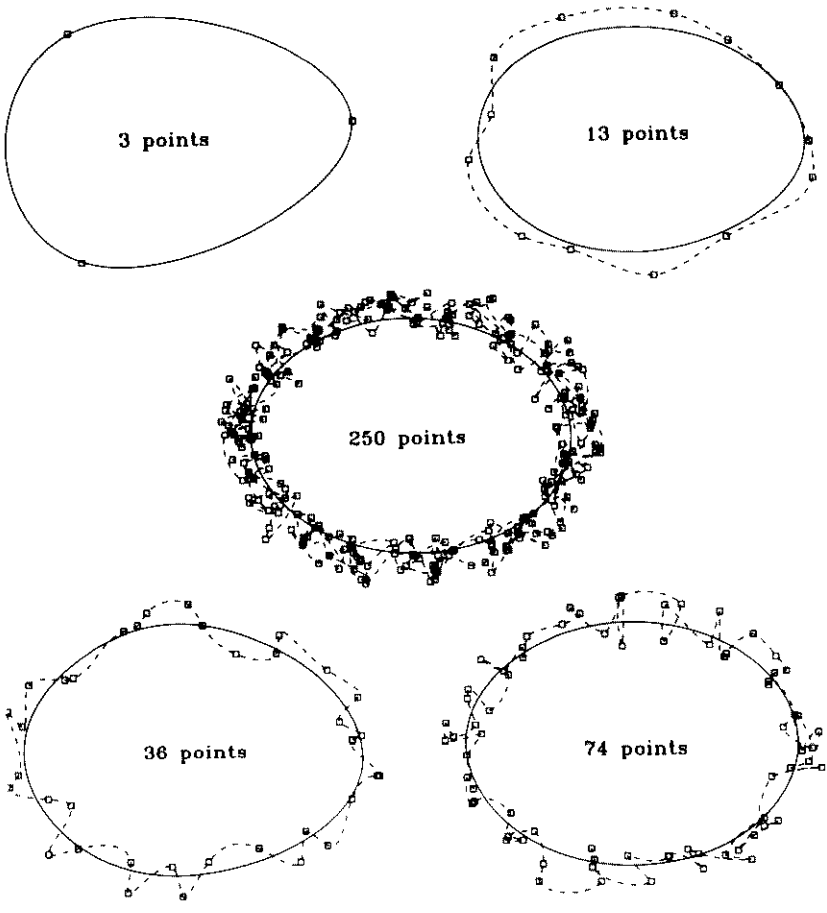



Fig. 2—Periodic cubic spline smoothing for a varying number of data points with uniform weights.

lation of the data (using periodic splines), while the solid curves represent cubic spline smoothing of the same data. In each case a parameter was introduced so that the curve could be represented as two separate single-valued functions of the parameter. Then smoothing was performed twice, once with the $x$ values as a function of the parameter and then again with the $y$ values as a function of the parameter. The smoothed $x$ and smoothed $y$ values were then plotted against the parameter to produce the closed curve. Figure 2 illustrates the algorithm with a varying number of data points, from only three distinct points to 250 points. In each case, uniform weights were used. A "tight" fit was chosen in the example with three points to show how the method can be used to simulate periodic interpolation of the points.

## REFERENCES


1. Carl de Boor, *A Practical Guide to Splines*, New York: Springer-Verlag, 1978.
2. Christian H. Reinsch, "Smoothing by Spline Functions," Numerische Mathematik, *10* (1967), pp. 177–83.
3. H. Spath, "Spline Algorithms for Curves and Surfaces," Winnepeg: Utilitas Mathematica Publishing Inc., 1974.
4. David I. Steinberg, *Computational Matrix Algebra*, New York: McGraw-Hill, Inc., 1974, p. 251.
5. V. N. Faddeeva, *Computational Methods of Linear Algebra*, New York: Dover Publications, Inc., 1959.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Hybrid-Mode, Shielded, Offset Parabolic Antenna

By R. A. SEMPLAK

(Manuscript received September 24, 1981)

*In this paper measurements are presented for both a hybrid-mode feed having corrugations in two opposite walls and a modified scale-model pyramidal horn-reflector antenna using this feed. Comparisons of the measured and theoretical data for the hybrid-mode shielded offset parabolic antenna with the theoretical data for the standard pyramidal horn-reflector antenna show that a net improvement can be obtained in sidelobe level of about 4 dB in the transverse polarization in the transverse plane and 4 to 13 dB in the longitudinal polarization in the longitudinal plane.*

## I. INTRODUCTION

In most respects, the standard pyramidal horn-reflector is an excellent antenna. It is a combination of a square pyramidal horn and a reflector that is a section of a paraboloid of revolution whose focus coincides with the apex of the square horn. Its geometry provides a shielded offset parabolic antenna that is broadband with good front-to-back discrimination and good return loss. However, inherent in the design of the pyramidal horn-reflector antenna is a problem that results from illuminating the reflector with a dominant waveguide mode.[1] The theoretically obtainable off-axis radiation levels for transverse polarization in the transverse plane and longitudinal polarization in the longitudinal plane are considerably higher than those obtained for longitudinal polarization in the transverse plane and transverse polarization in the longitudinal plane, i.e., the former are essentially the equivalent of an aperture with constant illumination, whereas the latter aperture field distributions are tapered to zero at the edges.

In the discussion of the horn-reflector antenna, it should be remembered that longitudinal polarization and longitudinal plane indicate that the electric field in the aperture and the plane of antenna rotation (for radiation measurements) are aligned with the pyramidal horn axis,

whereas transverse polarizations and transverse plane indicate that the electric field in the aperture and plane of antenna rotation are perpendicular to the horn axis.

As used in microwave radio-relay systems, the horn-reflector antenna is mounted with the axis of the horn normal to the earth's surface. Hence, longitudinal and transverse polarizations could be called vertical and horizontal, respectively. However, the aperture field distribution for each polarization is different. Moreover, when the antenna is used as an earth station antenna for satellite communications, or as a radiometer, or simply to obtain radiation characteristics in the longitudinal plane, the antenna is mounted on its side, i.e., the longitudinal axis of the horn is parallel with the earth's surface, and aperture field distributions for so-called vertical and horizontal polarizations are now interchanged. To avoid this ambiguity, longitudinal and transverse polarizations are referred to the axes of the horn.

Radio interference from adjacent paths limits the number of converging routes of a common-carrier microwave radio system. In recent years, demands have been made to improve the sidelobe performance of the pyramidal horn-reflector antenna. The use of blinders[2] (extensions to the side walls of the antenna aperture) provides a degree of far sidelobe reduction, i.e., lobes beyond 35 degrees from the axis of the main beam. A method now exists for eliminating the troublesome reflections from the flat weather cover of the horn-reflector antenna by using a focused weather cover.[3]

As used in terrestrial microwave radio systems, the horn-reflector antenna is mounted with the axis of the pyramidal horn normal to the earth's surface. Therefore, to address the problem of obtaining improvement in the sidelobe structure, primary consideration should be given to those sidelobes produced by transverse polarization in the transverse plane. Hence, one needs to consider those special feeds[4,5] that would decrease the field intensity at the two side walls.

In Section II a discussion of the design and radiation measurements of a 30-GHz, two-wall, hybrid-mode feed is presented; in Section III the radiation measurements obtained in both the transverse and longitudinal planes when this feed is used on a 30-GHz, scale-model, horn-reflector antenna are discussed;[6] and in Section IV the conclusion is made that the antenna as modified is no longer the familiar horn-reflector antenna, but a shielded, hybrid-mode-fed, offset paraboloid.

## II. TWO-WALL HYBRID-MODE FEED

The hybrid-mode feed fabricated for this experiment was designed at a frequency of 30 GHz as a 6-inch-square aperture with two opposite walls corrugated. The design of the corrugations at this frequency requires a tooth and groove width of 0.020 inches (0.508 mm) and a

tooth height or groove depth of 0.0997 inches (2.532 mm). These dimensions present a formidable machining task. Hence, other methods for fabrication were considered. As shown in Fig. 1, the approach used for this feed consisted of stacking alternating 0.020-inch- (0.508-mm) thick strips of brass and aluminum, with the height of the brass strips twice that of the aluminum strips. The assembly of alternating strips was electroformed and then machined to obtain the proper tooth height. The aluminum was chemically removed to provide the grooves. The rest of the fabrication proceeded along normal lines.

The measured vertical and horizontal polarization radiation characteristics for the two-wall hybrid-mode feed are shown in Figs. 2 and 3, respectively. The position of the two corrugated walls with respect to the electric field is indicated on the insert in each figure. For vertical polarization (Fig. 2), the corrugated walls are parallel to the electric field, whereas for horizontal polarization (Fig. 3) the two corrugated walls are normal to the electric field. An examination of these two figures indicates that, except for a few minor differences, the two patterns are essentially equal and symmetric. The dashed lines on these two figures indicate the respective cross-polarized response of the feed. From Fig. 3, we can conclude that the two-wall hybrid-mode feed provides a tapered field distribution for the transverse polarization in the transverse plane of the horn-reflector antenna. By rotating the hybrid-mode feed 90 degrees the same tapered field distribution can be provided for the longitudinal polarization in the longitudinal plane.

## III. MODIFIED HORN-REFLECTOR ANTENNA

### *3.1 Transverse polarization, transverse plane*

Using the radio range facilities at Holmdel, New Jersey, measurements were made of the far-field radiation characteristics of a 30-GHz,

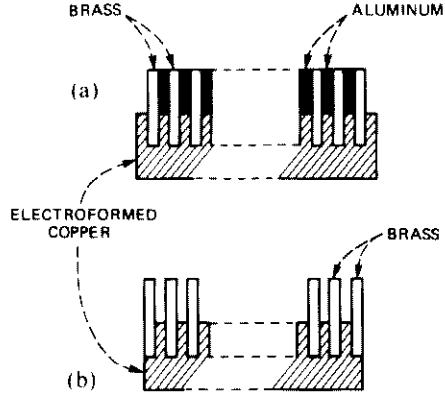



Fig. 1—(a) An assembly of alternating brass and aluminum strips used to achieve a corrugated wall. (b) The finished wall.

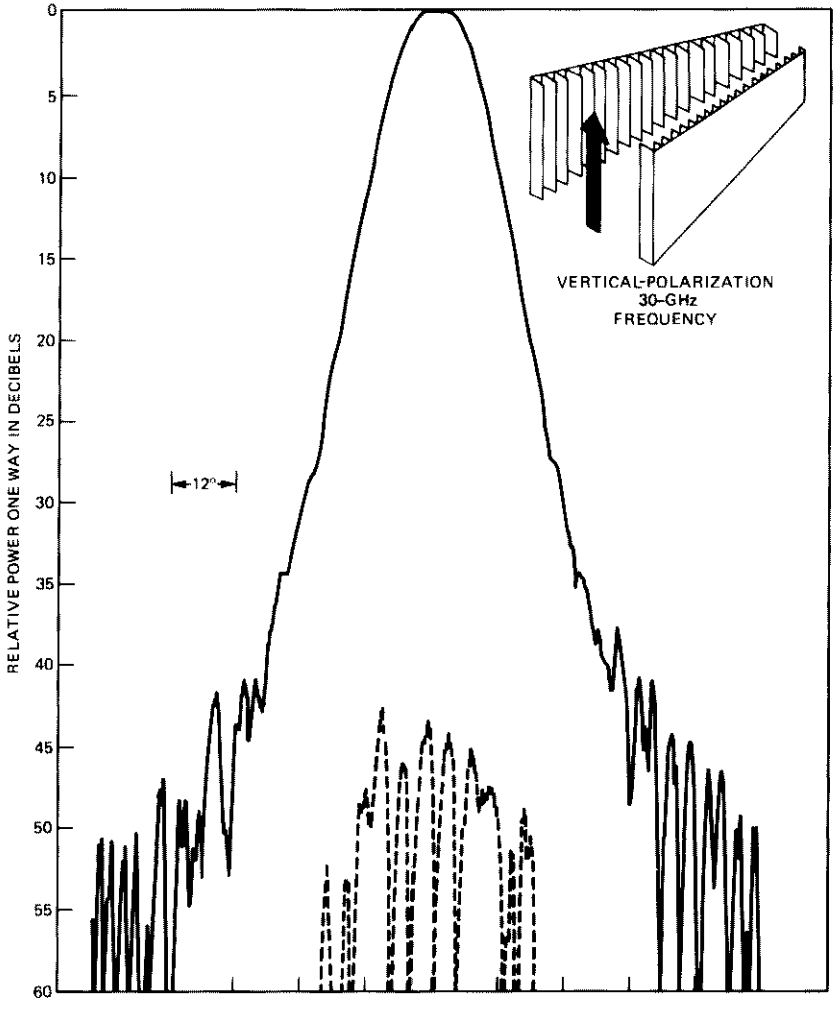



Fig. 2—The polarized radiation characteristics of the two-wall corrugated feed measured vertically are shown by the solid curve. The dashed curve shows the cross-polarized response. As depicted in the insert, the corrugated walls are parallel to the electric field.

scale-model, horn-reflector antenna[6] illuminated by the two-wall hybrid-mode feed discussed above. The solid line of Fig. 4 is the far-field radiation characteristic of this antenna for transverse polarization in the transverse plane. Recall that the two corrugated walls of the feed are normal to the transverse field. In Fig. 4, the dashed curve indicates the calculated envelope of peaks for the hybrid-mode antenna with ideal dual-mode excitation. Comparison of these two curves reveals good agreement over the main beam and first sidelobes but increasing disagreement for the sidelobes that are further out.

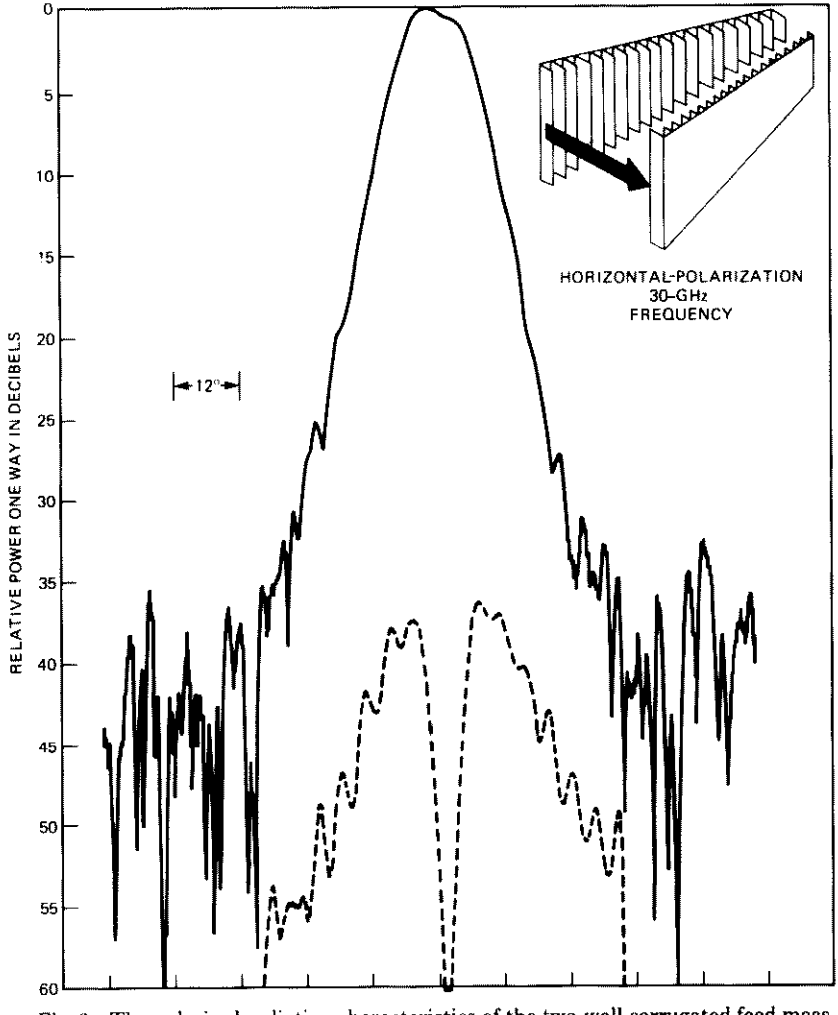



Fig. 3—The polarized radiation characteristics of the two-wall corrugated feed measured horizontally are shown by the solid curve. The dashed curve shows the cross-polarized response. As depicted in the insert, the corrugated walls are normal to the electric field.

A more extensive comparison is presented in Fig. 5. In this figure, the curves represent the characteristics obtained for transverse polarization in the transverse plane for the following: the broken curve is the theoretical response for the standard pyramidal horn-reflector antenna;[6] the dashed curve represents the theoretical response for the hybrid-mode antenna with ideal dual-mode excitation; and the solid curve is the measured data for the hybrid-mode antenna. An examination of the two theoretical curves indicates the possible improvement

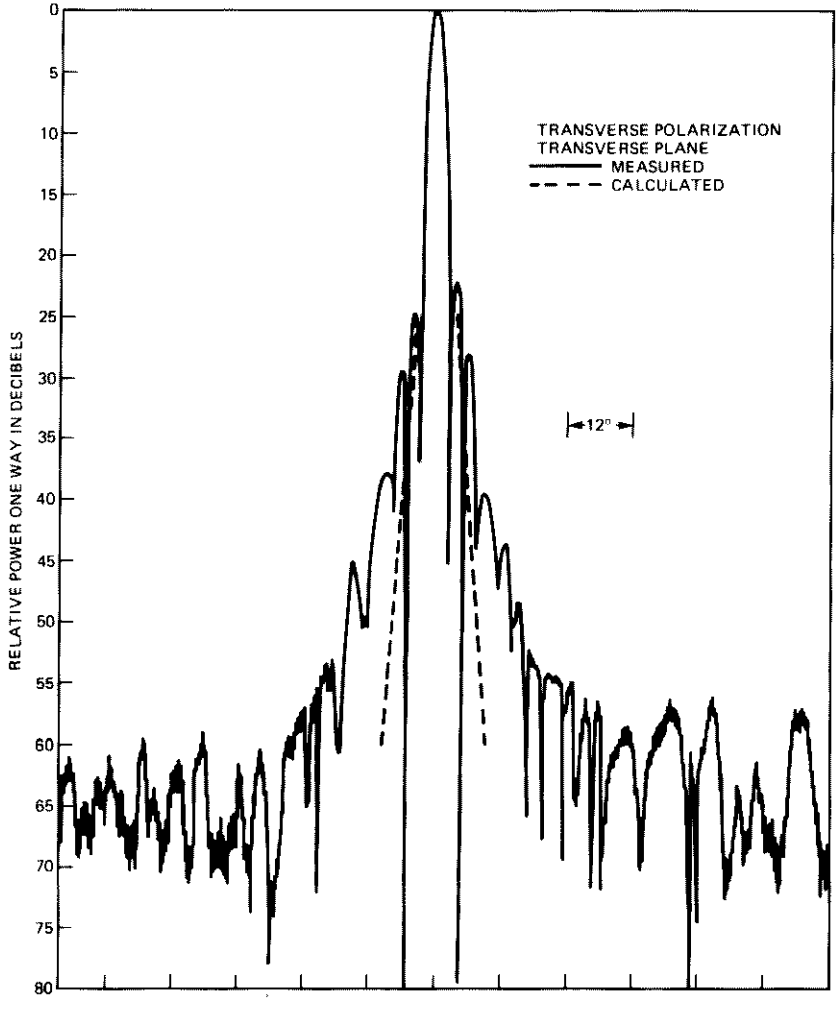



Fig. 4—The measured (solid curve) and calculated (dashed curve) radiation characteristics of the modified horn-reflector antenna for transverse polarization in the transverse plane.

in sidelobe structure that could be obtained by using the hybrid-mode feed instead of the standard horn feed of the standard horn-reflector. From comparison to the measured response (solid curve) one can see the actual improvement over that of the standard pyramidal horn-reflector antenna. However, the measured response is not quite as low as predicted by theory. The observed departure leads one to suspect that the two-wall corrugated structure used in these experiments may be flawed. By examining Figs. 2 and 3 more carefully, one observes regular small undulations in the radiation characteristics of the two-

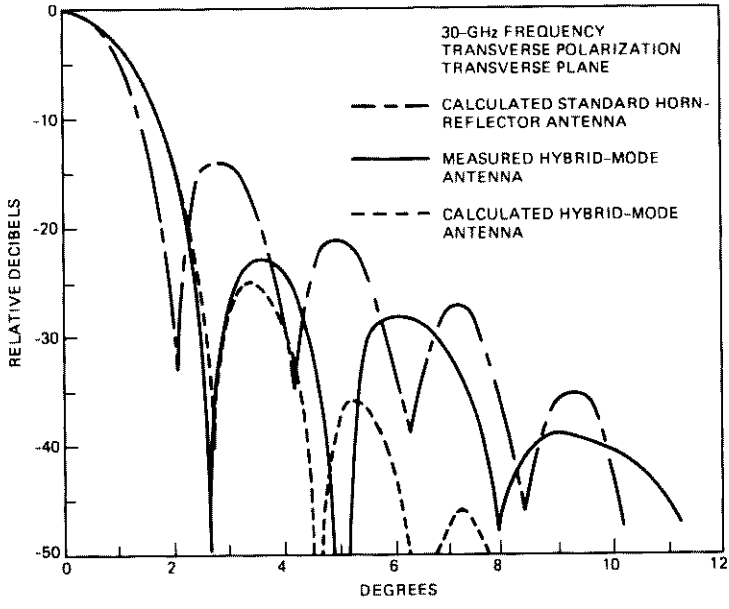



Fig. 5—Comparison of the measured response of the modified horn-reflector antenna (solid curve) with the theoretical response of the unmodified standard horn-reflector antenna (broken curve) and the theoretical response of the modified antenna (dashed curve).

wall corrugated feed that could indeed be indicative of the presence of modes other than the desired $HE_{11}$ mode.

A comparison of the hybride-mode antenna measurements made for longitudinal polarization in the transverse plane with those of Ref. 6 indicate only minor differences and therefore are not included here.

### *3.2 Longitudinal polarization, longitudinal plane*

To examine the possibility of improving radiation characteristics in the longitudinal plane, the hybrid-mode antenna is placed on the antenna positioner with its longitudinal axis parallel to the earth's surface. The two-wall hybrid-mode feed is rotated 90 degrees so that the corrugations are now on the front and back walls of the pyramidal portion of the antenna, i.e., the corrugations are now normal to the longitudinal field. The measured data obtained for this configuration are shown by the solid line in Fig. 6. The dashed line in this figure represents the calculated theoretical response for the hybrid-mode antenna with the ideal dual-mode excitation. Comparison of these two curves indicates good agreement.

A more extensive comparison is presented in Fig. 7. In this figure, the set of curves represents the characteristics obtained for longitudinal polarization in the longitudinal plane for the following: the broken

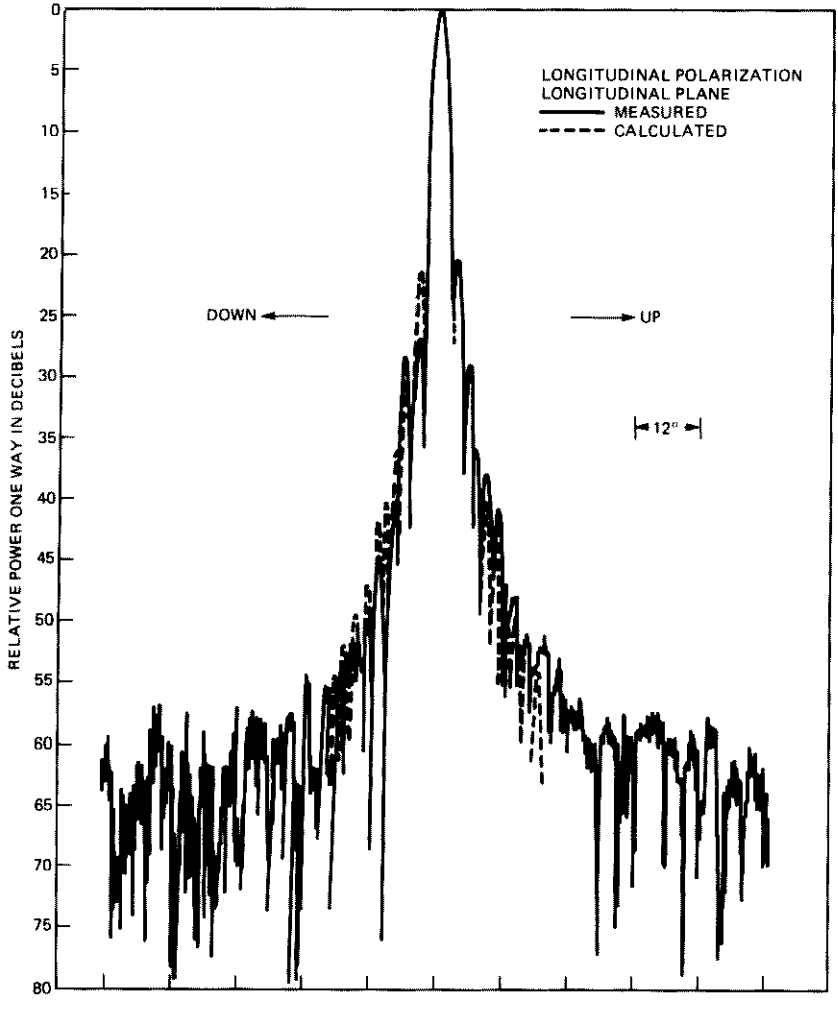



Fig. 6—The measured (solid curve) and calculated (dashed curve) radiation characteristics of the modified horn-reflector antenna for longitudinal polarization in the longitudinal plane.

curve is the theoretical response for the standard pyramidal horn-reflector antenna;[6] the dashed curve represents the theoretical response for the hybrid-mode antenna with ideal dual-mode excitation; and the solid curve is the measured data for the hybrid-mode antenna. Comparison of the two theoretical curves indicates the possible reduction in sidelobes that could be obtained by using a hybrid-mode-fed antenna. From the measured response (solid curve) one can observe the actual improvement over that of a standard pyramidal horn-reflector antenna and the good agreement with the theoretical values

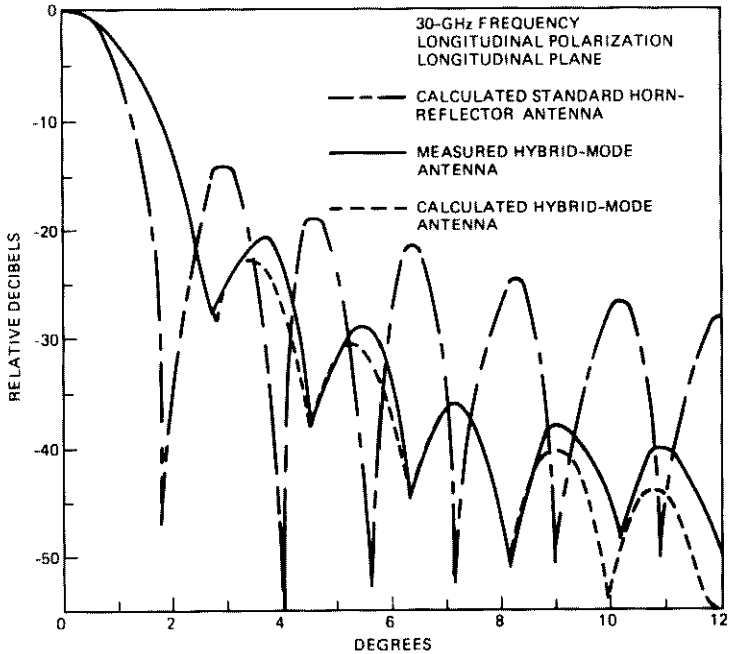



Fig. 7—Comparisons of the measured response of the modified horn-reflector antenna (solid curve) with the theoretical response of the unmodified standard horn-reflector antenna (broken curve) and the theoretical response of the modified antenna (dashed curve).

for the hybrid-mode antenna. The agreement here is better than that shown in Fig. 5, but one should recall that the pyramidal sidewalls contribute more to sidelobe structure than do the front and back walls. However, when compared to the transverse plane, the sidelobe structure in the longitudinal plane is, in general, higher. This can be attributed to the nearly parallel top and bottom edges of the antenna aperture.

A comparison of the hybrid-mode antenna radiation characteristics for transverse polarization in the longitudinal plane with those of Ref. 6 indicate only minor differences and therefore are not included here.

One should recall that, at present, the bandwidth of corrugated feeds is limited to less than two-to-one. This bandwidth limitation would tend to restrict the application of a two-wall corrugated feed to those system antennas where only two frequencies, i.e., 4 and 6 GHz, or 6 and 11 GHz were used. It should be noted that a two-wall hybrid-mode feed as discussed here will only reduce sidelobes in one plane; therefore one has the option when modifying existing antennas of applying the correction to that particular plane where interference in the near-in sidelobes is a problem. Of course, the use of a four-wall hybrid-mode feed reduces the sidelobes in both planes.

## IV. CONCLUSIONS

Measurements for both a two-wall hybrid-mode feed and a modified scale-model horn-reflector using the hybrid-mode feed (a hybrid-mode, shielded, offset parabolic antenna) are presented and discussed. The high-sidelobe characteristics of a standard pyramidal horn-reflector antenna displayed in the transverse plane for transverse polarization and in the longitudinal plane for longitudinal polarization can be improved by modifying the antenna with a two-wall or four-wall hybrid-mode feed.

## REFERENCES


1. A. B. Crawford, D. C. Hogg, and L. E. Hunt, "A Horn-Reflector Antenna for Space Communications," B.S.T.J., *40*, No. 4 (July 1961), pp. 1095–116.
2. D. T. Thomas, "Analysis and Design of Elementary Blinders for Larger Horn-Reflector Antennas," B.S.T.J., *50*, No. 9 (November 1971), pp. 2979–95.
3. R. A. Semplak, "Horn-Reflector Antenna—Eliminating Weather-Cover Reflections," B.S.T.J., *59*, No. 8 (October 1980), pp. 1333–42.
4. C. Dragone, "Attenuation and Radiation Characteristics of $HE_{11}$-Mode," IEEE Trans. Microwave Theory and Techniques, *MTT-28*, No. 7 (July 1980), pp. 704–10.
5. C. Dragone, "High-Frequency Behavior of Waveguides with Finite Surface Impedances," B.S.T.J., *60*, No. 1 (January 1981), pp. 89–116.
6. R. A. Semplak, "A 30-GHz Scale-Model, Pyramidal, Horn-Reflector Antenna," B.S.T.J., *58*, No. 6 (July–August 1979), pp. 1551–6.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# An Experimental Teleterminal—The Software Strategy

By D. L. BAYER and R. A. THOMPSON

(Manuscript received September 29, 1982)

*A research model of a synergism between a telephone and a computer terminal is described in this paper and its companion. The hardware design of this "teleterminal," described in the companion paper, includes an internal microprocessor and a data connection to a host computer. This paper describes the software in these two machines. The software resident in the teleterminal's internal microprocessor addresses internal issues like cursor control, tab expansion, and data modes. The software resident in the host controls screen content and menu selection. The teleterminal is a research tool for the investigation of the human interface for the access to experimental services. Such services include calling by name, directory retrieval, a variety of information (e.g., yellow pages, department store catalogs, community bulletin boards, newspapers, and libraries) and a variety of personal services such as mail, personal calendar, entertainment, and shopping. The user-interface is a conceptual tree-like structure that can be customized by the user.*

## I. INTRODUCTION

A *teleterminal* is defined to be a piece of equipment that merges the functionality of the traditional *tele*phone with that of a computer *terminal*. It is characterized by:

(*i*) A traditional telephone facility
(*ii*) Internal intelligence
(*iii*) A data communication facility
(*iv*) A general-purpose display
(*v*) Dynamic labeling of buttons (soft keys).

This merger of functionality has been seen to be synergistic. One such teleterminal, whose hardware is described in Ref. 1 and in the companion paper,[2] is illustrated in Fig. 1.

Fig. 1—A teleterminal.

The construction of and, more significantly, the experimentation with such a teleterminal is part of a large-scale, long-range research investigation into the systems, software, and applications aspects of telecommunications. Cognitive and social implications are a significant part of the study. An evolving futuristic "test-bed" environment[3] currently consists of a highly reliable host computer (a three-processor Tandem-16), an "intelligent" digital switching office,[4] and a collection of teleterminals. A number of systems-level principles have been uncovered in the course of this research and appropriate papers have been submitted to and published in appropriate journals. It is the purpose of this paper and its companion to describe the equipment used in this research and thereby serve as a common reference for those other papers of a more general nature.

Compatibility with today's environment would be of major importance if a real product were being described, but it is of little consequence in a research environment, except that it simplifies the dissemination of teleterminals to interested people. There are about three dozen of these teleterminals in an active user community. It must be emphasized that the teleterminal is *not* a proposed product; it is a research tool. Design decisions, and discussions in this paper about it, are based on this important point. If a product were anticipated, many decisions would have been made differently and the reasons discussed would change dramatically. Furthermore, neither this research nor this paper is intended to describe nor imply the future direction of the Bell System as it relates to advanced communications systems and services, nor is it intended to endorse any hardware or software offerings by any vendor.

The next section of this paper presents the research objectives of the project that have an impact on software. They include assumptions about the system environment and the impact on the user. In Section III, intelligence distribution is discussed and the internal and host programs are described. In Section IV, a scenario is presented to illustrate program execution, the user interface, and the kinds of capabilities provided. Section V contains a discussion of lessons learned and future directions.

## II. OBJECTIVES

Objectives that have software implications are emphasized. These are presented from the viewpoint of both the system and the user.

### *2.1 System environment*

In the current system configuration, each teleterminal has a standard terminal connection to a host computer and a standard telephone connection to a telephone office. The only assumed commonality

occurs at the teleterminal itself. Compatibility with the "POTS" (plain old telephone service) world requires that old-fashioned telephone hardware like the switchhook and *Touch-Tone** dialing equipment be necessary parts of the teleterminal.

Such a system configuration impacts software in that this required POTS hardware reduces available space for microprocessor memory inside the teleterminal and that special microprocessor software is needed to interact with this POTS hardware. Furthermore, the host software must transmit phone numbers to the teleterminal (with a special data mode) instead of directly to the telephone office controller and it must be concerned with such POTS functions as the switchhook state and dial tone. "POTS" functions, like "busy" or "network blocking", could be recognized by including certain tone-detecting filter circuits in the hardware, but to conserve space this was not done. As a consequence, the software is limited to assuming the presence of dial tone after a fixed delay and the software can make no branches on whether or not a call completes.

A second system aspect is our desire to use the teleterminals in a "test-bed" environment for the ongoing investigation of the human interface and the study of experimental services. This suggests that changes will be made to the software frequently and reflection of this in the initial design of the software architecture was felt to be wise. A distribution of intelligence was selected wherein the functional software is located in a centralized general-purpose host computer. Furthermore, this "access program" can be structured to simplify anticipated changes. Such a "host-centered" distribution of intelligence may not be ideal for a marketed product where the design goals would be different and the interface would be more stable. These kinds of items are discussed in Section V.

### 2.2 User impact

After system environment, the second area of objectives with software ramifications is the impact on the user. There are two such impacts: the physical cosmetics of the teleterminal and the conceptual interface to the software. Cosmetic features that affect software are the small size, the incorporation of both telephone and terminal characteristics, and the inclusion of function keys.

For the experiments with new applications, the teleterminal replaces a conventional business telephone. It is important that the set occupy approximately the same desk space as a conventional business telephone. Furthermore, the set is "real" in the sense that all electronics, except power supplies, reside within the set. This results in a limited

* Registered service mark of AT&T.

physical space for internal memory and thus restricts the size of the internal software. The placement of the function keys adjacent to the cathode ray tube (CRT) screen has major impact on the software. At the "low" level, internally executed "primitives" are required for labeling buttons and sensing button-pushes. At the "high" level, these buttons make possible the use of a tree-structured user interface to access various functions.[5]

From the user view, three general requirements of this access method are simplicity, convenience, and customizability. The interface must be as conceptually simple as possible, even to the most casual, unsophisticated user. The required convenience of the interface is attained by enhancing the access to often-used functions and the names of often-called people and by providing a "translation" capability between the user's name for some function and the "telephonese" or "computerese" for accessing it. Whether customization is done by the vendor or by the user, the need for real simplicity in the underlying data structure, and not just apparent simplicity in the user's view of it, is reinforced.

The structure of the user interface is tree-like in a deliberate attempt at "congruency"[5] with simple models of human cognition.[6] Similar "access trees" have appeared in the literature. References 7 and 8 are representative. In the marketplace, Hewlett-Packard's new line of terminals and the British PRESTEL are examples. In the mathematical sense, a "tree" is an acyclic, connected graph;[9] but the concept of tree to be presented is more like the "lay" notion of a large woody plant.

Informally, the parts of a tree include the root, branches, intermediate nodes, and leaves. Let a screen of button labels correspond to a node. Let the dynamic label of a button correspond to a branch or leaf, depending on whether its selection causes a traversal to a new node or the actuation of some function, respectively. The structure is not a mathematical tree because multiple branches to the same node are permitted, as are "backup" and "restart" branches (which make the "graph" cyclic). Analogous to the root, or initial node, of the tree is an initial labeling of the buttons by which the user perceives the "entering" of his structure of functions and directories. The root of an example access tree is illustrated in Fig. 2.

The organization of this access tree is intended to be defined or selected by the user. A *functional* organization would show a root menu with branches like **Telephone Functions**,* **Computer Functions**, **Calendar**, **Mail**, etc. A utilitarian organization would attempt to place highly used functions near the root and seldom used functions

* Labels identifying function keys are set in boldface type.

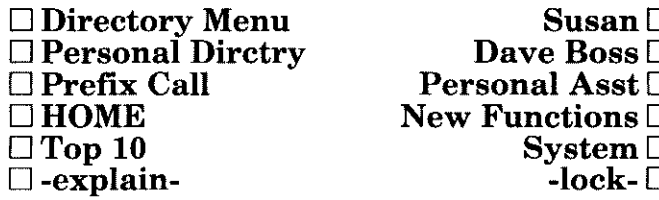



Fig. 2—A sample access tree.

far from the root. Some compromise of the two organizations is probably most appropriate, depending on the user's level of sophistication and the frequency of use.

## III. INTELLIGENCE DISTRIBUTION

The user interface to system capabilities is determined by the properties of the terminal and the network resources that the user can access through the terminal. The capabilities of the terminal are constrained by the computing power of the internal processor and by the data switching capabilities of the communications system. Currently available microprocessors span at least an order of magnitude in speed and several orders of magnitude in memory size. In the case of the teleterminal, power requirements and chip count were important considerations in the selection of the internal microprocessor. Many of the design considerations described in the previous section constrained the type of internal computations and thus focused our efforts on software issues relating to telephone applications vis-a-vis general-purpose word-processing systems.

In the next subsection, the capabilities of the processor within the teleterminal are presented. A list of functions performed by the instrument is given and the impact of this processing power on feature software is discussed. The limitations of, and alternatives to, the current design are presented in Section V.

### *3.1 Internal processing*

The teleterminal contains an Intel 8748 microprocessor with 2048 bytes of programmable read-only memory (PROM) for program storage and 288 bytes of random access memory (RAM) for data storage. The processor executes instructions in from four to eight microseconds. A common operation such as moving a byte from one location to another in RAM requires four instructions, occupies six bytes of program memory, and takes 32 microseconds. A single vectored interrupt is available for performing time-critical functions.

The internal program is written in assembly language and occupies most of the available program memory. The function of the internal program is to control the microprocessor's peripherals: a Matrox

display processor, a tone generator with muting relay, an ASCII keyboard with clicker relay, and a universal asynchronous receiver/transmitter (UART) for interface to the host computer.

The Matrox display processor cycles through a 512-byte dual-ported buffer memory displaying its contents as 16 rows of 32 characters on a CRT. In addition to character generation, the display processor provides character blinking at a half-hertz rate. The internal processor manages the contents of the buffer memory. Scrolling is implemented by moving the entire contents of the display buffer. A cursor is implemented by displaying an underscore character (_) when a space character lies at the cursor coordinates or by blinking the character at the cursor coordinates (inverse video would be more attractive but is not available in the current hardware). Button labels are cleared, positioned, and formatted by the internal processor.

The tone generator is used to dial over the telephone line associated with the teleterminal. The state of the switchhook is examined to determine if the receiver is off-hook. After a short delay to allow for dial tone, digits are dialed by activating the muting relay, activating the tone generator for approximately 100 milliseconds, then deactivating the muting relay, and waiting approximately 60 milliseconds before repeating.

The keyboard peripheral consists of three types of keys: standard keys that correspond to ASCII codes, special control characters within the control code set that correspond to the buttons adjacent to the screen, and polled control buttons. The first two types of keys are scanned, encoded into ASCII character codes, and latched by the keyboard circuitry.[2] The processor examines the latch approximately 60 times per second. When a character has been latched by the hardware, the processor reads the character, clears the latch, and activates the clicker relay to give the user audio feedback. Normal character codes are delivered to the output UART for transmission to the host. For codes corresponding to screen buttons, the processor emits a unique three-character sequence for each of the buttons.

Control buttons are wired directly to individual bits on one of the processor's input/output (I/O) ports. After debouncing, the buttons are used to (also see Section 3.3):

(*i*) Freeze the screen

(*ii*) Emit a special host-specified "delete previous character" code

(*iii*) Emit a special host-specified "delete current line" code

(*iv*) Emit a special host-specified "interrupt" code. The internal buffer of input characters from the host is also purged.

The input characters from the UART (data from the host) are read at interrupt level and collected in a 255-byte circular buffer. In full-duplex operation, flow control is implemented by emitting a single

symbol when the input buffer approaches overflow. This symbol can be set by the host with a control message. The characters in the circular buffer are processed by the base-level part of the control program. Data is separated into three classes: control messages, response messages, and ordinary text. Control messages change the "state" of the set and determine how ordinary text is to be processed. The following "states" exist:

(*i*) Label button—Data is positioned adjacent to the button, left or right justified, and constrained to fifteen characters per line.

(*ii*) No scroll—Data is constrained to the bottom line of the screen.

(*iii*) Dial—Data is interpreted as dial control: that is, a digit to dial, a request for dial delay (usually for second dial tone), a request to time and display call duration, or a request for current call status.

(*iv*) Terminal—Data is displayed much the same as on any simple CRT terminal.

Response messages are character strings generated by the teleterminal in response to either a button push or a host request for status. These messages appear in the teleterminal's input stream because the host computer, in full-duplex mode, echos all characters transmitted from the set. Status response messages are ignored by the set. Button-push response messages cause the set to start flashing the first character of the associated button label, providing a feedback cue equivalent to echoing ordinary characters.

In summary, the internal processing capabilities are used to control the CRT display and perform dialing functions. Support for telephone-related applications is manifested in functions that dial, label buttons, flash-button labels, and restrict output to the bottom line on the CRT.

### 3.2 Host processing

The description of the host software is contained in the next three subsections. First the host's software environment is discussed, then the structure of the user's data is defined, and finally the "access program" is described.

#### 3.2.1 Host software environment

At the system level, the host software environment is that of a transaction-oriented, time-sharing, general-purpose computer. The current implementation, under the *UNIX** operating system,[10] provides a convenient machinery for interfacing the access program not only to the standard commands, system calls, and supported programs of the *UNIX* operating system, such as those providing computer mail,

* Trademark of Bell Laboratories.

calendar functions, and various games, but also to current and future specialized application programs, such as have been written for mail access, call-memos, and personal calendar maintenance. The access program and the peripheral programs have been written in the C programming language[11] by a multiplicity of authors with program interfaces provided by the standard system calls and interprocess messages of the *UNIX* operating system. The complete environment resides in a Tandem computer system.

The user's tree-structured data are isolated from the access program that manages those data. This isolation is accomplished by keeping user-specific data in a separate file (per-user) and out of the program. This isolation permits the existence of a single copy of the program in a computer that supports multiple users, each with a unique and customized access structure. Furthermore, this software practice permits changes in directories or function procedures and similar modifications and customizations without requiring the recompiling of a program.

A data change to the access tree data file may be as trivial as an updated telephone number or name of an application program, or it may be purely cosmetic like renaming a button label, moving a familiar function to a new position in the tree, or placing a new label on some vacant button. Or, it may be more significant, like pruning the tree, attaching a new node to the tree, grafting an entire subtree to a new position, or adopting an entirely new access tree. Currently, these changes are made through the *UNIX* text editor to the data only and the program is not recompiled. An experimental editor with a potentially improved human interface is described in Ref. 12.

### *3.2.2 Data description*

The access tree is logically organized as a hierarchical data structure. The current implementation uses a magnetic disk medium and so the physical organization is linearized into a serial file of records. The data consist of an iteration of "record-groups" (each corresponds to a node of the tree or screen on the teleterminal), each in turn made up of ordered "records" (each corresponds to a button label). The actual data file read by the access program is obtained by "compiling" the file to be described here. A structure diagram of the data, in the sense of Michael Jackson,[13] is shown in Fig. 3.

The label is the text that appears adjacent to the corresponding button when the parent-node is active. The successor-node identifies the node to which the tree traverses when the corresponding branch is selected. The button-number is usually omitted but, if specified, is an integer between zero and eleven. The default is the "next" button in logical order. The command is executed by the program as described

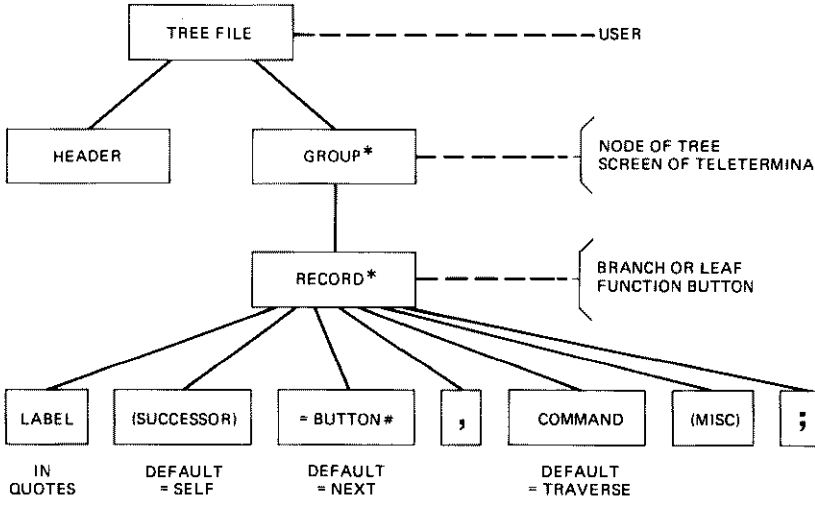



Fig. 3—Structure diagram of access tree data.

in the next subsection. The miscellaneous field contains data pertinent to the command, such as a telephone number or program name.

Data files in the format of such access trees also require "headers". Since each user is assumed to possess a personal tree file, it is appropriate to place user information in the header such as: a list of telephone numbers from which one is selected for "call-return" messages, the password for allowing access to "private" parts of the tree, and the default-assumed location used for adjusting call prefixes. The first several lines of a typical user's tree file are illustrated here in Fig. 4.

Record-groups are enclosed in brackets,"{ }", and symbolically named. The records are assigned to buttons in consecutive order within a record-group by default. The default successor-node is the current node and the default command is TRAVERSE. The correlation between a record-group and the user's perception on the CRT screen is seen by comparing the record-group named ROOT from the partial file of Fig. 4 with the "Root node" illustrated in Fig. 2. In the node corresponding to the PREROOT record-group of Fig. 4, buttons numbered one, seven, nine, and ten are left unlabeled.

From the header of the file in Fig. 4 it is seen that this tree belongs to a user whose Bell Laboratories phone number is 6170 at Murray Hill (represented by the mnemonic prefix "MH") and whose home phone number is 4649999 in dial area 201 (represented by the mne-

```
TREE heading
Numbers: murray-hill: MH6170, home: N4649999
Password: joesentme
Location: MH
%%
PREROOT
{
"introduction", CATFILE(expl/e.0,trav);
"Other User" =2; CATFILE(/get/dump/othus,exit);
"Other Phone", OTHERPHONE;
"Other Location", OTHERLOC;
"-explain-", EXPLAIN;
"open"(ROOT), CHANGELOCK;
"Bell logo" =8, CATFILE(/get/dump/bell,trav);
"-exit-" =11, CATFILE(none,exit);
}
ROOT
{
"Directory Menu" (DTORYMENU);
"Personal Dirctry" (PERSDIRY);
"Prefix Call" (PREFIX);
"HOME", CALL(N4649999,zhome);
"Top 10" (TOP10);
"-explain-", EXPLAIN;
"Susan"(PERSASST), CALL(MH4236,sst);
"Dave Boss", CALL(MH4235,db);
"Personal Asst"(PERSASST);
"New Functions"(NEWSVC);
"System", RUNSCROLL(/bin/sh,sh);
"-lock-"(PREROOT), CHANGELOCK;
}
PERSASST
{
"Today's Appnts", RUNSCROLL(/usr/lbin/caltoday,caltoday);
"Other Appnts", RUNSCROLL(/usr/lbin/calexam,calexam);
"Make Appnts", RUNSCROLL(/usr/lbin/calenter,calenter);
"Set Reminder", REMINDER;
"Time & Date", RUNBOTTOMLINE(/bin/date,date);
"2-month Cal", CAL2MONTH;
"Read Mail"(PREFIX), READMAIL;
"Send Mail", SENDMAIL;
"Send Call-Memo", SENDCALLMEMO;
"Std. Call-Memo", STDCALLMEMO;
"-backup-"(ROOT), TEMPBACKUP;
"-restart-"(ROOT);
}
DTORYMENU
{
"Emergency"(EMERGDTORY);
  .
  .
  .
```

Fig. 4—Example of a partial tree file.

monic prefix "N"). This user's password is "joesentme" and dial digits will be prefixed assuming (in the default) they are dialed from Bell Laboratories at MH. These default attributes may be changed by the user at the node called the PREROOT (see Fig. 4), first encountered when the file is initially opened at the beginning of the access program. Functions at the PREROOT allow the user to change his default location or add another phone number temporarily to those listed in the header, or allow a new user to identify himself.

Each user is assigned a computer user-identification and a corresponding file system "home" directory. The user's private access tree is expected to be found in this directory along with other pertinent files like a personal directory, a mailbox, and a personal calendar. Besides each private tree-file, there is a collection of "public" tree files, any of which may be accessed by any user. The real users have access structures in which only real or imminent functions are accessible. The running example throughout this paper is such a tree. One pseudo-user, called "demo", has indicated access to named functions that are not implemented but from which provocative demonstrations have been given.

#### *3.2.3 Program description*

The structure (also in the sense of Michael Jackson) of the host-resident access program is illustrated in Fig. 5. The uppermost level shows the initial processing of the tree-file, the initial labeling of the buttons, and an iteration of processing "button pushes". Each button push is processed by reading the button identification message from the teleterminal, checking the validity of the message and its "type," executing the corresponding command with respect to current "modes," traversing the tree to the successor-node, and displaying the button labels pertaining to that node. The formats of various messages to and from the teleterminal are discussed in the next subsection. Command "modes" and "types" are briefly described.

The lock mode and the explain mode are program states that apply to the imminent button push but which are manipulated by a previous button push. The lock mode disables most commands and is a means for providing privacy protection: it is toggled by the CHANGELOCK command (see Fig. 4), which prompts for a password when "unlocking." The explain mode is enabled by the EXPLAIN command (see Fig. 4) and causes the program to substitute an explanation of the subsequent button push for the regular action of that button push.

A very brief description of each of the currently implemented commands is tabularized in Fig. 6. These commands are usually requested by the user through the operation of an appropriately labeled button. However, as another "type" of request, unprompted

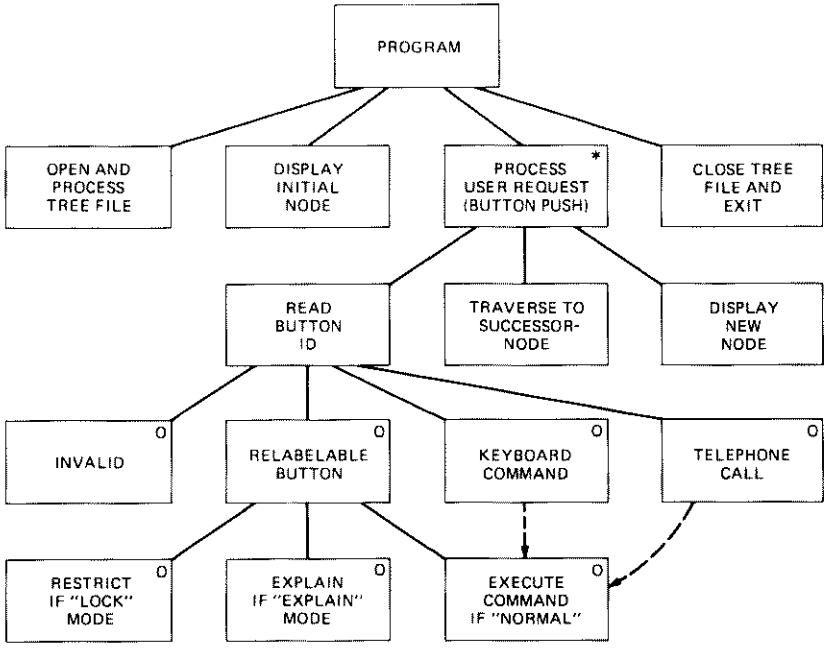



Fig. 5—Structure diagram of the access program.

keyboard entries are also interpreted as user-defined requests for specific commands. For example, unprompted numeric key entries are interpreted as telephone numbers.

Examination of Figs. 2, 4, and 6 will aid the reader in correlating the data file with the program. Several button pushes will be illustrated here, with more in the scenario presented in Section IV. In Fig. 4, the line beginning with "Personal Asst" (in the ROOT record-group) illustrates that **Personal Asst** is the label for button number eight on the root node (see Fig. 2). It is further seen from that record that when this button is selected by the user, the default TRAVERSE command is executed by the program and the tree traverses to the node whose record group is named PERSASST. Every command concludes with a traversal to the successor node; the TRAVERSE command does nothing else before that. The TRAVERSE command (see Fig. 6) requires no other data, and so none are present in the miscellaneous field of the record. The PERSASST record group is included in Fig. 4 and it corresponds to the button labels on the "Personal Assistant" node shown here in Fig. 7.

The line beginning with "System" on Fig. 4 illustrates that **System** is the label for button number ten on the root node (Fig. 2). When this button is selected by the user, the RUNSCROLL command is executed

| COMMAND | DESCRIPTION |
|---|---|
| **BACKRESTART** | backup or restart in directory |
| **CAL2MONTH** | special interface for two-month calendar |
| **CALENDAR** | special interface for general calendar |
| **CALL** | call from button label |
| **CALL2LINE** | call from two-line button label |
| **CALLLOCAL** | call with local prefix |
| **CALLPOTS** | "POTS" call |
| **CALLPREV** | call party previously called |
| **CALLPRFX** | call with labeled tie-line prefix |
| **CALLTRAV** | "POTS" call or traverse in directory |
| **CALLTREE** | call from temporary tree |
| **CATFILE** | cat indicated file and exit or traverse |
| **CHANGECALLTRAV** | toggle "call/traverse" mode in directory |
| **CHANGELOCK** | toggle "lock" mode |
| **DIRCATEGORY** | select directory category |
| **DIRINSTALL** | proceed with personal directory installation |
| **DIRTREE** | make temporary tree from temporary directory |
| **DUMMYCALL** | dummy version of CALL |
| **EXITENTRY** | traverse to TREE with selected directory entry |
| **EXITROOT** | OTHERFILEEXIT or OTHERFILEROOT via mode |
| **EXPLAIN** | toggle "explain" mode |
| **INSERTLABEL** | insert button label into textual files |
| **LEFTRIGHT** | left/right traversal in directory |
| **MAILRETURN** | "detour" return to mail program |
| **OTHERFILE** | use other file instead of TREE |
| **OTHERFILEEXIT** | leave other file to exit point in TREE |
| **OTHERFILEROOT** | leave other file to TREE root |
| **OTHERLOC** | change default location |
| **OTHERPHONE** | enter "other number" for call-memos |
| **READMAIL** | special interface to read mail |
| **REMINDER** | special interface for reminder service |
| **RUNBOTTOMLINE** | use bottom line, create child, continue |
| **RUNRETURN** | clear & scroll, create child, label return button |
| **RUNSCROLL** | clear & scroll, create child, wait |
| **SENDCALLMEMO** | special interface to send a call-memo |
| **SENDMAIL** | send mail via *UNIX* software |
| **STDCALLMEMO** | special interface for standard call-memo |
| **TEMPBACKUP** | temporal backup |
| **TEMPLATEBEGIN** | initiate template creation |
| **TEMPLATEFILL** | fill in directory template |
| **TEMPLATEINIT** | initialize template from previous call |
| **TEMPLATEMATCH** | count or list template matches in directory |
| **TRAVERSE** | simple tree traversal to successor |

Fig. 6—Current commands.

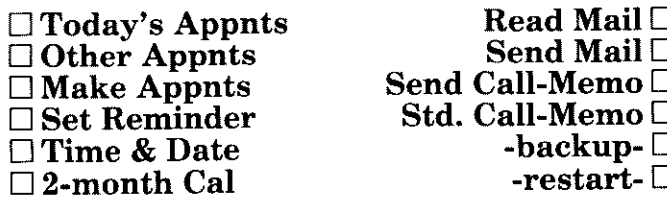



Fig. 7—The **Personal Assistant** node.

by the program and afterwards the tree traverses (by default) to the current node. In executing the RUNSCROLL command (see Fig. 6), the program clears the teleterminal of button labels and causes the indicated program (in this case, a *UNIX* program called the "shell"[14]) to execute from the teleterminal as a computer terminal. The screen now looks like that of a traditional terminal to the user, who is "talking" to the command interpreter of a supposedly familiar operating system. Upon termination of the shell, the access program gets control back and it executes the default traversal to the current, or root, node. In the "jargon" of the *UNIX* operating system, the activity is that the access program puts itself to sleep, and requests the spawning of a child process. The program expects to find the path to the particular child in the miscellaneous field of the record. When the user terminates the session with the child, it is killed and the parent (the access program) is awakened. The entire operation is perceived by the user as a leaf in that a function was requested, granted, and no traversal took place.

Changes to the data were discussed in the previous subsection. But not all changes to the access method can be implemented as simple "edits" of the access file. A new or changed command requires a program change and subsequent recompilation. However, the program has been structured specifically to simplify such changes. Command execution is implemented with a huge multiple branch. The C code in each branch is functionally decomposed. It should be relatively easy for a neophyte programmer (not our unsophisticated user, however) to modify this part of the program. A third kind of change is the unforeseen "nasty" kind of program change that requires a higher level of skill in the person making the change.

The user is expected to "log in" to his own account on the host from his teleterminal, and then to cause the access program to execute from his home directory. The program runs as a "shell" through which other programs, including the shell and other application programs, are invoked. It is planned for the invocation of the access tree program to be placed in the user's profile to facilitate "log in."

The internal firmware was designed so that communication between the user and the application software can proceed with a minimum of teleterminal-related details in the application programs. The areas of concern were system conventions imposed by the operating system, which impact the human interface to the application software and the potential for application software dependence on the detailed characteristics of the teleterminal. While the software was specifically implemented in the environment of the *UNIX* operating system, and the style of the paper assumes that, steps were taken in the design of the internal firmware to avoid assumptions about the specific operating system.

Operating systems establish conventions to control terminal I/O. Most systems usurp characters from the ASCII set for line deletion, character deletion, and interruption of program execution. In the *UNIX* operating system, the default symbols are: '@' to erase a line, '#' to erase a character, and DEL to interrupt program execution. Because our goal was to provide a consistent human interface independent of any operating system idiosyncrasies, the teleterminal allows the application to down-load the codes that should be used for these functions. Extra keys on the keyboard are provided for the erase-character, erase-line, and program-interrupt functions. Since conditions arise when a user may want to enter one of the special characters, the teleterminal will automatically prefix these characters with a host-specified escape-character (back-slash, by default, in the *UNIX* operating system).

Minimizing software dependence on the details of the teleterminal is achieved by defining a high-level interface to the functions:

(*i*) Labeling re-labelable buttons

(*ii*) Dialing

(*iii*) Forcing output to be placed on a single line of the display device.

By and large these functions are sufficient to allow all the tree-oriented software to execute on terminals that provide some form of relabelable buttons. Besides the use of a CRT to label buttons adjacent to the screen, alternatives are a button with embedded light-emitting diodes (LEDs) or a CRT with a touch-sensitive screen. Control of the screen is typically achieved by the host software through print statements. The teleterminal-specific details, such as the actual character strings defining control functions, can be isolated in one file. Using the macro facility of the C language, these control messages can be made to look like function calls. For example, a macro for labeling buttons may be defined such that the statement:

button_label(btn_num,"Prefix Call");

would expand to the C statement:

printf("%c%c%s\n",ESC,'a'-1+btn_num,"Prefix Call");

When executed (assuming btn_num is equal to 2) this statement results in the string

ESC b Prefix Call CR NL

being transmitted to the teleterminal, which causes the second button on the left-hand side of the screen to be cleared, then labeled with "Prefix Call". Similar macros may be defined for other primitives.

## IV. A SCENARIO

The functionality of the tree-like access method is demonstrated by a scenario wherein **A** calls **B**, but **B** is either busy or not home: **A** leaves **B** a message to return the call; then **B** becomes available, reads the message, and returns the call to **A**. We (a user named "us") are **A**, Susan is **B**, and the scenario begins at the root node (Fig. 2). A monologue for introducing a new user to the teleterminal, its services, and tree-like access is appropriate for the unsophisticated user before presenting this scenario. Such a monologue is illustrated in Ref. 5.

We select the **Susan** button and place a call to Susan, the department secretary. Customization of the tree-file is demonstrated by the presence of the **Susan** button in the root, indicating that this call is frequently made; and also by the button text itself: **Susan** is a "friendlier" label than initials and last name (like a typical directory entry) or **secretary**. As seen in Fig. 4, the miscellaneous field of the "Susan" record in the tree-file contains her telephone number and her computer user identification. The CALL command (see Fig. 6) causes the telephone number to be dialed and the id to be stored for possible subsequent use. The tree is traversed to the **Personal Assistant** (Fig. 7).

As an aside, this last traversal exemplifies two points already made. The user, in customizing his own tree-file, determines the successor node of *branches* like **Susan**. There are three appropriate choices:

(*i*) Traversal to the current node emphasizes the leaf nature of such a botton and, often, if such a call fails to complete, a "related" call (that is, to someone on the same node) may be the logical subsequent event.

(*ii*) Traversal to the root gives the perception of "resetting" the teleterminal for subsequent use.

(*iii*) Traversal to the **Personal Assistant** is appropriate if calendar and message functions are likely to be subsequently used.

With the use of **restart** and **backup** buttons, the decision is not all that critical. The second point that is exemplified is the fact that the access tree is not, strictly speaking, a tree: two branches to the **Personal Assistant** have already been demonstrated.

Continuing with the scenario, let **Susan** be busy and let us send her a "Call-Memo" requesting that she call us back. There are two "Call-Memo" buttons on the **Personal Asst** node (Fig. 7): **Send Call-Memo** prompts the user through the creation of a general call-memo and **Std Call-Memo** immediately sends a standard "I called you—Please call me back" call-memo. This standard message is created once by, or for, the user and is available in the user's directory along with a host of other handy files, like a personal directory and a personal calendar. We select **Std Call-Memo** and we are prompted on the bottom line of our teleterminal as shown here in Fig. 8.

The program can't be sure that the call-memo is intended for the person last called, but that is the default operation. If the user wishes to change the default, he identifies the desired user (typing is echoed after the colon on the bottom line). We simply type the "CR" key to continue the default operation and the scenario proceeds. The program responds on the bottom line as shown here in Fig. 9.

Proceeding to the second half of the scenario, **Susan** is tracked as she returns the call. Her first action is to read her mail. From the root (assuming that her tree-file resembles ours), she selects **Personal Asst** and then **Read Mail**. A special mail program is invoked that interfaces the mail program provided in the *UNIX* operating system

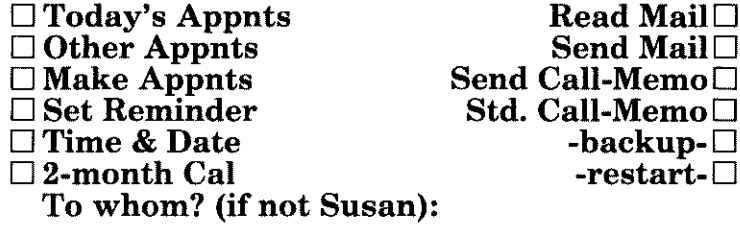



Fig. 8—The **Personal Assistant** node with **Std Call-Memo** prompt.

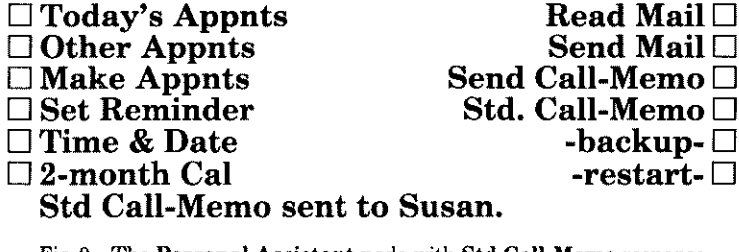



Fig. 9—The **Personal Assistant** node with **Std Call-Memo** response.

to the teleterminal. If our call-memo is her only new message, the screen appears as shown here in Fig. 10.
After selecting our message, the call-memo is scrolled onto the screen as shown here in Fig. 11.

There are a number of different responses to such a message: the choice of action is made after pressing the **process msg** button, as shown in Fig. 12. In our scenario, Susan elects to return the call and selects **Return Call**. Our telephone number is extracted from the call-memo by the program, the mnemonic prefix **"MH"** is appropriately translated into the correct dial prefix, depending on the location of Susan's teleterminal, and the call is made. The screen returns to the call-memo (Fig. 11) from which point the message would probably be **Thrown Out**, ending the scenario.

User perception and the importance of cosmetics cannot be overemphasized. The reader **must** be aware that the demonstration of this very scenario on an actual teleterminal is many times more appealing than a description of the demonstration in such a paper as this. Many

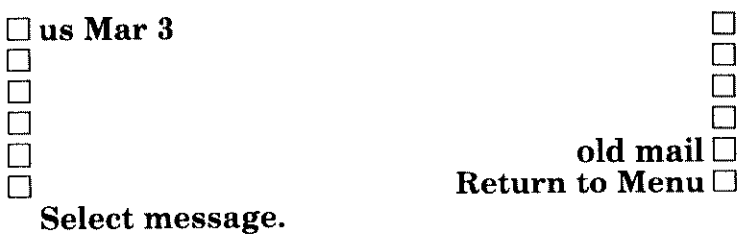



Fig. 10—Root → **Personal Assistant** → **Read Mail.**

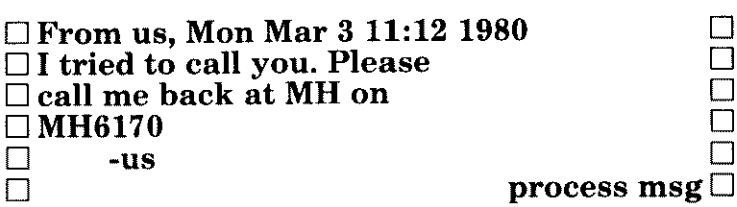



Fig. 11—The Standard Call-Memo.

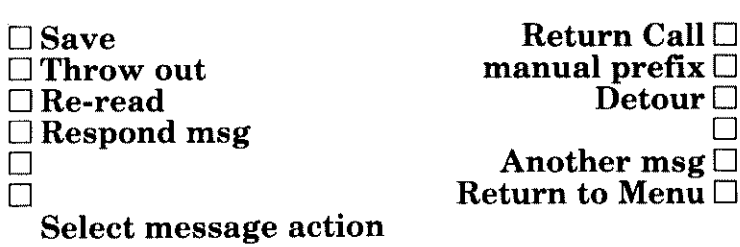



Fig. 12—The Message Action menu.

other functions have been implemented but the interest of brevity prevents any description. These include a personal calendar (see the first three buttons on Fig. 7), organizational and alphabetic directories, special-purpose directories (such as emergency numbers, dial-a-"x", and assistance), a personal telephone directory, manual selection of dial prefixes, and a food applications demonstration. In addition, the tree provides access to standard *UNIX* programs supplying facilities like a monthly calendar, time and date, a desk calculator simulation, and a large selection of computer games. Furthermore, accesses to a voice storage system and a dial-up dictation system have been implemented that present a highly satisfactory human interface.

## V. FUTURE DIRECTIONS

The functional split between the teleterminal and the host has provided a flexible and efficient interface for experimenting with new applications. We have "ported" the application software from a Digital Equipment PDP 11/45 system using the MERT[15] operating system to a Tandem computer using the Guardian[16] operating system. The human interface remained the same even though the I/O characteristics of the two systems are quite different.

Our experience exposed several areas where the current system is deficient. In the human interface, the teleterminal provides special keys for delete line, delete character, and program interrupt. Two additional special keys labeled "end of file" and "program abort" are needed to provide a completely operating-system-independent interface. Keys with the permanent labels "explain," "backup," and "restart," would improve the human interface to the access tree by freeing three of the relabelable buttons for other functions. An extension of the screen management allowing an arbitrary screen split between button labels and scrolled text would improve the human interface by allowing communications in excess of 32 characters.

The interface between the teleterminal and application programs could be improved by having the teleterminal emit a special button status message when the telephone goes off-hook. This would allow applications to respond directly to dial requests, eliminating a button push under some circumstances. A button label frame buffer capable of retaining about 30 button labels coupled with a special "more" key would remove the teleterminal-specific restriction that no more than twelve buttons can be labeled at one time.

The current teleterminal interface to the host supports a single character-oriented data stream. Application software must take special care to ensure that a message signaling the occurrence of an asynchronous event does not get interspersed with other messages. For example, if an electronic mail application wanted to inform the user of the

arrival of a message, the application software would have to coordinate an attempt to write the message "you have mail" with all other programs that might have reason to write to the instrument. This responsibility belongs in the network and in the teleterminal.

The solution requires the network to support some form of data multiplexing. For teleterminal applications, the network must support a control channel and several data channels. The control channel is a message-oriented channel used to establish the switch between data channels. The network must ensure that control messages are atomic, delivered error free, and are flow controlled. In order to ensure that control messages remain synchronized with the data channels, it is advisable that the control channel be implemented as a subchannel on a multiplexed data stream. The teleterminal would require enhancements to conform to the message protocol of the control channel and to route data over the appropriate data channel. It is sufficient that the teleterminal be able to maintain several data connections but only be able to receive (and send) data over one channel at a time.

For a network with this form of data multiplexing, asynchronous events, such as the arrival of electronic mail, could be implemented by sending an "alert" message to the teleterminal over the control channel. The teleterminal would respond by flashing a button with the **alert** label. The user would eventually respond to the alert by pressing the **alert** button, causing the teleterminal to send a "hold" status message over the current active data channel, switch to the mail data channel, and send a "proceed" status message over the mail channel. The mail program could then take control of the screen. To return to the program interrupted by the mail function, either the user would press a permanently-labeled **resume** buttom or the mail program would automatically return by sending an appropriate control message to the teleterminal. In either case, the teleterminal would send a status message over the previous data channel. The application program that had been put on hold would take over the screen by rewriting the last frame before the interruption.

A new design of the teleterminal has been recently completed and it incorporates many of the features described. Cosmetically, the screen is logically the same size as that of a conventional computer terminal and the keyboard is large enough for touch-typing (see Fig. 13). Thus, the set of services appropriately addressed is expanded from "enhanced telephony" to include information management and office automation.[17] Functionally, the new teleterminal provides a telephone interface like that of a traditional 6-button key telephone, thus allowing easy interface to call directors and speakerphones. The internal processor is a "microcomputer system" running under a *UNIX*-like operating system.[18] At this time it simply emulates the functions of the old

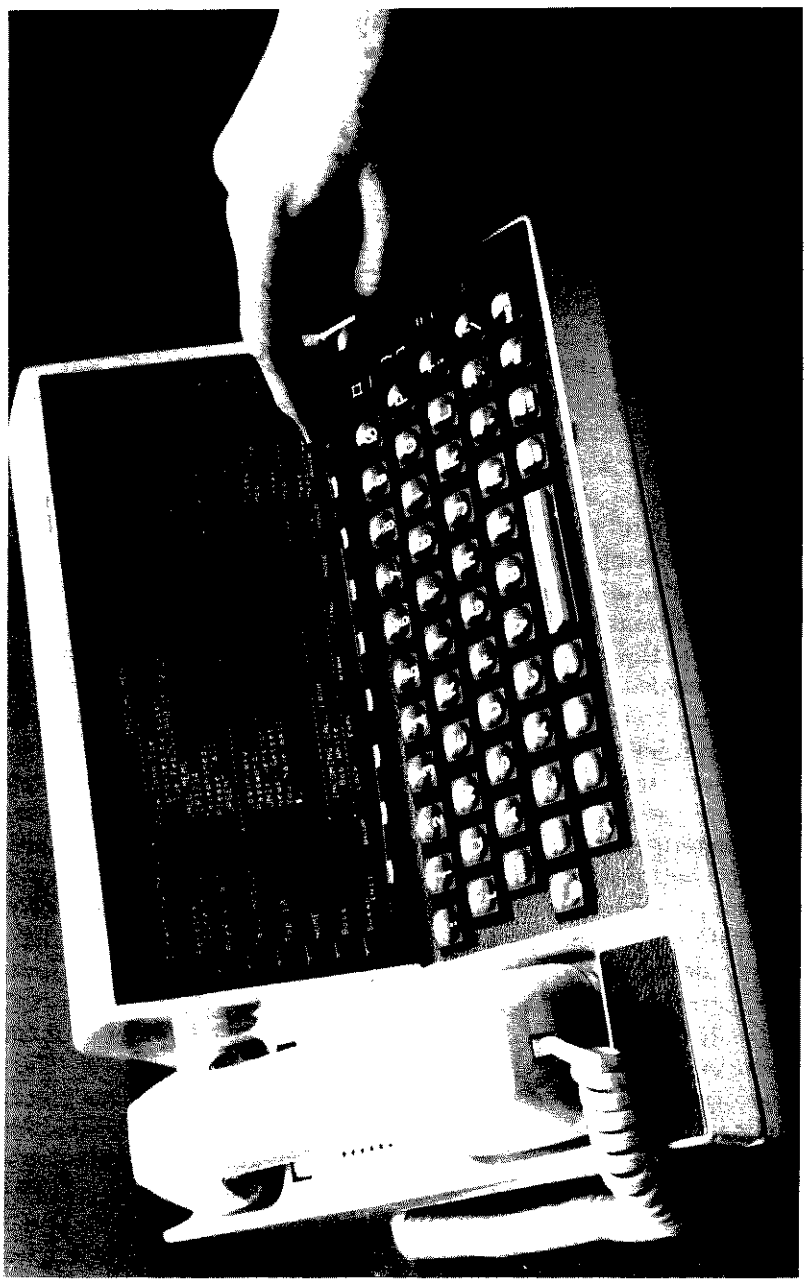


Fig. 13—Redesigned teleterminal.

internal software; however, many more elaborate functions are planned.

One of these is the capacity for a multiplicity of message "windows," each supporting one of the independent data connections mentioned above. Investigations into combining voice and data onto one digital channel are proceeding.[19] New applications become feasible once the teleterminal is connected to an intelligent switching office: for example, displaying the number of a calling party and a variety of transfer and screening functions. Many new capabilities, and improvements to existing capabilities, have been suggested by the teleterminal users and these are gradually being added.

## VI. CONCLUSION

This paper has described the software strategy behind the construction of a research tool called a teleterminal. The realization of this strategy lies in a software architecture whereby intelligence is distributed between the teleterminal's internal processor and a host computer. The programs resident in each of these locations, and their intercommunication, have been discussed.

More in the abstract, another part of this strategy is an access method whereby the user of a future teleterminal can conveniently interface to the abundance of applications that could be available. General concepts have been presented and a first implementation has been described from the user's viewpoint. The proposed access method has been found highly acceptable by colleagues and acquaintances but formal testing[20,21] on the general public has only just begun. This is ongoing work and this paper represents a "snapshot in time" that is already slightly out of date. The first working teleterminal was demonstrated internally in September of 1978. Since that time three dozen teleterminals have been constructed and a user group assembled. The software has undergone innumerable changes, and the newer "model" was designed and constructed.

Acknowledgments are owed to the following colleagues for their contributions to the software effort of the project: Bob Anderson for assistance with the internal software, Martin Sturzenbecker for the special mail program and a filter that enables the access program to work at a regular terminal, Ron Gordon for the tree "compiler" and the definition of the "high-level" data language illustrated in Fig. 4, Ron and Martin for a "Food Applications" program that clearly illustrates the value of organizing "atomic functions" into "generic capabilities," Misha Buric for the personal calendar program, Al Usas for support with the "*UNIX*-Tandem" environment, and Bob Allen and Donna Zanolla for their human factors testing.

## REFERENCES


1. D. W. Hagelbarger, R. V. Anderson, and P. S. Kubik, "Experiments with Teleterminals," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.1.1-5.
2. D. W. Hagelbarger, "Experiments with Teleterminals," *B.S.T.J.*, this issue.
3. G. D. Bergland, "An Experimental Telecommunications Test Bed," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.3.1-5.
4. R. W. Lucky, "A Flexible Experimental Digital Switching Office," *Proc. 1978 Int. Zurich Seminar on Digital Commun.*, Zurich, Switzerland, 1978.
5. R. A. Thompson, "Accessing Experimental Telecommunications Services," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.2.1-5.
6. G. Cohen, *The Psychology of Cognition*, New York: Academic Press, 1977.
7. G. T. Uber, et al., "The Organization and Formating of Hierarchical Displays for the Online Input of Data," *Proc. Fall Joint Computer Conf.*, 1968.
8. D. L. McCracken and G. G. Robertson, "Editing Tools for ZOG, a Highly Interactive Man-Machine Interface," *Proc. Int. Conf. Commun.*, Boston, MA, 1979.
9. M. Behzad and G. Chartrand, *Introduction to the Theory of Graphs*, Boston, MA: Allyn and Bacon, 1971.
10. "The UNIX Time-Sharing System," *57*, Part 2, a dedicated special issue of the *B.S.T.J.* (July-August 1978).
11. B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, Englewood Cliffs, NJ: Prentice-Hall, 1978.
12. R. D. Gordon and D. L. Smith, "An Access Tree Editor," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.7.1-5.
13. M. A. Jackson, *Principles of Program Design*, New York: Academic Press, 1975.
14. S. R. Bourne, "The UNIX Shell," *B.S.T.J., 57* (July-August 1978), pp. 1971-90.
15. H. Lycklama and D. L. Bayer, "The MERT Operating System," *B.S.T.J., 57* (July-August 1978), pp. 2049-86.
16. *Tandem 16, Guardian™ Operating Manual.* Tandem Computers Inc., 1980.
17. R. N. Klapman, "Enhanced Communications in an Executive Office," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.4.1-5.
18. W. M. Schell, "Control Software for an Experimental Teleterminal," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.8.1-5.
19. R. A. Thompson, "An Experimental User-Resident Communications Controller Supporting Sub-Rate Circuit-Switched Service," *Proc. Int. Symp. on Subscriber Loops and Services*, Munich, Germany, 1980, pp. 68-71, and *the IEEE Trans. Commun., COM-30*, Number 6 (June 1982), pp. 1399-408.
20. R. B. Allen, "Cognitive Factors in the Use of Menus and Trees: An Experiment," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.5.1-5.
21. R. A. Thompson, "User's Perceptions with Experimental Services and Terminals," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, 1981, pp. F2.6.1-5.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

# Experimental Teleterminals—Hardware

BY D. W. HAGELBARGER, R. V. ANDERSON, and P. S. KUBIK

(Manuscript received May 7, 1982)

*We have designed and built a series of experimental telephones that have facilities for both voice and data communication. The first of these has a 5-inch cathode ray tube character display and a small, full ASCII keyboard. Next to the display are twelve buttons whose labels are part of the display. The set is only a little larger than a standard telephone and can sit permanently on your desk without dominating it. It is a combination of an abbreviated computer terminal and a telephone and uses a microprocessor. Favorable experience with this first model led to several variations with larger displays and keyboards. We are still conducting experiments exploring the uses of these teleterminals.*

## I. INTRODUCTION

We have felt for some time now that a telephone set capable of providing integrated voice and data communications is desirable. A series of experiments to explore the possible uses of such a phone is under way. The key word here is "experiments"; we are not describing a development project and in fact one would not build a commercial product the way these experiments were built. The sets described here were internally referred to as GETSETS, an acronym for General-purpose Electronic Telephone SETS.

We have made a telephone with a microprocessor, a small keyboard, and a 5-inch cathode ray tube (CRT) display (see Fig. 1). The phone is small enough to sit permanently on your desk and measures about 9 inches wide, 10 inches deep, and 6 inches high. The keyboard has full American Standard Code for Information Interchange (ASCII) capability and the display has a capacity of 16 lines of 32 characters each. There are 12 buttons next to the display that can be labeled by writing text on the screen adjacent to the buttons. These buttons permit the user to perform any of a large number of complicated operations. The usual problem with having a large repertory of complicated operations

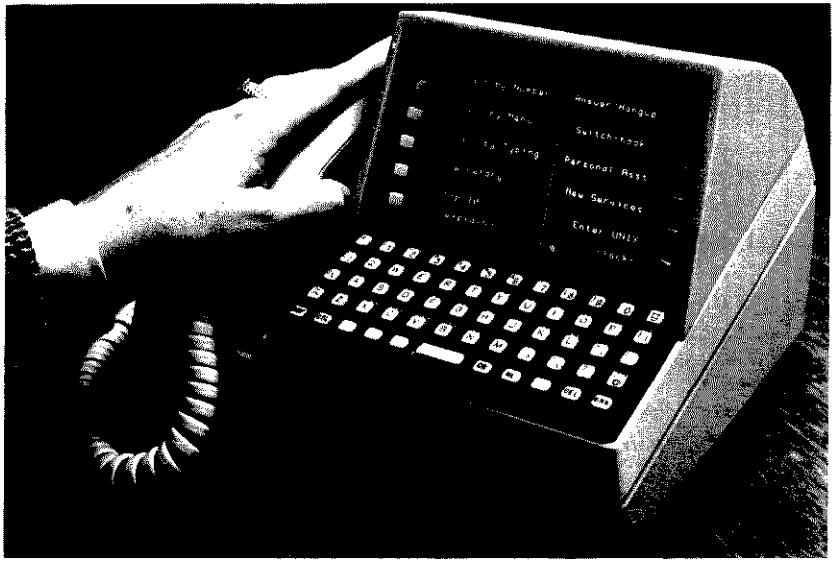


Fig. 1—Set 32.

is that it is difficult to remember the exact form of the commands if they are only used occasionally. It is much easier to recognize the command when it is presented as a button label, rather than having to recall it.

Even though the GETSET can function as a computer terminal, that is not our intent; the emphasis is on using the microprocessor to provide new and improved telephone functions. Modern electronic private branch exchanges, for example, have a processor powerful enough to provide abbreviated dialing, message forwarding, incoming party announcing, appointment calendar, and similar services. To access these functions, the telephone set needs a display and an alphanumeric keyboard. Since the set interacts with a host computer the keyboard needs full ASCII capability. All the ASCII characters, no matter how obscure, must be available. (It has been our experience that any apparently unneeded key, such as '$' in scientific computing, is given a critical function.) The display is a compromise between the desire for many characters and the limits of desk space.

In the future it will no doubt be possible to send mixed voice and data over a digital network; however, to experiment today we must use the existing analog network. In today's network there is a digital data path to the host computer and a separate voice line used for placing calls. The voice line uses the standard tone calling, dial tone, ringing, busy, etc.

## II. MECHANICAL CONSTRUCTION

We started with a 10-button Multibutton Electronic Telephone (MET) set and removed the flat plate that contains the lamps, keys, and tone caller from the right side. This was replaced with the keyboard and a hood covering the CRT monitor. The phone network, speaker for the tone ringer, and switchhook are, as before, under the handset; however, it was necessary to repackage the network with the hybrid transformer shielded with mu-metal and rotated approximately 60 degrees about a vertical axis in order to eliminate magnetic pickup from the vertical oscillator coil and deflection yoke of the monitor.

All the electronics except for the power supply are in the set, under the keyboard and CRT tube. In so far as possible all parts are mounted to the sturdy grey plastic frame of the MET set. This allows the set to be operated with the top and bottom covers removed. There is a small "dog house" on the back of the set that covers the neck of the CRT tube, which is about 1½ inches too long.

## III. FUNCTIONAL DESCRIPTION

Figure 2 shows a block diagram of the set. The keyboard chip, Standard Microsystems Corp. (SMC) KR2376-ST, converts key switch closures to 7-bit parallel ASCII codes, which are sent to the microprocessor. The buttons adjacent to the display are connected to this chip and are assigned control codes. The microprocessor, Intel 8748 with 8155 random access memory (RAM) and 8755A erasable programmable read only memory (EPROM), communicates with the host computer over a pair of serial data lines (typically 1200 baud) using a universal asynchronous receiver/transmitter (UART), Harris 6402-9.

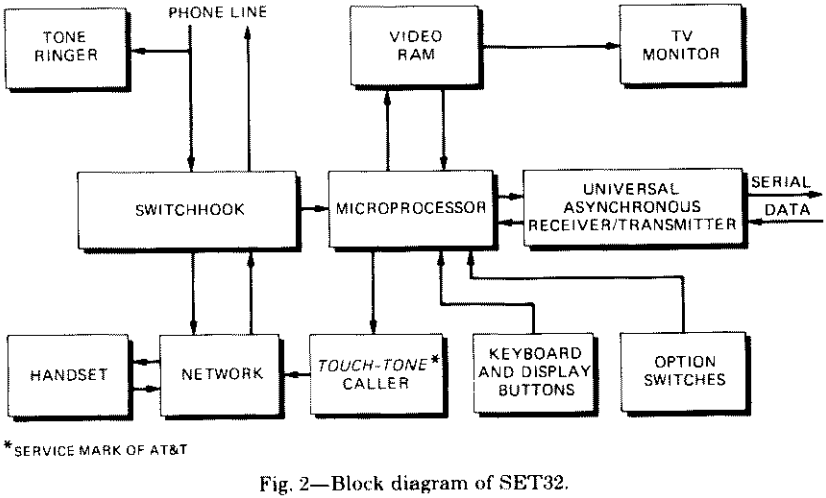



Fig. 2—Block diagram of SET32.

The microprocessor controls the display with a Matrox MTX 1632A Video RAM. To the microprocessor this looks like a 512 × 8-bit random access memory. Writing an ASCII code in location 000 causes the corresponding character to appear at the left end of the top line of the display. Location 001 comes next, and so on, with location 511 appearing at the right end of the bottom line. The Video RAM contains the character generator, refresh memory, and video and sync generators to drive the monitor.

The microprocessor also controls the tone caller, selecting the frequencies and closing a relay that mutes the phone transmitter, attenuates the receiver, and connects the caller through an isolation transformer to the network. The microprocessor can sense the state of the switchhook and make the clicker give audio feedback for the keyboard.

The monitor is a Ball Corp. TV-50 Data Display Monitor. We had to remake the printed circuit card in order to pack it into the set. This required mounting the components cordwood style and arranging the tall components to miss the neck of the CRT tube.

The tone ringer uses the American Microsystems Inc. (AMI) S2561 chip. It monitors the phone line for ringing current and converts it to tones driving the small standard speaker used in the MET set. It has a feature of ringing louder if the phone is not answered promptly. We have kept the rheostat from the MET set, which allows one to control the loudness of ringing.

Four of the seven option switches, which are mounted at the back of the set, control the baud rate, which can be set to any of the following speeds: 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600. The other switches are: local or on-line, half or full duplex, and RS 232 or current loop. There is also a reset push button that clears the screen and restarts the microprocessor program. Another control sets display brightness.

## IV. POWER

The set draws about 0.7 ampere at +5 volts, about 0.8 ampere at +12 volts, and about 25 milliamperes at −12 volts for a total of about 13 watts. We have added slots in the rear top cover below the hood. These allow air flow, which keeps the set from overheating. The −12 volts is used only by the keyboard chip and the Electronic Industries Association (EIA) interface. The power at +12 volts is dependent on the display adjustment; making the raster wider, for instance, increases the current needed.

## V. FIRMWARE

The program for the microprocessor is discussed in the companion paper.[1] Tables I and II of Ref. 1 summarize the command codes that the host uses to control the teleterminal.

## VI. FURTHER DETAILS

### *6.1 The keyboard*

The keyboard is based on a calculator keyboard. It has a molded silastic membrane with a hollow conical bump for each key. Inside each bump is a piece of conducting rubber; pressing on a button flattens the bump and pushes the conductive part against a printed circuit board where it electrically connects a pair of intermeshed fingers. The shape is such that the cone "snaps through" and gives a good feel. The keys are 12 mm on centers and the buttons are 5.5 by 6.5 mm. Making the keys relatively small allows even a very large finger to push a button without hitting the ones beside it. The buttons are too close together to permit touch-typing; it is really a two-finger keyboard.

The key array is 5 keys high and 12 wide with the space "bar" taking up two key positions. The top row is used for placing calls. In addition to the ten digits, the leftmost key has "*" and the rightmost is labeled with a modified "square" as well as "underline." (In ASCII use, the key produces "underline" in both upper and lower case.)

The clicker, which gives acoustic feedback, is a Western Electric MA5A relay with the contact stack removed. It is mounted so the armature strikes the frame to make the click. The microprocessor drives it with a 14-millisecond pulse.

The six buttons on each side of the display are also mounted 12 mm on centers. The vertical centering and height of the display are adjusted so that the buttons line up with the first line and every third line after that.

### *6.2 CRT monitor*

Since telephones are used in many different ambient light situations, it is desirable to improve the contrast of the screen. We have tried tubes with dark-colored phosphor and also bonded, neutral-grey filter faceplates. Both of these decrease the ambient light reflected from the screen. Current sets have the dark-colored phosphor and an etched front surface on the faceplate.

### *6.3 Microprocessor*

We started with just an Intel 8748; however, it soon became clear that it was not fast enough to keep up with incoming display data at 1200 baud. We therefore added the 8155 RAM, which has 256 bytes, as a buffer. This is enough for half of a solid screenful and seems to be adequate. It also gave us some more input/output (I/O) ports. Next we ran out of program space so we added the 8755A EPROM. This added 2K bytes, about half of which are used, and more I/O ports, which we are also using.

## VII. STATUS AND FURTHER EXPERIMENTS

We have built about three dozen of these sets. Approximately half of them are being used by members of our center in an experiment to test their use by a community of users and to develop new telephone functions. Others are mainly used for demonstrating the concept of a teleterminal. To test various proposed improvements, we have built three "breadboard" models. These models contain only keyboards and displays. We connect them to a Cromemco Z80 microprocessor. This has the advantage of allowing us to program in C language rather than assembly language.

Figure 3 shows the color set. It has the same keyboard as the original set. The display is a 5-inch color tube from a Sony color TV monitor. Since the color tube is larger than the black and white one, this makes the case deeper and boxier than the first set.

Figure 4 shows an intermediate-size set. The display is the same height but 1-1/2 inches wider than the first set. It has 16 lines of 64 characters each. We have added nine new buttons across the bottom of the screen that can be labeled with the last line of the display. We have used the same membrane keys but spread them sideways to take up the extra 1-1/2 inches. You can place your fingers on adjacent keys (with no extra space). It may be possible to touch-type on this keyboard.

Figure 5 shows the largest of the three models. It has almost a full-

Fig. 3—The color set.

Fig. 4—Set 64.

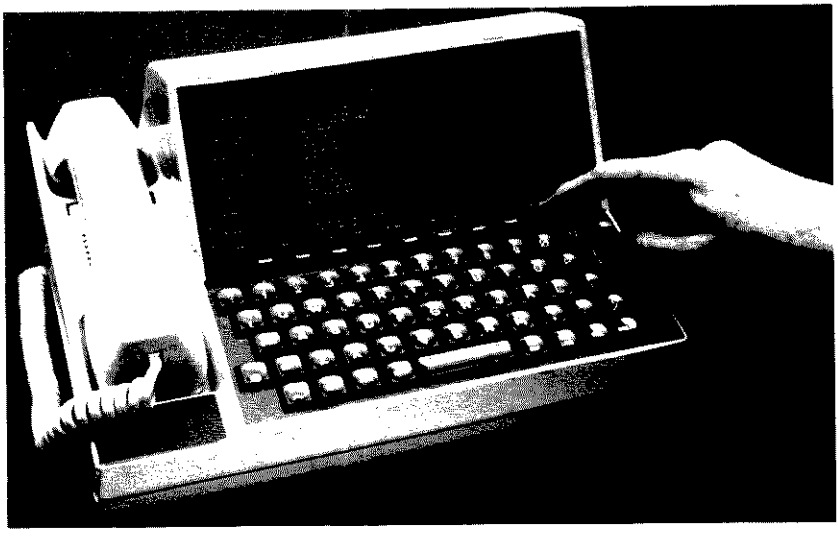


Fig. 5—Set 80.

sized keyboard. The display has 24 lines of 80 characters, the same as most computer terminals. It is a low-profile tube so the display is about 3-1/2 inches high and 7-1/4 wide; however, the definition is good and it is easy to read. The characters are about the same size as

ordinary typewriter output. There are eight buttons on each side and nine across the bottom of the display. With the addition of a speakerphone and a telephone interface that imitates a six-button keyset, this set is being used in the Executive Planning Information and Communication (EPIC) trial.[2]

## VIII. CONCLUSION

This paper has described a series of experimental teleterminals. These have been used to provide a new, computer-enhanced telephone and other services. Probably no one teleterminal suits all users' needs. Rather there should be a variety available, so that users can choose the keyboard display and other features that are most suited to their needs.

## IX. ACKNOWLEDGMENTS

Many people have made contributions to this project. We owe inspiration to H. S. McDonald for his notion of a "Superphone." We are indebted to D. L. Bayer for guidance and help in programming the microprocessor. R. A. Payne built the cases for the color set and the intermediate-size set; W. Kaminski designed the keyboards and displays for all the breadboard sets, and J. M. Gaughran contributed the layout for more than a dozen different printed circuit cards. He also expedited their production, often under a tight schedule. We particularly want to thank R. A. Thompson for the programs in the host computer, which make the teleterminals do something useful.

## REFERENCES

1. D. L. Bayer and R. A. Thompson, "An Experimental Teleterminal—The Software Strategy," B.S.T.J., this issue.
2. Richard N. Klapman, "Enhanced Communications in an Executive Office," Nat. Telecommun. Conf. Rec., *3*, 1981, pp. F2.4.1–5.

## CONTRIBUTORS TO THIS ISSUE

**Robert V. Anderson,** Bell Laboratories, 1954—. Mr. Anderson began his career as a Technical Aide. He has worked on a wide variety of projects including error-correcting code demonstrator, seismic encoder, world's fair exhibits, aids for the handicapped, computer control of the Murray Hill exhibit area, and computer terminals. He is currently a Member of Technical Staff.

**Douglas L. Bayer,** B.A., 1966, Knox College; M.S. (Physics), 1968, and Ph.D. (Nuclear Physics), 1970, Michigan State University; Rutgers University, 1971-1973; Bell Laboratories, 1973—. At Rutgers University, Mr. Bayer was involved in low-energy nuclear scattering experiments. At Bell Laboratories, he has been involved in computer operating systems research on mini- and microcomputers. Member, ACM.

**Aland K. Chin,** B.A., 1972, Brandeis University; M.S., 1975, Ph.D., 1977, Cornell University; Senior Research Engineer, Honeywell Electro-Optics Center, 1977–1978; Bell Laboratories, 1978—. Mr. Chin is involved in the design, processing, and characterization of light-emitting diodes for optical communication systems. Member, American Physical Society, American Association for the Advancement of Science, Phi Beta Kappa, New York Academy of Science, Electrochemical Society.

**Richard V. Cox,** B.S. (Electrical Engineering), 1970, Rutgers University; M.A., 1972, Ph.D., 1974 (Electrical Engineering), Princeton University; The Aerospace Corporation, 1973–1977; Assistant Professor, Rutgers University, 1977–1979; Bell Laboratories, 1979—. Mr. Cox is a member of the Acoustics Research Department. His current research interests are in digital speech coding, analog speech scrambling, and real-time speech processing systems.

**Michael A. DiGiuseppe,** B.S. (Chemistry), 1968, Polytechnic Institute of New York; Ph.D. (Chemistry), 1975, Brown University; Allied Corporation, 1973–1980, Bell Laboratories, 1980—. At Allied Corporation Mr. DiGiuseppe was engaged in the crystal growth of garnets for use as laser-hosts and substrates for magnetic films. Research efforts focused on high temperature phase equilibria in oxide melts. At Bell Laboratories Mr. DiGiuseppe is engaged in the crystal growth research of alloys for lightwave applications. Member, AACG (current AACG-NJ chairman), ACS, AAAS, Sigma Xi.

**Nancy Y. Graham,** A.B., 1959, M.A., 1962 (Mathematics), University of California, Berkeley; Bell Laboratories, 1970–1975, 1979—. Ms. Graham was a part-time employee in the Acoustics Research Department from 1970 to 1975, where she became interested in computer graphics. Ms. Graham returned to Bell Laboratories in 1979 and is a Member of Technical Staff in the Computer Graphics Group.

**David W. Hagelbarger,** A. B. (Chemistry, Mathematics, Physics), 1942, Hiram College; Ph.D. (Physics), 1947, California Institute of Technology; Bell Laboratories, 1949—. Mr. Hagelbarger has done research in a variety of fields including learning machines, stability of molten zones, magnet design, error-correcting codes, information retrieval, seismic monitoring of nuclear test bans, world's fair exhibits, switching networks, aids for the handicapped, and teaching aids. He is currently interested in improving the interface between computers and people.

**Nuggehally S. Jayant,** B.Sc. (Physics and Mathematics), 1962, Mysore University; B.E., 1965, and Ph.D. (Electrical Communication Engineering), 1970, Indian Institute of Science, Bangalore; Research Associate at Stanford University, 1967–1968; Bell Laboratories, 1968—. Mr. Jayant was a visiting scientist at the Indian Institute of Science in 1972 and 1975. He has worked in the field of digital coding and transmission of waveforms, with special reference to robust speech communications. He is also editor of the IEEE Reprint Book, *Waveform Quantization and Coding.*

**Peter D. Karabinis,** B.E.(E.E.), 1974, M.E.(E.E.), 1976, The City College of New York; Bell Laboratories, 1976—. Mr. Karabinis is a member of the Exploratory Radio Systems Department of the Radio Transmission Laboratory. His interests lie in wideband digital transmission over dispersive channels, optimum transmitter-receiver design techniques, and digital signal processing. He has worked on space diversity combiners and intermediate frequency equalizers, and is currently involved in design and development of components for high-speed digital radio systems. Member, Eta Kappa Nu.

**Vassilis G. Keramidas,** Ph.D. (Solid State Science), 1973, Materials Research Laboratory, Pennsylvania State University; Bell Laboratories, 1973—. Mr. Keramidas has worked on LEDs for displays and optoelectronics, on ohmic contacts to compound semiconductors and on the crystal growth, by liquid phase epitaxy, and characterization of

materials for LEDs for lightwave communications. He is currently Supervisor of a Special Materials Group. Member of American Physical Society, Electrochemical Society, American Association for Crystal Growth.

**Peter S. Kubik,** Western Electric, 1942–1947; Bell Laboratories, 1947—. Mr. Kubik started his career as a screw machine operator at Western Electric. He attended evening classes in mechanical design at Stevens Institute and Newark College of Engineering. Mr. Kubik did the physical design of two experimental switching offices, ESSEX and XDS. He has worked on a variety of projects including magnetostrictive delay lines, the first successful gas laser, world's fair exhibits, educational aids, aids for the handicapped, and computer terminals. He is currently a Member of Technical Staff.

**Barbara J. McDermott,** B.A., 1949, University of Michigan, Ann Arbor; M.A., 1962, Columbia University, New York; Haskins Laboratories, 1950–1959; Bell Laboratories, 1959—. From 1950–1959 Ms. McDermott was a research assistant at Haskins Laboratories, New York. Since 1959, she has been a Member of the Speech Research Department, Bell Laboratories, Murray Hill, NJ, where her principal research interest has been the perception of transmitted speech.

**John A. Morrison,** B.Sc., 1952, King's College, University of London; Sc.M., 1954 and Ph.D., 1956, Brown University; Bell Laboratories, 1956—. Mr. Morrison has done research in a number of different areas of applied mathematics and mathematical physics. He has recently been interested in probability theory, and various queueing problems in particular. He was a Visiting Professor of Mechanics at Lehigh University during the Fall semester, 1968. He is currently the Managing Editor of SIAM Review. Member, American Mathematical Society, SIAM, IEEE, Sigma Xi.

**Ann Marie S. Quinn,** B.S. (Linguistics and Speech Science), Rutgers University, New Brunswick, NJ; Bell Laboratories, 1969—. Ms. Quinn is presently working in the Acoustics Research Department.

**Robert H. Saul,** Ph.D. (Metallurgy and Materials Science), 1967, Carnegie-Mellon University; Bell Laboratories, 1967—. Initially Mr. Saul was engaged in characterization and development of epitaxial growth techniques for opto-electronic materials and devices. In 1972

he became Supervisor of a group responsible for developing visible light-emitting diodes. That work pioneered the use of multi-slice epitaxial techniques for achieving state of the art performance in a manufacturable growth system. Since 1975 Mr. Saul has supervised a group responsible for developing a variety of infrared light-emitting diodes that are used in optical-isolators and fiber optic systems. His current interests include development of long wavelength sources for lightwave transmission systems and reliability of detectors. Mr. Saul holds eight patents in the area of materials growth and opto-electronic devices. Senior member, IEEE; member, American Physical Society, Sigma Xi, Tau Beta Pi. Chairman, 1982 IEEE Specialist Conference on Light-Emitting Diodes and Photodetectors.

**R. A. Semplak,** B.S. (Physics), 1961, Monmouth College; Bell Laboratories, 1955—. Mr. Semplak's main research interest is in studies of atmospheric effects on micro- and millimeter-wave propagation. Currently, he is a member of the Radio Communications Research Department. Member, Sigma Xi and Commission F of the International Union of Radio Science (URSI/USNC).

**Henryk Temkin,** Ph.D. (Physics), 1975, Stevens Institute of Technology; Cornell University, 1975–1977; Bell Laboratories, 1977—. Mr. Temkin is involved in the development of materials and devices for fiber communications. He is currently a member of the Semiconductor Electronics Research Department. Member, American Physical Society, Electrochemical Society.

**Richard A. Thompson,** B.S., 1964 (Electrical Engineering), Lafayette College; M.S., 1966 (Electrical Engineering), Columbia University; Ph.D., 1971 (Computer Science), University of Connecticut; Bell Laboratories, 1963–1968, 1977—. Mr. Thompson is currently in the Digital Systems Research Department at Murray Hill, NJ. He was a member of the Electrical Engineering Department at Virginia Polytechnic Institute and State University from 1971–1977, achieving the rank of Associate Professor. His research interests include probabilistic formal languages, fault tolerance and cellular automata, the human-machine interface, and communications systems. Member, IEEE, active participant in Computer and Communications Societies.

**José M. Tribolet,** B.S. (Electrical Engineering), 1972, Instituto Superior Técnico, Lisbon, Portugal; M.S., E.E., and Sc.D degrees (Electrical Engineering) from the Massachusetts Institute of Technol-

ogy, Cambridge, in 1974, 1975, and 1977, respectively. From 1972 to 1977 Mr. Tribolet was a member of the Massachusetts Institute of Technology Research Laboratory of Electronics, where his research activities involved the application of homomorphic signal processing to speech and seismic data analysis. From 1977 to 1978 he was with the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ, as a post-doctoral fellow, working on adaptive transform coding of speech. He is presently Full Professor of Electrical Engineering and Computer Science at the Instituto Superior Técnico, Lisbon, Portugal, where he directs the Research Institute in Systems Engineering and Computer Science (INESC). Mr. Tribolet was recently on sabbatical leave at the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ, from July through December 1981, where he worked on speech recognition, coding, and scrambling. Member, Sigma Xi.

**Wilson W. Yale,** B.S. (Mathematics), 1973, M.S. (Mgt. Science), 1976, Ph.D. (Industrial Engineering), 1978, Lehigh University; Bell Laboratories, 1978—. Mr. Yale has done research in non-linear programming, cutting tool engineering, finance, decision support systems, and customer decision modeling. More recently he has been interested in stochastic problems arising from office automation and from estimating the demand for virtual private-line networks. He was a Visiting Professor of Industrial Engineering at Lehigh University during the spring semester of 1979. Member, ORSA, TIMS.

**Christie L. Zipfel,** A.B. (Physics), 1963, Vassar College; M.S., 1965, and Ph.D., 1969 (Physics), University of Michigan; Bell Laboratories, 1974—. Mrs. Zipfel is a member of the Lightwave LED Group.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Fishburn P. C., Farquhar P. H., **Finite-Degree Utility Independence.** Math Oper R 7(3):348–353, 1982.

Gehrlein W. V., Gopinath B., Lagarias, J. C., Fishburn P. C., **Optimal Pairs of Score Vectors for Positional Scoring Rules.** Appl Math O 8(4):309–324, 1982.

Graham R. L., Foulds L. R., **Unlikelihood That Minimal Phylogenies for a Realistic Biological Study Can Be Constructed in Reasonable Computational Time.** Math Biosci 60(2):133–142, 1982.

Harrison M. L., Davidson R. P., Wadsack R. L., **Bellmactm-32 - A Testable 32-Bit Microprocessor.** J Digit Sys 6(2–3):131–138, 1982.

Jagpal H. S., Sudit E. F., Vinod H. D., **Measuring Dynamic Marketing Mix Interactions Using Translog Functions.** J Bus 55(3):401–415, 1982.

Kantor W. M., **Spreads, Translation-Planes and Kerdock Sets.** Siam J Alg 3(3):308–318, 1982.

Lagarias J. C., **Best Simultaneous Diophantine Approximations. 1. Growth-Rates of Best Approximation Denominators.** T Am Math S 272(2):545–554, 1982.

Radner R., Rosenthal R. W., **Private Information and Pure-Strategy Equilibria.** Math Oper R 7(3):401–409, 1982.

Shepp L. A., **The XYZ Conjecture and the FKG Inequality.** Ann Probab 10(3):824–827, 1982.

Sobotka L. J., **Special Issue—International Test Conference (Editorial).** J Digit Sys 6(2–3):103–105, 1982.

Vardi Y., **Nonparametric-Estimation in Renewal Processes.** Ann Statist 10(3):772–785, 1982.

Walley P., Fine T. L., **Towards A Frequentist Theory of Upper and Lower Probability.** Ann Statist 10(3):741–761, 1982.

Wei M. C., Sholl H. A., **An Expression Model for Extraction and Evaluation of Parallelism in Control-Structures.** IEEE Comput 31(9):851–863, 1982.

Yannakak M., **The Complexity of the Partial Order Dimension Problem.** Siam J Alg 3(3):351–358, 1982.

## ENGINEERING

Capasso F., Logan R. A., Tsang W. T., **Interdigitated pn Junction Device With Novel Capacitance Voltage Characteristic, Ultralow Capacitance and Low Punch-Through Voltage.** Electr Lett 18(18):760–761, 1982.

Cheng C. L., Coldren L. A., Miller B. I., Rentschl J. A., Shen C. C., **Low-Resistance Ohmic Contacts to P-InP.** Electr Lett 18(17):755–756, 1982.

Kamgar A., **Miniaturization of Si MOSFET at 77-K.** IEEE Device 29(8):1226–1228, 1982.

Kartalopoulos S. V., **The Minimization of Logic Functions Utilizing 2-Dimensional Representations of Hypercubes.** Int J Elect 53(3):233–245, 1982.

Kenney J. J., **Digital Radio for Transmission at 6 and 11 GHz.** Microwave J 25(8):71+, 1982.

Levine B. F., et al., **High-Quantum-Efficiency Low-Threshold Microcleaved $Al_xGa_{1-x}As$ Lasers.** Electr Lett 18(16):690–691, 1982.

Lin C., Glodis P. F., **Tunable Fiber Raman Oscillator in the 1.32–1.41 μm Spectral Region Using a Low-Loss, Low OH-Single-Mode Fiber.** Electr Lett 18(16):696–697, 1982.

Logan, R. A., et al., **Low-Threshold GaInAsP-InP Mesa Lasers.** Electr Lett 18(18):782–783, 1982.

Ng K. K., Taylor G. W., Sinha A. K., **Instability of MOSFET due to Redistribution of Oxide Charges.** IEEE Device 29(8):1323–1330, 1982.

Stolen R. H., Howard R. E., Pleibel W., **Substrate-Tube Lithography for Optical Fibers.** Electr Lett 18(18):764–765, 1982.

Stone J., Cohen L. G., **Minimum-Dispersion Spectra of Single-Mode Fibers Measured With Subpicosecond Resolution by White-Light Cross-Correlation.** Electr Lett 18(16):716–718, 1982.

Temkin H., Chin A. K., Dutt B. V., **Schottky-Barrier Restricted GaAlAs Laser.** Electr Lett 18(16):701–703, 1982.

Tsai Y. K. et al., **Structure-Preserving Model-Reduction With Applications to Power-System Dynamic Equivalents.** IEEE Circ S 29(8):525–535, 1982.

Vanderziel J. P., Logan R. A., **Generation of Short Optical Pulses in Semiconductor-Lasers by Combined dc and Microwave Current Injection.** IEEE J Q EL 18(9):1340–1350, 1982.

## MANAGEMENT/ECONOMICS

Hsiao C., **Autoregressive Modeling and Causal Ordering of Economic Variables.** J Econ Dyn 4(3):243–259, 1982.

Rosenthal R. W., **A Dynamic Oligopoly Game With Lags in Demand—More on the Monotonicity of Price in the Number of Sellers.** Int Econ R 23(2):353–360, 1982.

## PHYSICAL SCIENCES

Anderson P. W., **Localization—A Growth Industry (Editorial).** Physica B&C 110(1–3):1830–1836, 1982.

Archer V. D., **Methods for Defect Evaluation of Thin (100) Oriented Silicon Epitaxial Layers Using a Wet Chemical Etch.** J Elchem So 129(9):2074–2076, 1982.

Banavar J. R., Cieplak M., Cieplak M. Z., **Influence of Boundary-Conditions on Random Unfrustrated Magnetic Systems.** Phys Rev B 26(5):2482–2489, 1982.

Beck S. M., Brus L. E., **Transient Raman-Scattering Study of the Initial Semi-Quinone Radical Kinetics Following Photolysis of Aqueous Benzoquinone and Hydroquinone.** J Am Chem S 104(18):4789–4792, 1982.

Beni G., Scrosatti B., **(IT) Recent Advances in Electrochemical Energetics. 4. Electrochromic Displays.** Chim Ind M 64(7–8):487–491, 1982.

Benton J. L., Kimerling L. C., **Capacitance Transient Spectroscopy of Trace Contamination in Silicon.** J Elchem So 129(9):2098–2102, 1982.

Bhatt R. N., Ramakrishnan T. V., **Effect of Mass Anisotropy on the Low-Temperature Conductivity of Disordered-Systems in Two Dimensions.** Physica B&C 110(1–3):2078–2080, 1982.

Bhatt R. N., **Frustration in Highly Random Magnetic Systems.** Physica B&C 110(1–3):2145–2147, 1982.

Bondybey V. E., **Laser Vaporization of Silicon-Carbide—Lifetime and Spectroscopy of $SiC_2$.** J Phys Chem 86(17):3396–3399, 1982.

Bondybey V. E., English J. H., Miller T. A., **The Absorption-Spectrum of the Perfluoropyridine Cation in a Ne Matrix.** Chem P Lett 90(5):394–396, 1982.

Bosch M. A., Dayem A. H., Harrison T. R., Lemons R. A., **Lamellar Solidification of Laser Melted Co-Si Films.** Appl Phys L 41(4):363–364, 1982.

Bowmer T. N., Reichmanis E., Wilkins C. W., Hellman M. Y., **Radiation Degradation of Copolymers of Methyl-Methacrylate and 3-Oximino-2-Butanone Methacrylates.** J Pol Sc Pc 20(9):2661–2668, 1982.

Brown B. L., Mills A. P., Tyson J. A., **Results of a 440-Day Search for Gravitational Radiation.** Phys Rev D 26(6):1209–1218, 1982.

Bruckenstein S., Miller B., **Current-Voltage Analysis of Photo-Electrochemical Cells Under Mass and Light-Flux Variation.** J Elchem So 129(9):2029–2034, 1982.

Bucksbaum P. H., Boker J., Storz R. H., White J. C., **Amplification of Ultrashort Pulses in Krypton Fluoride at 248-NM.** Optics Lett 7(9):399–401, 1982.

Cais R. E., Kometani J. M., **Ethylene Vinyl Bromide Copolymers by Reductive Debromination of Polyvinyl Bromide—A C-13 NMR-Study.** Macromolec 15(4):954-960, 1982.

Capasso F., Tsang W. T., Hutchinson A., Williams G. F., **Enhancement of Electron-Impact Ionization in Super-Lattices—A New Avalanche Photo-Diode With A Large Ionization Rate Ratio.** Inst Phys C1982(63):569-570, 1982.

Capasso F., Tsang W. T., Hutchinson A. L., Foy P. W., **The Graded Bandgap Avalanche-Diode—A New Molecular-Beam Epitaxial Structure With A Large Ionization Rate Ratio.** Inst Phys C1982(63):473-478, 1982.

Chen C. Y., Cho A. Y., Gossard A. C., Garbinski P. A., **Offset Channel Insulated Gate Field-Effect Transistors.** Appl Phys L 41(4):360-362, 1982.

Chen L. J., Mayer J. W., Tu K. N., Sheng T. T., **Lattice Imaging of Silicide Silicon Interfaces.** Thin Sol Fi 93(1-2):91-97, 1982.

Chu S. N. G., **A Simple Damage-Free Grooving Method for Revealing the Quality of InP/InGaAsP Multilayer Structure.** J Elchem So 129(9);2082-2085, 1982.

Davies J. H., Lee P. A., Rice T. M., **Electron Glass.** Phys Rev L 49(10):758-761, 1982.

Disalvo F. J., Waszczak J. V., **Magnetic-Properties of Copper Chalcogenide Spinels.** Phys Rev B 26(5):2501-2506, 1982.

Dubois L. H., **Oxygen-Chemisorption and Cuprous-Oxide Formation on Cu(111)—A High-Resolution EELS Study.** Surf Sci 119(2-3):399-410, 1982.

Duncan A., Roskies R., Vaidya H., **Monte-Carlo Study of Long-Range Chiral Structure in QCD.** Phys Lett B 114(6):439-444, 1982.

Dynes R. C., **Localization and Correlation—Effects in Metals and Semiconductors Experiment.** Physica B&C 110(1-3):1857-1865, 1982.

Espinosa G. P., Cooper A. S., Barz H., **Isomorphs of the Superconducting—Magnetic Ternary Stannides.** Mater Res B 17(8):963-969, 1982.

Frankenthal R. P., Siconolfi D. J., **The Equilibrium Surface—Composition of Tin Lead Alloys.** Surf Sci 119(2-3):331-348, 1982.

Girvin S. M., Jonson M., Lee P. A., **Interaction Effects in Disordered Landau-Level Systems in Two Dimensions.** Phys Rev B 26(4):1651-1659, 1982.

Golding B., Fox D. L., Haemmerle W. H., **Dephasing of Extrinsic Tunneling Systems in Silica Glass.** Physica B&C 110(1-3):2039-2040, 1982.

Goldsmith P. F., Snell R. L., DeGuchi S., Kortkov R., Linke R. A., **Vibrationally Excited Cyanoacetylene in the Orion Molecular Cloud.** Astrophys J 260(1):147-158, 1982.

Golovchenko J. A., Cox D. E., Goland A. N., **Critical Analysis of the Charge-State Dependence of the Energy-Loss of Channeled Ions.** Phys Rev B 26(5):2335-2340, 1982.

Greenside H. S., Coughran W. M., Schryer N. L., **Non-Linear Pattern-Formation Near the Onset of Rayleigh-Benard Convection.** Phys Rev L 49(10):726-729, 1982.

Griffiths J. E., Sunder W. A., **Raman-Spectra and Phase-Transitions in $O_2PF_6$.** J Chem Phys 77(6):2753-2756, 1982.

Hall C. K., Helfand E., **Conformational State Relaxation in Polymers—Time-Correlation Functions.** J Chem Phys 77(6):3275-3282, 1982.

Hebard A. F., Fiory A. T., **Vortex Dynamics in Two-Dimensional Superconductors.** Physica B&C 110(1-3):1637-1648, 1982.

Hegarty J., Sturge M. D., Weisbuch C., Gossard A. C., Wiegmann W., **Resonant Rayleigh-Scattering From an Inhomogeneously Broadened Transition—A New Probe of the Homogeneous Linewidth.** Phys Rev L 49(13):930-932, 1982.

Heller A., **Electrochemical Solar-Cells.** Solar Energ 29(2):153-162, 1982.

Ho K. M., Fu C. L., Harmon B. N., Weber W., Hamann D. R., **Vibrational Frequencies and Structural-Properties of Transition-Metals Via Total-Energy Calculations.** Phys Rev L 49(9):673-676, 1982.

Kaplan A. E., Smith P. W., Tomlinson W. J., **Switching of Reflection of Light at Non-Linear Interfaces.** P Soc Photo 317:305-310, 1981.

Khanarian G., **Direct-Current Electric-Field Induced 2nd Harmonic-Generation in Flexible Molecules and Polymers.** J Chem Phys 77(5):2684-2687, 1982.

Lanzerotti L. J., et al., **$SO_2$ Ice and Applications to the Frosts of IO.** Astrophys J 259(2):920-929, 1982.

Leventhal M., et al., **Time-Variable Positron-Annihilation Radiation From the Galactic-Center Direction.** Astrophys J 260(1):L 1+,1982.

Levine B. F., Tsang W. T., Bethea C. G., Capasso F., **Electron-Drift Velocity-Measurement in Compositionally Graded $Al_xGa_{1-x}As$ by Time-Resolved Optical Picosecond Reflectivity.** Appl Phys L 41(5):470–472, 1982.

Lloyd J. R., Nakahara S., **Some Remarks on Void Growth in Thin Silver Films (Letter).** Thin Sol FI 92(1–2):L 61, 1982.

Loponen M. T., Dynes R. C., Narayanamurti V., Garno J. P., **The Time-Dependent Specific-Heat of Dielectric Glasses.** Physica B&C 110(1–3):1873–1879, 1982.

Marcus M. A., **Lack of 2nd-Order Phase-Transitions in Cubic Blue Phases in Landau Theory.** Molec Cryst 82(5):167–171, 1982.

McAfee K. B., Szmanda C. R., Hozack R. S., Johnson R. E., **Dynamics of Energy-Transfer During Charge-Exchange in Symmetric Diatomic Systems.** J Chem Phys 77(5):2399–2407, 1982.

McBrierty V. J., Douglass D. C., Wudl F., **A Nuclear Magnetic-Resonance Study of $(TMTSF)_2PF_6$.** Sol St Comm 43(9):679–682, 1982.

McBrierty V. J., Douglass D. C., Furukawa T., **Magnetic-Resonance and Relaxation in a Vinylidene Fluoride Trifluoroethylene Co-Polymer.** Macromolec 15(4):1063–1067, 1982.

McCrory J. C., **Electroanalytical Chemistry of Chlorinated Phenols at a Glassy-Carbon Electrode.** Analyt Chim 141(SEP):105–114, 1982.

Miller R. C., Gossard A. C., Tsang W. T., Munteau O., **Bound Excitons in P-Doped GaAs Quantum Wells.** Sol St Comm 43(7):519–522, 1982.

Miller R. C., Tsang W. T., Munteau O., **Extrinsic Layer at $Al_xGa_{1-x}As$-GaAs Interfaces.** Appl Phys L 41(4):374–376, 1982.

Miller T. A., Bondybey V. E., **Spectroscopy of Molecular-Ions (Review or Bibliog.)** Appl Sp Rev 18(1):105–169, 1982.

Mollenauer L. F., Vieira N. D., Szeto L., **Mode-Locking by Synchronous Pumping Using a Gain Medium with Microsecond Decay Times.** Optics Lett 7(9):414–416, 1982.

Murray C. A., Allara D. L., Hebard A. F., Padden F. J., **Determination of Sample Morphology of Multilayered Structures Used in Surface Enhanced Raman-Scattering Experiments.** Surf Sci 119(2–3):449–478, 1982.

Nelson E. D., Kaufman S., **Release Rate Calorimetry of PVC Compounds Containing Antimony Oxide and Iron-Oxide.** J Fire Flam 13(2):79–103, 1982.

Olego D., Chang T. Y., Silberg E., Caridi E. A., Pinczuk A., **Compositional Dependence of Band-Gap Energy and Conduction-Band Effective Mass of $In_{1-x-y}Ga_xAl_yAs$ Lattice Matched to InP.** Appl Phys L 41(5):476–478, 1982.

Penzias A. A., **Laser Patents (Letter).** Science 217(4565):1082, 1982.

Raghavachari K., Haddon R. C., Starnes W. H., **Primary Event in the Thermal Dehydrochlorination of Pristine Polyvinyl-Chloride—Intermediacy of A Cyclic Chloronium Ion.** J Am Chem S 104(19):5054–5056, 1982.

Ramakrishnan T. V., Sur K., **Theory of a Mixed-Valent Impurity.** Phys Rev B 26(4):1798–1811, 1982.

Riley J. E., **Ultrahigh Sensitivity Uranium Analyses Using Fission-Track Counting—Further Analyses of Semiconductor Packaging Materials.** J Rad Chem 72(1–2):89–99, 1982.

Schluter M., **Theoretical—Models of Schottky Barriers.** Thin Sol Fi 93(1–2):3–19, 1982.

Schmidt P. F., Glascock M. D., **Applications and Problems of Parametric Counting.** J Rad Chem 72(1–2):231–244, 1982.

Stiles K. R., Lee J. W., **Automated Conductivity Profiler for Multilayer GaAs-(AlGa)As Structures.** Rev Sci Ins 53(9):1449–1451, 1982.

Stillinger F. H., Weber T. A., **Asymmetry Between Protons and Proton Holes in Gas Phase Neutralization Reactions.** Molec Phys 46(6):1325–1333, 1982.

Stone F. T., **Separation of Total-Loss Data Into Its Absorption and Scattering Components—A More Accurate Model for Fiber Loss.** Appl Optics 21(15):2721–2726, 1982.

Sturge M. D., **Citation Classic—Optical-Absorption of Gallium—Arsenide Between 0.6 and 2.75 eV.** CC/Phy Chem1982(38):20, 1982.

Swaminathan V., Wagner W. R., Schumaker N. E., Miller R. C., **Anomalous Bands in the Photo-Luminescent Spectra From GaAs-(Al,Ga)As Double Heterostructures.** Thin Sol Fi 93(1-2):195-205, 1982.

Thomas G. A., et al., **Temperature-Dependent Conductivity of Metallic Doped Semiconductors.** Phys Rev B 26(4):2113-2119, 1982.

Tolk N. H., et al., **Optical Radiation From Photon-Stimulated Desorption of Excited Atoms.** Phys Rev L 49(11):812-815, 1982.

Tomita A., **Interferometric-Technique for Measuring Small Gaps in Single-Mode and Multimode Fiber Connectors (Letter).** Appl Optics 21(15):2655-2656, 1982.

Tsang W. T., **Novel Optoelectronic Devices Prepared by Molecular-Beam Expitaxy.** P Soc Photo 317:66-73, 1981.

Tsang W. T., Miller R. C., Capasso F., Bonner W. A., **High-Quality InP Grown by Molecular-Beam Epitaxy.** Appl Phys L 41(5):467-469, 1982.

Tsang W. T., et al., **Molecular-Beam Epitaxially Grown 1.3 $\mu$m GaInAsP/InP Double-Heterostructure Lasers.** Electr Lett 18(18):785-786, 1982.

Tung R. T., Poate J. M., Bean J. C., Gibson J. M., Jacobson D. C., **Epitaxial Silicides.** Thin Sol Fi 93(1-2):77-90, 1982.

Wagner R. E., Tomlinson W. J., **Coupling Efficiency of Optics in Single-Mode Fiber Components.** Appl Optics 21(15):2671-2688, 1982.

Walsh W. M., Wudl F., Aharon-Shalom H. E., Rupp L. W., Vandenberg J. M., Andres K., Torrance J. B., **Itinerant-Electron Antiferromagnetism Precursor to Superconductivity in an Organic Conductor.** Phys Rev L 49(12):885-888, 1982.

Wemple S. H., Flahive P. G., Allyn C. L., Schlosser W. O., Iglesias D. E., **Design Principles for Source Via Power GaAs-FETs.** Inst Phys C1982(63):437-442, 1982.

Wertheim G. K., Remeika J. P., Buchanan D. N., **Electronic-Structure of $BaPb_{1-x}Bi_xO_3$.** Phys Rev B 26(4):2120-2123, 1982.

Whangbo M. H., Walsh W. M., Haddon R. C., Wudl F., **Band-Structure of $(TMISF)_2X$.** Sol St Comm 43(8):637-639, 1982.

Wood D. L., Nassau K., **Refractive-Index of Cubic Zirconia Stabilized With Yttria.** Appl Optics 21(16):2978-2981, 1982.

Zweier J. L., Peisach J., Mims W. B., **Electron-Spin Echo Studies of the Copper-Complexes of Conalbumin.** J Biol Chem 257(17):314-316, 1982.

## SOCIAL AND LIFE SCIENCES

Bauer D. W., Miller J., **Stimulus-Response Compatibility and the Motor System.** Q J Exp P-A 34(AUG):367-380, 1982.

Blumberg W. E., **A Statistical-Analysis of the Publications and Collaborations of Fanelli, Alessandro, Rossi (Review or Bibliog.).** Adv Exp Med 148:7-19, 1982.

Chance B., Powers L., Ching Y., **Structure and Function of the Redox Site of Cytochrome-Oxidase (Review or Bibliog.).** Adv Exp Med 148:95-109, 1982.

Daniels S. R., Bates S., Lukin R. R., Benton C., Third J., Glueck C. J., **Cerebrovascular Arteriopathy (Arteriosclerosis) and Ischemic Childhood Stroke.** Stroke 13(3):360-365, 1982.

Egan D. E., Grimes-Farrow D. D., **Differences in Mental Representations Spontaneously Adopted for Reasoning.** Mem Cognit 10(4):297-307, 1982.

Kitagawa T., et al., **Evidence for Hydrogen-Bonding of Bound Dioxygen to the Distal Histidine of Oxycobalt Myoglobin and Hemoglobin.** Nature 298(5877):869-871, 1982.

Lund A. M., **Nebraska Symposium on Motivation—Cognitive Processes (1980) (Book Review).** Am J Psycho 95(2):333-335, 1982.

Meyer D. E., Smith J. E. K., Wright C. E., **Models for the Speed and Accuracy of Aimed Movements.** Psychol Rev 89(5):449-482, 1982.

Ogawa S., Lee T. M., **Proton Stoichiometry of Adenosine 5'-Triphosphate Synthesis in Rat-Liver Mitochondria Studied by P-31 Nuclear Magnetic-Resonance.** Biochem 21(18):4467-4473, 1982.

Teich M. C., Prucnal P. R., Vannucci G., Breton M. E., McGill W. J., **Multiplication Noise in the Human Visual-System at Threshold. 3. The Role of Non-Poisson Quantum Fluctuations.** Biol Cybern 44(3):157-165, 1982.

## SPEECH AND ACOUSTICS

Fay D., **Substitutions and Splices—A Study of Sentence Blends.** Linguistics 19(7–8):717–749, 1981.

Pierrehumbert J., Liberman M., **Fundamental—Frequency in Sentence Production—Cooper W. E., Sorensen J. M (Book Review).** Cont Psycho 27(9):690–692, 1982.

Xydeas C. S., Evci C. C., Steele R., **Sequential Adaptive Predictors for ADPCM Speech Encoders.** IEEE Commun 30(8):1942–1954, 1982.

THE BELL SYSTEM
TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF COMPUTING

Volume 62 January 1983 Number 1, Part 2

Copyright © 1983 American Telephone and Telegraph Company. Printed in U.S.A.

***The 3B20D Processor & DMERT Operating System:***

# Prologue

By J. M. SCANLON

(Manuscript received June 23, 1982)

The 3B20 Duplex Processor (3B20D) and its DMERT operating system represents a new and very important building block for the Bell System's stored program network. The first system to cut over was the network control point application in Kansas City, Missouri, on September 3, 1981. By year end 1982, the common 3B20D/DMERT processor system will have been in service supporting six different telecommunication applications across 65 different sites in 17 of the 20 Bell Operating Companies, including the Long Lines Division of AT&T. By year-end 1983, the number of different applications is expected to grow to ten and the number of sites to 300, covering all Bell Operating Companies. This very rapid buildup from first introduction to substantial deployment throughout the Bell System establishes the 3B20D/DMERT processor as a key element for the continued evolution of the Bell System's stored program network.[1]

The 3B20D Processor had its origins in exploratory work, begun in 1976, to establish a successor to the 3A processor[2] deployed with No. 2B ESS and No. 3 ESS.[3] At that time, a specialized replacement processor for 3A was envisioned, which is the origin of the designation "3B." At about the same time, research in operating systems charac-

terized by the *UNIX** operating system[4] and MERT[5], and portable higher-level languages characterized by C,[6] had matured to the point that the use of these techniques in real-time telecommunications applications became feasible and highly desirable. In addition, it was becoming clear that processor capabilities to support the future needs of the stored program network would have to be significantly enhanced beyond the capabilities of then-deployed ESS processors. These enhancements were required to support what was foreseen as the broadening role of the ESS processor from primarily a highly reliable call-handling and switch-maintenance controller to a sophisticated data base manager and terminal/data link controller as well. The use of a common processor and operating system throughout the telecommunications network also was seen as providing significant economic benefits occurring both from manufacturing and operational standardization, and most importantly, from achieving a high degree of software standardization. By early 1978, these factors had crystallized into five major development goals to:

(*i*) Provide the high degree of reliability and fault tolerance traditionally expected of ESS processors.

(*ii*) Provide efficient support of the high-level language, C, and a new real-time operating system, DMERT.

(*iii*) Provide the ability to directly execute programs written for earlier ESS processors, permitting these ESSs to follow a low-cost migration path to new processor and software technology.

(*iv*) Provide hardware and software architectural features to efficiently support large real-time data base and extensive data link capabilities.

(*v*) Provide a highly efficient software-development system based on the *UNIX* operating system as an integrated capability of the processor.

These goals were met successfully by a development program that designed the processor, 3B20D, and the operating system, DMERT, in parallel. The design approach used to bring the processor and operating system into fruition are discussed in the papers included in this issue.

The issue begins with a paper that discusses the first four applications of the 3B20D/DMERT system, followed by an overview and a discussion of the architecture of the complete system. There are then 15 other papers, grouped in sequence along the major topics of hardware (5 papers), software (4), maintenance and craft interface (5), and, finally, system integration and test (1). While omitting design-level details, this collection of papers provides a comprehensive overview of the 3B20D Processor and DMERT Operating System.

* Trademark of Bell Laboratories.

It is impossible to adequately acknowledge the contributions of everyone involved in a project of the magnitude of 3B20D/DMERT—people from many organizations of Bell Laboratories, Western Electric, AT&T, Long Lines, and the operating telephone companies, all participated in important ways. The authors of this volume would like to express their gratitude to all of these people for the unity of purpose and free communication that overcame the complex organizational interfaces and technical problems, and permitted successful project completion.

## REFERENCES

1. *Stored Program Controlled Network*, B.S.T.J., *61*, No. 7 (September 1982).
2. W. N. Toy, "Fault Tolerant Design of ESS Processors," Proc. IEEE, *66* (October 1978), pp. 1126–45.
3. E. A. Irland and U. K. Stagg, "New Developments in Suburban and Rural ESS (No. 2 and No. 3 ESS)," 1974 ISS Symposium Record, pp 512/1–6.
4. D. M. Ritchie and K. Thompson, "The *UNIX* Time-Sharing System," B.S.T.J., *57* (July–August 1978), pp. 1905–29.
5. H. Lycklama and D. L. Bayer, "The MERT Operating System," B.S.T.J., *57* (July–August 1978), pp. 2049–86.
6. D. M. Ritchie, S. C. Johnson, M. E. Lesk, and B. W. Kernighan, "The C Programming Language," B.S.T.J., *57* (July–August 1978), pp. 1991–2019.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# The 3B20D Processor & DMERT As a Base for Telecommunications Applications

By R. W. MITZE, H. L. BOSCO, N. X. DeLESSIO, R. J. FRANK, N. A. MARTELLOTTO, W. C. SCHWARTZ, and R. M. WOLFE

(Manuscript received March 18, 1982)

*The 3B20D Processor and the Duplex Multienvironment Real-Time (DMERT) Operating System provide a versatile processing base to fulfill the varied processing needs of telecommunications systems. To illustrate the versatility of the 3B20D/DMERT system, this article describes the structuring of four diverse telecommunication system applications.*

## I. INTRODUCTION

The 3B20D Processor and the Duplex Multienvironment Real-Time (DMERT) operating system were designed to provide a base for a wide variety of high-availability real-time and time-sharing applications. The wide applicability of the 3B20D/DMERT combination is due to the flexible hardware configuration of the processor coupled with the multilevel structure of the DMERT operating system. The versatile, intelligent input/output processor and the high-speed dedicated dual-serial channels provide for varying peripheral configuration interconnection needs.[1,2] The microcode support for multiple resident instruction sets and the operating system support for application software at several different operating system functionality levels provide the diverse operating environment needs of teleprocessing system applications as well as data processing oriented applications.[3]

The combined 3B20D Processor and DMERT operating system can be viewed as a hierarchy of capability levels. This is depicted in Fig. 1. The lowest capability level is the 3B20D Processor hardware and

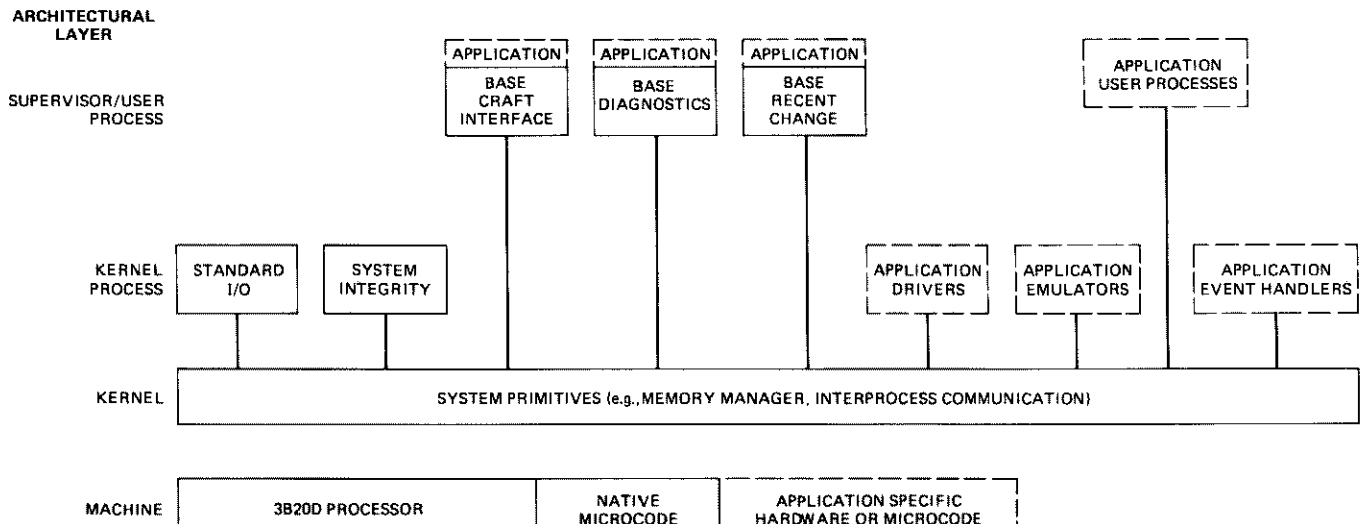



Fig. 1—3B20D/DMERT structure.

microcode. This level provides the machine instruction primitives. The next level of capability is the DMERT operating system kernel. This level provides the basic operating system primitive functions such as memory management and interprocess communication and scheduling. The next level of capability is the kernel process level. This level provides the ability to directly interface with the hardware for maximum real-time efficiency. The highest level of capability is the supervisor/user process level. This level provides full user-oriented input/output capability combined with the powerful and convenient development environment available to *UNIX** operating system users.

The combination of a multilevel operating system and support of multiple resident instruction sets gives 3B20D/DMERT exceptional adaptability. We will describe how these two capabilities can be used to provide different kinds of services to applications and illustrate their use in four systems applications.

### *1.1 Multilevel operating system*

From an application viewpoint the multilevel capability of DMERT allows an application to tune its software according to the most appropriate trade-off of real-time control versus ease of programming and maintainability. The DMERT kernel process interface allows applications to attach directly to interrupts in order to support application-specific peripherals or other custom hardware. It also allows applications to write real-time event software handlers that run at any desired hardware priority level in order to provide real-time response to events. In effect, the application shares the hardware-scheduling decisions with the DMERT operating system.

If direct hardware scheduling or control is not needed, DMERT offers the supervisor/user level of the *UNIX* operating system for time-sharing applications. This level provides a large amount of program protection and development support to make writing and maintaining user-level processes easier. At the user level all access to machine resources is controlled by DMERT so that unauthorized accesses are prevented and the user is shielded from most of the specific hardware details. In most applications, the bulk of the software is at the user level and only a small percentage needs to be written at the kernel process level.

The user-level environment of the *UNIX* operating system supplied with DMERT need not be the only user-level environment. Applications also have the ability to construct additional environments to provide specific application services for programmers. This can be especially useful as a means of ensuring uniform treatment of appli-

* Trademark of Bell Laboratories.

cation-specific resources and as a way to provide a standard development base in the situation where multiple systems may be used by a single application.

### *1.2 Multiple resident instruction sets*

The 3B20D Processor is a microcoded machine that provides sufficient microstore and control to support multiple instruction sets resident on the processor at the same time. The DMERT operating system running in 3B native code is designed to coexist with emulated software running on a different instruction set. This allows the 3B20D Processor to be used by certain applications to preserve any software investment involved in previously released systems. Since the processor dynamically shifts between the emulated and native environment by executing a single instruction, new software can be written in either emulated or native mode as needed.

## II. NO. 5 ESS

No. 5 ESS[4] is a Bell System-developed local digital switching system designed as a world-class product for both domestic and international markets and employing the latest technology. The No. 5 ESS system can be configured to serve rural offices with as few as 1000 customer lines, as well as metropolitan offices with up to 100,000 lines. Early applications of No. 5 ESS are as replacements of electromechanical switching systems. The first such system was placed in commercial service in March, 1982. This represents an important first step in the evolution of the Bell System's nationwide digital network.

In the overall No. 5 ESS architecture, the 3B20D Processor is used as the central processor complex and is responsible for a variety of functions including maintenance control, administration, human interface, system integrity, and certain centralized call-processing functions such as routing and control of global data. Some of these functions are performed by the No. 5 ESS application software, some are performed by DMERT, and some are joint responsibilities requiring close coordination between the No. 5 ESS application and DMERT. In this capacity, the 3B20D Processor is playing several different roles, and the multilevel hierarchy in DMERT allows the different functions to be placed at the most appropriate level. For each function, the tradeoffs involved were evaluated and the most appropriate capability level for No. 5 ESS processes was selected. For example, many functions such as administration, recent change, and basic diagnostics are not real-time sensitive. Therefore, the performance of the kernel process level was not required and the easier development environment of the user-process level was considered to be the deciding criterion. On the other hand, the call-processing function required the best possible real-

time performance even at the cost of lesser development convenience, so the kernel process level was chosen.

The No. 5 ESS application added an application-specific environment having features such as task management, message handling, and timing specifically oriented to No. 5 ESS call-processing needs. This same environment is provided on other processing components within the distributed No. 5 ESS architecture. This resulted in a uniform interface for No. 5 ESS developers with subsequent advantages in training, portability, and ease of development. Thus, the No. 5 ESS application demonstrates how the multiple-level architecture of DMERT allows applications the capability of making their own choices and trade-offs between performance versus flexibility and ease of development. Also, this approach shows how the kernel process interface allows an application to implement an environment most suitable for its particular needs.

## III. TRAFFIC SERVICE POSITION SYSTEM

Since its introduction in 1969, the Traffic Service Position System No. 1[5, 6] (TSPS No. 1) has been deployed rapidly throughout the Bell System's nationwide telecommunications network to the point where there are more than 150 systems in the continental United States. Over 95 percent of Bell System customers and a large number of customers of other telephone companies are served by TSPS No. 1. The continuing growth of operator services traffic, plus the continuing addition of new features, have steadily reduced the remaining real-time capacity and available memory in the TSPS No. 1 Processor. Additionally, plans to utilize TSPS as an Action Point in the emerging stored program controlled network dictated the need for new processor peripherals, such as a mass memory disk that cannot be provided by the current TSPS No. 1 processor.

To meet these needs, a major evolution from TSPS No. 1 to a new system, called TSPS No. 1B, was required. The goals of this evolution were to provide substantially increased call capacity, memory, and processor peripheral capability.

The size of the investment in existing TSPS No. 1 software programs and peripheral hardware is very large. To preserve as much of this investment as possible, the existing software and peripheral hardware were retained in the transition to TSPS No. 1B. At the same time, the capability of adding new features using high-level language software was provided.

The 3B20D Processor and DMERT provide the basis for meeting the goals of TSPS No. 1B. The 3B20D Processor replaces the existing TSPS No. 1 processor. The existing TSPS No. 1 periphery is retained and, through emulation, the existing TSPS software is preserved. The

retention of existing TSPS No. 1 periphery is achieved through the use of a Peripheral System Interface (PSI) circuit designed to interface the TSPS buses with the 3B20D Processor. The software preservation is accomplished by defining (through microcode) one of the multiple 3B20D processor instruction sets to be that of the existing processor, thus emulating that processor and allowing existing TSPS software to be transported to the 3B20D Processor almost intact. The ability exists to switch between the emulated instruction set and the 3B20D native instruction set within a single process and with a single instruction. Thus, new software can be added into either environment, as appropriate. Both emulated and native-mode software are run under the DMERT operating system, allowing operating system services to be available to both forms of software. The emulated existing TSPS No. 1 assembly language code is structured as a single kernel process executing under DMERT. As explained in the No. 5 ESS discussion, this permits efficient control of real-time resources where required. Other administrative and diagnostic functions that are not real-time sensitive are implemented as user-level processes. The DMERT operating system provides processor maintenance and administration. Thus, both the multilevel operating system and multiple-resident instruction set capabilities of 3B20D/DMERT are essential elements of the TSPS No. 1B design.

The first TSPS No. 1B started serving telephone customers in Fresno, California in November, 1981. As of September 1982, about 20 TSPS No. 1B's are in service and, by year-end, it is planned to have a total of about 35 in service. Performance data from all TSPS No. 1B sites indicate that all design objectives have been achieved.

## IV. NETWORK CONTROL POINT

The Network Control Point[7] (NCP) is a new Bell System development that adds an on-line real-time data base to the Common Channel Interoffice Signaling (CCIS) network. Using the alternate-routing and direct-signaling features of the CCIS network, the NCP provides a high-reliability, distributed data-base capability. The first applications of the NCP are to support 800 Service and Expanded 800 Service and to provide billing validation for the Automated Calling Card Service.

Since the NCP was designed to use the 3B20D Processor and the DMERT operating system, the software is written entirely in the C programming language. Those portions of the NCP software that are real-time critical are implemented at the kernel process level, while those portions with less stringent real-time requirements are implemented at user level. Thus, all features of the DMERT operating system are used to create an efficient, homogeneous system at a significant savings in project development cost.

The primary purpose of the NCP is to provide reliable, fast access to a data base. Hence, special emphasis is placed on the access to the moving head disks and to the communication via data links to the data base administration centers. A cache algorithm is implemented that allows the most frequently used entries to be queried without access to the disk. Multiple copies of the data base, beyond the two copies maintained by DMERT, are kept to protect against loss of the data base. Special spoolers are provided to aid in the communication between the NCP and the data base administration centers. The flexibility of DMERT permits these application-specific features to be easily implemented and integrated into the system.

Hardware for the NCP is composed of standard 3B20D Processor units except for one special peripheral controller board used to link the NCP to the signal transfer point of the CCIS network. The DMERT input/output driver module is modified to handle this board, and a diagnostic module is integrated with the standard DMERT diagnostics. The remainder of the NCP hardware consists of duplex processor systems equipped with 10 to 16 megabytes of memory each, four input/output processors, five moving head disk units, six to ten BX.25 data links, and a magnetic tape unit, making the typical NCP one of the larger 3B20D Processor systems in operation.

The NCP was the first user of the 3B20D Processor and the DMERT operating system to go into commercial service. The straightforward architecture of the NCP, both hardware and software, and the system test capability at the NCP development laboratory allowed sufficient operational confidence to be established so that four systems were placed into service on September 3, 1981. By November 1, 1981, 14 systems were in service, handling 15 million queries per business day across the United States from all calls prefixed by 800.

## V. ATTACHED PROCESSOR SYSTEM FOR NO. 4 ESS AND NO. 1A ESS

The 1A Processor is the central processing unit of both the No. 4 Electronic Switching System (No. 4 ESS) and the No. 1A Electronic Switching System (No. 1A ESS).[8, 9] The No. 4 ESS is designed to handle toll and tandem switching functions while the No. 1A ESS is designed to handle local, tandem, and toll switching functions; both systems have been in service for more than six years. Currently there are about 90 No. 4 ESS and 800 No. 1A ESS systems in service. The growth of telephone traffic and customer demand for new services on these systems established a need to increase processing capacity, implement new service features, and expand the memory spectrum. A new system architecture involving the attachment of an additional processor to the 1A Processor configuration was designed to meet these needs. The 3B20D Processor with the DMERT operating system

was chosen as the appropriate processor because of its low cost, reliability, processor and memory resource capacity, and the availability of a high-level language in the software development environment.

The No. 4 ESS and No. 1A ESS are designed to use the powerful 1A Processor for central control and memory. There are three types of memory: program store containing the fixed set of software instructions for operational functions; call store containing translation data describing office configuration and parameters; and file store on disk containing backup copies for both program and call store as well as less frequently exercised programs, such as diagnostic routines. The 1A Processor file store was the earliest resource to exhaust under the pressure of increased traffic load and new feature development. The Attached Processor System (APS), with the 3B20D Processor and DMERT operating system, has been designed as a replacement for the 1A Processor file store. In addition, APS also provides a mechanism to deload the 1A Processor of operational and administrative tasks which potentially limit its real-time performance at high traffic loads. It should be emphasized that the 1A and 3B20D Processor support different programming languages and new development software can be written in the more appropriate language.

The APS system includes hardware and software to connect the 3B20D Processor to the 1A Processor. The hardware includes an Attached Processor Interface (API) unit to interconnect the Direct Memory Access (DMA) channels of the two processors. The software includes an API driver consisting of 3B20D Processor program modules and corresponding 1A Processor program modules. Together these modules administer and maintain a 10 megabit/second fully duplicated interprocessor communication link. The 3B20D Processor modules are designed at the kernel process level to meet the stringent real-time requirements for processor intercommunication. To maintain integrity with the existing No. 4 ESS and No. 1A ESS software environment, the 1A Processor disk administration mechanisms were provided in the APS. The 1A Processor was also provided with full access to the 3B20D/DMERT File Manager and the entire 3B20D file system addressing spectrum.

The first No. 4 ESS with APS was scheduled for commercial service in 1982. The first No. 1A ESS with APS is scheduled for service in 1983. To take full advantage of the 3B20D Processor capabilities for future development, several new No. 4 ESS and No. 1A ESS features are planned to be implemented using DMERT and the high-level C language.

## VI. SUMMARY

The 3B20D/DMERT system has achieved its objective of providing a cost and real-time effective base for a wide variety of telecommuni-

cations systems. The key concepts of multiple levels of functional support, emulation microcode support, and versatile input/output interfaces as combined in the 3B20D/DMERT system provide an adaptable base that can be tailored to many differing needs. In addition to the four applications described above, the 3B20D/DMERT system is the basis of a number of other telecommunication system designs currently under way in the Bell System. This widespread use of 3B20D/DMERT marks it as a processor/operating system combination of significance in the Bell System.

## REFERENCES


1. W. N. Toy and L. E. Gallaher, "Overview and Architecture of the 3B20D Processor", B.S.T.J., this issue.
2. A. H. Budlong and F. W. Wendland, "3B20D Input/Output Systems," B.S.T.J., this issue.
3. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman, "DMERT Operating System," B.S.T.J., this issue.
4. W. B. Smith and F. T. Andrews, "No. 5 ESS Overview," Int. Switching Symp., 1981, Montreal, Canada.
5. N. X. DeLessio and R. E. Staehler, "The 3B Duplex Processor System and Its Application to TSPS, Part 2," Int. Switching Symp., 1981, Montreal, Canada.
6. R. E. Staehler, "Traffic Service Position System No. 1B—Overview and Objectives," B.S.T.J., *62*, No. 3 (March 1983).
7. "Common Channel Interoffice Signaling," B.S.T.J., *57*, No. 2 (February 1978).
8. "The 1A Processor," B.S.T.J., *56*, No. 2 (February 1977), pp. 119–327.
9. "No. 4 ESS: Prologue," B.S.T.J., *60*, No. 6, Part 2 (July–August 1981), pp. 1041–1048.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# Overview and Architecture of the 3B20D Processor

By W. N. TOY and L. E. GALLAHER

(Manuscript received March 8, 1982)

*The 3B20D Processor is designed to meet a broad range of telecommunication applications. New features such as memory management are incorporated into its design to support a modern operating system. Hardware supports are provided to efficiently execute a high-level language. A major design objective is to meet the stringent reliability requirements of electronic switching systems. The proven technique of self-checking duplex operation forms the basic architecture structure of the 3B20D Processor.*

## I. INTRODUCTION

The 3B20D Processor is the first member of a family of processors designed for a broad range of Bell System applications. Its development is a natural outgrowth of the continued need for high availability and real-time control of Electronic Switching Systems (ESSs),[1-3] including existing as well as new telecommunication applications. With the rapid growth of integrated-circuit technology, the processor architecture is evolving to include as many features as possible to significantly reduce software development and maintenance costs.

The 3B20D Processor architecture takes advantage of the LSI technology to expand its functionality and yet maintain a high reliability standard. Some of the design goals are to:

(*i*) Achieve highest performance that is consistent with system cost, e.g., provide hardware facilities such as data cache, high-speed interrupt stack, address-translation cache, and microprogramming for critical functions that require too much time in software.

(*ii*) Reduce software complexity, e.g., provide a modern real-time

operating system to manage system resources, thereby creating a more useful and more reliable programming environment for the user.

(*iii*) Reduce programming effort, e.g., provide both an efficient high-level language, such as the C language,[4] and a comprehensive set of software development tools.

(*iv*) Provide a high level of reliability and fault tolerance, e.g., built-in error-detection and correction codes, recovery features, and fault diagnostics.

(*v*) Provide features for system integrity and security, e.g., memory management protection and privileged instructions.

These goals are considered from the viewpoints of both hardware and software architecture in order to realize the most cost effective system for a wide spectrum of applications. Much of the development effort has been directed to achieve these goals.

## II. GENERAL DESCRIPTION

High availability is one of the major objectives in the design of the 3B20D Processor. The successful deployment and field operation of many ESS systems have demonstrated the simplicity and robustness of duplex configuration in meeting the ESS reliability requirements.[5] Hence, duplex configuration forms the basic structure for both the hardware and software architecture. Experience gained in the design and field operation of the No. 3A Processor provided valuable input for the 3B20D Processor design.[6]

The 3B20D Processor employs concurrent self-checking design. Extensive checking hardware is incorporated as an integral part of the processor. Faults occurring during normal operation are quickly discovered by detection hardware. This eliminates the need both to run the standby processor in the synchronous mode of operation and to run the fault-recognition program to identify the defective unit when a mismatch occurs. Self-checking implementation simplifies the maintenance program. Reconfiguration into a working system is immediate, without extensive diagnostic programs to determine which subsystem unit contains the fault. Furthermore, the self-checking design will permit more straightforward expansion from simplex to duplex, or multiple processor arrangements.

### *2.1 Duplex configuration*[7]

Figure 1 shows the general block diagram of the 3B20D Processor. The Central Control (CC), the memory, and the I/O disk system are duplicated and grouped as a switchable entity although each CC can access each disk system. The quantity of equipment within the switchable block is small enough to meet the reliability requirement; therefore, the complexity of a recovery program to manage additional

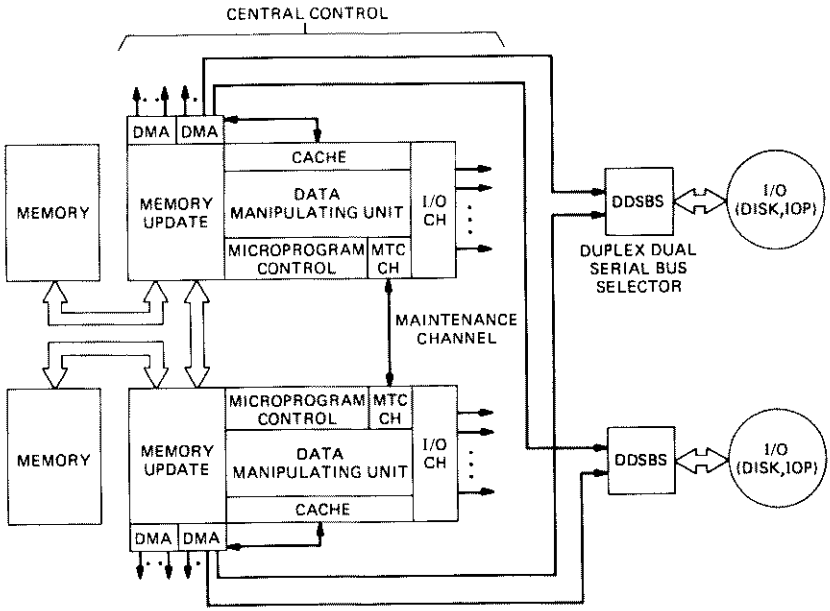



Fig. 1—General block diagram of the 3B20 Processor.

working states is avoided. Although each CC has direct access to both disk systems, this capability is used mainly to provide a valid data source for memory reload under trouble conditions. The processors are not run in the synchronous and match mode of operation as is done in early systems.[1-3] However, both stores (on-line and standby) are kept up-to-date by the memory update hardware concurrent with instruction execution. This is achieved by having the on-line memory-update circuit write into both memories simultaneously when memory data are written by the CC. Under trouble conditions, when the control is switched to the standby processor, its memory will contain up-to-date information without performing a complete transfer from one processor to another. The Direct Memory Access (DMA) circuits interface directly with the memory update circuit in order to have access to both memories. A DMA write also updates the standby memory. Communication between the DMA and the peripheral devices is accomplished by using a high-speed dual-serial channel (DSCH). The duplex dual-serial bus selector (DDSBS) allows both processors to access a single I/O device. For maintenance purposes, the duplex 3B central controls are interconnected via the Maintenance (MTC) channel. This high-speed serial path provides diagnostic access at the microcode level. It has the capability of transmitting a stream of microinstructions to exercise the processor from the on-line processor or from an external unit for diagnostic purposes.

### 2.2 Peripheral devices[8,9]

A broad range of general-purpose peripheral devices is provided for the 3B20D Processor system. High reliability and maintainability continues to be the design philosophy of the 3B20D peripheral system.

The critical components are duplicated and the software ensures that valid data sources are maintained. The DDSBSs permit controlled switching of a working standby device for a faulty on-line device when duplication of peripheral devices is needed. Some major peripheral devices developed for the 3B20D system are:

(*i*) Moving head disk system—The disk system provides a reliable and flexible mass storage medium for program and data. A backup copy of system programs and critical parameters can be reloaded quickly in the event of a duplex main-store failure. The disk system comprises the Disk File Controller (DFC) and the Moving Head Disk drive (MHD). The DFC interprets and executes commands from the processor to cause information transfer from and to the MHD. Each DFC occupies 1 of 128 channel slots and supports up to 16 MHD drives which are available in 80 and 300 megabyte sizes.

(*ii*) I/O processor (IOP)—The IOP provides the control for a wide range of data-link facilities and is the most flexible of the family of devices. An IOP supports up to 16 Peripheral Controllers (PCs) with each being a microprocessor-based controller programmed to handle a specific terminal or device. For example, one type of PC is the Line Controller (LC); each LC can support up to four independent lines (data links or terminals).

(*iii*) Magnetic tape system—The tape drive accepts the industry-standard (IBM compatible) 9-track tapes at a density of 1600 bits per inch. The tape controller is derived from the basic PC and occupies one of the 16 slots of the IOP.

(*iv*) Scanner/signal distributor (SC/SD)—This device is useful in monitoring and controlling power, equipment states, environment conditions, etc. The SC/SD circuit board provides 48 scan points and 32 signal-distributer points. It occupies one of the PC slots of the IOP. When an IOP is fully equipped with 16 SC/SD circuit packs, a total of 768 scan points and 512 signal-distributor points are provided.

## III. SOFTWARE SUPPORT FEATURES[10]

The high cost of designing, updating, and maintaining software dominates the cost of producing computer systems. Considerable attention has been focused on providing various types of support, i.e., high-level language, operating system, and software test, in the development of 3B20D Processor. The combined software and hardware effort has yielded an integrated and cost-effective system.

### 3.1 High-level language support

The most common approach to increasing software productivity and reducing software maintenance cost is the extensive use of a high-level language suitable for the application. The design of the 3B20D Processor instruction set was based on the fact that C language programs would dominate the programming environment. C is a general-purpose programming language featuring economy of expression, modern control flow and data structures, and a rich set of operators. Many studies were directed to measure and determine the characteristics of a large, diverse sample of C programs. Based on the result of these studies, the instruction set was optimized to be space and time efficient for compiled C programs. Some features provided for the instruction set are concerned with:

(*i*) Symmetrical resources
(*ii*) Addressing modes
(*iii*) Address manipulation
(*iv*) Flexible data structure
(*v*) Stack instructions
(*vi*) Procedural instructions.

From the compiler's viewpoint, the most important attribute of a processor instruction set is regularity. It is the key feature needed to abstract the various processor resources for uniform treatment by the complier. The 3B20D instruction set includes a wide range of address modes—i.e., indexing, direct, indirect—covering various data structures. The treatment of the addressing modes, applied identically to all data types (bytes, half words, full words, and instructions) without exceptions, makes it possible to compile compact and efficient codes.

The subroutine is one of the most important concepts in software. The principal idea in modular, structured programming is the partitioning of large programs into many small, understandable procedures or subroutines. Efficient instructions have been provided to handle subroutine entry and exit in addition to stack manipulation.

### 3.2 Operating system support

Higher productivity in application programming is made possible by the high-level, simplified facilities provided by the operating system. The Duplex Multienvironment Real Time (DMERT) operating system for the 3B20D Processor is a general manager of processor, memory, input/output, and software processes. The functional description of DMERT will be described in more detail in the next section. This section describes the hardware that has been incorporated into the design to reduce the overhead of the operating system.

As previously indicated, a high-speed address translation cache

memory called the Address Translation Buffer (ATB) is provided to reduce the overhead associated with the address translation function.

Context switching is necessary upon interrupt. A memory stack is provided to facilitate the saving and restoring of the hardware context. In the 3B20D Processor, a local high-speed 8K-byte RAM is provided for this function. The addressing of the stack is part of the kernel virtual address space and has been assigned a fixed segment number and pages 0 to 3. Whenever the kernel virtual address falls into this range, the store operation is directed to the high-speed RAM; otherwise, the virtual address is translated by the ATB and pointed to the main memory. The combination of fixed mapping by special circuit and dynamic address translation by ATB allows the high-speed stack to be extended into the main memory when the use of the high-speed RAM is exceeded.

### *3.3 Software test support*

The software test facility is an option provided at both the microprogram level and the macroprogram level. The Microlevel Test Set (MLTS) is attached to the microcontrol section of the central control. It has the capability of direct access to a support computer system for assembling and loading the writable microstore through the MLTS. The primary purpose of the MLTS used in the development of the 3B20D Processor is for initial debugging and troubleshooting the processor core hardware and, subsequently, the microprogram sequences. Features incorporated into the MLTS allow stepping, freezing, examining, and tracing the execution of a microprogram sequence.

The Utility Circuit (UC), on the other hand, provides a similar set of facilities, except at the macroprogram level for software debugging and troubleshooting. The UC and its associated software form an extensive Test Utility System (TUS) for software testing. A small number of matchers are incorporated into the UC for the tracing and monitoring of a variety of system conditions so that a programmer can observe and follow the execution of a program sequence. Much of the program debugging can take place in real time concurrent with program execution. The UC thus directly extracts and records information such as transfer trace from the internal data buses, thereby "capturing" the history of the machine while it is running at normal speed.

## IV. DMERT OPERATING SYSTEM[11]

DMERT is a general-purpose operating system. It is structured as cooperating processes that provide different levels of virtual machines. Protection is built into the structure, preventing these virtual machines from destructively interfering with each other. For processes to cooperate in accomplishing their task, DMERT provides a rich set of

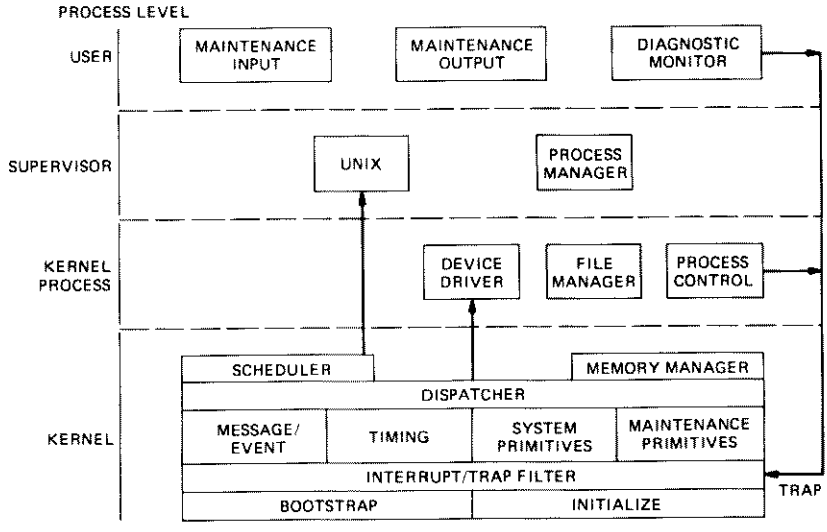



Fig. 2—The DMERT structure.

interprocess communication and synchronization mechanisms, including messages, events, process ports, interprocess traps, and shared memory. By means of these communication primitives, system services can be provided easily to a requesting process.

### *4.1 Multilayered virtual machines*

DMERT provides four levels of virtual machines. They are the kernel, kernel process, supervisor, and user (Fig. 2). Successive levels put additional restrictions on access rights of system resources. This helps free programmers from the details of the physical machine. The higher level may take advantage of services provided by the lower levels.

(*i*) Kernel—The kernel provides the most primitive virtual machine. Programs at this low level are closest to the hardware. They directly control the system hardware and do not have access to other system functions. The kernel services are primitive, yet are efficient in their execution. The user cannot introduce code at this level.

(*ii*) Kernel process—Kernel processes are also strongly hardware related and are structured to provide time-critical processing in a real-time environment. DMERT uses this level for the file manager and device drivers. In addition, there are several special processes that provide scheduling, memory management, and other services. Users may add a kernel process that communicates efficiently with other kernel processes making efficient use of their services.

(*iii*) Supervisor—This third level is comprised of programs that are

generally hardware independent. These processes can use all the services provided by the kernel and its processes. The process manager is implemented in this level. So is a *UNIX** operating system supervisor, which provides time-shared usage of processor hardware through services provided by a scheduler. In general, supervisor segments are not locked in memory and can be moved out onto the disk. Consequently, supervisor processes take a much longer time to dispatch than do kernel processes.

(*iv*) User—All applications programs for which time is not a critical factor are written at the user level. The user process is linked directly with a *UNIX* operating system supervisor in which DMERT treats the supervisor/user pair as a single process. These user programs only see the software environment and services of the supervisor and are well protected from other user processes. Since user-level code must interface its own supervisor to use the lower-level primitives, additional overhead is added for user-level processes.

#### *4.2 Multiple environment support*

An application may add code at the level of the kernel process, the supervisor process, or the user. The multilevel structure makes DMERT a flexible system for real-time use. In general, the higher the level, the more services that are available to an application; the lower the level, the more efficient is the program execution. This level structure of virtual machines permits DMERT to manage real-time applications, while at the same time providing the flexibility of a time-sharing system for background tasks. Figure 3 shows how telephone switching software can be allocated to these different levels. By means of this hierarchical execution-level structure, application programs can customize their control and distribution of real time.

The portion of real time that is not utilized by the kernel or kernel processes is time shared among supervisor and user processes. Deferrable jobs such as traffic reports, recent changes, and diagnostics, as shown in Fig. 3, are implemented at the highest user level. The DMERT architecture thus simultaneously supports both a real-time and a time-sharing environment to fully utilize physical resources in the most efficient manner.

## V. MAINTENANCE FEATURES

Increased support in this area is most appropriate to facilitate a more reliable and more maintainable system, thereby reducing the maintenance cost. For real-time applications, as in the Electronic Switching Systems (ESS), high availability and uninterrupted opera-

* Trademark of Bell Laboratories.

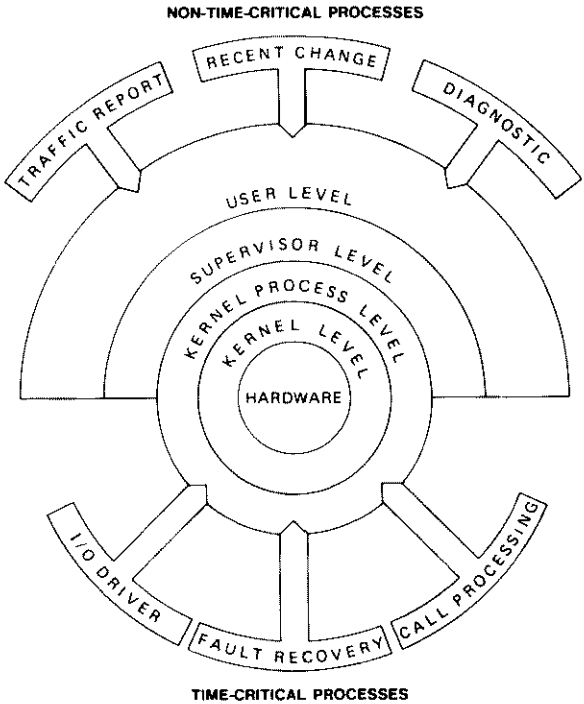



Fig. 3—Example of DMERT multienvironment structure.

tion is essential. This requires the system to function correctly even when a fault is present and maintenance is being performed. The 3B20D is designed to meet the ESS standard so that the expected amount of accumulated processor downtime does not exceed an average of 2 minutes per system per year.[5] Software and hardware are designed to function jointly to ensure that reliability objectives are met. Software features include such components as fault recovery programs, audits, and diagnostics.[12] Hardware features include redundant processors, error-detection circuits, maintenance-access controls, and diagnostic microcode. These components contribute to the effective maintenance design.

## VI. SUMMARY

The 3B20D Processor is a high-availability system capable of supporting a broad spectrum of applications. A comprehensive set of software tools and facilities is provided to improve programming productivity and also to reduce the cost of software development and maintenance. The hardware architecture is designed to efficiently support high-level languages, particularly the C language.

A wide range of general-purpose peripherals have been provided with the 3B20D Processor. Some of these are the moving head-disk system, magnetic tape system, high-speed printer, scanner and signal distributor, and data terminals.

DMERT is the duplex multienvironment operating system for the 3B20D Processor. It was designed concurrently with the hardware to meet the high-availability demands of real-time switching and telecommunication systems. DMERT provides a set of procedures that enables users to efficiently share the 3B20D Processor and the physical resources such as processor time, storage space, and peripheral devices. The multiple environment permits time-critical code to coexist with time-shared software as background tasks.

An important provision in the 3B20D Processor is a complete set of maintenance facilities, from error detection through fault recovery and diagnostics. Approximately 30 percent of the internal central control logic is devoted to self-checking. This allows concurrent error detection and immediate recovery. The combined hardware and software features give an integrated package of maintenance facilities to meet the high ESS reliability requirements.

## VII. ACKNOWLEDGMENTS

The design of the 3B20D Processor was accomplished through the combined efforts of many designers in Bell Laboratories and Western Electric. The authors wish to acknowledge the leadership of R. J. Watters who helped direct the early development of the 3B20D Processor system, and the design efforts of C. W. Hoffner, D. E. Gearhart, C. M. Narayanan, D. J. Matter, and D. J. McGraw.

## REFERENCES

1. "No. 1 ESS," B.S.T.J., *43*, No. 5 (September 1964).
2. "No. 2 ESS," B.S.T.J., *48*, No. 8 (October 1969).
3. "No. 1A Processor," B.S.T.J., *56*, No. 2 (February 1977).
4. B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, Englewood Cliffs, NJ: Prentice-Hall, 1978.
5. W. N. Toy, "Fault Tolerant Design of ESS Processors," *Proc. IEEE*, *66* (October 1978), pp. 1126–45.
6. T. F. Storey, "Design of a Microprogram Control for a Processor in an Electronic-Switching System," B.S.T.J., *55*, (February 1976), pp. 183–232.
7. M. W. Rolund, J. T. Beckett, and D. A. Harms, "3B20D Central Processing Unit," B.S.T.J., this issue.
8. A. H. Budlong and F. W. Wendland, "3B20D Input/Output System," B.S.T.J., this issue.
9. R. E. Haglund and L. D. Peterson, "3B20D File Memory System," B.S.T.J., this issue.
10. B. R. Rowland and R. J. Welsch, "Software Development System," B.S.T.J., this issue.
11. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman, "DMERT Operating System," B.S.T.J., this issue.
12. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "Fault Detection and Recovery," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# 3B20D Central Processing Unit

By M. W. ROLUND, J. T. BECKETT, and D. A. HARMS

(Manuscript received March 18, 1982)

*The 3B20D Processor has been developed to meet the need for very reliable, real-time control of a variety of Bell System applications. To achieve its high-reliability goals, most of the major subsystems within the processor are duplicated, including the Central Processing Unit (CPU). The CPU uses a 32-bit architecture throughout, including the memory and input/output buses. Extensive self-checking logic is employed. The 3B20D CPU is microprogrammed to select dynamically up to four instruction sets. The microstore uses a 64-bit word with up to 16K words of high-speed bipolar PROM or RAM available. This rich emulation capability makes the 3B20D Processor ideal for emulating existing instruction sets and porting existing software. Peripheral units are connected to the CPU via the Direct Memory Access unit (DMA). The DMA controllers provide direct memory transfers between the main store and peripheral devices, reducing the load placed on the central control to process input/output requests.*

## I. CENTRAL CONTROL STRUCTURE

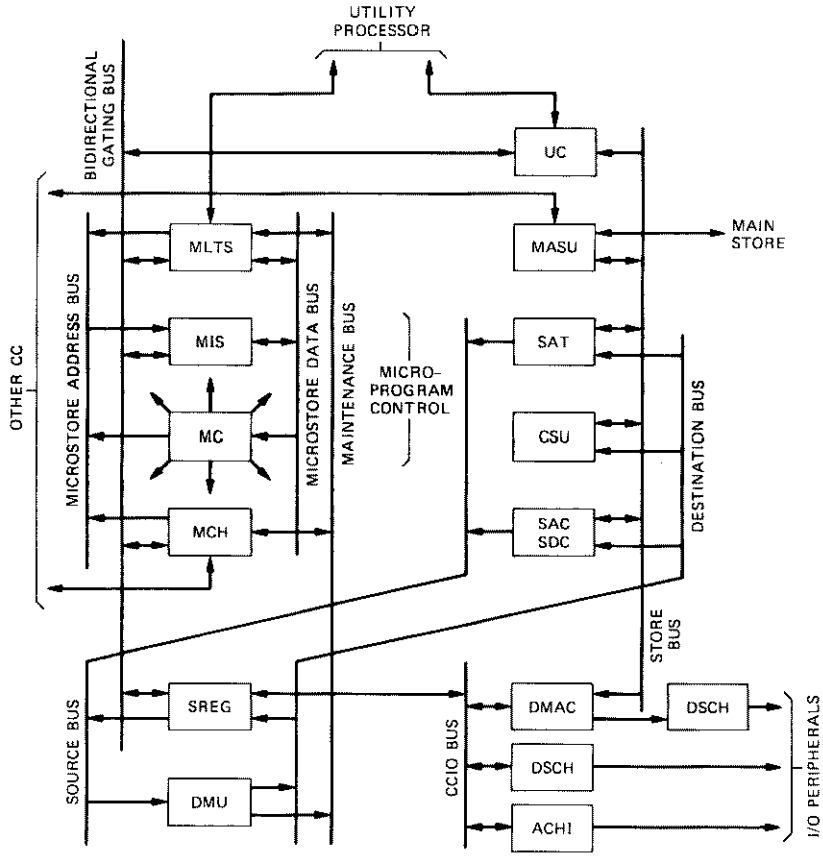
The 3B20D Processor performs all the functions normally associated with a CPU and several unique functions as well. The unique functions include the features necessary for reliable duplex operation, efficient emulation of other machines, and communication with a flexible and intelligent periphery. The processor employs an extensive microprogram capability to minimize the amount of hardwired decoding and to simplify the writing of microcode. There is substantial flexibility in the choice of instruction formats that can be efficiently interpreted.

The CPU is a 32-bit machine with a 24-bit address. Most of the data paths in the Central Control (CC) are 32 bits wide (plus 4 parity-check

bits). A fully self-checking philosophy is applied within each CPU without dependence upon matching between the duplex CPU's. The CC design is based on the register type of architecture, with multiple buses to allow concurrent data transfers. Separate I/O and store buses provide the capability of concurrent store and I/O operations.

### *1.1 Major subsystems*

A detailed block diagram of the CC structure is shown in Fig. 1. The




ACHI – APPLICATION CHANNEL INTERFACE
CC – CENTRAL CONTROL
CCIO – CENTRAL CONTROL INPUT/OUTPUT
CSU – CACHE STORE UNIT
DMAC – DIRECT MEMORY ACCESS CONTROLLER
DMU – DATA MANIPULATION UNIT
DSCH – DUAL SERIAL CHANNEL
MASU – MAIN STORE UPDATE
MC – MICROCONTROL UNIT
MCH – MAINTENANCE CHANNEL
MIS – MICROINSTRUCTION STORE
MLTS – MICROLEVEL TEST SET
SAC – STORE ADDRESS CONTROLLER
SAT – STORE ADDRESS TRANSLATOR
SDC – STORE DATA CONTROLLER
SREG – SPECIAL REGISTERS
UC – UTILITY CIRCUIT

Fig. 1—The 3B20D Processor central control.

major subsystems and their associated functions are described in the following sections.

### 1.1.1 Microcontrol

This subsystem provides nearly all of the complex control and sequencing operations required to implement the instruction set. The microcode can support up to three different emulations in addition to its native instruction set. Other complicated sequencing functions are also stored in the microstore (MIS). The Microcontrol (MC) unit sequences the microstore and interprets each of its words to generate the needed control signals specified by the microinstruction. To optimize the execution of microinstructions, execution time depends upon the complexity of the microinstruction. Each is allocated sufficient time to implement the microinstruction in multiples of 50 nanoseconds. These times are 150, 200, 250, and 300 nanoseconds. The wide 64-bit word allows a sufficient number of independent fields within the microinstruction to perform a number of simultaneous operations. Some common and high-runner instructions are implemented with a single microinstruction.

### 1.1.2 Data manipulation unit

The arithmetic and logic operations are carried out in the Rotate Mask Unit (RMU) and the Arithmetic/Logic Unit (ALU). These two units are connected in series to comprise the Data Manipulation Unit (DMU) shown in Fig. 2. The RMU can rotate or shift any number of bit positions from 0 to 31 through a two-stage barrel-shift network. In addition, a selection of AND/OR operations can be performed on bits, nibbles (4 bits), bytes, half words, full words, and miscellaneous predefined patterns. The RMU outputs feed directly into the ALU. Any bit fields within a word can easily be manipulated and processed by the DMU, greatly enhancing the power of the microcode. The ALU is implemented using 2901 bipolar 4-bit processor elements (see Fig. 3).[1] Eight such chips provide two key elements: the 2-port (A and B), 16-word (RAM) and the high speed ALU. Data in any of the 16 words addressed by the 4-bit A-address-field input can be used as an operand to the ALU. Likewise, data in any of the 16 words defined by the 4-bit B-address-field input can be simultaneously read and used as a second operand to the ALU. The result can be directed to the RAM word specified by the B-field. To take advantage of the above feature, the internal 16-word RAM is dedicated as general registers. This enables the arithmetic and logical operation involving general registers and/or the output of the RMU to be performed at the optimum speed.

The logic blocks of Fig. 2 reveal the way the RMU-ALU is self-checked. The rotate unit is checked by word parity, which it preserves,

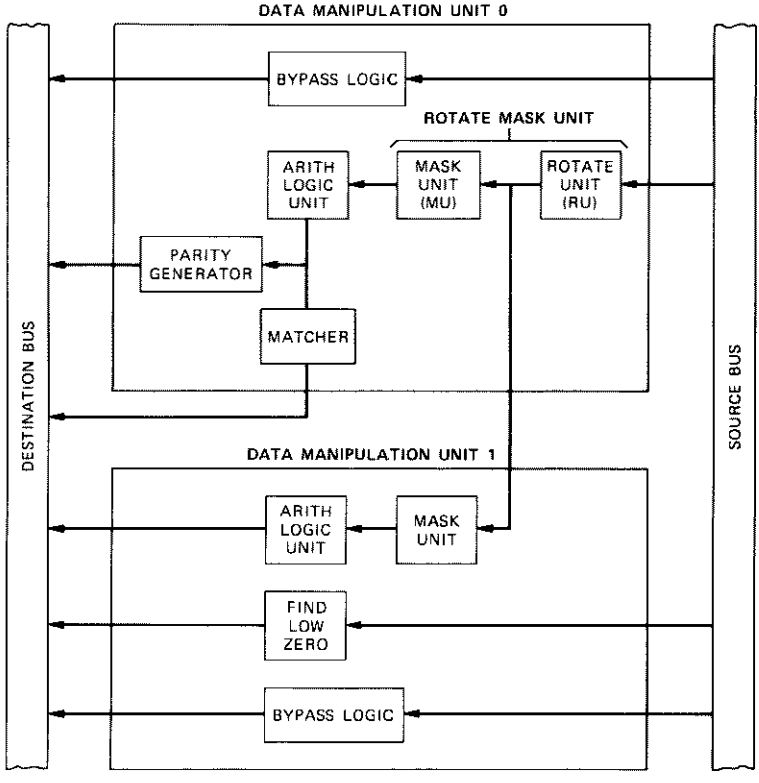



Fig. 2—Block diagram of the data manipulation unit.

and the mask unit is checked by duplication. The ALU is also duplicated. The data is taken from one ALU and parity is generated from the other. The data from one ALU is also matched with the duplicate. The underlying self-checking strategy, illustrated here and used throughout the CPU, is to use parity checking in those cases where parity is preserved and duplication of logic elements where parity is not preserved.

### *1.1.3 Special registers*

The 16 general registers reside inside the DMU and are available to the programmer. A number of Special Registers (SREG) associated with the operation of the CC are external to the DMU. Most of them are not explicitly specified by the instruction. They are characterized by their special dedicated functions with additional inputs from sources other than the internal data bus. Their outputs are used to control and direct the operation of the processor. Some of the SREG's are: the Error Register, Program Status Word, the Hardware Status Register,

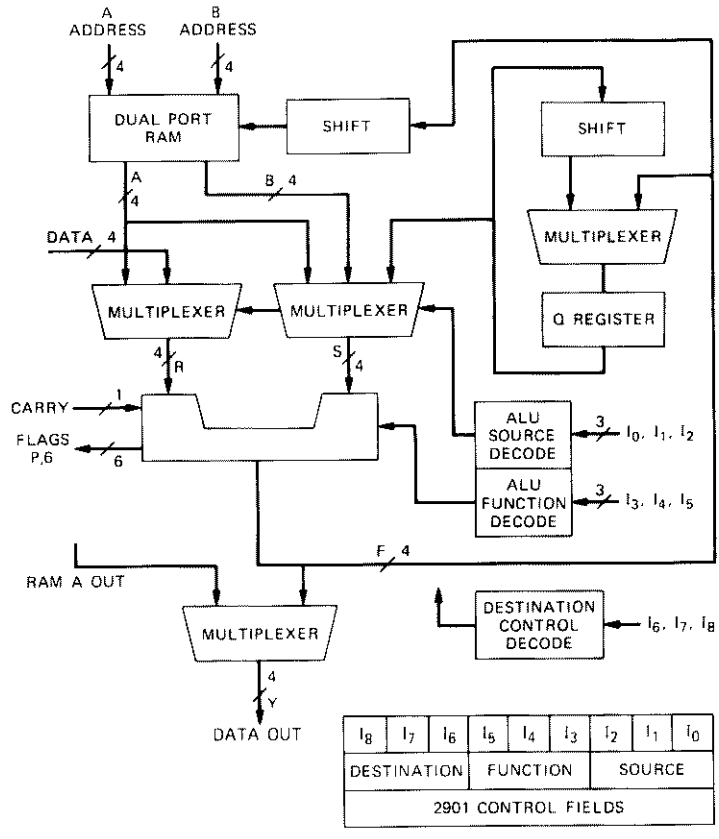



Fig. 3—The 2901 internal architecture.

the System Status Register, the Interrupt Register, timers, etc. In addition, a 32-word RAM is provided within the SREG block and is essentially available only at the microcode level. It is used for scratch-pad space and preassigned registers to facilitate and enhance the power of microprogram sequences. The registers supporting memory management[2] are an example of such preassigned registers.

### *1.1.4 Store interface control*

The store interface circuit controls the transfer of data or instructions from the memory system[2] to the CC. Several SREGs are associated with the store interface. Associated with the Store Address Control (SAC) are the Program Address (PA), the Store Address Register (SAR), and the Store Control Register (SCR). Associated with the Store Data Control (SDC) are the Store Data Register (SDR), the Store Instruction Register (SIR), and the Instruction Buffer (IB).

The SAC and SDC together make up the store interface that handles the memory addressing, the updating of the program counter, and the fetching and prefetching of instructions. The circuit ensures a continuous flow of instructions to be interpreted by the microcontrol unit.

#### *1.1.5 Store address translation*

Memory mapping is required in the implementation of a virtual addressed multiprogramming system. The Store Address Translation (SAT) facility containing the Address Translation Buffer (ATB) is the mechanism that provides a mapping between a program-specific virtual address and its corresponding physical address. Address translation hardware[2] is provided to facilitate memory management in a more efficient manner. The store address space is divided into 128 segments, each having up to 64 pages, with a page containing 2K bytes. Both the virtual and the physical address spaces are 24 bits. The complete set of virtual-to-physical address translation tables are stored in the main memory. A significant amount of time would be required by the CC in the repetitive task of dynamic address translation in using the main store tables. This translation time is reduced substantially by storing the likely-to-be-used physical address translations in a high-speed cache-like Address Translation Buffer (ATB).

#### *1.1.6 Main store update*

The Main Store Update (MASU) unit[2] provides a multiport interface to the memories as both DMA and CC attempt to use the memory. The update circuit arbitrates asynchronous requests from the on-line CC, both DMAs, and the off-line CC. The cross coupling between the memory update units permits the on-line CPU to access either memory or both memories for concurrent write operations.

#### *1.1.7 Input/output interface*

The communication path between the CC and the I/O channels is through the CCIO bus. It is a local, high-speed, direct-coupled, parallel bus. Direct memory access between the main store and peripheral units is provided by a Direct Memory Access Controller (DMAC) that communicates with intelligent peripheral units via Dual Serial Channels (DSCH). I/O channels including user-specific interfaces can be connected directly to the CC via the CCIO bus. Two such standard interfaces are the DSCH, a high-speed multiport serial interface, and the Application Channel Interface (ACHI), a high-throughput, parallel bus, peripheral communication path.

#### *1.1.8 Cache*

The cache[2] is an optional circuit equipped to improve the overall

system performance by reducing the effective memory access time. The cache is a fourway set associative memory containing a total of 8K bytes.

#### *1.1.9 Maintenance channel*

This circuit provides diagnostic access to the CC at the microinstruction level. It also is used to control basic fault recovery and system sanity functions in the off-line processor.

### *1.2 Major buses*

The processor structure diagrammed in Fig. 1 allows three key transfers to proceed simultaneously. The first is a microcontrol path through the Microinstruction Store (MIS). The second is a computation path through the Data Manipulation Unit (DMU). The third is a memory path through the cache. The CPU is pipelined at both the microstore and store levels: microinstruction execution is overlapped with microstore read and instruction execution is overlapped with fetch. In addition, separate transfer paths are provided for maintenance and input/output.

Microaddresses pass from the Microcontrol Unit (MC) to the MIS over the 16-bit Microstore Address (MSA) Bus. Microinstructions pass from the MIS to the MC over the 64-bit Microstore Data (MSD) Bus. Control signals derived from the microinstruction fan out to the subsystems that connect to the 32-bit Source (SRC) bus and 32-bit Destination (DST) bus.

The SRC and DST buses are the primary gating paths for computations. Data and addresses pass from the DMU to the SDC and SAC, respectively, over the DST bus. The SREG are also accessed over the SRC and DST buses. The Store Bus includes both a 32-bit data field and a 24-bit address field. The store can also be accessed by the DMAC over the same bus structure.

The Maintenance Channel (MCH) controls the processor using the MSA and MSD buses. The 32-bit Bidirectional gating bus (BGB) and Maintenance (MTC) bus are used by the MCH to observe the CPU. The BGB is also used to load the writeable microstore.

The CCIO bus is a bus linking the CC with the I/O channels. Operations over the bus are programmed I/O instructions. Control of the DMA circuits is exercised over this bus but the data transfers via DMA are routed directly onto the main store bus.

## II. I/O FACILITIES

The I/O facilities[3,4] are designed to meet a wide range of applications with different needs and capabilities. A modular and flexible I/O communication structure is provided by means of dedicated point-to-

point channels. The loose coupling of the processor to the peripherals allows considerable freedom to grow and expand the system with minimum physical constraints. The I/O architecture is shown in Fig. 4. Two channels have been designed for the CCIO bus, the dual serial channel and the application channel interface.

### 2.1 Input/output channels

#### 2.1.1 Dual serial channel

The Dual Serial Channel (DSCH) (see Fig. 5) is a modular circuit that provides high-speed serial links between the CC or DMA and the peripheral devices. Direct memory access is only supported on devices connected to DSCHs on a DMAC. Transfers for devices directly connected to the CCIO bus are completely controlled by CC software using specific I/O instructions.

The DSCH supports word (36 bits) or block (16 words) transfers at either of two rates. For peripheral devices located within 100 feet of the DSCH a 10-MHz clock rate is used. For devices up to 250 feet away a 5-MHz clock rate must be used.

The DSCH provides a private serial point-to-point data path to each of the 16 intelligent devices it supports. Each link uses a five twisted-pair cable as the transmission media. Two of the pairs are used for bidirectional data transmission, two pairs for unidirectional clocks and the fifth pair for the peripheral device to set flags in the DSCH. Data is transmitted using RS422 compatible signaling. The other end of the cable connects to a Duplex Dual-Serial Bus Selector (DDSBS) that converts the signals from the DSCH into a parallel format. Each DDSBS can interface to two DSCHs allowing the peripheral device to be connected to both CPU's of the duplex processor.

#### 2.1.2 Application channel interface

The Application Channel Interface (ACHI) (see Fig. 6) provides a differential dc parallel bus and control signals for high-throughput programmed I/O. The circuit can only reside on the CCIO bus and has no autonomous sequencer. The ACHI's relies on command sequences from the CC to carry out peripheral operations.

The ACHI's CCIO bus interface is similar to that of the DSCH. The same data bus, addressing, control, and response signaling is used. The parallel bus generated on the peripheral side of the ACHI consists of two unidirectional 36-bit data buses and 11 control and response leads. One data bus is for input to the ACHI and one is for output with concurrent data transfers allowed.

### 2.2 Direct memory access controller

The Direct Memory Access Controller (DMAC) (see Fig. 7) provides

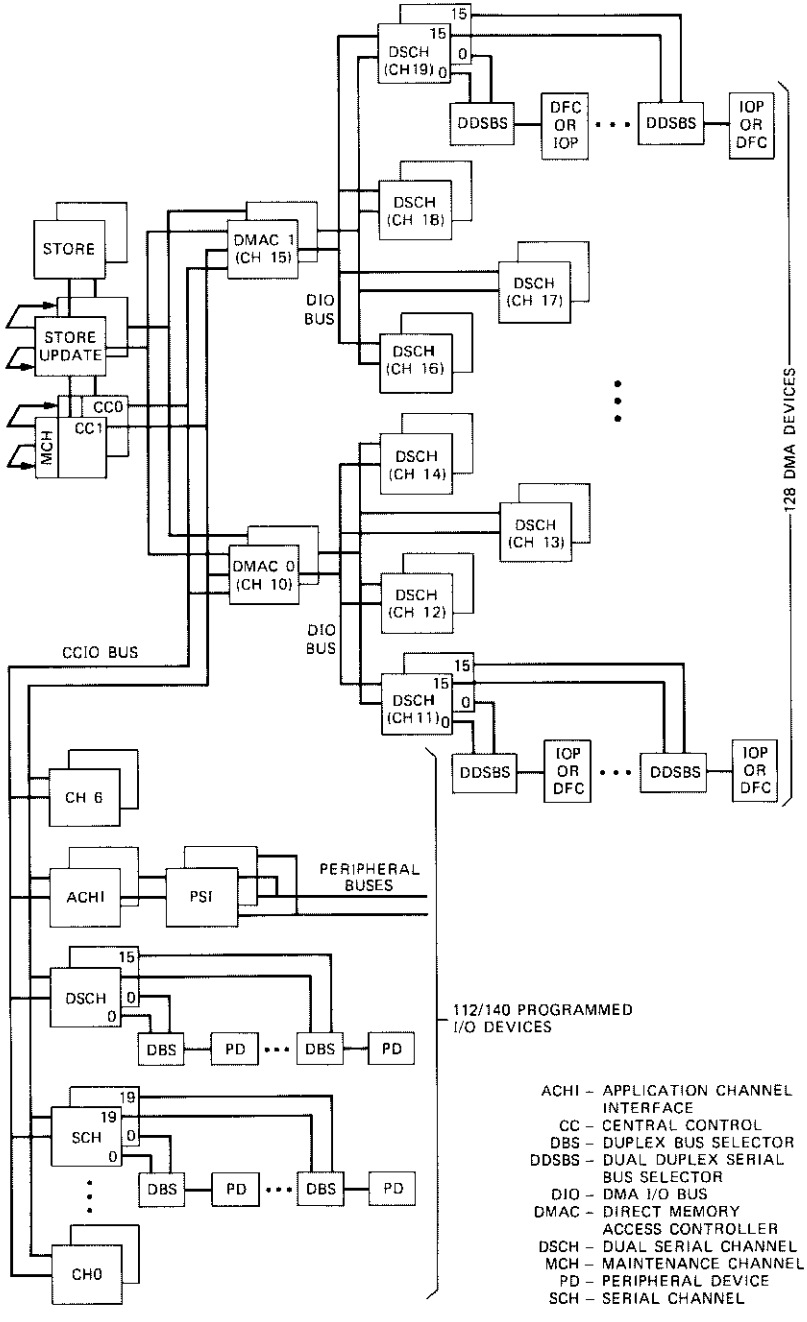



Fig. 4—The 3B20D Processor input/output architecture.

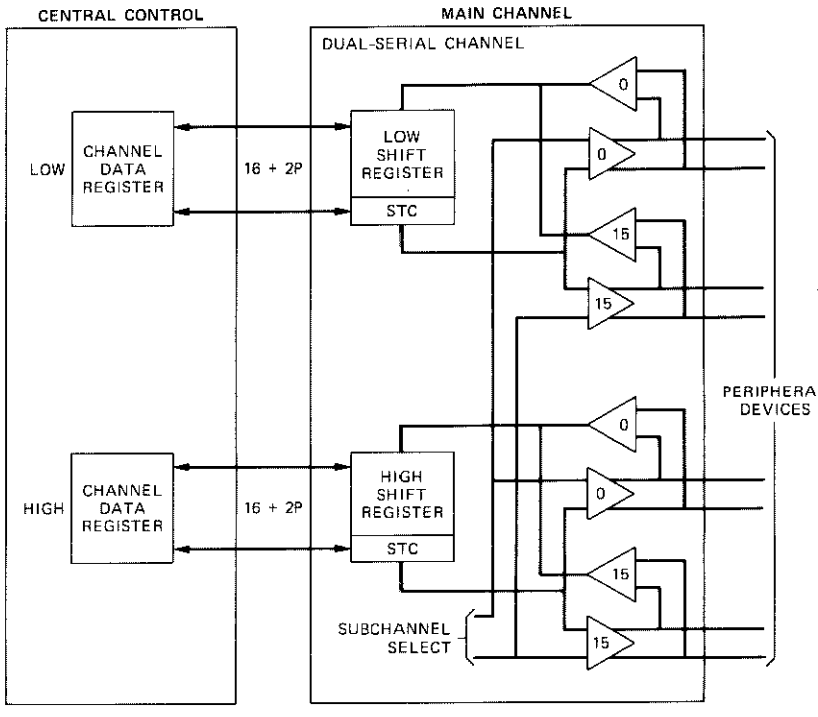



Fig. 5—Dual-serial channel.

the facility for moving data between I/O devices and main memory without having the processor involved in transferring the data. The DMA circuit consists of a DMAC and one to four DSCHs. A DMA can accommodate up to 64 devices, all of which may be active concurrently. The DMAC supports virtual addressing, word or block (16 word) transfers, device-initiated transfers, and multiple jobs for a given device. The latter function allocates a unique segment in memory to each job for a device. This prevents one job from mutilating another job's memory.

A DMA transfer is a two-step operation. Half the time is spent with the channel and peripheral device passing data and half with the channel-DMAC-main store passing data. While the former operation is in progress, the DMAC can be loading or unloading another channel from main store. The DMAC has a hardware priority circuit that gives channel 0 priority over channel 1, which has a higher priority than channel 2, etc. The devices on a given channel are prioritized in the same manner, i.e., device 0 having the highest priority and device 15 the lowest. The channel does not permit interleaving devices on less than a block or word boundary. Once a transfer of a block or word is

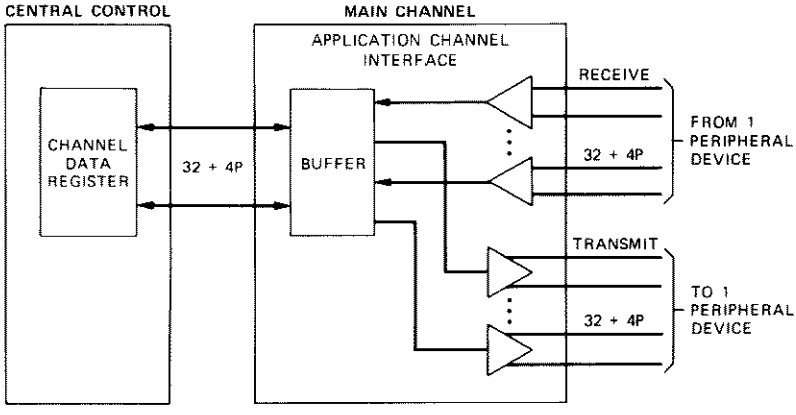



Fig. 6—Application channel interface.

initiated to a device, that transfer must be completed before another device on that channel can be serviced.

A DMA transfer is initiated by the device passing the DMAC the starting virtual address and, optionally, a transfer count. The DMAC translates the address into a physical address as described below. The transfer count is used as a check verifying that the device and DMAC are in agreement as to the number of transfers to take place.

The address-translation process used by the DMAC is the same as that used by the CC with both using the translation-page tables that are stored in main memory. Each page has protection bits defining DMA read/write access capability. The maximum size transfers that can be accomplished with a single address setup is one segment or 64 2K-byte pages. As part of the initialization process for the DMAC, the processor passes a unique page table pointer to the DMA for each of its active devices. The DMAC uses the page table pointer and the virtual address to obtain the desired physical page pointer. As the DMA transfer crosses a page boundary, the DMAC automatically accesses the page table to obtain the next physical page pointer.

After setting up the DMAC the device initiates the transfer by sending a transfer request. The DMAC will ask for the data from the device or send the data to the device in a word (32 bits) or block (16 words) mode. The device then sends another transfer request and the handshaking continues until the entire job is completed.

## III. MAINTENANCE FEATURES

The 3B20D places great emphasis on maintenance features. A complete, fully developed package of such features includes self-checking, fault recovery, and diagnostic capabilities.

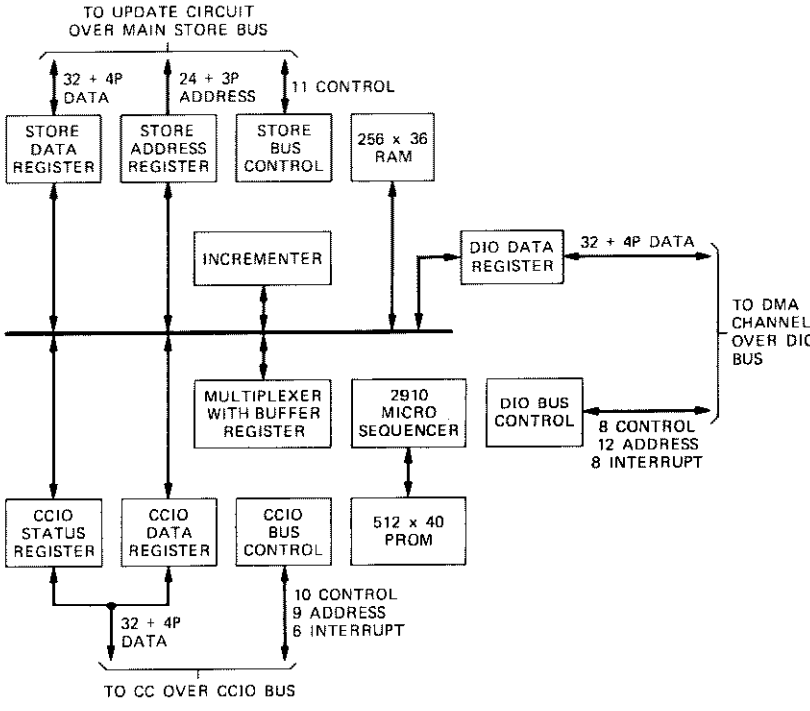



Fig. 7—Direct memory access controller.

### *3.1 Self-checking hardware*

Self-checking is implemented in the 3B20D design for concurrent error detection. The maintenance philosophy is to provide a sufficient amount of hardware to enable detection of nearly all service-affecting single hardware faults.[5] To minimize the potential sources of errors in the main memory,[2] single-bit Hamming correction and double-bit error detection are employed. Most software faults such as memory-protection violations, illegal instructions, and out-of-range addresses also are detected. Some of the fault-detection techniques used in the 3B20D Processor are:

(*i*) Parity per byte on data paths throughout the internal CC, memory buses, and peripherals

(*ii*) Single-bit Hamming correction and double-bit error detection on the main store data

(*iii*) Duplicated arithmetic and logic unit and other control logic

(*iv*) "Watchdog" sanity timers for software faults.

Faults detected by the processor check hardware are collected together into a single error register. The action taken for a particular fault depends on its impact on the system. These actions range from

microinterrupts handled in firmware to a stop-and-switch response where control is transferred to the standby half of the duplex processor.

### 3.2 Hardware fault-recovery features

Fault detection is the first and most important step in realizing a highly reliable system. Almost of equal importance is the rapid recovery by the system. Recovery is achieved by a coordinated combination of hardware, microcode, and software actions. As soon as an error is detected, immediate action takes place to reconfigure the system into an error-free working system. The recovery process involves two steps: reconfiguration and initialization. To facilitate this rapid recovery from system faults, the 3B20D Processor provides the following.

#### 3.2.1 A memory update unit

This unit couples the on-line memory to the off-line memory. The off-line processor's memory is updated on each memory write operation, whether CC or DMA originated, to provide continuous agreement with the on-line memory.

#### 3.2.2 A maintenance channel

This unit directs the off-line processor to initialize and recover when the on-line processor has detected critical error conditions.

#### 3.2.3 Initialization microcode

This nondestructive microcode makes critical recovery decisions when error conditions are detected. This microcode is particularly important if total software sanity is lost.

### 3.3 Diagnostic features

Hardware has been incorporated into the design of this system to permit a systematic approach for identifying failures via software. This diagnostic software[6] depends heavily on the maintenance channel and its associated circuitry. The primary function of the MCH is the diagnosis of one CPU by the other. The MCH is an autonomous portion of the processor which, under control of the other CPU, can provide information about the state of the machine. The MCH thus exercises the machine at its most basic level by direct access to the microprogram control. The MCH characteristics are like those of the DSCH, and the access protocols are compatible with each other.

## IV. MICROCODE FEATURES

The large microstore address space in the 3B20D Processor provides a relatively inexpensive means of specifying complex instructions and

special system functions. These can be added or modified with little difficulty compared to hardwired functions.

### 4.1 Types of features

Microcode can reside in either PROM or in writeable control store. The following sequences have been implemented in the 3B microcode:

(*i*) Microboot—This sequence of code initializes the processor and loads both the Writable Control Store (WCS) and the first-level bootstrap program from disk. If it is unable to load from the primary disk, it automatically tries to load from the backup disk.

(*ii*) Tapeboot—This sequence of microcode initializes the processor and then copies data from a standard nine-track magnetic tape to the disk. The microboot routine can then be used to initiate program execution.

(*iii*) Basic instruction set—This set of microcode implements the native instruction set and special instructions that are specific to the 3B20D Processor.

(*iv*) External interrupt routine—This sequence is invoked only upon the completion of an instruction with an interrupt pending. The microcode saves the state of the system, then transfers to an interrupt event handler routine.

(*v*) Error interrupt routine—This section of microcode is entered in case of a hardware or software error. The microcode saves the state of the machine and then transfers control to a routine that attempts to recover processing without switching to the other machine. This interrupt can be encountered between any microinstructions.

(*vi*) Memory management trap—This sequence of microcode is entered if the virtual address of the memory fetch cannot be translated directly to a physical address by the address-translation hardware. The microcode reads the segment and page tables of the active process, loads the hardware translation unit with the translation information and then reactivates the memory access.

(*vii*) Maintenance channel instructions—These instructions allow the processor to communicate with its duplex mate via the maintenance channel.

(*viii*) Diagnostic sequences—A section of the WCS is reserved for the diagnostic programs to load and then execute special microcode sequences.

(*ix*) Miscellaneous routines—There are several miscellaneous microcode routines that are used for functions such as to load the writable control store and to load the hardware matcher registers in the utility circuit.

(*x*) Emulation routines—Up to three additional sets of microcode can be loaded that allow the 3B20D to emulate other machines.

### 4.2 Microcode development

Several software and hardware tools were developed that allowed microcode to be generated and debugged with relative ease. Of principal importance are the MICA[7] bit-sliced assembler and the Micro-level Test Set (MLTS).[6] These tools allowed a laboratory utility computer to be used to assemble microcode, and to control the 3B20D via a link to the MLTS. The facility was used to load writable microstore, access main store, access registers and control sequencing at the microcode level, set breakpoints and provide trace capability.

During the development phase of the 3B20D processor, substantial advantages were realized by the fact that the system was microprogrammed. Major improvements in the instruction set architecture were accommodated by microcode changes. Features have been added to enhance error recovery, to permit transferring bootstrap programs from magnetic tape to disk, and to improve system performance.

## V. SUMMARY

The 3B20D CPU is a 32-bit machine with 24-bit addressing. Hardware features have been provided to support a modern general-purpose operating system, e.g., virtual-to-physical address translation. Other features include microprogram implementation, emulation capability, high-speed data cache, high-speed interrupt stack, self-checking circuits, extensive diagnostic access, and high-availability duplex operation.

The standard I/O communication between the CC and the peripherals is by means of a DMA controlling dual-serial channels, capable of transmitting an effective rate of 2M-bytes/s. The DMA can operate in a word-transfer mode or a block mode of 16 words per block. The loose coupling of the channels between the processor and the peripherals permits considerable freedom in expanding a system.

## VI. ACKNOWLEDGMENTS

Many individuals have contributed in a significant way to the organization, design, and implementation of the 3B20D central processing unit. The authors would like to particularly acknowledge the contributions of C. A. Borcher and C. W. Hoffner to both the development of the 3B20D Processor and to the content of this paper.

## REFERENCES

1. *The AM2900 Family Data Book*, Sunnyvale, California: Advanced Micro Devices, 1979.
2. I. K. Hetherington and P. Kusulas, "3B20D Memory Systems," B.S.T.J., this issue.
3. A. H. Budlong and F. W. Wendland, "3B20D Input/Output System," B.S.T.J., this issue.

4. R. E. Haglund and L. D. Peterson, "3B20D File Memory Systems," B.S.T.J., this issue.
5. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "Fault Detection and Recovery," B.S.T.J., this issue.
6. J. L. Quinn, R. L. Engram, and F. M. Goetz, "Diagnostic Tests and Control Software," B.S.T.J., this issue.
7. J. T. Beckett and S. W. Ng, "A General Purpose Microcode Assembler," Proc. IEEE/COMPSAC, 1978, pp. 84–89.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# 3B20D Processor Memory Systems

By I. K. HETHERINGTON and P. KUSULAS

(Manuscript received March 17, 1982)

*The memory system supplied with the 3B20D Processor provides a high-reliability, high-performance, main-frame memory for use by the 3B20D Central Control and Input/Output system. The memory system is designed using a collection of high-speed, static and dynamic memory devices and appropriate logic controllers. In addition to providing basic on-line storage for program text and data, the memory system provides hardware assistance for virtual-to-physical address translation, access protection, memory resource arbitration, and performance enhancement utilizing a high-speed cache memory. The technology used in implementing these functions includes state-of-the-art 64K dynamic random access memory devices and high-speed TTL-compatible gate-array integrated circuits.*

## I. INTRODUCTION

The memory system associated with the 3B20D Processor[1] includes a 16-megabyte memory, a high-speed cache memory, and hardware assistance for the virtual-to-physical address translation process, access protection, and memory resource arbitration functions.[2] The memory system utilizes high-speed, static and dynamic memory devices and appropriate logic controllers.

The block diagram shown in Fig. 1 highlights the major components and interconnections of the 3B20D Memory System. The diagram indicates the memory system related control, address, and data paths, including the interconnection to the fully duplicated system. Internal central control data paths associated with the Store Address Translator (SAT) are not shown.

As indicated, the 3B20D Memory System is comprised of a 16-

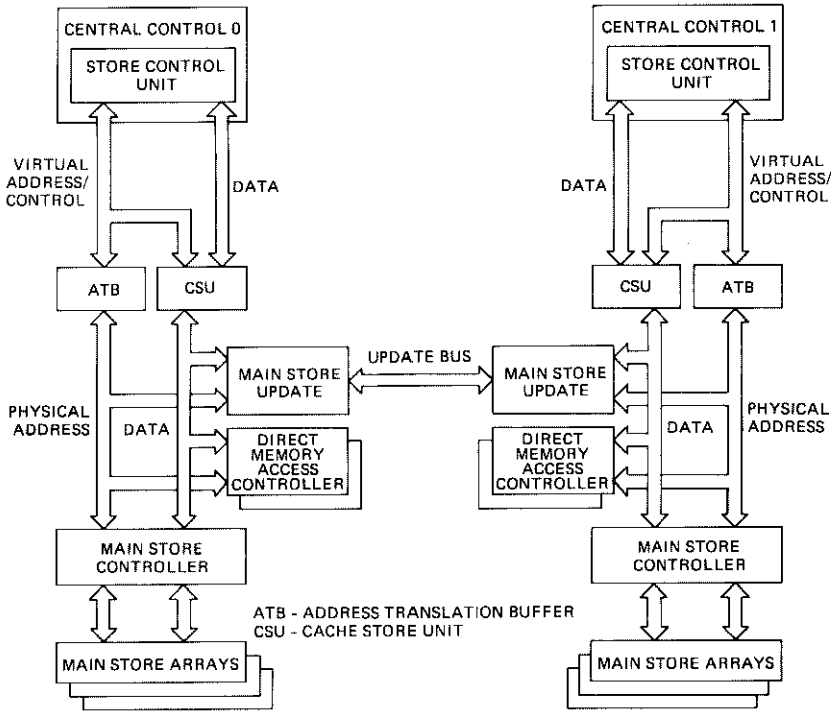



Fig. 1—Block diagram of the 3B20D Processor memory system.

megabyte main store, a Main Store Update (MASU) circuit providing interconnection control between the duplicated memory systems, a store address translator, and an optional Cache Store Unit (CSU). The subsystems that use the memory system are the central control and the Direct Memory Access Controller (DMAC). Memory operations are initiated by either of the duplicated central controls or DMACs. In the DMERT environment, one CC/DMAC combination has control of the system and initiates all memory operations. The MASU circuit performs write operations to both duplicated main stores. In this way, both main stores are kept up to date with currently executing programs, data, and I/O transactions. If a central control switch is needed, it can be accomplished quickly since the off-line central control can immediately use a fully updated main store.[3]

The 3B20D Processor supports memory management that allows programs to be written using virtual addresses without regard to where they actually reside in memory. An address translation hardware assistance circuit, the SAT, is provided in the 3B20D. The SAT provides high-speed access to the most recently used address-translation and access parameters. It serves as a cache memory for the

translation parameters that apply to the software processes executing in the central control. The same address-translation mechanism implemented by the SAT is used by the DMAC.[4,5] Thus, processes executing in the central control can share virtual address spaces with peripheral devices communicating through the DMAC. In addition, once this address-translation mechanism is established by the operating system, the central control or the peripheral devices may initiate memory operations independently.

The remainder of this paper discusses, in more detail, the major components of the memory system. The discussions cover the operational aspects of the design, self-checking, error reporting, error recovery, and diagnostic features[6] provided in the memory system design. Other topics addressed include performance, reliability, device technology, and environmental considerations.

## II. STORE ADDRESS TRANSLATOR

In any computer system, the main memory is an expensive resource and has to be efficiently managed for improved system performance. Because the 3B20D Processor supports a multiprogramming environment, the memory may be shared by several user programs, each having access to the full virtual memory spectrum. Hence, a mechanism for relocation and protection of a user's address space (text and data) is required. The 3B20D translates the virtual addresses used within the processor and I/O subsystem to a real or physical address used within the main store. To reduce the overhead associated with the address-translation and protection mechanism, the 3B20D uses an SAT that contains a high-speed address cache called the Address Translation Buffer (ATB).

### *2.1 Address space partitioning*

For the purpose of sharing, relocating, and protecting, the 24-bit address space is partitioned into "segments." A segment is a contiguous block of sequential virtual addresses the size of which may range from 1 byte to 128K bytes. To reduce memory breakage due to fragmentation, each segment is further partitioned into 2K-byte blocks called "pages." A segment has up to 64 pages and an address space can have up to 128 segments. The virtual address generated by the processor is divided into a 7-bit segment field, a 6-bit page field, and an 11-bit byte offset (displacement) as shown in Fig. 2.

### *2.2 Address translation process*

The SAT translates the 24-bit virtual address to a 24-bit physical address used by the CSU and the main store. This translation is conceptually a two-step table-lookup operation and is controlled by

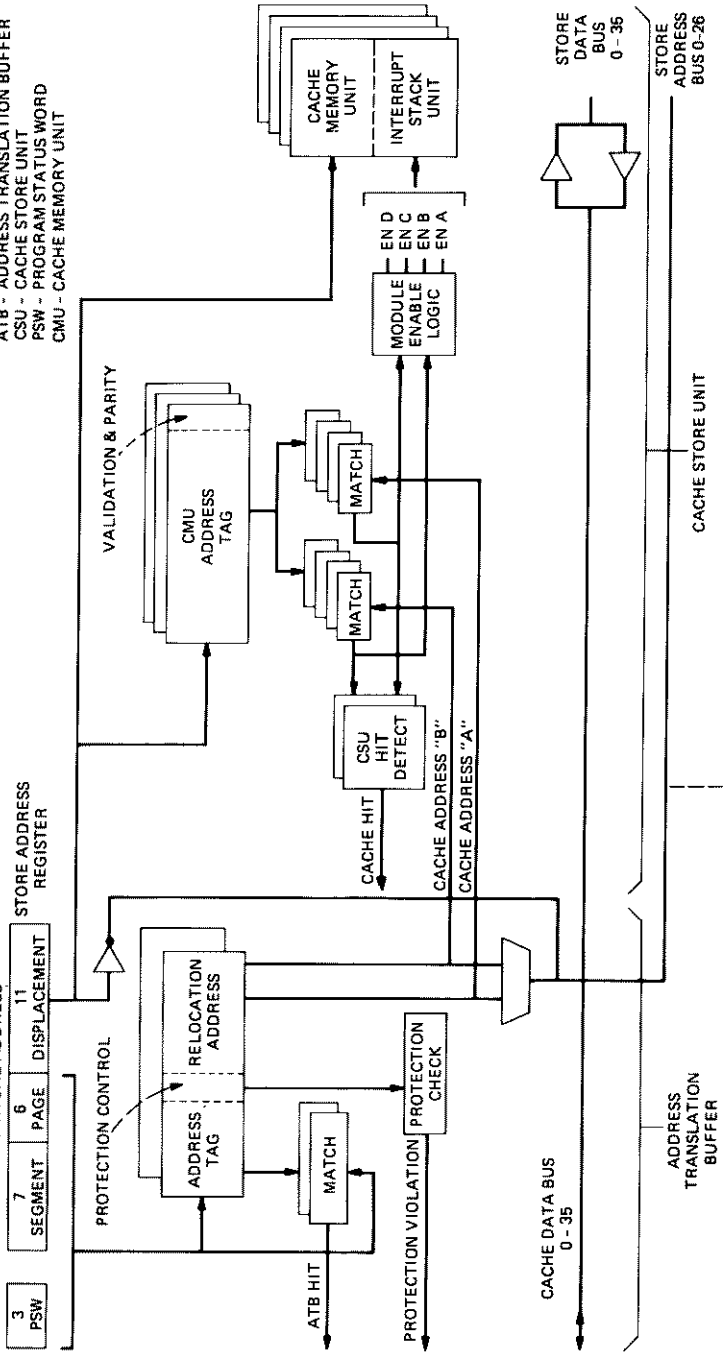



Fig. 2—Block diagram of the 3B20D Processor store address translator and cache store unit.

the Program Status Word (PSW), the segment base register (SBR), and two main store resident tables (the segment table and the page table). The resident operating system manages these registers and tables. Recently used translation information is kept in the high-speed ATB and is accessed concurrently with the CSU.

### 2.3 Segment base register

The currently executing process's[2] address space is defined by the SBR. The SBR is a 32-bit register that contains the physical address pointer to the beginning of the process's segment table and its length (which is equal to the number of segments allocated to the process). There are eight such SBRs in the processor to accommodate eight independent address spaces at any time. These eight address spaces are assigned by software, and in a DMERT environment, a minimum subset of four is allocated to the kernel, a kernel process, a supervisor process, and a user process. The fields in an SBR are allocated as follows:

(*i*) Segment table address (22 bits)—This field points to the beginning of the segment table in main store.

(*ii*) Unused (3 bits)—This field is presently not used.

(*iii*) Segment limit (7 bits)—This field designates the length of the segment table.

### 2.4 Segment table

The segment table (one exists for each process) contains a descriptor for each segment of the process. Each entry is 4 bytes long and resides on a full word boundary in the main store. It is partitioned into three fields as follows:

(*i*) Page table address (22 bits)—This field points to the beginning of the page table.

(*ii*) Page table length (6 bits)—The length of the page table is one more than the value in this field.

(*iii*) Protection bits (4 bits)—Three of these bits indicate whether the segment is readable, writable, or executable. One bit indicates the validity of the entry.

### 2.5 Page table

The page table (one exists for each segment) contains a descriptor for each page in the segment. Each entry is 4 bytes long and resides on a full word boundary in the main store. The entry has four fields that indicate:

(*i*) Relocation Address

(*ii*) DMA usage

(*iii*) Protection
(*iv*) Unused.

The process of virtual to physical address translation is described in the following sections.

#### *2.5.1 Segment table lookup*

The segment field of the virtual address is first compared against the segment limit field of the SBR to establish whether the addressed entry is within the table. If the segment limit is less than the segment field, a "segment length error" exception is recognized and control of the processor is transferred to handle the exception. If there is no such error, the segment field of the virtual address and the segment table address in the SBR are used to index into the segment table in the main store and access an entry.

The fetched segment table entry is checked for validity. If the entry is invalid, control is transferred to exception handling microcode. If the entry is valid, the page field of the virtual address is compared against the page table length field in the segment table entry. If the value of the page field is greater than the maximum number of pages in the segment, the control is transferred to a microcode routine to handle the error. In case of no such error, a page table lookup is initiated.

#### *2.5.2 Page table lookup*

The contents of the page-field of the virtual address and the page table address field of the fetched segment table entry are used to index into the page table to fetch a page table entry. The protection bits obtained from the segment table entry and the page table entry are ANDed, and a check is made for a possible protection violation. In case of such an error, an error-handling routine is initiated. If there is no protection violation, then the physical address is generated by concatenating the relocation address field of the fetched page table entry and the byte offset field of the virtual address.

### *2.6 Address translation buffer*

The Address Translation Buffer (ATB) is provided to reduce the overhead associated with the address-translation mechanism. The ATB is capable of holding 128-page table entries for eight different address spaces (for a total of 1024 entries) and is organized as a two-way set associative memory. When the processor initiates a store access, the ATB checks whether the corresponding physical address is available in its memory. If the physical address is available (called a "hit"), it is sent to the main store. But, if the physical address is not available (called a "miss"), an "ATB Miss" processing microroutine is

initiated which does the segment and page table lookups and loads the obtained physical address into the ATB. The access is then restarted. Subsequent access to the same page results in hits.

### *2.7 ATB control bits in PSW*

The PSW contains a field that provides control for the memory management. The ATB functions can be enabled or disabled. When disabled, address-translation and protection check functions are disabled, and the virtual addresses are directed to the CSU and the main store as physical addresses.

The PSW also controls process communication capabilities between virtual address spaces. The PSW designates the Primary Segment Base Register (PSBR) and the Secondary Segment Base Register (SSBR). Each of these registers defines a virtual address space. Generally, the PSBR is used to select one of the eight blocks of the ATB for address translation. Under PSW control, however, the SSBR can be used for read, write, or both read and write memory operations. All instruction fetches use the PSBR irrespective of the contents of the PSW. Special instructions are provided to manipulate the PSW. This hardware feature can be used to move data between two address spaces very efficiently.

### *2.8 ATB operation*

Figure 2 shows the block diagram of the ATB. When an access is initiated by the processor, the ATB is accessed using 3 bits of the PSW (selecting one of eight address spaces), low order 3 bits of the segment field, and low order 3 bits of the page field of the virtual address. The address tag fields of the ATB are matched against the corresponding bits of the virtual address. Simultaneously, the two relocation address fields are directed to the cache. A hit is generated in case of a successful match.

If a hit is detected, then a check is done to see whether the access is allowed on that page. The processor and the cache are informed in the case of any protection violation.

If a miss is detected, the cache ignores the relocation addresses. An "ATB Miss" microroutine is initiated by the processor and the new ATB entry is loaded into one set of the ATB using a defined replacement algorithm. The access is then restarted.

As mentioned before, the ATB is capable of handling translations for eight tasks at any instant. But, when a new task is allocated to a block of the ATB, the entries associated with the previous task have to be invalidated. A dedicated 8-bit counter is provided to do this invalidation with minimum microcode overhead.

The address translation mechanism provided by the SAT is com-

patible with that provded by the DMAC. In this way, processes executing on peripheral controllers can share virtual address spaces with processes executing on the central control.

### *2.9 Redundancy*

The ATB hardware is completely self-checked. Parity is checked over the Store Address Register (SAR), the ATB entries and the ATB related bits in the PSW. Parity bits are regenerated for the physical addresses to be checked at the cache and main store. The hit and protection check logics are duplicated and matched, and can be exercised under maintenance control. In case of any hardware faults in the circuit-pack, the access is aborted and control is transferred to a fault-handling routine. The ATB memory can be written and read over the source and destination buses of the processor.

## III. CACHE STORE UNIT

The Cache Store Unit (CSU) reduces the effective access and cycle time of store operations for the 3B20D Processor. Combining a relatively small, high-speed "cache" memory with a large main store results in a system with an average access time approaching that of the high-speed cache but with the low cost per bit and storage capacity of the Main Store.

The concept of a cache takes advantage of a general programming characteristic of locality of reference. Most references to memory tend to be highly localized or clustered into small groups at any given time, and regions tend to change relatively slowly during the course of program execution. Thus, a relatively small, high-speed CSU contains the most often used words from the main store and thereby reduces the average access time of the reference.

### *3.1 Organization*

The CSU is organized into two sections: an interrupt stack section and a cache section. Since the main store contains a much larger storage capcity than the CSU, a mapping function is required to compress the main store address range into the much smaller CSU. The compression is achieved by adding a tag which contains the physical page address to each word of cache data storage. To increase the probability that a main store word is in the CSU (a "hit"), the cache is organized as a four-way set-associative memory. Each one of the four cache modules contains 2K bytes of high-speed storage. The interrupt stack section also contains 8K bytes of memory.

Since the cache is four-way set-associative with the main store, there is a one for one correspondence to the page offset address but full associativity for the page portion of the address. Thus, word 0 from

any of the four cache modules may correspond to word 0 from any page in main store.

### 3.2 Operation

Since the cache section of the CSU functions as an associative memory, a match search in the cache is performed in parallel with the translation of the virtual-to-physical address by the ATB. This hardware parallelism is illustrated in Fig. 2. An access causes the low 11 bits of the virtual address to select a unique page offset on each of the four cache modules. The page portion of the virtual address (high 13 address bits) is translated by the two-way set-associative ATB. Each of the four cache tag modules is matched to the two translated physical page addresses. The ATB will indicate to the cache which of the two translated addresses is valid. If one of the four cache tag modules matches this translated page address, the CSU will generate a hit signal to the CPU and gate the associated word onto the cache data bus.

Functionally, the CSU interconnects the CPU and main store. The CSU connects to the CPU via the cache address bus, cache data bus, and control leads. Because the CSU interconnects the CPU and main store, DMA transfers to main store will not prevent the CSU from being accessed by the CPU. Thus, CPU contention for the main store is reduced since most references will result in a CSU hit.

During system initialization, all locations in the CSU are invalidated. When the CPU references the memory system and a cache miss results, the referenced word is automatically copied into the CSU. If a word needs to be copied from the main store and all four cache modules contain valid data, a random replacement algorithm selects the cache group in which a word will be replaced. Once the cache has been initialized by the system, the operation is transparent to the software with the exception of enhanced performance.

The CSU contains arbitration and sequencer control logic to automatically update its data and arbitrate between CPU and DMA write operations. When the DMA writes into the main store the CSU checks if the DMA reference is in the cache. If the word is in the cache, it is automatically invalidated by the CSU sequencer. While the CSU is checking for a DMA write hit, the CSU indicates a busy condition to the CPU. DMA reads from the main store do not result in any operation from the CSU.

### 3.3 Interrupt stack

The CSU also contains an 8K-byte high-speed memory which is used by the CPU to reduce the interrupt response time when the CPU is in the kernel operating mode. When the CSU is in the interrupt

stack mode, the CSU will generate 100-percent hits for both read and write operations.

### *3.4 Software support*

The CSU is designed to support the software organization of the DMERT operating system and the "C" programming language. Call and Return are frequent operations with the C language. The CSU has built-in hardware algorithms to function as a high-speed data stack. A stack write operation (CSAV) will force the data to be copied into both the cache and main store. A stack read operation (CRET) will cause the data to be read from the cache and then invalidated. This location in the cache is then available for use by new data. The data stack is part of the cache section and is totally separate from the interrupt stack.

### *3.5 Self-checking*

The self-checking philosophy of the CSU is to provide immediate detection of faults that cause errors and nonimmediate detection of faults that affect the performance of the CSU. The CSU has built-in self-checking hardware that monitors the operation of the CSU. Faults detected by the self-checking hardware include CSU sequencer errors, multiple writes into the CSU, multiple cache hits, accessing errors in the tag memories, and cache write errors.

A multiple hit is an example of a catastrophic fault since it would result in the contents of two or more cache words being ORed onto the cache data bus. This type of failure is prevented by duplication of the hit logic and by providing multiple hit detectors that monitor the hit logic and the module enable logic. The cache also generates and checks parity over the tag and address bits.

The CSU also provides diagnostic access to the "internals" of the circuit. When the CSU is configured in the maintenance mode, the normally associative memory tages are configured to function as conventional RAM memories. In addition, special access is provided to the counters, CSU sequencer, and various status bits.

## IV. MAIN STORE

### *4.1 Configuration*

Physically the main store can consist of one or two modules. Each module contains one Main Store Controller (MASC) and up to sixteen Main Store Arrays (MASA). The central control can directly address 16 megabytes of text or data.

The preproduction design of the main store was based on the initial use of 16K Dynamic Random Access Memory (DRAM) devices. The design was organized to allow evolution to higher-density devices. The

production design initially used a Western Electric 64K DRAM[7] that utilized two power supply voltages. The current design uses a newer Western Electric 64K DRAM that permits denser packaging. This design is implemented using a single circuit pack main store controller and a 1-megabyte MASA circuit pack. Thus, all 16 megabytes of addressable memory are contained in one module.

#### *4.2 MASU operational aspects*

The MASU circuit provides control of main store bus communication between duplex Control Units (CU), allowing the MAS in the standby CU to be kept up to date. The MASU also controls the use of the main store buses by the central control, DMA, and other CUs. The MASU in the active CU in a duplex configuration gives the highest priority to the central control followed by DMA 0, DMA 1, and operations from the other CU, respectively.

Communication with the Main Store is over the address bus, data bus and the command bus. The command leads indicate whether the operation to be performed is a write, read, clear, byte, halfword, or maintenance operation. Valid operations to the MAS include:

(*i*) Write full word
(*ii*) Write halfword/byte
(*iii*) Read word
(*iv*) Read and clear fullword
(*v*) Read and clear halfword/byte
(*vi*) Maintenance write (nonmemory operation)
(*vii*) Maintenance read (nonmemory operation).

The MASU communicates asynchronously with the MASC by the use of the Store Go signal (SGO) and Store Complete signal (SCM). Prior to issuing the SGO to the MASC, the MASU issues an address and data bus enable to the requesting unit with the highest priority. The MASU then issues the SGO to the MASC. The MASC upon receiving the SGO begins a timing sequence that selects the addressed MASA and issues a GO signal (GOI) to the MASA. The selected MASA then allows data to be read or written at the specified address depending upon which main store operation was decoded by the MASC.

The MASC performs various error checks during the timing sequence to ensure the integrity of the MASC and MASA. In the event an error does occur, the MASC sends an error signal to the requesting unit. Depending upon the operation being performed and the state of the error sources, the MASC at a specific point in the timing sequence issues an SCM to the MASU to indicate that the operation has been completed. The MASU can then grant the main store buses to the next requesting unit with the highest priority.

### 4.3 Self-checking

The 3B20D main store was designed with a significant amount of self-check circuitry. The partitioning of the circuitry on the circuit packs also was designed to improve fault detection. For example, the data bus transceivers on the memory array have only one bit from each parity field partitioned in each device. In this way a fault affecting either one of the bits or all the bits in the device will be detected by a simple parity check. Similarly the address, RAS, and CAS drivers for the memory devices on the MASA were partioned so that a fault associated with one of these drivers would affect at least two bits in each of the data-parity fields. By affecting more than one byte the probability of detecting the error by failing a byte-parity check increases.

To ensure bus integrity the MASC checks the address, data, and command bus parity between the MASC and the MASU. If the address parity check fails during a write cycle to the memory, circuitry in the MASC prevents the writing of data into the addressed memory location. Thus, invalid information will not be written into the memory. If the other parity checks fail, an error is signaled but an undesired memory operation may take place. The MASC internally monitors the timing sequencer and refresh address counters to ensure that they are functioning properly. Circuitry also checks the SGQ and SCM signals between the MASC and MASU to ensure that "handshaking" that takes place between the two units is functioning properly. The MASC by checking four selected responses from the MASA also can check communication to the MASA. The MASC can determine from the select responses whether an MASA was selected or not, and it can also detect if more than one MASA responded.

### 4.4 Error correction

The 3B20D main store performs error detection and correction. Error correction circuitry on the MASC and Error Correction Coding (ECC) of data in the MASA result in the correction of all single bit errors. This circuitry also flags all double and detectable multi-bit errors.

In the 3B20D central control, byte parity is maintained over each byte of the data word. The main store uses the existing byte-parity bits in a modified form of the Hamming code.[8] By adding four additional ECC bits (in addition to the byte-parity bits) the main store can then perform single-bit correction and double-bit failure detection. When presenting data to the system the MASC maintains byte parity by gating the byte parity bits to the MASA from the MASU, and vice versa. On the MASC five gate arrays are used to implement the error-checking and correcting circuitry. One gate array code is used for each

of the four bytes. The fifth gate array code is used for the additional four ECC bits.

The MASC issues an error signal when it detects a location in MAS in need of correction. The MASC presents the corrected data to the system but will not automatically rewrite correct data at the location. The actual rewriting of data is handled by the software error interrupt handler. In the event of a noncorrectable error, the error interrupt handler reads the data from the standby CU—if it is in a duplex configuration—and uses that data to rewrite the faulty location.

### *4.5 Diagnostics*

The 3B20D MASC provides maintenance access to a significant portion of the circuitry in the main store. This maintenance access is used by microcode to initialize the MASC and by diagnostics to test the functional operation of the MAS.

The MASC issues various maintenance commands within the MAS when the maintenance command, address, and SGO signal are presented to the MAS. The maintenance operation is decoded off the address and is only valid for one MAS cycle. By using an address decoded maintenance command, the state of certain data bits can be latched, providing the latched maintenance state until cleared. The MASC under diagnostic control uses the various maintenance commands and states to perform the following operations:

(*i*) Control address loop-around (address returns on data bus)
(*ii*) Control the refresh circuitry
(*iii*) Control the error correction circuitry
(*iv*) Control the error detection/reporting circuitry.

By using the maintenance capability, the diagnostics can verify the integrity of the bus structure, the MASA, and the MASC. The data integrity of each MASA is tested by performing a series of data pattern tests on each MASA equipped.

## V. ACKNOWLEDGMENTS

The authors would like to thank D. J. Matter, C. M. Narayanan, and D. M. Trampel for their assistance in the preparation of this manuscript and their contributions to the 3B20D memory system design. D. R. Draper and W. A. Read also have made significant contributions to the main store design.

## REFERENCES

1. M. W. Rolund, J. T. Beckett, and D. A. Harsm, "3B20D Central Processing Unit," B.S.T.J., this issue.
2. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman, "DMERT Operating System," B.S.T.J., this issue.

3. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "Fault Detection and Recovery," B.S.T.J., this issue.
4. A. H. Budlong and F. W. Wendland, "3B20D Input/Output System," B.S.T.J., this issue.
5. R. E. Haglund and L. D. Peterson, "3B20D File Memory Systems," B.S.T.J., this issue.
6. J. L. Quinn, R. L. Engram, and F. M. Goetz, "Diagnostic Tests and Control Software," B.S.T.J., this issue.
7. R. P. Cenker, D. G. Clemons, W. R. Huber, J. B. Petrizzi, F. J. Procyk, G. M. Trout, "Fault Tolerant 64K Dynamic Random Access Memory," IEEE Transactions of Electronic Devices, *ED26*, No. 6 (June 1979), pp. 853–60.
8. R. W. Hamming, *Coding and Information Theory*, Englewood Cliffs, N.J.: Prentice-Hall, 1980.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# 3B20D Packaging and Technology

By S. H. KULPA, J. M. BROWN, and A. W. FULTON

(Manuscript received March 18, 1982)

*The 3B20D Processor is built using a broad range of complex and high-performance integrated circuits. These integrated circuit devices are packaged and interconnected utilizing* Bellpac™ *packaging system technology. Used throughout the processor are high-density multilayer printed wiring boards with high pin-out connectors. The elements of the technology were combined using computer-aided design tools to assure optimized system thermal and electrical performance.*

## I. INTRODUCTION

The cost, performance, and schedule objectives of the 3B20D Processor and the overall complexity of the design required:

- A full spectrum of circuit integration
- Common integrated circuit specifications
- A broad range of semiconductor memory devices
- A standard hardware packaging technology
- Quick turnaround prototype circuit packs
- High interconnection capability at the circuit pack level
- A hierarchy computer aided design (CAD) process
- Comprehensive design audits in the CAD process.

The use of *Bellpac** packaging system technology provides a dense, high performance packaging system.[1] This includes a substantial increase in the number of devices allowed per circuit pack and the number of contacts per pack over previous technologies. The use of *Bellpac* packaging technology also allowed a low system cost since the

* *Bellpac* is a trademark of Western Electric.

hardware is Bell System standard and manufactured in high volume. Complex integrated circuits from a variety of vendors were employed throughout the 3B20D Processor. A wide variety of integrated circuits and comprehensive specifications were used to optimize processor performance, cost, and reliability. *Bellpac* packaging technology and the integrated circuits were coupled together through a sophisticated CAD system. The CAD system is comprised of design, analysis, simulation, and audit tools at device, circuit pack, unit, and frame levels.

## II. INTEGRATED CIRCUITS AND APPARATUS PERFORMANCE

### *2.1 General device requirements*

The integrated circuit device family used in the 3B20D Processor consists of approximately 280 codes of TTL (Transistor-Transistor Logic) compatible SSI (Small-Scale Integration), MSI (Medium-Scale Integration), and LSI (Large-Scale Integration) devices. These codes were selected to optimize the cost and performance of the wide range of processor functions included in the central control, memory, and peripherals. Device performance range varies greatly from the high-speed, speed-selected, bipolar Schottky TTL parts in the central control to relatively slow metal oxide semiconductor microprocessor peripherals in the I/O area. The scale of integration varies from SSI devices driving heavily loaded buses to VLSI (Very Large-Scale Integration) memory and microprocessor devices. Whenever possible, where system performance and/or cost was not jeopardized, the same parts were used in several different areas of the design to minimize the number of unique integrated circuits. Where the same device could be used for multiple functions, the device specification was written to cover the extreme needs of all applications.

To assure that all of these devices would interface with each other, common device specifications were established. The temperature range for the devices was specified as 0–95°C to accommodate central office equipment requirements and to keep cooling costs low. Supply voltage limits were established at 5 volts ± 10 percent to minimize power generation and distribution costs. The interface levels were established based on TTL standards (i.e. Vol $\leq$ 0.4 volt, Voh $\geq$ 3.4 volts) consistent with the temperature and supply standards described above. Other common standards included dual in-line package (DIP) dimensions and lead finish specifications to ensure manufacturability and reliability.

As the project progressed through its phases, the reliability requirements for devices varied. For the initial system models, high-reliability devices with good electrical performance were essential to minimize

interference with software/hardware integration. This was particularly true with devices on prototype wire-wrap hardware, which generate substantially more noise and crosstalk than the production multilayer designs. Since much of the prototype hardware was built before final device specifications could be negotiated with vendors, high-quality military grade parts were procured wherever possible and special screens were imposed where high-reliability parts could not otherwise be obtained. For production units, reliability requirements were established based on system availability objectives.[2] For most of the codes this worked out to a 100 FIT* objective per device. This objective was translated into specific packaging, burn-in, screens, and life-testing requirements in the device specifications.

Special devices required creation of a number of expanded specifications. For example, to simplify circuit pack testing or to provide a low-cost method for changing information contained in PROMs, reliable socketing was required which, in turn, required special gold lead finishes on the devices. A small number of other devices, for cost or availability reasons, were specified at reduced temperature and/or supply voltage ranges.

### *2.2 Small-scale and medium-scale integration devices*

There are about 160 codes of devices that can be classified as having small-scale integration (SSI) or medium-scale integration (MSI). These consist of two basic device groups: LSTTL and STTL. The LSTTL gates are typically characterized as having 10 ns, 2 mW parts, and the STTL gates as having 5 ns, 20 mW parts. Both families meet industry standards and are widely available. The identical functions are generally available in both families. Due to the much lower power of the LSTTL devices, they were preferred except where speed was critical. To maximize the use of the LSTTL devices a special output current drive requirement was specified that effectively doubled their capacitive drive performance under worst-case conditions. Since the STTL devices were used in the speed-critical paths, their performance had a major part in establishing system performance. On a subset of these devices, special speed screens at high-capacitive loads were specified. The increased cost of these screens on these few device codes was minimal at the system level, but the system performance gain was substantial.

In addition to the standard logic devices in the SSI and MSI category, there were a variety of special devices such as delay lines (10 codes) and oscillators (10 codes). These devices generally had STTL and LSTTL interfaces to the other devices.

---

* FIT is defined as one failure in $10^9$ operating hours.

### 2.3 Memory devices

The semiconductor memory device requirements for the various processor functions were broad. They included dynamic RAMs, static RAMs, PROMs, Electrically alterable PROM (EPROM), and First In First Out (FIFO) memories in a variety of functional organizations and speed ranges.

The dynamic RAMs used in the 3B20D for main storage are 64K-bit chips manufactured by Western Electric. They are organized 64K words by 1 bit and have an internal redundancy that was used to increase yield.

Static RAMs are used in a variety of applications including cache, microstore, memory management, and the address/data storage for the microprocessor-based circuits. The first three applications require high-speed memory, while the address/data storage requires high density at moderate speed (16K bits at 200 ns). There are 10 codes of static RAM in the 3B20D Processor.

The programmable memories (PROM, EPROM) are used in the microstore and as program memory in microprocessor-based designs. In addition, smaller PROMs have been used in sequencer designs. In many designs these parts are used interchangeably with static RAMs. Again organization, speed, and power requirements are varied and result in 17 codes of programmable memories. The FIFO memories used in the 3B20D Processor are moderate-speed devices, primarily used for data buffering.

### 2.4 Large-scale integration logic

Because of the low cost per gate available with LSI logic its use was chosen wherever available parts could meet performance and functional requirements. The parts that were chosen provide large general-purpose functional blocks, such as microprocessors and associated peripherals, protocol controllers, dynamic RAM controllers, etc. Because of the relatively long development time and high development cost for LSI parts, new custom LSI designs were not undertaken during the initial 3B20D development stages. Cost reduction and new feature designs completed after the initial design phase have tended to use LSI gate arrays. There are a total of 30 catalog LSI codes and 16 LSI gate array codes in the 3B20D.

### 2.5 Circuit packs

The smallest replaceable module of the 3B20D is the circuit pack. Each pack consists of a collection of integrated circuits, in dual in-line packages, interconnected on circuit boards using the *Bellpac* packaging system (see Fig. 1). Typical circuit boards use a multilayer printed wiring board structure, which consists of six layers with plated through

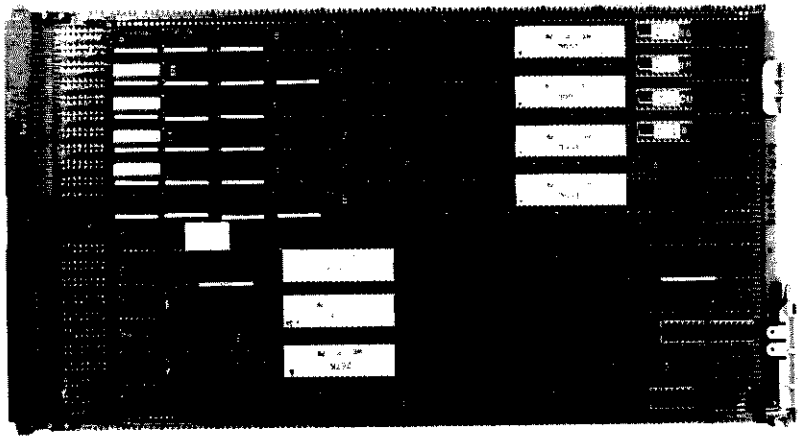


Fig. 1—The 3B20D Processor circuit pack.

holes to interconnect the layers. The structure of the printed circuit board is as follows: a component surface/signal layer, a signal layer, a power layer, a ground layer, a signal layer, and a wiring surface/signal layer. Board design used a standardized 0.100-inch grid to facilitate computer aided design and also to permit automated assembly and test. Protection for surface-layer printed wiring is provided by a cover coat that is applied prior to component assembly. The circuit pack multilayer board is normally 7.67 by 13.375 inches (although a few 7.67 x 9.375-inch boards are also used). All boards are 0.062 inch thick.

The circuit pack designs use either a 200-contact or 300-contact connector to interconnect the pack to a unit backplane. The connectors consist of a matrix of contacts contained in a plastic housing soldered to the edge of the printed circuit board. These contacts are bifurcated and selectively gold plated to provide a low-resistance, low-cost connection that will perform well over a 40-year design life. The circuit pack connector mates with a matrix of 0.025-inch-square pins mounted in the backplane on a 0.125-inch grid. The mating of the circuit pack contact with the backplane pin results in a loading of the bifurcated contact with a minimum normal contact force of 100 grams at the end of a 40-year life. The backplane pin is also selectively gold plated to provide a high-reliability, low-cost contact.

### *2.6 Frame unit*

The next level of circuit packaging is the frame unit. Circuit packs mount into a frame unit that consists of a backplane printed wiring board with pins and apparatus mountings to support the packs. The backplane board is approximately 8.00 by 22.55 inches in size and

typically interconnects 21 circuit packs. Three apparatus mountings mount across the backplane to guide and support the circuit packs. The structure of the printed wiring backplane consists of six circuit layers that are interconnected with plated through holes. Pins are staked into the plated through holes on a 0.125-inch grid to match the circuit pack connector. These pins have low assembly cost and make highly reliable contact with the circuit boards.

The apparatus mounting provides circuit pack support and alignment with the backplane pins (Fig. 2). The apparatus mounting is designed to provide minimum impedance to air flow moving vertically through the unit. A designation strip is located at the top front of the mounting identifying the appropriate circuit packs. Matching the designation strip with the plastic faceplate of each circuit pack ensures proper installation.

The pins in the backplane extend on both sides of the printed wiring board. On the side of the backplane away from the circuit packs, additional connections between circuit packs can be made by wire, automatically wrapped to the pins. The majority of circuit pack

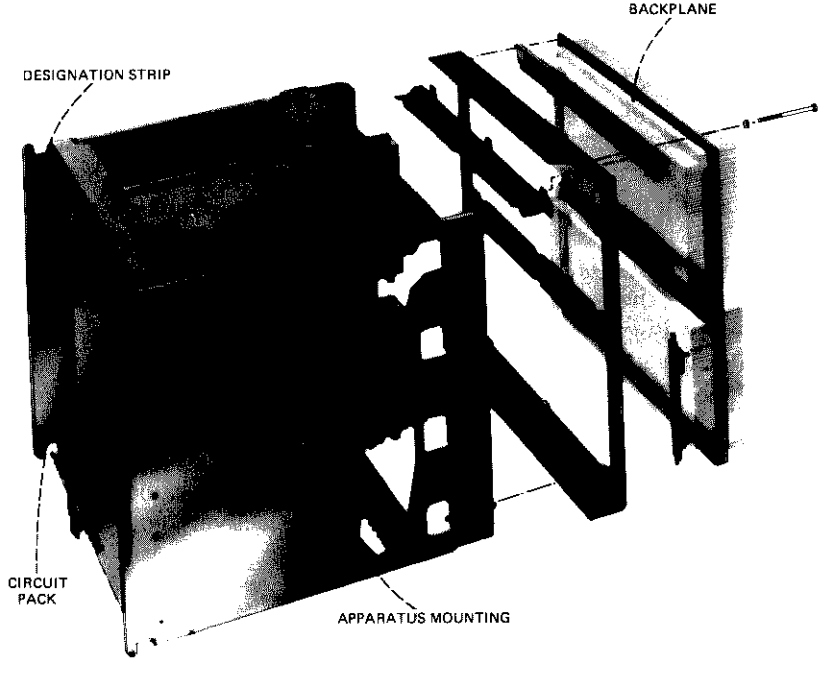



Fig. 2—The 3B20D Processor frame unit.

interconnections are made by signal paths within the backplane multilayer board; wires are used for changes or for connections that could not be routed in the printed circuit board. Also this side of the backplane is used for interconnecting different frame units or frames via connectorized cables. Many different connectors are used varying in size from 6 contacts to 24 contacts. This cable connector family supports both discrete wire (or switchboard) cable as well as multiconductor tape cable. The cable connectors are guided and retained on the backplane with special cable connector apparatus mountings that are attached to the backplane.

## III. 3B20D UNIT PHYSICAL DESIGNS

### *3.1 Control unit frame*

The 3B20D control unit frame is the core of the processor. Its modular equipment design permits the equipage of optional hardware features suitable to support a wide range of applications. The control unit frame in its maximum configuration contains the following units: central control, main store module 0 and 1, direct memory access input/output, cooling, and power. Figure 3 shows a maximally configured frame and denotes those units not required in a basic 3B20D Processor.

The 3B20D Processor frame units are mounted in a 7-foot-high, 24-inch-deep, 2-foot 2-inch-wide framework as shown in Fig. 3. Optional units and/or circuit packs are simply omitted where host system requirements do not require them. Subsequent paragraphs provide a description of the control unit frame units and their functions.

#### *3.1.1 Central control unit*

The central control unit is 10 inches high by 2 feet 2 inches wide. It contains the following required circuit functions:

(*i*) Central processing unit
(*ii*) Microstore
(*iii*) Main store update
(*iv*) Maintenance channel.

It has circuit pack positions allocated for the following optional features:

(*i*) Two input/output channel boards
(*ii*) Growth microstore
(*iii*) Cache
(*iv*) Utility circuit.

Also a circuit pack position has been reserved for the connection of a test set. The unit includes a fuse block, extending the full width of the unit, which provides individual fusing for each pack.[3]

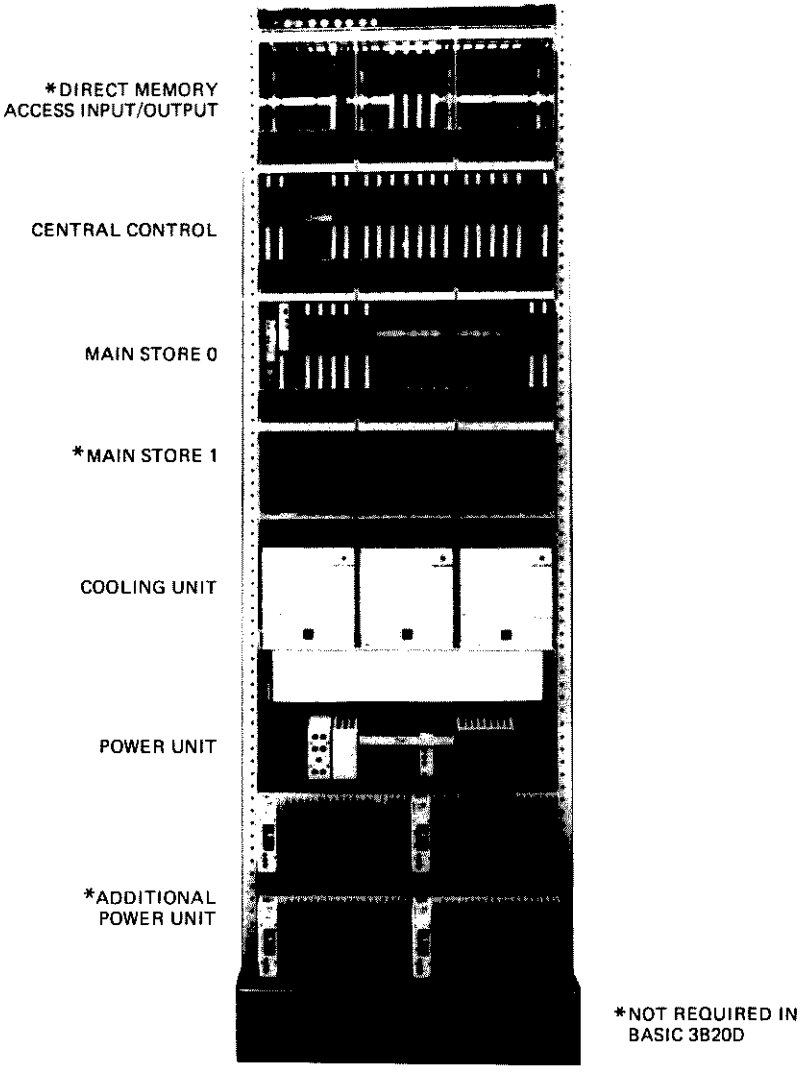



Fig. 3—The 3B20D Processor central control frame.

### *3.1.2 Main store unit*

The main store units are each 10 inches high by 2 feet 2 inches wide and may be equipped with a maximum of 16 megabytes of memory. The unit is minimally equipped with store controller packs and with one array pack. Additional main store array packs can be added one at a time to form a maximum complement of 16 array packs per module. When a unit is using 512K-byte arrays, two main store units

may be used per frame to provide up to 16 megabytes of memory; when 1024K-byte arrays are used, only one memory unit is required for the maximum addressable 16 megabytes of memory.[4]

#### *3.1.3 Direct memory access unit*

The Direct Memory Access (DMA) unit is 10 inches high by 2 feet 2 inches wide and is used for expansion of I/O channel requirements beyond the two positions provided in the central control unit. It also houses the DMA controllers and their associated I/O channels.[3] A total of five positions are available for I/O expansion. I/O channels that can be equipped in these positions include the dual-serial channel, serial channel, and applications channel interface packs. There is room in the unit for two DMA controllers, although normally only one is equipped. Each DMA controller is equipped with a dual-serial channel with the option to add up to three additional dual-serial channel packs.

#### *3.1.4 Cooling unit*

The cooling unit is 8 inches high by 2 feet 2 inches wide and provides forced air cooling to keep device temperatures within specifications. The cooling unit is divided into three replaceable fan module assemblies, each with a fan and filter.

#### *3.1.5 Frame power system*

Power for the control frame is provided by the power unit located at the base of the frame. The power unit includes bulk power converters, a reference supply, and a power distribution/fusing system. Two to four −48 to +5 volts dc power converters are tied in parallel to provide power to all frame units via a frame and fuse panel bus bar system. The +12 volt power required by the devices used on the 512K-byte store arrays is supplied by a −48 to +12 volt dc converter located in the power unit. The power unit is also equipped with input fuses for each of the converters and for the reference supply. The power switch for the frame is implemented as a common design circuit pack which provides control and power sequencing for the frame and is located in the main store unit.

### *3.2 Peripheral control frame*

The Peripheral Control Frame (PCF) houses those units which provide the interface communications between the 3B20D Processor and its periphery. This frame in its maximum configuration is equipped with an Input/Output Processor (IOP) basic unit, IOP growth unit, a Disk File Controller (DFC) unit, and a cooling unit. A port switch unit is also provided in one of the peripheral control frames. The frame is

a framework 7 feet high, 24 inches deep by 2 feet 2 inches wide. The following paragraphs describe the various units within this frame.

#### *3.2.1 Input/output processor basic unit*

The IOP basic unit is 16 inches high by 2 feet 2 inches wide and has a self-contained power system. This unit is subdivided into a processor section and two peripheral controller (PC) communities; each of the latter can be equipped with up to four PC circuit packs. The power switch circuit pack provides power control and sequencing functions to this unit and to the growth unit if present. DC to DC converters and a fuse system similar to the one used in the control frame complete the power system.[5]

#### *3.2.2 Input/output processor growth unit*

The IOP growth unit can be added to the basic unit to provide additional PC communities. It is 10 inches high by 2 feet 2 inches wide. This unit provides two communities of up to four PCs each to meet additional host system peripheral requirements. The self-contained power system for this unit is controlled by the power switch in the IOP basic unit.[5]

#### *3.2.3 Disk file control unit*

The DFC unit is 16 inches high by 2 feet 2 inches wide and includes a self-contained power system. The DFC is used to interface the 3B20D control unit with the disk memory system.[6] Control and data communications are provided for the moving head disk drives.

#### *3.2.4 Cooling unit*

The cooling unit used in the PCF is the same design as used in the control unit frame (see Section 3.1). This unit is located below the IOP basic or growth unit as appropriate to provide forced air cooling.

#### *3.2.5 Port switch unit*

The port switch unit is 4 inches high by 2 feet 2 inches. This self-powered unit provides one or two communities of up to three port switches each and also has positions for up to five scan and signal distribution interface circuits. One community is required for the port switching of the maintenance terminals.

## IV. ELECTRICAL AND PHYSICAL DESIGN PROCESS

The electrical and physical design of the 3B20D Processor was based heavily on the application of a computer aided design process and the use of design standards and audits. Design standards were established and used to control the design process and to assure commonality

among separate design activities. The design audits were used to provide a predicted measure of system performance based on a given design and thus allowed feedback for making design changes to maximize system performance.

### *4.1 Circuit pack, unit, and frame design*

The fundamental hardware building blocks (circuit packs, units and frames) were designed and tied together with the assistance of a CAD system. After the initial paper design, a circuit pack design file was created in the CAD system that contained all connectivity information. The design file represents the official record for the circuit. Circuit board manufacturing information, schematic drawings, design audits, and system connectivity can be created from these files. As the design progressed and various information was created, such as board routing, schematic representation, etc., it was added to the file. This circuit pack design file was also used to generate the software simulation files which were used to functionally verify the design and generate circuit pack test information.[7]

Fundamental to the CAD system were the design standards used throughout the 3B20D Processor. The design standards include a project library with integrated circuit and circuit board topology information, fixed power and ground pinouts for circuit packs, and common hardware definitions. The standards also provided specific rules for electrical design (fanout, crosstalk, etc.) that could be audited by the CAD system.

The interconnection of the different circuit packs within a unit was captured in a unit design file in the CAD system. These files contained all connectivity information and, like the circuit pack files, provided the basis for developing and storing detailed wiring information for the unit backplanes.

Frame drawings were a composite of design information from the unit files, with connectivity between the units added. Also included in the frame drawings was power distribution information.

### *4.2 Designing for system crosstalk and noise margin*

The rapid switching transitions in TTL logic made noise and crosstalk performance a critical design criterion. For the system to operate successfully, noise had to be kept well below the switching thresholds of the integrated circuit devices used over a wide range of device temperature, supply voltage, and packaging variations. The large number of signal nets and the difficulty of tracking down subtle noise problems in a system as complex as the 3B20D dictated a set of design rules based on worst-case conditions. Comprehensive audits were made to assure compliance with these rules.

Two major noise audits were provided for the 3B20D. The first was a crosstalk audit (parallelism audit) that searched the circuit pack and backplane routing files to identify for each net all signal nets immediately adjacent and parallel to the net of interest and sum the exposures. Through theoretical calculations and experimental results a maximum safe exposure limit was established for each device type. All nets exceeding this crosstalk exposure limit were referred for further analysis to the circuit designer. The designer's analysis would determine whether rerouting was required to assure that system performance was not adversely affected. The second major noise audit was for current shuttling. Analysis for this audit is a complex function of the device types and detailed routing information. Using a device parameter library and the circuit pack routing files, the current shuttling audit generates a list of all nets with potential noise problems. These nets were then reviewed by the circuit designer.

A fanout audit also was provided. This audit is based on detailed device drive and loading information in the device library and on the circuit pack and unit connectivity files. Violations of the fanout rules can result in reduced noise margins, or in extreme cases, malfunction of the logic.

### *4.3 Timing analysis*

Timing analysis consisted of computing the minimum and maximum delay for each device in a path of interest under actual load conditions. The delay consists of an intrinsic component through the device itself and an interconnection component that depends on the wiring parameters, fanout, and the output drive capabilities. CAD programs and libraries were developed to extract the interconnection data to compute the device and path delays.

### *4.4 Thermal analysis*

Computer aided tools also were used for thermal analysis. In particular they were used to predict device temperatures on certain heat-sensitive devices in the frame environment. The tools used an experimentally determined heat transfer coefficient and integrated circuit power dissipations to calculate device temperatures. In addition to confirming the adequacies of the design, these tools were used to specify temperatures for selected heat sensitive devices where device cost, performance, and temperature were important trade-offs.

## V. FLEXIBLE FLOOR PLAN

Because the 3B20D Processor was designed as a cost-effective processor for a wide range of Bell System applications, it usually will be installed as a group of frames within the host system environment; in

other cases, such as retrofit environments, the processor frames may have to fit into the space available. The 3B20D has been designed to support both of these situations. From an environmental standpoint, no special air-conditioning diffusers are required for the 3B20D equipment. However, sufficient cooling and ventilation must be provided so that the aisle air temperature does not exceed 37°C on a long term basis or 49°C for a short term.

The recommended typical floor plan for the 3B20D is shown in Fig. 4. This plan depicts a basic processor with its most likely growth patterns and satisfies most system applications. Moving head disk cables and associated duct work have been designed to support the typical plan. Other floor plans are permitted providing cable length restrictions are observed. These cases may require special cables and/or duct work which would have to be engineered as part of the host system.

## VI. TECHNOLOGY PERFORMANCE

### *6.1 Rapid development*

To assure a rapid development of the 3B20D Processor, standardized hardware components and special prototype hardware were used. The basic *Bellpac* packaging system hardware building blocks were used wherever possible to reduce availability intervals. Quick-turnaround wire-wrap boards were used for prototype circuit packs and backplanes. The use of a CAD system allowed the transition from wire-wrap to multilayer design to be a minimal design effort. The net result

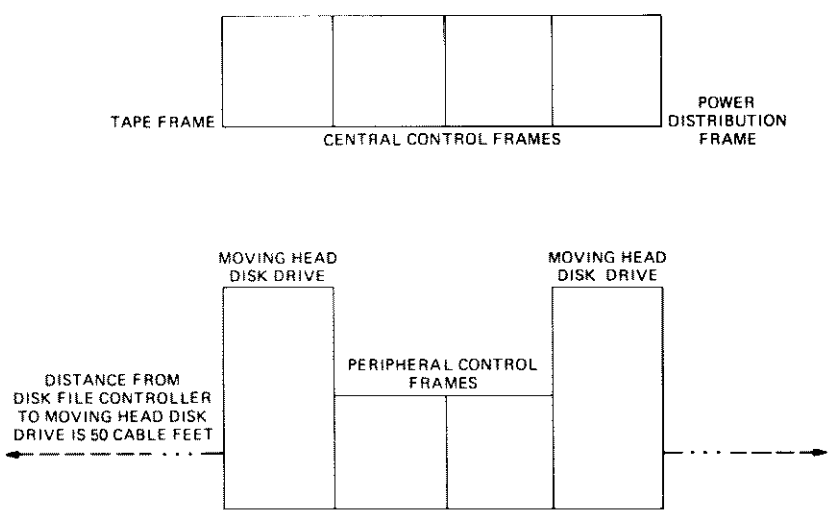



Fig. 4—Recommended floor plan for the 3B20D Processor.

of the use of off-the-shelf hardware, computer aided designs, and wire wrap models was that the objective project development schedule was achieved.

### 6.2 System electrical performance

The electrical performance of a system is best measured by the successful achievement of system performance objectives and the lack of timing and noise problems. Using the technology system described, all 3B20D performance objectives have been met. Further, despite the high degree of complexity, few noise and timing problems were encountered in the actual hardware. This successful result was achieved by the coordinated and comprehensive set of device and packaging specifications and design aids and audits.

### 6.3 Thermal performance

The 3B20D Processor is designed to operate in a normal environment of less than 37°C and at a maximum elevation of 6000 feet above sea level. In the event of a building cooling failure, the processor will remain operable in air temperatures of up to 49°C. During these building cooling failures, the circuit board temperatures do not exceed the design goal of 95°C in the control frame and DFC. In the IOP units there are a number of critical LSI devices, which are kept below 70°C to maintain good timing margins. All peripheral equipment (disk drives, tape drives, printers, and terminals) is tested by the respective vendor for operation up to 49°C.

Final thermal acceptability of each 3B20D system to be shipped is verified in a factory heat test where each processor system, with peripherals, is heated to 49°C for a 6-hour functional test.

## VII. ACKNOWLEDGMENT

This paper comprises the work of many individuals, in addition to the authors. Special thanks for their participation is given to Messrs. H. A. Devine, S. C. Zemke, and C. E. Miller.

## REFERENCES

1. W. L. Harrod and A. G. Lubowe, "The BELLPAC Modular Electronic Packaging System," B.S.T.J., *58*, No. 10 (December 1979), pp. 2271–88.
2. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "Fault Detection and Recovery," BSTJ, this issue.
3. M. W. Rolund, J. T. Beckett, and D. A. Harms, "3B20D Central Processing Unit," B.S.T.J., this issue.
4. I. K. Hetherington and P. Kusulas, "3B20D Memory Systems," B.S.T.J., this issue.
5. A. H. Budlong and F. W. Wendland, "3B20D Input/Output System," B.S.T.J., this issue.
6. R. E. Haglund and L. D. Peterson, "3B20D File Memory System," B.S.T.J., this issue.
7. "LAMP: Logic Analyzer for Maintenance Planning," B.S.T.J., *53*, No. 8 (October 1974).

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# 3B20D File Memory Systems

By R. E. Haglund and L. D. Peterson

(Manuscript received March 10, 1982)

*The 3B20D file memory system employs microprocessor-based intelligent Disk File Controllers (DFC), each supporting a complement of one to eight 300-megabyte disk drives. The DFC communicates with the system via a high-capacity dual-serial channel link to the Direct Memory Access Controller (DMAC) that accesses the processor's main memory. The DFC communicates with its disk files via an industry standard interface. The disk power system is also microprocessor based. The system operates disk drives on commercial ac power until a power failure occurs. The disks are then switched under microprocessor control to inverter power until the commercial ac fault is cleared, at which time the disk is returned to commercial ac power.*

## I. INTRODUCTION

The 3B20D file store consists of a microprocessor based Disk File Controller (DFC) plus one to eight 300-megabyte disk files. The files are powered by a microprocessor-based power system consisting of an inverter, a cycloconverter and a static switch. The DFC communicates via the 3B20D Dual-Serial Channel (DSCH) and through the Direct Memory Access Controller (DMAC) to the processor's main memory (see Ref 1). The DFC communicates with its disk files via a disk interface that utilizes the industry standard Storage Module Drive (SMD) interface.

The DFC provides for stand-alone processing of disk accesses via approximately 30,000 bytes of code stored in its PROM memory. Also stored in the PROM memory are 30,000 bytes of diagnostic code.[2]

The DFC uses a bit-sliced bipolar microprocessor to manage the 10-

MHz serial data streams to/from the disk files, the disk control functions, and the system interface. In addition, the processor runs a disk exerciser program to test disk drives during periods when no system accesses are in progress.

## II. SUBSYSTEM COMPONENT DESCRIPTION

The disk file memory system for the 3B20D Processor consists of three major elements: a DFC, 300M-byte moving head disk memories, and a 208 Vac power-control system for the disk memory units. The power system consists of a Power Control Unit (PCU) and an emergency -48 Vdc to 208 Vac Disk File Inverter (DFI).

### *2.1 The 300-megabyte disk drive*

The disk drives used in 3B20D Processor are high-speed, random-access, data-storage devices (Fig. 1) supplied by Century Data Systems and the Control Data Corporation. The disk drive unit consists of two physical parts. The basic drive assembly contains the chassis, power supply, air-filtration system, read/write head assembly, control circuit cards, and the input/output interface. The second part is the removable disk pack upon which data is recorded.

The disk pack contains 12 stacked platters. The inner ten are coated on both sides with the recording medium, ferrous oxide. The top and bottom platters are only for disk pack protection. The remaining ten platters have a total of 20 recording surfaces, of which 19 are used for recording data while the other contains prerecorded servo information that is used for head positioning and timing. Each disk pack is capable of storing 300M bytes (unformatted) of information. Each read/write head may be positioned over 815 different cylinders. The 19 read/write heads plus the servo head are mechanically attached together on a single carriage arm. Therefore, all heads are moved in parallel such that all heads are always positioned over the same cylinder. By selecting one of the 19 data surfaces, an individual track is selected. Each track is divided into 32 sectors or blocks of 512 bytes each. A sector is the smallest quantity of data written or read by a disk file controller.

Communication to and from the drive is via the industry-standard SMD type of interface. Two cables interconnect the drive to the disk file controller. One contains the control and status signal lines while the other contains the read and write data signal lines. The disk drive is powered from 208-volt single-phase power. Starting current is a maximum of 50 amperes, with running current of approximately 9 amperes. The disk pack is rotated at approximately 3600 RPM; thus the average rotational latency is 8.5 ms. The average head positioning

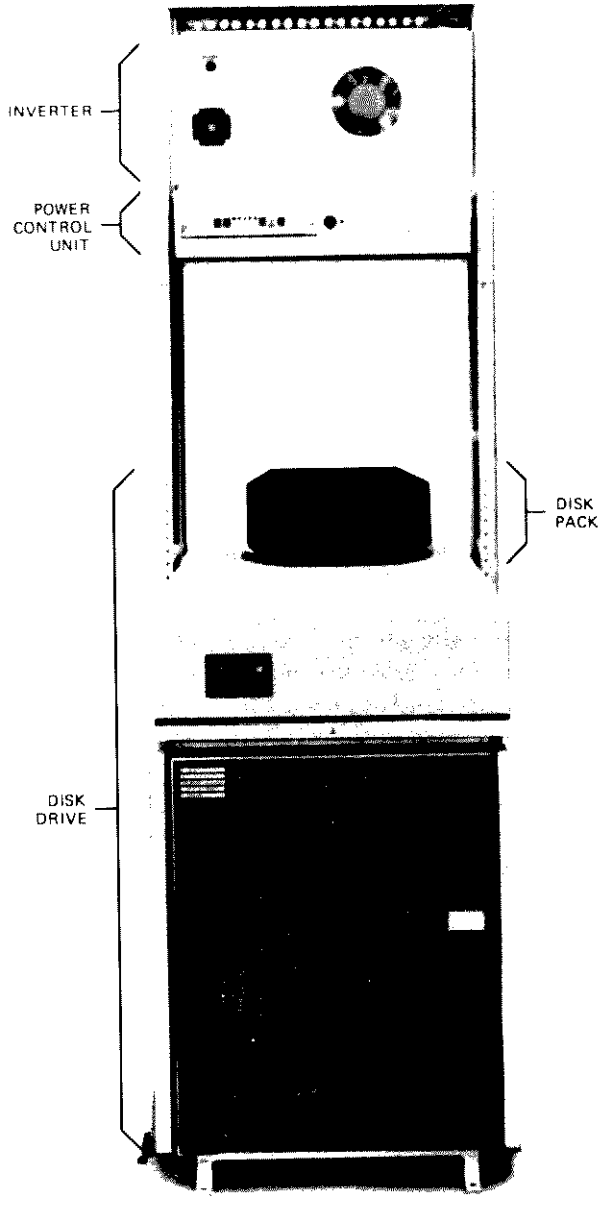



Fig. 1—Moving head disk frame.

(seek) time is 30 ms. Data is transferred to and from the disk drive at a 9.67-MHz serial data rate.

The disk units are designed to meet Bell System environmental requirements. Acoustical noise generated by a disk unit is less than 65

dBA. RFI gasketing material was used extensively to limit electromagnetic radiation from the units and reduce the susceptibility of the units to external electromagnetic radiation sources.

### *2.2 Disk power control and backup power system*

The PCU and the −48 Vdc to 208 Vac DFI are used in combination to supply uninterruptable power to a single 300M-byte disk. The units interconnect with each other and the disk as shown in Fig. 2.

Inputs to the PCU are −48 Vdc, two sources of 208 Vac, and control signals from the DFI. Power for the logic circuitry is derived from the −48 Vdc. One source of the 208 Vac is from the commercial power grid or the essential ac grid in a central office. The other source of 208 Vac is from the DFI. Status and control functions also are received by the PCU from the DFI.

Cooling for the DFI is provided by two fans. The fans are active only when the unit is supplying 208 Vac. The power dissipated by the DFI under a load condition is 500 watts when supplying 2000 watts to the disk. Normally power to the disk is supplied by the commercial or essential ac. Under this condition, the inverter dissipates no more than 150 watts.

Even though nominally continuous, commercial power is subject to minor interruptions of short duration. Since the disk pack has substantial inertia it is possible to power the motor from the commercial ac mains despite such interruptions. The disk drives used in the 3B20D Processor have ferroresonant supplies for their own internal dc power, and these supplies are tolerant of minor interruptions in input voltage.

The 3B20D disk power system takes advantage of these characteristics to deliver power to the disk drive from the most efficient source.

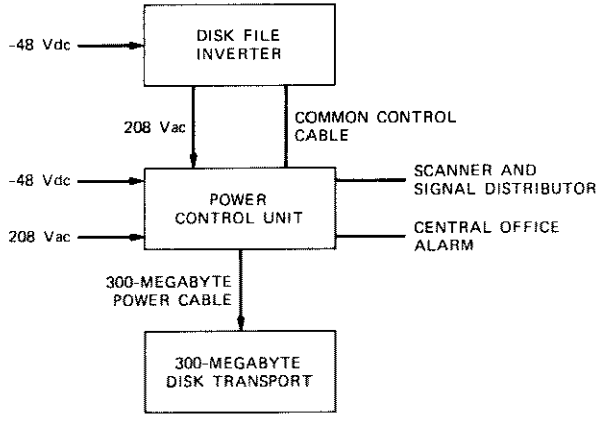



Fig. 2—Interconnections between the control panel, inverter, and disk transport cabling units.

The DFI powered from the battery plant supplies the power for the disk drive when the ac mains are not available or out of tolerance in voltage or frequency. A silicon controlled rectifier static switch provides the changeover within two cycles of the time the ac mains are found to be out of tolerance. The inverter is phase locked to the ac mains voltage when it is present so that the transient due to the switchover to battery power is minimized. The operation of the disk drive is unaffected by the switchover between ac and inverter power.

The standby inverter (Fig. 3) operates at 600 Hz; its output is converted to the required 60 Hz ac by means of a tap-changing cycloconverter. This provides a piecewise approximation to the desired 60-Hz sine wave (Fig. 4). The inverter frequency was chosen high enough to allow the use of a small, efficient inverter transformer and also to allow a sufficient number of intervals in which the output voltage could be selected to make a good approximation to a sine wave. The cycloconverter output has the same peak and rms values and same average value over a half cycle as the sine wave it replaces. The third and fifth harmonics are also reduced to small values by appropriate choice of cycloconverter parameters.

The disk power system is controlled by a microprocessor housed in the inverter. This processor administers the operation of the inverter, cycloconverter, and static switch so that the proper power source is connected to the disk drive. The choice of power source is made on

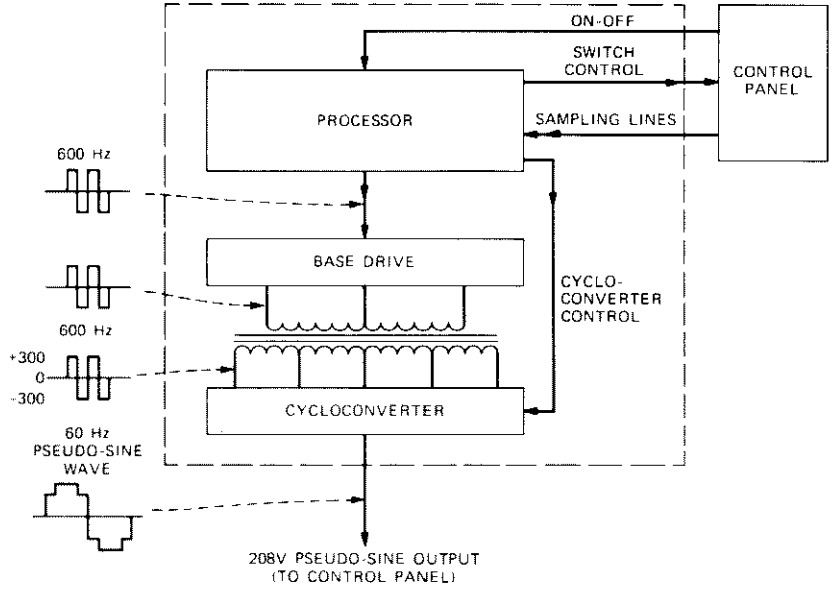



Fig. 3—Disk inverter operation.

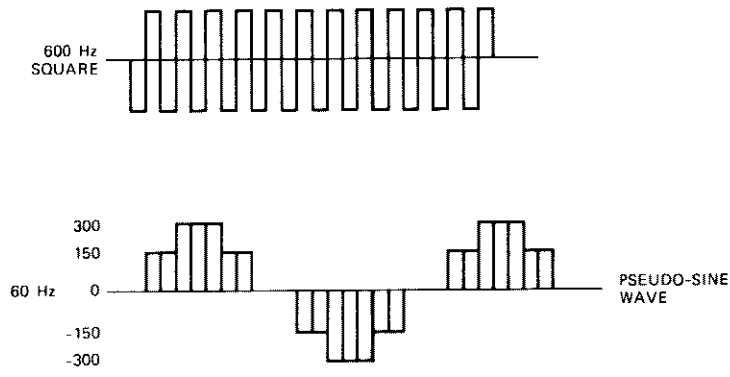



Fig. 4—Pseudo-sine wave generation.

the basis of sampling the ac mains for voltage above the programmed threshold value ten times per half cycle. If the mains voltage supply is found to be deficient for two samples in any half cycle, the static switch disconnects the ac mains and the cycloconverter supplies power to the disk drive. When the mains voltage has been determined to be within acceptable limits in voltage and frequency for a sufficient time, the cycloconverter output is inhibited and the static switch again allows the mains to power the disk drive.

While operating from the battery plant, the inverter regulates its output voltage to compensate for battery variations. A reflex regulator circuit uses some of the 600-Hz inverter output voltage in a phase-controlled rectifier to control the apparent input voltage to the inverter to maintain it constant at −52.2 volts in the face of battery voltages in the range of −42.75 to −52.5 volts. The microprocessor controls the operation of the regulator so that the peak current demands on the inverter transistors are not increased by the operation of the regulator. The regulator operation is inhibited during the times the peak output current is delivered to the load, and regulator operation is only allowed when the 60-Hz output is in the part of the cycle where the output voltage, and therefore the inverter load current, is near zero.

Since it may be necessary to change the disk pack while ac power is not available, the inverter must be able to start the drive motor. The current required to start the disk drive motor is much larger than the current required to run the motor. Since it would be uneconomical to size the inverter to carry the starting load, the microprocessor controls the cycloconverter output voltage during the time the motor is starting so that the starting current is greatly reduced. This power reduction is done at the expense of starting time; the time required to start the drive motor is 45 seconds on battery and 15 seconds on the commercial mains.

With the inverter always powered up even while the disk is being

powered by commercial ac, the possibility exists that the power-handling components could suffer a failure that would be undetected until the inverter was required to supply power. A diagnostic triggered by a disk being out of service is available to detect this failure. The microprocessor continually monitors the state of the power switch to detect when the disk has been taken out of service. When this condition is detected, the microprocessor starts the diagnostic routine, which causes the inverter to supply power to the disk drive for a few minutes while monitoring the cycloconverter output voltage for the correct levels. Thus, any failure of power-handling components will be detected during the regular daily diagnostic exercise of the disk frame.

### *2.3 Moving head disk frame*

The Moving Head Disk (MHD) frame (Fig. 1) is the frame configuration used to mount a disk, PCU and DFI.

The PCU and DFI share a common interface cable for transmitting status and control information. The DFI 208-Vac output is connected to the PCU via a flexible conduit. The disk unit is in turn connected to the PCU via a flexible power cord. The −48 Vdc input for the PCU is connectorized at the top of the frame. However, the DFI −48 Vdc input is routed directly to a power-distribution frame due to the wire gauge and currents involved.

The disk drive is rolled onto and secured to a metal base plate fastened in the base of the MHD frame. The drive protrudes out the front of the frame 18 inches. The base plate is hinged and is raised or lowered by jackscrews on both sides of the base plate.

The I/O cables between disks and the DFC are routed in a rectangular duct at the rear of the MHD frame. The duct serves to provide additional electromagnetic shielding and physical protection for the cables. In multiple, adjacent MHD frame applications, the cable ducts form a continuous trough for the I/O cables.

### *2.4 Disk growth and floor plan*

The MHD frame was designed to provide for growth in a modular fashion. Depicted in Fig. 5 is a typical floor plan for the 3B20D Processor. The Peripheral Control (PC) frame is situated between two sets of MHD frames. Each PC frame contains a DFC (Fig. 6).

The total cumulative control cable bus length between a DFC and all disks controlled by the DFC is 100 cable feet. Thus, all MHDs controlled by a particular DFC are placed adjacent to each other so as not to exceed the 100-foot limit.

### *2.5 3B20D disk file controller*

The 3B20D Disk File Controller (DFC) is a high-performance, microprogrammed processor whose design provides efficient control of

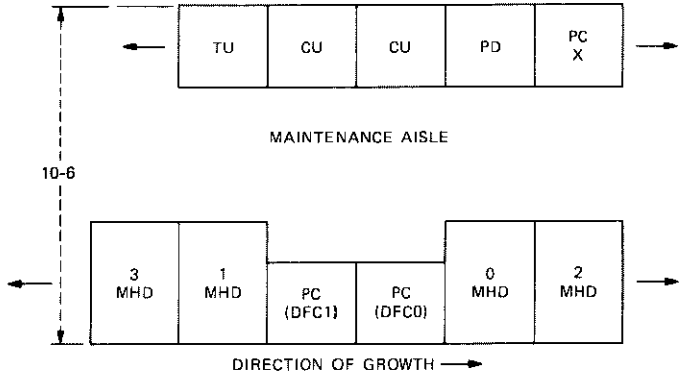



Fig. 5—Typical floor plan of the 3B20D Processor.

MHD drives and data transfers between the 3B20D Control Unit (CU) and MHD units. Each DFC supports a maximum of eight MHD units and utilizes a standard 3B20D input/output peripheral interface. The following sections discuss the hardware architecture and functionality of the DFC.

### *2.5.1 3B20D DFC architecture*

***2.5.1.1 DFC communications bus.*** A DFC with its associated MHD units—as shown in Fig. 7—consists of a processor, MHD interface, and the two circuit boards that provide the interface to the 3B20D Processor. These are linked together via a common communications bus.

The bus, which is dc coupled, provides 16 data bits plus 2 parity bits, 5 bits for source address, 5 bits for destination address, and 1 synchronization clock line. The source/destination addresses are binary encoded allowing a maximum of 32 unique hardware registers to be addressed on the bus. The DFC's processor and the 3B20D Processor interface each utilize 8 address codes, and the remaining 16 codes are assigned to the MHD interface. Since the bus provides unique source and destination address buses, only one bus cycle, which is defined by the synchronizing clock, is required to transfer a 16-bit data word from one hardware register to another. All data transfers on this bus are initiated and controlled by the DFC's processor. The period of the synchronizing clock may be 150, 200, 250, or 300 ns and is dependent on the instruction being executed by the processor, the capabilities of the hardware, and the timing requirements of specific firmware routines.

***2.5.1.2 DFC/MHD interface.*** Each DFC will support a maximum of eight MHD units equipped with an SMD interface. As depicted in Fig. 7, each MHD is connected to the DFC via two cables, one of which is common to all MHDs and is daisy chained from one unit to the next.

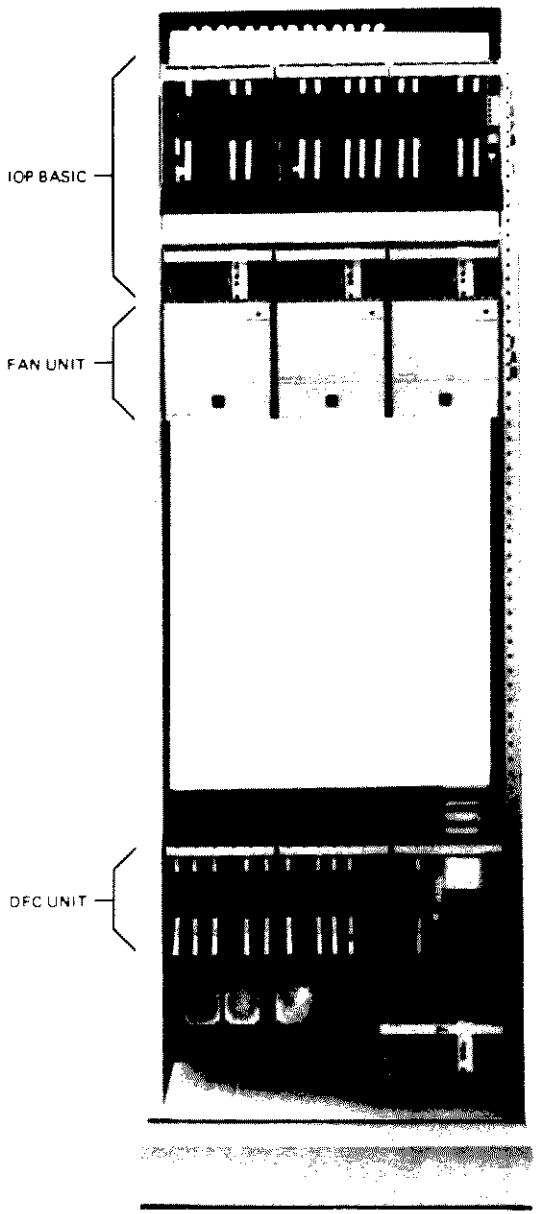



Fig. 6—Peripheral control frame.

This cable provides a communication path between the DFC and MHDs for command and status information. The transfer of data between the MHDs and the DFC occurs serially over a private path provided by the other cable. The data transferred is in a Non-Return

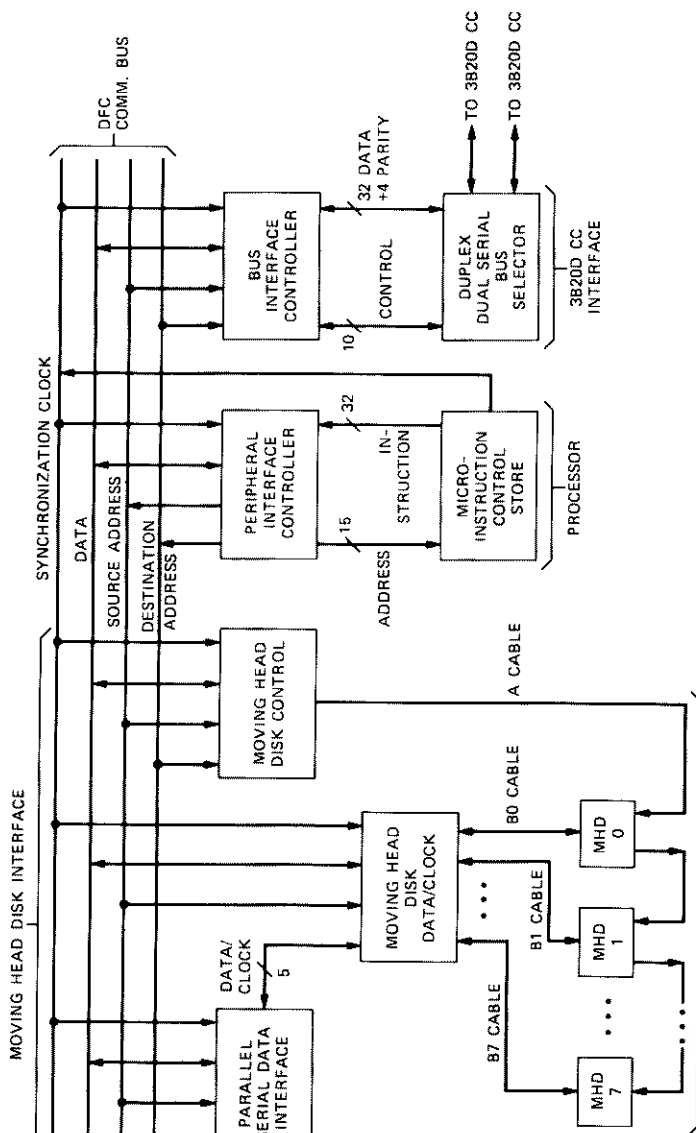



Fig. 7—Disk system block diagram.

to Zero (NRZ) format and a synchronous bit clock accompanies the data to simplify the circuit requirements within the DFC and MHD.

Because the MHD units may be located up to 100 cable feet from the DFC, signaling paths in both cables are implemented as twisted pairs, and both the DFC and MHDs interface with the cable using differential line drivers and receivers.

***2.5.1.3 DFC/3B20D control unit interface.*** The DFC communicates with the 3B20D CU via a DSCH interface consisting of the Duplex Dual Serial Bus Selector(DDSBS) in the DFC and DSCH in the 3B20D CU. The communications path between the DDSBS and the DSCH consists of five twisted-pair dc-coupled signaling paths. The pairs are divided into two bidirectional data pairs, two unidirectional clock pairs, and a single pair used to transmit service requests to the DMAC and interrupt requests to the 3B20D CU.

Using the clock and data pairs, the minimum amount of information that is transferred is 32 data plus 4 parity bits. The data is transmitted serially on the two data pairs (16 data plus 2 parity on each pair) in an NRZ format. A synchronous bit clock is transmitted on the appropriate clock pair by either the DDSBS (DFC to CU transfer) or the DSCH (CU to DFC transfer). The interface does support a block transfer mode where 16 words of 32 bits may be transferred as a single block between the DFC and CU. This mode of operation is useful in minimizing the overhead associated with transferring large blocks of data between the DFC and CU.

The DFC's processor does not access the DDSBS directly, but rather has read/write access to several registers in the Bus Interface Controller (BIC). The BIC functions as a buffer between the DDSBS (32 bits) and the DFC's processor (16 bits). The BIC buffers data and commands for the processor and performs the necessary handshaking to communicate with the DDSBS.

### *2.5.2 DFC processor*

The DFC's processor consists of the Peripheral Interface Controller (PIC) and its associated Microinstruction Control Store (MCS). The PIC is a 16-bit, bit-sliced microprocessor that is capable of performing all the common arithmetic, logic, and sequencer flow-control operations found in many 16-bit minicomputers. The PIC contains a microprogram data register, Arithmetic Logic Unit (ALU), 8K bytes of RAM that serve as a temporary data store, vectored interrupt control, and microprogram address generator.

The MCS stores instructions that the PIC executes in Programmable Read-Only Memory (PROM) devices. Each MCS circuit pack has the capability of storing a maximum of 4K microinstructions. A total of three MCS circuit packs are utilized. In addition to functioning as a

store, the MCS also generates the synchronizing clock for the PIC and DFC communication bus based on bits stored with each microinstruction.

On the rising edge of each synchronizing clock signal, the PIC generates a new microprogram address that is sent to the MCS. The MCS provides the PIC with the instruction stored at the requested address before the next synchronizing clock, at which time the PIC loads the instruction in its microprogram data register and sends a new address to the MCS. The PIC then executes the registered instruction while the next succeeding instruction is being fetched from the MCS. This pipelined operation that is provided by the microprogram data register allows a higher instruction execution rate than would otherwise be possible and contributes to a higher level of DFC performance.

Each PIC microinstruction is 40 bits wide and is divided into several fields as shown in Fig. 8. The PIC registers bits 0 through 31 in its microprogram data register while bits 32 through 35 are used by the MCS to verify parity on each instruction that is fetched, and bits 36 through 39 determine the period of the synchronizing clock generated by the MCS. Bits 0–4 and 5–9 are the source and destination address bus fields, and since these bit fields are dedicated, each PIC instruction specifies the movement of data from one register to another. Bit 12, when set, disables all hardware interrupts. The capability to disable interrupts on a per-instruction basis is required for the proper functioning of instructions that implement "JUMPS" and also allows the programmer to construct code segments that are critical from a real-time viewpoint and cannot be interrupted. Bits 10 and 11 determine how the overlay field, bits 13 through 31, are to be interpreted by the PIC.

### *2.5.3 MHD interface*

The MHD interface consists of three circuit packs that allow the PIC to communicate with its MHD units. These circuit packs are the MHD Control (MHDC), MHD Data/Clock (MHDDC), and Parallel Serial Data Interface (PSDI).

***2.5.3.1 MHD control.*** The MHDC consists of four registers, varying in width, whose outputs drive the control bus of an MHD with an SMD interface. These registers may be loaded and read by the PIC via the internal DFC communication bus. In addition to these registers, the

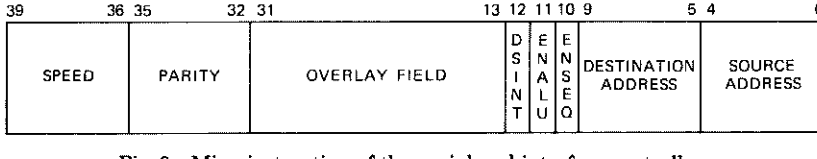



Fig. 8—Microinstruction of the peripheral interface controller.

circuit pack also provides a port that allows the PIC to sample 8 bits of status information from the MHD that is currently selected by the DFC. The output of each register drives the input of the differential line driver whose outputs drive the control cable to the MHD units. Likewise, the 8 bits of MHD status are driven onto the cable by the selected MHD unit and differential line receivers in the MHDC pack translate the differential voltage to a TTL level.

The device-enable register is a single bit register that is reset by the DFC during a power-up sequence, thus disabling the interface between the DFC and MHDs. The PIC writes this register to a one, enabling the interface, after executing its power-up microcode.

The unit-select register is a 4-bit register that contains the address of the MHD unit communicating with the PIC.

The data register is 11-bits wide and its outputs form the data portion of the control cable. This register is used with the disk tag register to transfer command information to the MHD unit consisting of cylinder number, read/write head number, and read/write/initialization commands.

The tag register's outputs form strobe signals that cause the MHD units to use information present on the data portion of the control cable. For example, bit 3 going active causes all MHDs on the control cable to compare the state of the unit-select bits with the MHD's address code. If a match occurs, then that MHD becomes selected and honors commands sent using bits 0 through 2 of the tag register. Since each MHD connected to a DFC has a unique address code, only one MHD may be actively communicating with the DFC at a given time. time.

***2.5.3.2 MHD data/clock.*** The MHDDC interfaces the serial data and clock between the MHDs and the PSDI, and provides the PIC access to a select acknowledge signal generated by each MHD. The select acknowledge signal is made active by an MHD whenever it is selected, and by providing one acknowledge signal per MHD allows the PIC to verify that the correct disk volume has been selected for use. With respect to the serial data and clock signals, the MHDDC functions as a voltage level translator (differential signaling to TTL and vice versa) and signal multiplexer, routing the data/clock signals between the selected MHD and the PSDI.

***2.5.3.3 Parallel serial data interface.*** The PSDI allows the PIC to transmit data to and receive data from the MHD. A block diagram of the PSDI is shown in Fig. 9.

Data exchange between a DFC and an MHD drive takes place over a 10-MHz serial data path. During a disk write, the PSDI performs parallel to serial data conversion, checks data parity as the data is being shifted out serially, and computes a 32-bit cyclic ECC that is

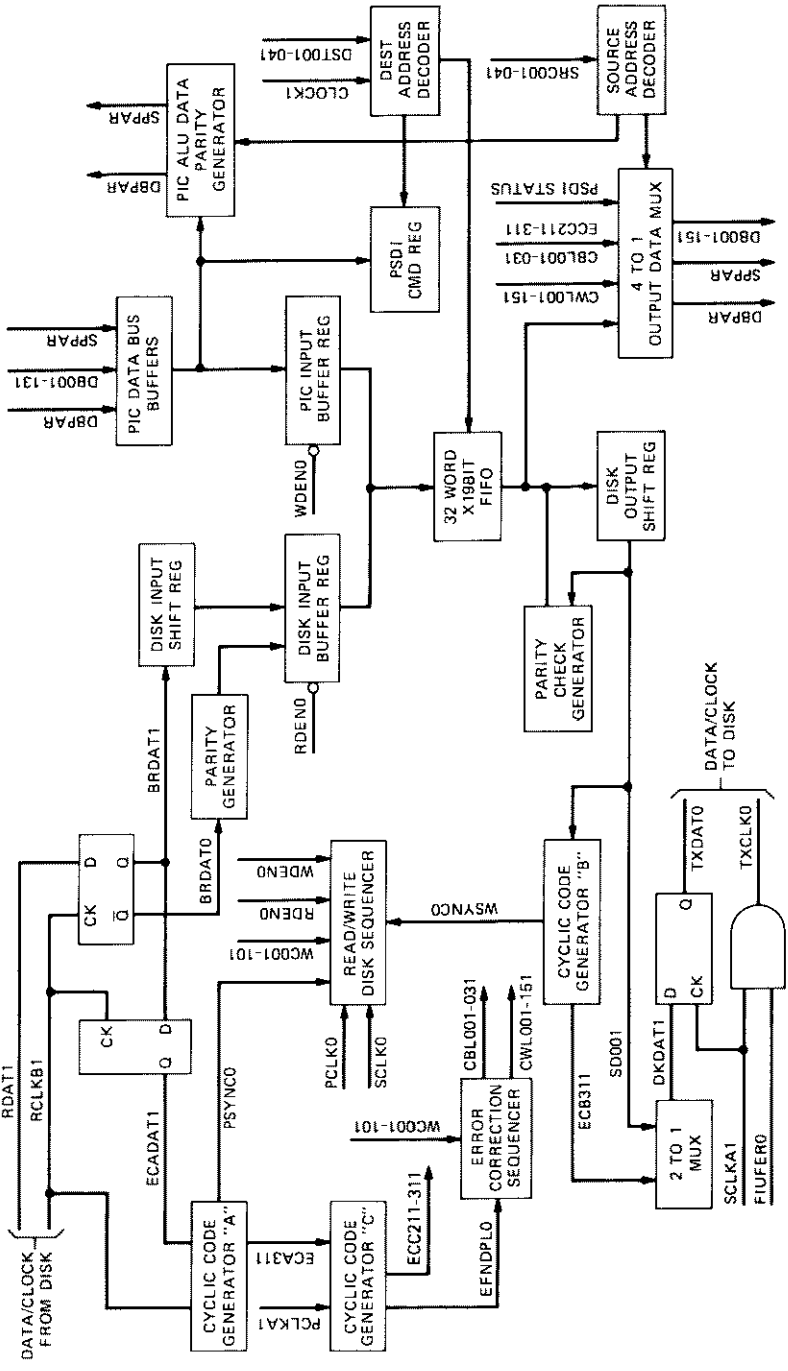



Fig. 9—Parallel/serial data interface.

also transmitted to the disk after a predefined block of data has been transmitted. During a disk read, the PSDI performs serial-to-parallel data conversion, computes data byte parity, and evaluates the ECC read from disk to determine if any errors are present in the data. While the PSDI itself does not perform the error-correction function, it does provide the PIC with the following information about each data block read from the disk: (*i*) The presence of errors, (*ii*) whether the error is correctable or uncorrectable, (*iii*) the location of the error relative to the start of the block, and (*iv*) an 11-bit correction mask that is to be exclusive ORed with the received data to affect the correction.

### *2.6 Disk read and write operations*

#### *2.6.1 Disk read operation*

The PSDI functions as a serial data interface between the PIC and an MHD drive. During disk reads serial data and clock transmitted by the disk drive are assembled into 16-bit words in the disk input shift register, and stored in the data FIFO. Simultaneously, the serial data is also presented to the input of a parity generator where data byte parity is assigned and stored in the FIFO together with the data, and to cyclic code generator A where a 32-bit code word is computed to be utilized during the error correction search process. In addition, this cyclic code generator searches the incoming data stream for the presence of data synchronization bits. The generator searches the synchronization characters following the preamble to locate the beginning of the address/data fields in the serial data stream. Once sync is found, the read/write disk sequencer begins assembling 16-bit words and storing them in the data FIFO. The only information stored in the FIFO is the address and data bytes. Once the ECC field from disk has been input to the code generator, the contents are serially transmitted to a second code generator C. Then, to start error correction generation, the first code generator begins its search for synchronization characters again.

Cyclic code generator C and the error correction sequencer function together to determine the presence and location of an error in the received serial data stream. Code generator C is a 32-bit shift register which has several feedback paths around it. The error correction sequencer supplies shift pulses to generator C and if after each shift the contents of the low-order 21 bits of generator C are zero, the correction sequencer stops and indicates that a correctable error has been found. If they are nonzero and both the bit and word location counters have not reached their maximum counts, the sequencer will generate another shift pulse. If the 21 bits remain nonzero and both the bit and word location counters have reached their maximum

counts, the sequencer stops and indicates that an uncorrectable error is present in the data stream.

An uncorrectable data error is one where the first and last data bits in error are separated by more than nine consecutive bits. A correctable data error is one where the first and last data bits in error are separated by nine or less consecutive bits. The first case of no error is distinguished from the error case by the value of the upper-order 11 bits for code generator C. With no bits in error, these 11 bits will all be zero.

As mentioned earlier, the PSDI does not perform the error correction function, but does provide the PIC access to the information necessary to perform this task. The correction mask code generator C high-order 11 bits, the bit location counter, word location counter, and correction sequencer status (correctable or uncorrectable data error) are made available to the PIC through the output data multiplexer whose outputs are the PIC data bus. The error correction code contained in each section of the disk is utilized together with code generators A and C, and the error correction sequencer are designed to detect and locate errors in the address, data, and the error correction code fields recorded on disk.

#### *2.6.2 Disk write operation*

When writing the disk, information contained in the FIFO is loaded into the disk output shift register where it is serialized and transmitted to the disk drive under the control of the read/write disk sequencer. Simultaneously, the serial data stream is applied to the input of the parity check generator and cyclic code generator B. Although each data word in the FIFO has two parity bits associated with it, they are not transmitted to the disk, but are loaded into the parity check generator each time the output shift register is loaded from FIFO. In addition to the parity bits, each data word also has associated with it a parity check control bit, which is loaded into the parity check generator during the FIFO to output shift register transfer. The parity check control bit either enables or disables the parity generator on a per-word basis allowing words with known bad parity to be transmitted without causing parity errors.

Cyclic code generator B computes the 32-bit ECC field to be written on disk based on information transmitted in the address and data fields. Since writing a sector on disk requires all fields shown in Fig. 6 to be written, code generator B also searches for the synchronization characters. When sync is found, the read/write sequencer configures code generator B to operate as a feedback shift register to compute the ECC field for the address and data fields that follow the sync characters. In addition, at sync time, the read/write sequencer begins counting the words it loads into the output shift register from the

FIFO. After the last data word has been serially transmitted from the output shift register, the read/write sequencer disables all the code generator B feedback paths, enables the contents of the generator to be shifted out to the disk, and then redirects the serial stream back to the FIFO/output shift register to transmit the post-amble characters. Once the ECC field is transmitted, both the read/write sequencer and code generator B revert back to their initial states searching for synchronization characters.

For diagnostic purposes it is useful to be able to write an invalid ECC field when writing a sector on disk. Setting a bit in the PSDI command/status register prevents the read/write sequencer from transmitting the contents of code generator B to the disk. Instead, the read/write sequencer continues to transmit information to the disk contained in the FIFO. This scheme allows any two 16-bit words to be written in the ECC field in place of the normally computed value provided by code generator B.

#### *2.6.3 Data FIFO*

A 32-word FIFO memory is loaded by the read/write sequencer and read by the PIC during a disk read operation, with the reverse being true during a disk write operation. Each word in the FIFO is 19 bits wide: 16 data, 2 parity, and 1 parity check enable bit. As explained earlier the parity check enable bit is significant only during a disk write operation. FIFO write access is controlled by the mode bit in the PSDI status register. Depending on the state of this mode bit, data is written into the FIFO by the read/write sequencer or by the PIC. Because the FIFO requires that valid data be present for 70 ns after the write pulse makes its low-to-high transition, two input data buffer registers are provided for use by the read/write sequencer and PIC.

Associated with the FIFO are four status flags. The FIFO ready and FIFO half full/empty flags provide the PIC information to load or unload the FIFO properly and efficiently. When the PSDI is in the read mode, FIFO ready low/high controls whether the PIC may or may not read a word from the FIFO. Likewise, when FIFO half full/empty occurs, it allows or prevents the PIC from reading 16 words from the FIFO.

In write mode, these flags function similarly with FIFO ready low, indicating that the PIC may load a single word in the FIFO, and FIFO half empty low, indicating that the PIC may load 16 words in the FIFO. In addition to these FIFO status indicators, FIFO underflow and overflow error flags have been provided. A FIFO underflow error occurs whenever the read/write sequencer or PIC reads data from the FIFO when the FIFO is empty or valid data is not present at the bottom of the FIFO. FIFO overflow occurs when either the read/write

sequencer or PIC writes the FIFO when it is already full or data in the top memory location has not rippled down to a lower memory cell. Both of these errors are latched into individual flip-flops that can only be cleared when the PSDI is master cleared.

### *2.7 DFC firmware structure*

#### *2.7.1 Introduction*

The disk file controller firmware provides the interface between the 3B20D Processor and the moving head disk for storage and retrieval of information. The firmware can make certain decisions of its own regarding the integrity of the MHDs or of itself, but will not take any actions on data content. It has a routine diagnostic exercise to check itself and a disk exerciser to check disk data format integrity. It also has sanity timers on data transfers to/from the processor and timers on disk accesses. Data is transferred between the DFC and the processor at an average rate of 1 megabyte per second. The DFC can handle up to 8 MHDs on a single controller. There are two job-processing options that may be invoked by the 3B20D. The DFC will process jobs on either a first-in/first-out basis or by a modified elevator algorithm. The elevator algorithm is used to keep the number of long head seeks to a minimum. There is also an autoread retry to help in recovering marginal data from the media.

The firmware will do as much as possible to guarantee the data integrity on a disk pack and will not allow data with bad parity to be written either to the disk or the 3B20D. The DFC will try, as much as possible, to perform 'overlapped' seeks. That is, if there are two or more requests, each to a different disk, the DFC will issue a seek to each disk before trying to do the job on any one of the disks. Thus, if a seek requires a long time to complete, and another disk has completed its seek, then the job for the disk which has completed its seek will be done.

To assure the operational capabilities of a disk, a disk exerciser is run periodically. The disk exerciser assures that the data within a given area of the disk is usable. By sequentially addressing various areas of the disk with the exerciser, eventually all areas of the disk will be verified for data integrity.

#### *2.7.2 Timing considerations*

The time it takes to get or put data on a disk file is determined primarily by how far away the data block is from the current position of the disk heads. The majority of disk accesses are less than 20 blocks in length, thus the rotational latency and seek times of the disk file become the determining factors in data transfer. In the optimal case, there is about 2 milliseconds of delay from the time the disk file

controller gets the command until it is through transferring the data. This is the sum of verifying the disk head location, doing error correction, and reading the requested block from a disk back to the processor. The worst case, assuming no DMA contention problems, is 74 milliseconds: the sum of the maximum seek time plus one rotation of the disk. The average time is 41 milliseconds.

#### *2.7.3 Firmware tables*

The DFC maintains an orderly record of data within the controller by using six tables: SYSGEN data table, circular queue table, Active Job Table (AJT), read data buffers, write data buffers, and the disk verify data table.

SYSGEN is a programmed input/output command received by the DFC. This command contains the address in 3B20D main memory of the SYSGEN table. The SYSGEN data table contains information for the DFC. The information includes the:

unload pointer
load pointer
disk driver queue address (virtual)
maximum cylinder allowed
optimization data
maximum number of jobs in disk driver's queue.

The job queue is an image of the data from the disk drivers queue. There are 16 bytes in each disk job description. There can be a maximum of 64 such jobs in the DFC at any one time.

The AJT has a series of entries to describe to the firmware the status of a particular drive. There is one such entry for each disk allowed on the controller. There are 32 words associated with each disk entry in the AJT. Some of the more important items in the list are: command, current disk maintenance state, current cylinder number, and where on the disk to start the job.

There are eight buffers set aside for the storage of data to be transferred to or from the disk. Each buffer is 263 words (16 bits each). There are seven status words and 256 data words in each buffer.

The disk verify routine checks the validity of data on the disk. This is done by reading a sector and determining if the header and data are correct. If they are not, then that disk block number is recorded in the disk verify buffer. Up to 64 such bad blocks are allowed before the verify routine will terminate.

## III. SUMMARY

The 3B20D file memory system provides a maximum formatted storage capacity of 2,030 megabytes of storage on eight 300-megabyte

disk drives. The disks are managed by a disk file controller using a high-speed bit-sliced microprocessor, which utilizes 30,000 bytes of microcode to asynchronously process data transfers to and from the 3B20D main store via DMA. The disk controller interface to the 3B20D processor is via the high-speed dual-serial channel.

## IV. ACKNOWLEDGMENTS

Acknowledgment is made to R. D. Adams, C. F. Ault, T. J. Clarey, R. A. Ehle, R. M. Gagliardi, T. S. Greenwood, D. H. Leonard, P. D. Olson, and E. M. Schaefer for their design contributions; and to R. C. Hatfield and N. C. Zielke for their supporting activities.

## REFERENCES

1. M. W. Rolund, J. T. Beckett, and D. A. Harms, "3B20D Central Processing Unit," BSTJ, this issue.
2. J. L. Quinn and F. M. Goetz, "Deferrable Maintenance and Diagnostics," BSTJ, this issue.
3. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman "DMERT Operating System," BSTJ, this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# 3B20D Input/Output System

By A. H. BUDLONG and F. W. WENDLAND

(Manuscript received March 18, 1982)

*The 3B20D Input/Output system supports a large number of peripheral devices of various types. The 3B20D Input/Output Processor was designed for maximum flexibility because of the diversity of the peripheral interface requirements. Because of this diversity the basic structure was developed around a high-speed bipolar multiplexor. This multiplexor interfaces with a standard microprocessor bus to numerous specialized one- or two-board computers, which interface to the peripheral devices. Power and unit designs of the input/output processor also provide flexibility in power control and ground isolation of the peripherals. The variety of peripheral units include slow- and medium-speed tape drives (including streamers), low- and high-speed data links, medium to very-high-speed printers, and scanner and signal distributor circuits.*

## I. INTRODUCTION

The 3B20D Input/Output system was planned from conception to be very flexible, and to support a large number of different peripheral devices. To accomplish this, several difficult system and design-level problems had to be solved.

(*i*) Processor real time—Previous experience in system input/output had shown that even low-speed input/output could be very real-time intensive. Terminal input/output, in particular, requires considerable per-character processing. The 3B20D Input/Output system, therefore, had to provide for front-end processing.

(*ii*) Lack of interface standards—Input/output standards at the physical, electrical, and protocol levels for certain peripheral devices, even those of the same functional class, i.e., tapes, floppy disks, and so

on, are relatively nonexistent. Where there are standards, such as Electronics Industries Association's RS 232 (electrical and physical standards for data link and terminal access), they often are interpreted differently by various vendors, cover only a portion of the market for a class of devices, or are subject to periodic revision. The input/output system, therefore, had to provide a standard interface that could easily support a wide range of nonstandard interfaces and power/ground systems.

(*iii*) Variations in operation and maintenance processing—Error handling, as well as operational processing, is significantly different for each peripheral device.

(*iv*) Noninterruptive growth and maintenance—The input/output system had to permit easy addition of new or replacement devices without interrupting service.

The sheer number of various devices available, their mismatch in performance to the central processing unit bus structures, and their often low cost and rapid obsolescence required an extremely flexible and adaptive interface structure.

## II. THE 3B20D INPUT/OUTPUT PROCESSOR

The 3B20D input/output processor resolves the many problems stated above with five major hardware components, and a modular software structure in the host. The hardware components are: duplicated access to each 3B20D half over the two dual-serial channel interfaces; a common high-speed memory access multiplexor, the peripheral interface controller; communities of single-board computers, called peripheral controllers, which provide preprocessing intelligence and specialized device input/output interfaces; the input/output microprocessor interface bus, which provides a common Peripheral Controller interface to the Peripheral Interface Controller; and unit power design structured for flexibility and growth (see Fig. 1).

### *2.1 Circuit partitioning*

The dual-serial channel is described in detail in this issue including the Duplex Serial Bus Selector.[1,2] The duplex serial bus selector acts in concert with the bus interface controller circuit pack to communicate with the peripheral interface controller. The bus interface controller buffers processor data and commands to the peripheral interface controller, as well as data and status information from the peripheral interface controller. It also performs the mandatory "handshaking" to communicate with the duplex serial bus selector. The bus interface controller allows the 16-bit peripheral interface controller to transmit and receive data from the higher-capacity 32-bit duplex serial bus selector at a rate the peripheral interface controller is able to accept.

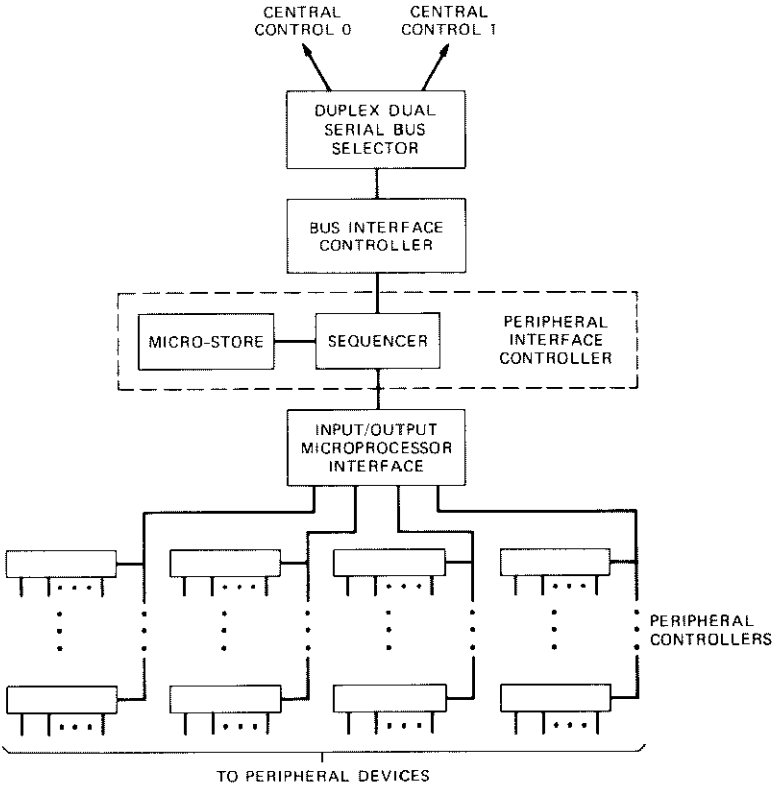



Fig. 1—The 3B20D peripheral interface.

The bus interface controller verifies the integrity of data transferred between the duplex serial bus selector and the peripheral interface controller via per-byte data-parity checks. The bus interface controller uses a 16-word by 32-bit data first in-first out register that can be alternately accessed by the peripheral interface controller and duplex serial bus selector for the transfer of data. The bus interface controller contains a 32-bit command register that records 3B20D processor commands to the peripheral interface controller, and a 16-bit status flag register and a 16-bit error flag register. The status flags are used to request and interrupt direct memory access service from the 3B20D Processor, and inform the peripheral interface controller of 3B20D commands or requests for data transfer. The error flags record the occurrence of errors and aid in fault resolution and recovery. Peripheral interface controller sanity and interval timing functions are also provided by the bus interface controller.

The peripheral interface controller is a simplex, high-speed, bipolar microprocessor. The peripheral interface controller, with its associated

bus interface circuits, handles all common maintenance and operational input/output functions between the various peripheral controllers, and the 3B20D direct memory access.

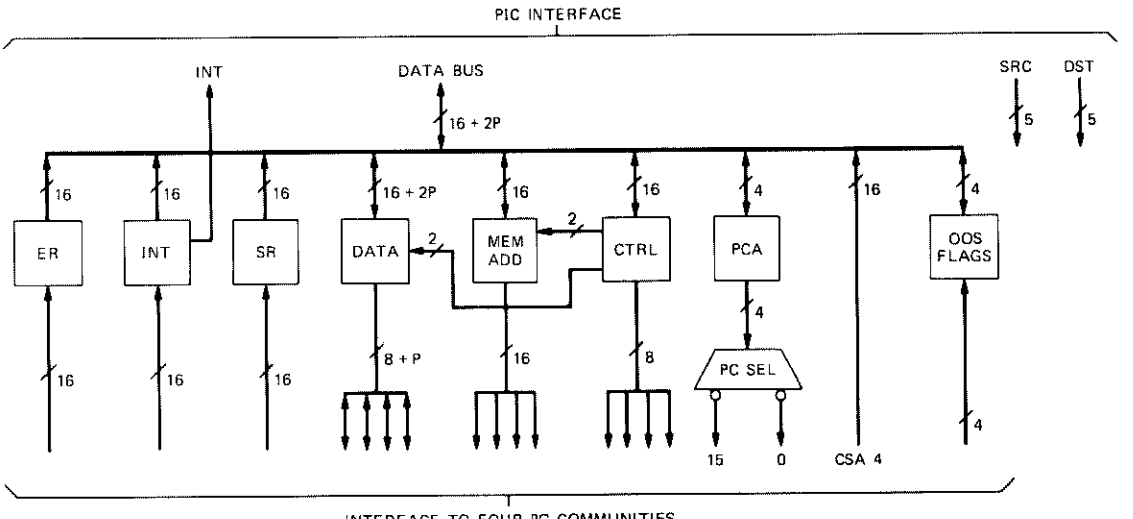
The peripheral interface controller consists of two circuit packs, a controller and one 8K-word micro control memory. The peripheral interface controller contains an arithmetic and logic unit, a 16-word general register file, a 4K-word data store, an 8-level interrupt controller, a microcontroller sequencer, and a pipe-line register. The 8K-by 40-bit micro control memory pack stores the peripheral interface controller operational and diagnostic firmware.

The input/output microprocessor interface circuit serves to interface the 16-bit peripheral interface controller with up to four communities of four peripheral controllers each. The interface between the input/output microprocessor interface and each community consists of a 16-bit memory address, an 8-data plus 1-parity-bit bidirectional data bus, and eight control pulses. Figure 2 details the input/output microprocessor interface bus structure. In addition, a private peripheral controller select signal is connected to each of the 16 peripheral controllers. The selected peripheral controller acknowledges the receipt of a control pulse by setting its control signal acknowledge. The 16 control signal acknowledge bits are compared to the 16 peripheral controller select signals to indicate to the peripheral interface controller when the acknowledgment has been received.

Each peripheral controller also has three request leads that are employed to report errors or to request service. Each peripheral controller request lead is assigned a bit in each of three 16-bit request registers: peripheral controller error, interrupt, and service request. A summary bit is provided over the interrupt register contents to trigger a peripheral interface controller interrupt whenever any peripheral controller interrupts are set. The peripheral interface controller can read each of the request registers to resolve requests to the peripheral controller level.

The four fanout branches of data, address, and control signals (one per community) are driven from three common registers. The 16-bit address register is implemented using binary up/down counters to allow autoincrement and autodecrement capability. The mode of operation is specified by bits in the control signal register. Special input/output microprocessor interface circuitry has been provided to guarantee address setup and hold times of the peripheral controller memory read and write control signals.

The peripheral interface controller data register is segmented by the input/output microprocessor interface into high and low bytes to be shipped a byte at a time to and from the 8-data and 1-parity-bit peripheral controller data buses. A flag in the control register selects




ER – ERROR REGISTER
INT – INTERRUPT REGISTER
MEM ADD – MEMORY ADDRESS
P – PARITY
PC – PERIPHERAL CONTROLLER
PIC – PERIPHERAL INTERFACE CONTROLLER
SR – SERVICE REQUEST REGISTER

Fig. 2—Microprocessor interface.

the high or low byte as the source or destination of the data transfer between the input/output microprocessor interface and the selected peripheral controller. An additional bit in the control register is used to toggle the data byte selector flag after a peripheral controller memory read or write cycle. The circuit incorporates a delay timing chain to guarantee data hold time during peripheral controller memory write operations.

The control register is partitioned into two fields; an 8-bit register, the outputs of which directly drive the control signal buses; and an 8-bit field that controls internal functions in the input/output microprocessor interface.

An out-of-service bit is provided for each of the four peripheral controller communities. This 4-bit register in the input/output microprocessor interface can be read and written by the peripheral interface controller. Additionally, a power-down condition within a community will automatically set its out-of-service flag. When out of service, the interrupt requests of the four peripheral controllers within that community are not ORed into the interrupt register summary.

### *2.2 Peripheral controllers*

The peripheral controller is a microcomputer system that serves as an intelligent interface between the peripheral interface controller and the slow- to medium-speed peripheral units. The peripheral controller is responsible for all device-specific front-end processing and per-byte interrupt handling. This relieves the 3B20D central processing unit from low-level tasks and greatly increases the number of peripheral devices the 3B20D can support. Figure 3 shows a typical one-board peripheral controller design. Up to four subdevices can be separately addressed within a peripheral controller by the peripheral interface controller.

#### *2.2.1 Peripheral interface controller-to-peripheral controller communications*

Subdevices are capable of initiating several types of work requests. A peripheral controller may initiate a service request that results in the peripheral interface controller reading service-request status information to determine the type of work required. A subdevice activates a service request to transfer data to or from 3B20D main memory or to indicate a job is completed. In addition to service requests, a subdevice may at a high priority level, initiate an interrupt request that results in interrupt-request status information being read by the peripheral interface controller. The peripheral controller interrupt request is activated by a subdevice to transfer high-priority informa-

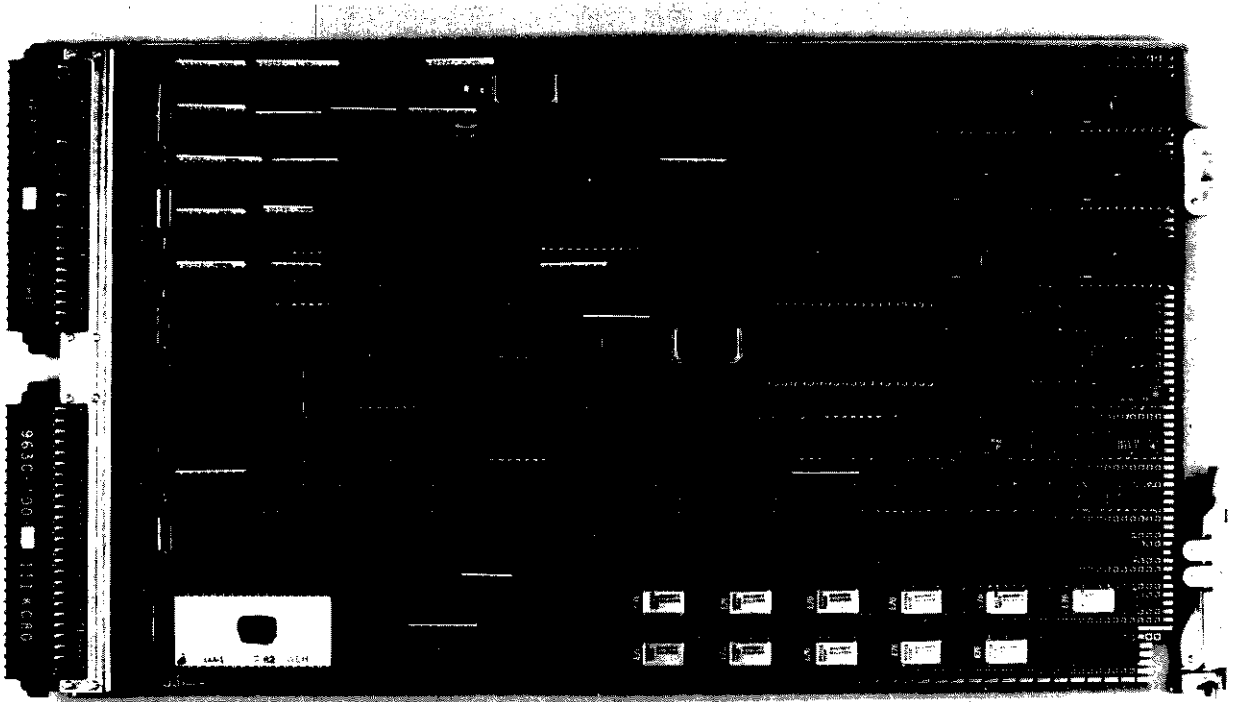


Fig. 3—Typical one-board peripheral controller design.

tion to or from 3B20D main memory or to report a high-priority job has completed.

All operational communication between the peripheral interface controller and the peripheral controller is done through a dual-access memory resident on each peripheral controller. Dedicated memory locations are reserved within the dual access memory that contain pointers used by the peripheral interface controller to locate each subdevice data buffers, data transfer parameters, and status and command areas. The dual access memory also contains memory locations that contain information at the peripheral controller level such as service and interrupt request data and peripheral controller command area pointers.

Each of the peripheral controllers is an independent single-board microcomputer. The hardware is specialized to handle the particular interface and is not necessarily the same as any other peripheral controller. However, all peripheral controllers have a functionally common front-end design and maintain the same communication protocol with the peripheral interface controller. The input/output processor has been designed to allow the peripheral controllers to exist as autonomously as possible. This capability will enable future peripheral controller designers to use state of the art microprocessors and microprocessor-controlled peripherals.

#### *2.2.2 Peripheral controller hardware architecture*

Although peripheral controllers designed for specific interfaces are different, all peripheral controllers have the following components: microprocessor, read-only memory, random access memory, service and interrupt sources, isolation circuitry, error detection logic, and scan back circuitry.

The peripheral controller employs a microprocessor to administer the transfer of data between its peripheral units and 3B20D main memory. There are no requirements dictating the use of a specific microprocessor type. The intent is to allow a peripheral controller design to take advantage of the microprocessor that most efficiently handles a particular peripheral unit.

The peripheral controller houses a bootstrap program stored in read-only memory that is entered when power is initially applied or when the peripheral controller reset function is activated by the peripheral interface controller. The bootstrap program causes the microprocessor to initialize its periphery and to recognize peripheral interface controller commands. In addition to the bootstrap program, the read-only memory may contain operational and diagnostic programs.

The peripheral controller uses random access memory to store the operational program, and buffer transient data, and to provide storage

for peripheral interface controller communication parameters. The portion of random access memory containing the data buffers and communication parameters is accessed by both the peripheral interface controller and peripheral controller microprocessor. Therefore, this portion of random access memory must provide dual-access capability.

To prevent an insane microprocessor condition from babbling on the common bus, the peripheral controller is provided with an isolation state. In this state, isolation circuitry removes all response signals generated by the associated peripheral controller from the bus. In some peripheral controller applications the isolation state also prevents data transmission to the peripheral units connected to the peripheral controller.

#### *2.2.3 Service and interrupt request generation*

The peripheral controller generates a service request or an interrupt request whenever a data transfer to or from 3B20D main memory is required, or to indicate that a command completion response is available. The peripheral controller provides a separate pulse source for the interrupt request and service request indication. The peripheral controller must refrain from sending a service request or interrupt request until the previous one has been recognized by the peripheral interface controller. The peripheral interface controller indicates recognition of a service request or interrupt request by clearing the associated request pending flag in the peripheral controller dual-access memory.

Each subdevice within the peripheral controller is assigned 4 bytes of dual-access memory, which contain the status information for the associated subdevice. The status information is comprised of generic and user-defined data, and is read as a result of subdevice initiated service requests and interrupt requests.

#### *2.2.4 Hardware error detection*

The peripheral controller provides parity checking and generation for data contained in the dual-access memory. Since most microprocessor and microprocessor peripherals available today do not carry parity within the device, routine diagnostics must take over the responsibility of error detection for these devices. Parity over read-only memory data is not a requirement. This enables peripheral controller designs to take advantage of the available large, 8-bit-wide, read-only memories. However, a program sanity check must be provided to detect a read-only memory data failure. A parity error flip/flop is used to store the indication of a dual-access memory parity failure. The peripheral controller also provides control logic to invert parity generation in order to check the parity circuits.

Peripheral controllers operate autonomously with regard to peripheral interface controller activity and, therefore, provide their own system clock. A system clock check circuit is provided by the peripheral controller to detect clock activity abnormalities. A clock error flip-flop is used to store the clock failure indication. The error lead is employed to report the detection of such failures.

Due to the complexity of the peripheral controller and the limited amount of parity checking provided, the peripheral controller uses routine maintenance diagnostics to enhance error-detection capabilities. Redundant software, rather than redundant hardware, is employed to detect fault conditions. The peripheral controller microprocessor periodically executes routine maintenance diagnostics during nonpeak data transmission time intervals. A routine maintenance diagnostics error flip-flop is provided to store the routine maintenance diagnostics failure.

To assure program execution sanity, the peripheral controller provides a sanity check mechanism. The sanity check complexity is directly proportional to the destructive powers of an insane peripheral controller microprocessor. A sanity failure flip-flop is used to store the sanity failure indication.

#### *2.2.5 Scan back circuitry*

The peripheral controller provides four directly addressable read-only functions: status and error information, address lower loop-around data, address upper loop-around data, and peripheral controller type-identity code.

The peripheral interface controller uses the scan back address data in a loop-around mode to check the integrity of the direct memory access address and data buses and to verify that a particular peripheral controller is receiving all the address information.

## III. PERIPHERAL INTERFACE CONTROLLER AND PERIPHERAL CONTROLLER FIRMWARE AND SOFTWARE

### *3.1 Peripheral interface controller firmware*

The functions performed by the peripheral interface controller are carried out at two levels: base level and interrupt level. Deferrable tasks are performed during base level processing and nondeferrable at interrupt level (see Fig. 4).

Peripheral controller service requests are handled during the peripheral interface controller base level loop. The peripheral interface controller interrogates the peripheral controller service request register to ascertain which peripheral controllers are requesting service. The peripheral interface controller then reads the peripheral controller status bytes in dual-access memory to determine the type of job to be

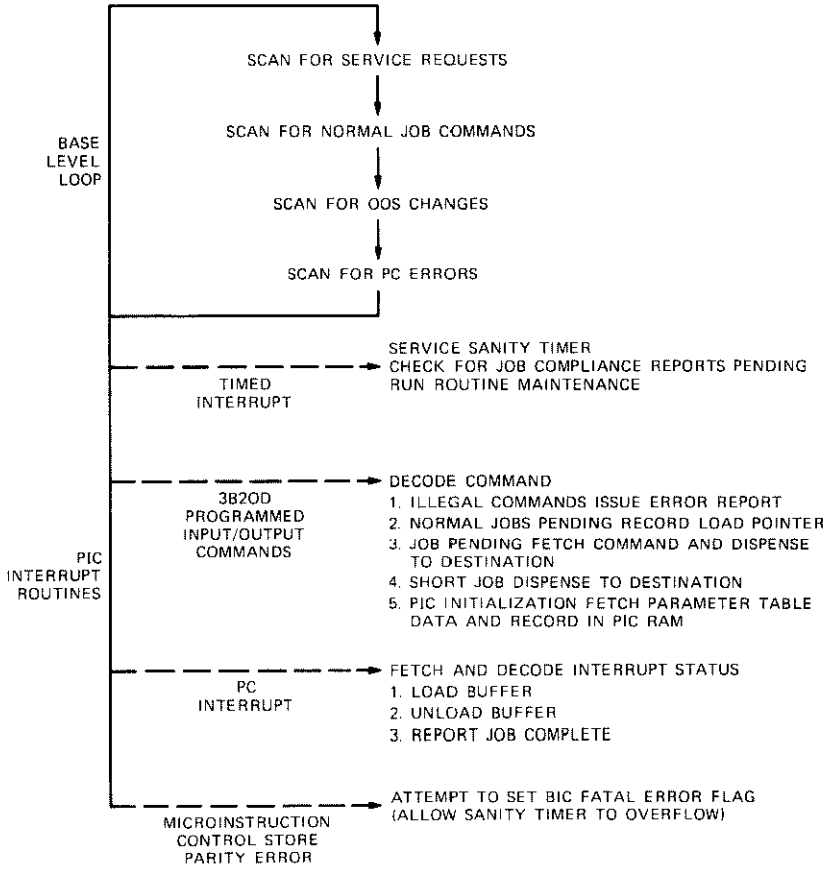



Fig. 4—Peripheral interface controller firmware structure.

performed and executes that job. Job sizes are restricted to a maximum transfer size of 256 bytes to guarantee the maximum time the peripheral controller is held out of dual-access memory. The peripheral controller breaks the transfer of a large block into small blocks to satisfy this requirement. Reports of completed jobs are buffered in an area of the peripheral interface controller data store.

After servicing peripheral controller requests, the peripheral interface controller compares the load and unload pointers in a job queue stored in 3B20D main memory. If the queue is not empty, the peripheral interface controller unloads up to eight jobs and issues them to the appropriate peripheral controllers or to the peripheral interface controller itself.

The peripheral interface controller next scans for changes in the peripheral controller community out-of-service status. A power con-

verter out of tolerance causes an out-of-service indicator to be automatically set in the input/output microprocessor interface. A high-priority error response is issued to report such a condition.

The peripheral interface controller next scans for peripheral controller error reports. Again, the peripheral interface controller reports this via a high-priority error response. The loop is complete at this point and the peripheral interface controller begins scanning for peripheral controller service requests once again.

Interrupts are employed to initiate time-critical tasks. A timed interrupt routine is entered at a fixed rate. The sanity timer is serviced at this time. Additionally, the job-completion queue is interrogated and, if not empty, is loaded into the response buffer in 3B20D main memory. The peripheral interface controller then interrupts the 3B20D.

High-priority jobs are handled using the interrupt mechanism. Upon registration of the message in the high-priority buffer in the peripheral controller dual-access memory, the peripheral controller interrupts the peripheral interface controller. The peripheral interface controller moves the buffer contents into 3B20D main memory. This process continues until the entire message has been transferred into 3B20D main memory. The peripheral controller then enters a job-completion report into the appropriate entry in the peripheral controller dual-access memory and again interrupts the peripheral interface controller. The peripheral interface controller then moves the report into the high-priority response buffer in 3B20D main memory and interrupts the 3B20D.

Programmed input/output commands arriving from the 3B20D also trigger a peripheral interface controller interrupt. The peripheral interface controller decodes the command to determine its type. New entries in the 3B20D main memory job queue are reported to the peripheral interface controller in this manner. A portion of the command contains the 3B20D load pointer for the job queue. The peripheral interface controller records this entry into the peripheral interface controller random access memory to be used during base level processing.

The 3B20D also uses programmed input/output commands to alert the peripheral interface controller of the presence of a high-priority job awaiting execution in the 3B20D main memory high-priority job register. The peripheral interface controller retrieves the command and immediately issues it to the specified peripheral controller.

Additionally, short jobs can be transferred to the peripheral interface controller via programmed input/output. Jobs of this sort are constrained to contain no more than 28 bits of command information. Only a single 32-bit programmed input/output command is used.

### 3.2 Peripheral controller software

#### 3.2.1 Initialization

Peripheral controller initialization is initiated by a 3B20D Processor resident input/output processor fault recovery program. The fault recovery program can direct the peripheral interface controller, via a configuration command, to generate a clear signal for any one of the 16 addressable peripheral controllers within the input/output processor. The peripheral controller responds to the clear signal by transferring control to a bootstrap program resident in on-board read-only memory. Execution of the bootstrap program results in initialization of address pointers, flags, timers, interrupt control circuitry, and other parameters as required. In addition to the above mentioned items, the peripheral controller must initialize associated work pointers. On-board peripheral controller diagnostics are downloaded from the central processing unit during this stage.

Upon completion of initialization the peripheral controller microprocessor monitors sanity and responds to any commands placed in the peripheral controller work queues.

#### 3.2.2 Peripheral interface controller/peripheral controller microprocessor command execution

A peripheral controller microprocessor can be directed to perform certain tasks through the use of a command queue system. A peripheral controller can have as many as 15 queues, three of which are for generic type commands. The remaining queues are double ended (deques) and provide communication to peripheral controller subdevices. Each of the four peripheral controller subdevices has a transmit, receive, and asychronous report deque. Jobs are sent to the transmit and receive deques informing the peripheral controller subdevice to send and receive data respectively. Unlike the transmit and receive deques the asychronous report deque only sends subdevice reports, it never receives jobs for execution. The command queues and their associated pointers—load, unload, and current—reside in the peripheral controller dual-access memory. The peripheral interface controller, after entering a command in the queue, informs the peripheral controller by updating the queue load pointer. The peripheral controller detects the entry, processes the job, and inserts a completion report in the queue. The completion report includes a 2-byte completion code. The peripheral controller receives knowledge of the completion report via a service request. Retrieving the report, the peripheral interface controller sends it to 3B20D main memory and updates the unload pointer. The peripheral controller maintains a current job pointer, which allows the peripheral interface controller knowledge of which peripheral controller job is currently running.

A unique completion code has been reserved as an indication of an invalid command. Upon detection of this condition, the microprocessor immediately loads the invalid-command completion code byte into the appropriate field in the command/response entry and signals a job-completion report via a service request or interrupt request. At this point, the microprocessor is ready to accept the next entry in this command deque.

For high-priority command entries, the microprocessor may set an interrupt request indicator instead of setting a command completion service request indicator. The peripheral interface controller recognizes and responds to all peripheral controller interrupt request work before responding to peripheral controller service request work.

#### *3.2.3 Peripheral controller microprocessor base level and interrupt level work*

There are three modes of communication between the peripheral controller microprocessor, subdevices, and peripheral interface controller:

(*i*) Demand interrupt mode—The peripheral interface controller interrupts the peripheral controller microprocessor to indicate a high-priority command has been placed in the high-priority work deque. The peripheral controller microprocessor immediately responds to the command interrupt by accepting the command and executing the associated code as required. The subdevice may interrupt the peripheral controller microprocessor, indicating receive data is present. In this mode the microprocessor checks the data for errors and, if none are found, moves the data to an associated input buffer. When the input-buffer-full threshold is reached, the microprocessor uses a command address pointer to locate a receive command. The receive command contents are required to permit the peripheral controller microprocessor to initiate a direct memory access transfer of data.

(*ii*) Base level polling mode—In the base level program, all subdevices are polled in sequence for status and any required processing is performed. The status of all currently active direct memory access transfers is updated and new direct memory access jobs are initiated as required. Typically, all subdevices transmit data, and low-priority maintenance work is done during the base level loop.

(*iii*) Time interrupt mode—A real-time clock provides periodic interrupts to the peripheral controller microprocessor for the purpose of handling low-level periodic functions. Some examples of the functions processed in this mode are: checking subdevice status, maintenance of a sanity timer, and administration of routine maintenance diagnostics.

### *3.3 Peripheral interface controller firmware development tools*

Available for development of peripheral interface controller firmware are (*i*) the Read-Only Memory Emulator and Trace Control Unit circuit packs and (*ii*) the Read-Only Memory Emulator and Trace System software system. These function jointly to support the peripheral interface controller as a test tool during hardware and firmware debugging. The read-only memory emulator and trace system has hardware control over the peripheral interface controller and provides program single-stepping and breakpointing capabilities. Program memory can be examined, modified, and downloaded, via a commercial microprocessor based system. A trace system is provided with three 16-bit by 256-word trace memories.

## IV. INPUT/OUTPUT PROCESSOR UNIT AND POWER

The input/output processor unit and power is structured around the peripheral interface controller and the four communities of peripheral controllers as separate items. Each is located within its own housing and is separately powered. An overall power control switch is provided with the peripheral interface controller; and each peripheral controller community may be individually switched without affecting the others (see Fig. 5). Although several standard voltages are supplied in the backplane, other voltages can be supplied as part of the peripheral controller design, if required.

Since several of the problems of interfacing to various peripherals are related to the wide range of electrical and physical interfaces required, the input/output microprocessor interface bus connector field was separated from the peripheral device defined connector area for each peripheral controller slot. The entire upper half of each peripheral controller's backplane connector was reserved for definition by the particular application. Standard Bell System connectors are then used to generate a cable or harness that performs the translation to the specific connector of each physical device. The foundation peripheral controller for No. 5 ESS and the programmable link controller for the signal transfer point system used in the Common Channel Interoffice Signaling network are two examples of application peripheral controllers made possible by the flexibility of the input/output processor. Multiple board peripheral controllers have also been designed for special applications using connectorized straps in the upper connector area to interconnect the circuit packs that comprise the multiple board peripheral controller. Circuit pack slots are thus system defined, depending upon the peripherals needed by each application.

Because all four peripheral controller communities are on separate

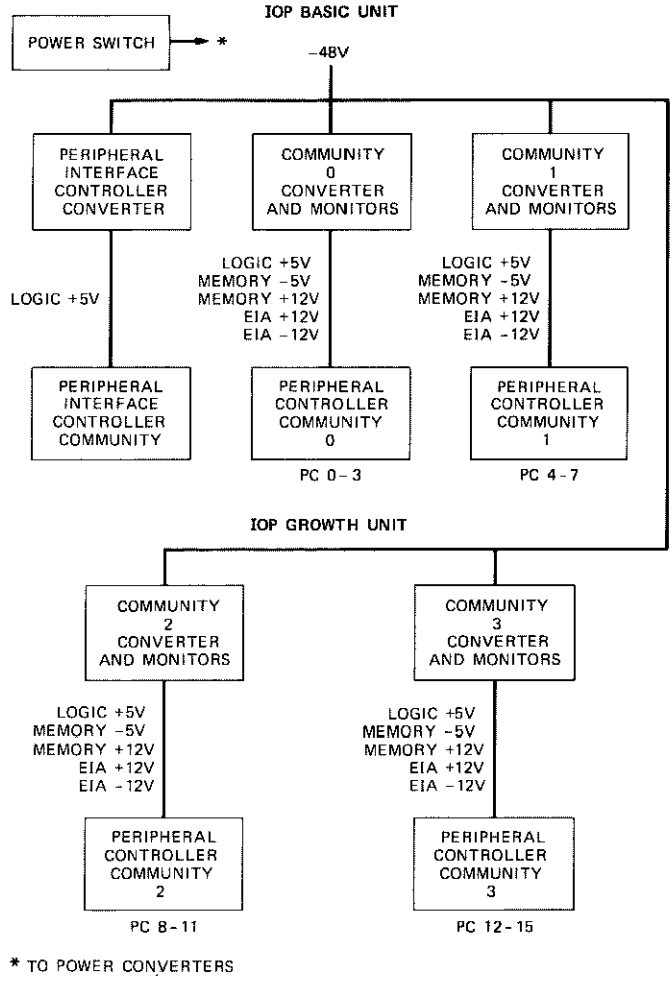



Fig. 5—Input/output processor power diagram.

power systems, peripheral controllers with special grounding or power requirements can be assigned to separate peripheral controller communities, effectively isolating them from the remaining communities, and providing separate power control.

## V. 3B20D PERIPHERALS CURRENTLY SUPPORTED

The peripheral controllers and peripheral devices available on the 3B20D Processor are discussed in the following paragraphs.

### *5.1 Nine-track magnetic tape controller*

Up to four 25-inches-per-second (IPS), 1600-bits-per-inch (BPI), phase-encoded (PE) nine-track tape transports can be accommodated per peripheral controller (only one transport may be accessed at a time via a shared formatter). The drive accepts up to 10.5-inch-diameter reels (2400 feet by 0.5-inch tapes). The interface is an RS422 differential bus up to 250 feet long using an industry standard data and control format. The controller can read and write any block size up to 2K bytes long, write file marks, seek forward or reverse, and can rewrite and reread blocks of data that exhibit errors. The instantaneous data rate is 40K bytes per second. The average data transfer rate for 2K data blocks is 26K bytes per second.

### *5.2 Scanner-signal distributor controller*

Forty-eight scan points and 32 signal distributor points—which may be located up to 1000 feet away (assuming 26-gauge twisted pair)—are provided by this controller. Scanning is carried on autonomously or it is directed. A change of a scan state must remain in the new state for two consecutive scan cycles before it is recognized. The signal distributor can operate or release a point, or it can flash a point indefinitely or for a timed period. Both the scanner and the signal distributor are self protected to 140 volts, and can detect a foreign potential in the same range. A point scanned must be a floating contact or saturated transistor switch. Points are scanned once every 48 ms. The signal distributor can supply up to 5 mA into a floating load or loop. The scanner signal distributor was designed primarily to operate optical couplers and low-power relays. Scan or distribute points that are remote to the 3B20D Processor are interfaced through an optical coupler isolation circuit.

### *5.3 High speed nine-track magnetic tape controller*

Up to four 12.5- to 125-IPS, 1600-BPI, phase encoded, start-stop transports, or up to four 25-IPS streaming transports with embedded formatter, in any mix, may be accommodated per peripheral controller. Access is via an RS 322 bus limited to 20 feet using an industry standard data and control format. Only one transport may be accessed at a time. The controller can write or read any block size up to 6K bytes as well as perform the same control functions as the peripheral controller described in 5.1. The instantaneous data transfer rate is 200K bytes per second. The average data transfer rate when reading 6K blocks of data is 150K bytes per second.

### *5.4 Synchronous data link peripheral controller*

This circuit pack is a BX.25 level-2 synchronous link arranged for full duplex private line or dial backup operation, and it provides two independent channels at bit rates up to 9.6K b/s. The channels differ in that one supplies an automatic call unit port for dial-up and dial-in operations. The backup configuration is supported when equipped with Dataphone II generation data sets. The peripheral controller capacity is 9600 b/s full duplex. Each of the two BX.25 level-2 channels has an associated level-1 interface that is RS449/RS232C compatible.

### *5.5 Asynchronous data link peripheral controller*

This peripheral controller provides two independently programmable full-duplex or half-duplex, asynchronous or isochronous channels at data rates up to 9.6K b/s. It can communicate with a local terminal directly or via a "null modem" connection, with a remote terminal via a modem and private line, or with a remote terminal via a modem and the public switched telephone facility. The user can specify the data rate, parity and stop code format in addition to input, output, and line discipline processing options. Each channel drives an optically isolated EIA RS232C standard port.

### *5.6 Maintenance teletypewriter peripheral controller*

This peripheral controller is designed to permit manual intervention and control of the 3B20D control unit and peripherals through the maintenance terminal. It contains sufficient code in read-only memory to allow control of the system during routine or emergency maintenance procedures. Read-only memory code performs memory and device initialization, parsing of the 3B20D emergency action interface input and support for the maintenance terminal display and a remote maintenance center data link using BX.25 protocol. Down-loaded random access memory code provides additional functions such as support for teleterminal input/output syntax and other teleterminal display capabilities.

### *5.7 BX.25 data link interface*

The data link controller provides a general-purpose high-speed synchronous BX.25 level-2 interface and a special high-capacity software interface to the 3B20D. The peripheral controller is equipped with an RS232C and a CCITT V.35 interface having data rates of up to 56K b/s.

## VI. SUMMARY

The 3B20D Processor's input/output structure was designed around a high-speed direct memory access input/output channel, a high speed

multiplexor and direct memory access interface, and single- and multiple-board intelligent controllers, which provide device-specific front-end processing and interrupt handling. These components, associated with a common driver with device-specific handlers in the 3B20D Processor's operating system, successfully provide a flexible and effective interface to a wide range of peripheral devices.

## VII. ACKNOWLEDGMENTS

The 3B20D Input/Output Processor and its peripherals are the result of a team effort of circuit, physical, and software design groups. The authors wish to acknowledge T. S. Greenwood, J. O. Becker, and R. T. Watters for their management support of the project, and the particular design efforts of T. T. Butler, D. J. Brahm, W. V. Lindquist, N. E. Stohs, J. L. Snapp, T. T. Towle, S. P. Michiels, M. G. Bradac, S. P. Davison, W. A. Hommowun, W. R. Hudgins, and R. M. Venzon, C. A. Mennitt, C. B. Kelley, W. G. Reichert, R. Huisman, all of whom helped to make the concepts become a reality.

## REFERENCES

1. W. N. Toy and L. E. Gallaher, "Overview and Architecture of the 3B20D Processor," B.S.T.J., this issue.
2. M. W. Rolund, J. T. Beckett, and D. A. Harms, "3B20D Central Processing Unit," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# Software Development System

By B. R. ROWLAND and R. J. WELSCH

(Manuscript received March 22, 1982)

*The 3B20D Processor software development system is an integrated collection of tools and procedures that is used in the development and administration of all 3B20D Processor software. This article describes the tools and procedures and their use in developing the 3B20D Processor software. These tools and procedures include compilers, assemblers, and loaders, as well as change-administration and load-building procedures. The most important characteristic of the development system is the balance between the enforcement of the project standards and the flexibility offered to developers.*

## I. INTRODUCTION

The software that comprises the operating system, diagnostics and fault recovery, configuration data base, field utilities, and craft interface for the 3B20D Processor has been undergoing development and change over several hundred developer years. Without a strict change-administration strategy and a productive development environment, the tight schedules and reliable deliveries characterizing this system would not have been possible.

This paper presents a description of the software-development environment used by the project's programmers and those administrating 3B20D Processor subsystems. Without the administrative control, these developers might otherwise be limited to simply compiling or assembling programs and attempting to test software with an ad hoc version of the total system software. The emphasis presented here is that of the roles of those involved in the software change process, the tools they each use, the flexibility offered by the tools and administration structure, and the standards and strategies enforced in the development environment.

## II. THE ADMINISTRATION OF SOFTWARE DEVELOPMENT

The 3B20D Processor project started with relatively small teams of engineers, compilation tools, and a primitive testing environment. The project has grown to a size that now requires considerable machine support, sophisticated tools, and administrative control. Over 100 developers are responsible for nearly 8,000 source files resident across eight support computers. Compilation of all of these files into a full load takes over 24 hours of machine time on a commercial 16-bit minicomputer. This large amount of software has been partitioned into 23 logically manageable subsystems, the basis for administration and testing.

The 3B20D Processor is the target for output from a software development and administrative environment that resides on the host support computers. All object code for the target is archived, edited, and cross-compiled on the host and then transported by tape or data link for testing on target 3B20D Processors. This separation of host and target ensures a stable development environment as the target machine evolves with new features and performance improvements that may consist of changes in hardware, software, or firmware.

The software administration and development strategy can best be described as one of well-defined and closely tracked data movement between nodes on development host machines and it requires a specific scenario dictating when the data movement takes place. The data are requests for software changes and program files, and the nodes are instances of the software structure reflecting some portion of the total software in one of several development states. Development activity is spread across the host machines by partitioning the software into nearly independent subsystems.

In the development environment, a file system structure containing copies of controlled software source and generated object is referred to as a *node*. When a node is populated with source, object, and target products—all in a like state of development or approval—the node is called a *view* of the software. For each major release, the development system supports three views of the software.

(*i*) The *official* view, or node, contains all of the official source, object, and target products released to customers.

(*ii*) The *approved* view contains those source, object, and target products that have been released as emergency corrections to the last major release. The approved view is an incremental addition to the official view.

(*iii*) The *under-test* view contains those source, object, and target products that will be sent out with the next major release. The under-test view contains everything from the approved view and is otherwise an increment from the combination of the official and approved views.

A final view, that of the developer, may include a mixture of the above views in combination with privately modified versions of various files undergoing a development change.

The approval of a software change submitted by a developer involves the successful integration of that change into a known stable software base. The data movement and approval strategies have a common thread: the incremental modification of a known good software base to produce the next iteration of that base.

An important decision made in the design of the 3B20D Processor development environment was the centralization of the activities used to construct the collection of libraries and objects for installation (i.e., the *load*) on a machine isolated from development activity. This allowed efficient load-building techniques to be developed that do not interfere with developer activity. The implication of centralized load building is that developer changes must be collected and moved to a common point where official load building takes place. In fact, it is only source changes that are moved for load-building purposes. All changes to products are then reconstructed from changed source. Results are then redistributed back to the appropriate development machines.

## III. MAJOR SOFTWARE DEVELOPMENT SYSTEM COMPONENTS

It is the interaction of four distinct software administration and generation systems that creates the 3B20D Processor Software Development System (SDS). This section presents an overview of these systems; Section IV describes their use; and Section V describes the major standards that contribute to the environment and are enforced by the project's tools.

### *3.1 The Modification Request System*

The Modification Request (MR) data base system is a general-purpose hierarchical system tailored to 3B20D Processor change-tracking requirements. The MR is the entity that identifies all requests for 3B20D Processor software changes and tags all source changes made by developers and administrators to official 3B20D Processor software. The MR system serves as a central master data base from which administrators control and track all software changes and generate appropriate tracking and status reports.

As implemented for this project, the MR data base is actually a three-level data base comprised of an MR level, one or more release levels per MR, and one or more subsystem levels per release. The higher the level (the MR level is highest), the more global the scope of the information maintained.

The MR level information includes an identification number, origin-

ator information, problem description, priority, severity, and due date. At the release level, the data base includes due date, feature engineer and developer, and priority. Subsystem level data includes responsible developer, modified source files, status, solution description, and load-building instructions. The majority of the information is automatically generated and stored in response to developer and administrator actions. For the most part, direct data entry is not necessary.

The MR system includes a high-level query language and report-formatting facilities for producing reports from the data base. The query language is very powerful in that it allows for selection specification and sorting levels. The reporting facilities process the output of the query language. Additional capabilities exist that cause brief reports, called synopsis reports, to be generated automatically and sent to appropriate administrators, supervisors, and/or developers in response to particular status changes. For example, when an MR requires a change in a particular subsystem, the responsible developer receives a synopsis report. When the responsible developer submits changes in response to an MR for system test, appropriate administrative personnel receive synopsis reports.

### *3.2 The Change Management System*

The Change Management System (CMS) is a facility compatible with the *UNIX** operating system aimed at controlling the activities of both developers and administrators related to software change and change approval. The CMS uses the *UNIX* operating system Source Code Control System (SCCS) and its own relational data base to track and relate each change made in a source file to an MR.[1]

All official 3B20D Processor source files are maintained via CMS on a per-subsystem basis. A CMS instance is defined to be a set of SCCS-controlled source files and the relational data base tracking the MR-based changes. Each subsystem is a unique instance of CMS. There are, therefore, one or more CMS instances on each development machine (see Fig. 1). This strategy was chosen over a strategy having only one CMS instance per machine for the following reasons:

(*i*) Corruption of a CMS data base would only affect one subsystem.

(*ii*) The majority of the developers need access to the official source files for only one subsystem.

(*iii*) Separate CMS instances per subsystem allow load balancing on the development machines to be more easily implemented. Instead of having to extract a portion of a large data base and move it with developers to another machine, an entire CMS data base can be moved.

---

* Trademark of Bell Laboratories.

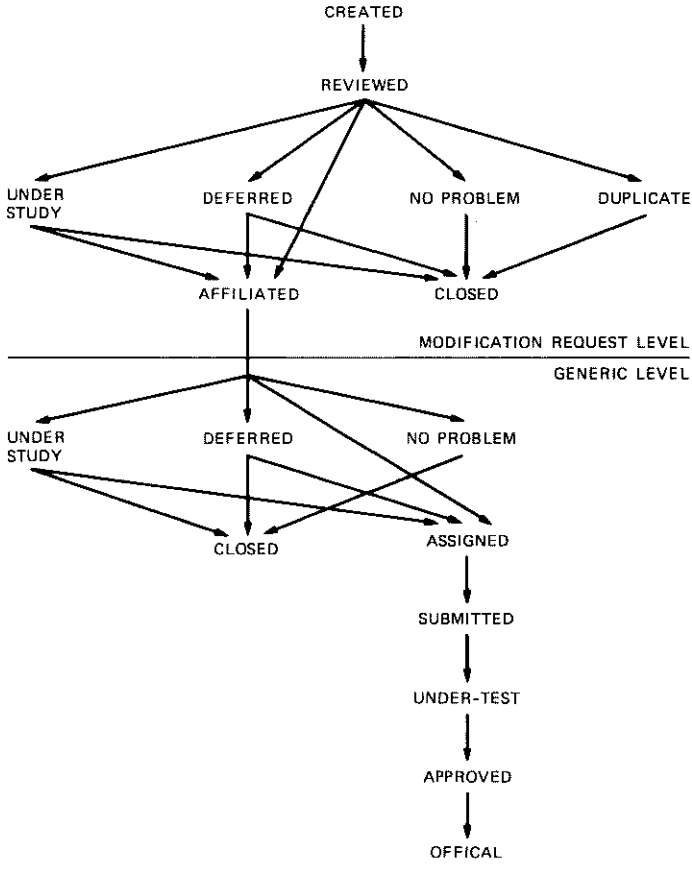



Fig. 1—Diagram of the state changes in the Modification Request System.

### *3.3 The Software Generation System*

The software development system is geared to construct executable software for the following three target environments:

(*i*) A 3B20D Processor running the DMERT operating system.[2] This is the primary target for official system software construction.

(*ii*) A PDP 11/70 running a real-time version of the *UNIX* operating system. This is the recognized 3B20D test laboratory support processor.

(*iii*) A PDP 11/70 running the *UNIX* operating system. This is the primary machine and operating system for development and official software construction.

The primary target for which the SDS generates code is the 3B20D Processor. A Software Generation System (SGS) containing a compiler, assembler, and link editor is provided for generating and sup-

porting 3B20D Processor binary executable object modules on the development and load-building machines.

### *3.4 Building software views*

The major tool used by both developers and administrators in conjunction with the SGS for generating 3B20D Processor loads is the build tool. This tool invokes the minimum number of commands necessary to create a requested object incorporating selectable states of related software from its components. It determines, according to a specified set of dependency information, exactly which objects must be rebuilt (because they may be out of date) and which objects are current with respect to the changes desired in the requested object to avoid unnecessary compilation steps. This dependency information is automatically generated by a utility from the software components. This ability to create an object with particular versions of software that reflect its state or submitted MR software changes is known as *building a software view.*

## IV. THE SOFTWARE DEVELOPMENT PROCESS

All software in the DMERT environment is created, corrected, or enhanced in conjunction with an MR. With this association of an MR to every software introduction or change, all development on the project can be adequately monitored. The assignment of an MR allows a developer to make modifications to official project source and to ultimately submit the changes for approval by project administrators. The development activities that occur and the tools used in the process between the assignment of an MR and the submission of changes are described in this section.

### *4.1 MR handling*

The MR system maintains modification requests in a variety of states (see Fig. 1) that reflect the activity being taken on the request. An MR is *created* by other developers or applications and then initially *reviewed* by administrators. It may be found that an MR is a *duplicate* of a previously resolved MR; it may be required to be placed *under study*; it can be *deferred* for later consideration; or it may be found to be *no problem* in which case the MR is ultimately *closed*. The under study or deferred MRs eventually will be either *closed* or *affiliated* with generics for action. Once the MR level state is affiliated, then generic level action is allowed.

MR assignment is made to one or more subsystems, as many as are required to implement a solution. When an MR is assigned, a developer then creates a development node in which to maintain copies of source to be created or modified in response to the modification request and

to build a view of the subsystem affected by the MR. This private development node gives the developer the freedom to make any changes with any set of tools on local copies of source files. The developer is now free to experiment with solutions to satisfy the MR, while official source remains protected by CMS. Source files intended to be modified by the developer for the MR are requested via CMS commands that associate the changes with the MR and the release to which the MR is being applied. The association of MR to source files changed is kept in the CMS data base to allow administrators or other developers to request versions of software containing solutions to particular problems. While a developer has copies of official source to be worked on in a private node, all other developers are prevented by CMS from obtaining copies of the same source to avoid creating unsynchronized changes. Using private source copies, the developer is ready to make changes and create new subsystem objects for testing.

#### *4.2 Languages and tools*

Software for the 3B20D Processor target is written primarily in the high-level programming language C with occasional use of the 3B20D Processor assembly language. The C language contains many modern control and data structures found in languages such as PASCAL. It is characterized by its brevity of expression, direct access to data type representation, and operations and declarations available to the programmer to enhance the generation of efficient assembly level code. The 3B20D Processor assembly language is a member of IS25, a 3B20D family standard specifying:

(*i*) Activation stack format
(*ii*) Data type representations
(*iii*) Registers
(*iv*) Operations and addressing modes for accessing and manipulating data objects ranging in size from a single bit to 32-bit words.

The tools used to compile C programs and assemble assembly code into user level objects are modeled after the tools used with the *UNIX* operating system and for the development of the *BELLMAC**-8 microprocessor.[3] An additional complement of tools is used to create and modify special process files and prepare the developer with information that will be very valuable in the testing environment.

A single command can be used to control the C program compilation process. This command invokes a source code preprocessor, the C compiler and optimizer, assembler, and link editor. These four tools in turn convert a collection of C programs into a single object with addresses that are either relocatable or absolute (Fig. 2).

* Trademark of Western Electric.

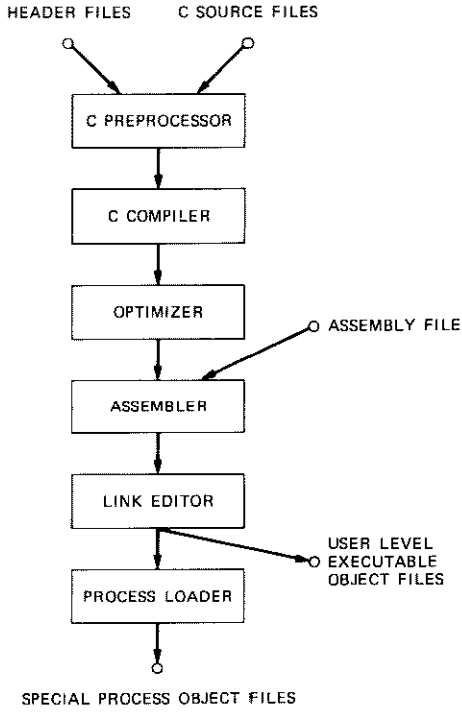



Fig. 2—Compilation tools.

The preprocessor provides a macro expansion facility and directives for sharing common source (or *header*) files. Header files typically contain data and macro declarations shared by many programs. This mechanism ensures that common definitions are consistent across all programs using them because they are accessed from the same source.

The C compiler and optimizer are based on the portable C compiler[4] and a similarly structured portable code improver. The compiler generates assembly code from preprocessed C source using syntax-directed parsing (from *YACC*[5]) and modified Sethi-Ullman[6] and Aho-Johnson[7] tree-matching and expression-optimization techniques. The generated assembly code can then optionally be passed through an optimizer (or, more correctly, code improver) that eliminates unnecessary branching, removes redundant register loads, and converts certain instructions or sequences of instructions into simpler or more efficient, yet semantically equivalent ones. Both the compiler and optimizer leave their result in the form of an assembly language file.

From assembly language source, the assembler creates an object file containing object code text and data (with optional relocation information), symbol and source line number tables to communicate with

testing tools, and a section layout dictionary that provides information on the structure of the file.

The link editor is capable of combining a collection of object files into a single object by resolving interprogram symbol references and binding symbols to virtual or absolute addresses as specified. The object file for the 3B20D Processor may contain multiple text and data sections. This facility is used in the creation of special process files and libraries.

Process files for the DMERT operating system are generated using a special tool that uses the link editor to create special data sections used in process communication and entry. All kernel, supervisor, and special DMERT processes require use of this tool as a final construction step. User level processes require only the normal link edit step for completion.

#### *4.3 Building a view of a subsystem*

The object construction tool, build, (based on the *UNIX* operating system make command), exists to create objects according to a previously specified set of object construction commands and a list of dependencies. This specification is used to identify which software components are needed to construct others.

Using modification time-stamps provided by the file system, build determines which objects are current and which need to be rebuilt. When an object needs to be rebuilt, its specification must be examined, and so on down the line until all necessary objects are rebuilt or current. These object construction instructions constitute a makefile for the particular DMERT subsystem.

The advantage that build offers over make is that the developer in a private development node need only have copies of the source being changed and can access the remainder of the objects, header files, or C or assembler source files from other nodes as specified in the developer's *viewpath*. A viewpath is an ordered set of directory names indicating the nodes to be searched for missing software components. The viewpath is an extension of the *UNIX* operating system *searchpath* variable that contains a list of directories containing commands. By specifying a viewpath, a developer can create a desired view of a subsystem product to be tested. The view may be, for example, that of the developer's changes integrated with only officially approved software or with any other software currently submitted to the test team.

#### *4.4 Preparation for testing*

Once a subsystem has been changed in response to an MR within a developer's node and is ready to be taken into the laboratory for

testing, there are listings that can be prepared to aid in the software testing process. A set of SGS utilities exist to generate these listings. A C program breakpoint source listing contains a source listing of all the functions in a file augmented by line numbers indicating C source lines at which a breakpoint can be set during testing. This listing is important due to the fact that object code may have been significantly rearranged during the optimization step of the compiler rendering certain C source lines shuffled in a semantically equivalent, but non-obvious, fashion. The listing helps a developer in a test laboratory who needs to correlate high-level C source to 3B20D Processor assembly code.

A namelist utility creates a listing of all the C language symbols residing in the object file's symbol table with their addresses and types. Not all symbols associated with an object need reside in the symbol table since they can be removed during object creation by another SGS utility.

A developer can obtain assembly source listings associated with an object file in either of two ways. The listing can be generated by the compiler during compilation or it can be created by a *disassembler* from the object file. This latter listing will contain C program source information in terms of symbols, labels, and line numbers if they have not been stripped from the object. This listing can be of particular interest after software has executed on the machine to determine where text and data may have been accidently altered by a process out of control in a development laboratory.

With these listings augmenting the original source files, the developer is prepared to enter the testing environment.

#### *4.5 Submitting changed files for approval*

When all desired changes have been made and unit or subsystem testing has convinced the developer that the modification request has been satisfied, the files extracted from the official CMS data base (and any files newly created as a result of the MR) are submitted to the test team through the use of CMS commands (refer again to Fig. 1). The activity of the developer on the files in question can be temporarily or permanently suspended by other CMS commands that place the changes made into the CMS data base and allow other developers access to the same files for editing if necessary.

The software approval process in the SDS is initiated when the developer submits an MR-related software change for inclusion in the next release. The developer submittal begins a three-step approval process:

(*i*) Independent Certification—The source change is used by administrators to reconstruct the changed products. This is done in such

a way as to not affect anything currently approved or under test. The changed products are then independently tested (certified) by a member of the project's system test group. Certification failure implies rejection of the MR and further work for the developer.

(*ii*) Integration—Those changes passing certification are incorporated into the under-test view where they are integrated with the rest of the under-test changes. This integration is performed by members of the project's integration group. Failure during integration also implies MR rejection, as well as recertification following additional development.

(*iii*) Approval—When all certified changes have been integrated and soak tested, the under-test view will be approved. This implies releasing a software update to 3B20D customers, updating all the nodes, and approving MRs.

## V. DEVELOPMENT ENVIRONMENT AND STANDARDS

The software development system must establish a balance between the flexibility given to developers and administrators to enhance their productivity and restrictive standards so that administration is manageable and effective. Considerable development flexibilities already have been identified. Among these are:

(*i*) The capability to select a particular version of a source file by specifying one or more MR numbers whose related source changes are applied to the last released version of the source file.

(*ii*) The use of any editing facilities once a source file has been obtained from CMS.

(*iii*) The C language, which encourages programmer optimization and functional modularization.

(*iv*) The protection from concurrent source changes by independent developers ensured by CMS.

(*v*) The independent, private development node allowing experimentation that does not interfere with other development activity.

(*vi*) The flexible capabilities for view construction provided by build.

(*vii*) The ability administrators have through CMS to back out changes associated with particular MRs and return to previous software states.

The software environment and its SDS tools also impose an important set of standards on development activities that guarantee safe, productive, and orderly administration of all change activity.

### *5.1 Host machine configuration*

The host machines used in software development and administration are each standardized with respect to the activities that occur on the

machines. This standard ensures that all development activity can be monitored adequately and that the proper tools can be made available to those needing them. With a fixed set of machines, updates to the software tools can be synchronized making sure that all development is compatible.

The 3B20D Processor software development system resides on a network of eight general-purpose, 16-bit minicomputers, all of which run the *UNIX* operating system. The functional configuration of these eight machines is shown in Fig. 3.

(*i*) Developer activity is confined to the six machines labeled SD (software development). Here developers utilize CMS, the SGS, and the build facility, as well as the standard tools available with the *UNIX* operating system.

(*ii*) The machine labeled SC (software control) is for administrative use only. This machine contains the modification request data base system from which all change activity is controlled.

(*iii*) The SP (software production) machine is an administrative machine. All official product construction takes place on this machine.

The test environment combines a 3B20D Processor and a support machine running a real-time version of the *UNIX* operating system as its support processor.

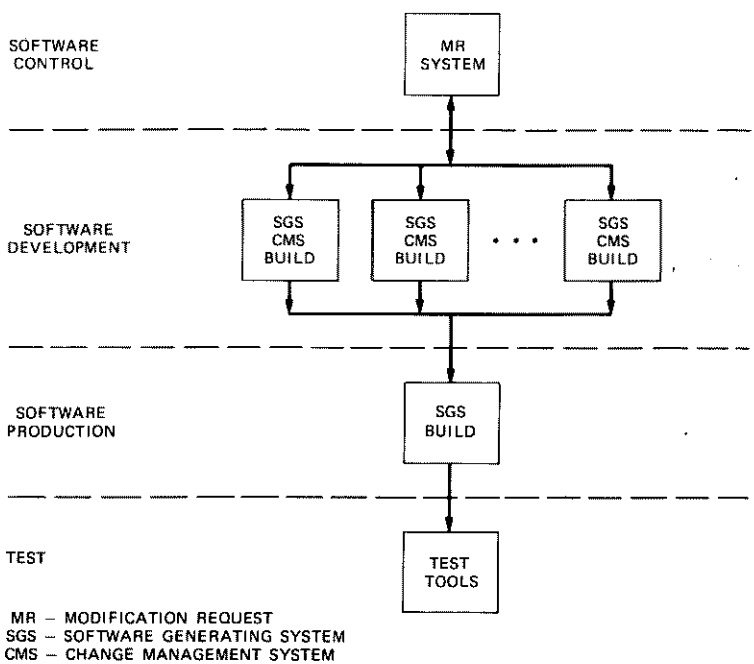



Fig. 3—Host machine configuration.

### 5.2 Software configuration

A standard *UNIX* operating system directory structure configuration has been established for the 3B20D Processor software subsystems. This configuration is repeated in each CMS node so that file access and object construction techniques are fully repeatable no matter where the view building activity takes place.

The 3B20D software is logically partitioned into distinct subsystems. Typical subsystems include the operating system, diagnostics, and peripheral unit drivers. Software development for the subsystems is evenly spread over the six SD machines described in the previous section. A single subsystem is fully contained on one of the SD machines. Each SD contains the software development activities for one or more subsystems.

The only machine on which the full directory structure combining all of the subsystems exists is the SP machine where official load construction takes place. Subsystem development can take place on the individual SD machines, independent of other subsystems, because the following software construction standard is enforced: the only source files shared between subsystems during load construction are global header files (common definitions required by more more than one subsystem) or global libraries (collections of common object modules).

Global header files and global libraries have fixed, standard positions in the SD machines' directory structure. These shared resources are distributed from the SP machine to the various SD machines after validation by administrators. In this way, all SD machines are kept synchronized with respect to this critical data.

Other top level directories or directory structures are used during the load-construction process as installation points for revised tools, tool usage descriptions in manual page format, and 3B20D Processor core software components.

### 5.3 Interactions between CMS and the MR system

The major strength of CMS and the MR system lies in the standardization of source-change administration. The following major standards are enforced with these tools:

(*i*) Every incremental change is tagged with an MR number.

(*ii*) The developer cannot modify official source until an MR has been assigned to him or her by an administrator. This assigning capability can be used by project administrators to funnel developer activity.

(*iii*) Once a developer is satisfied with a change, it is submitted against the corresponding MR for independent verification. Once submitted, no further source changes for the MR are allowed.

The MR data base system on the SC machine and the various CMS instances, each identified via a subsystem name, on the six SD machines are connected in the SDS via a high-speed network and a remote job-execution facility. Specific actions on the SC machine cause remote jobs to be sent to one of the SD machines and executed for a particular CMS instance. The reverse is also true. The major triggers interconnecting the MR system and the CMS instances are:

(*i*) When an MR is assigned to a particular subsystem for a particular generic, a remote job is sent to the SD machine on which that subsystem's development is based. When the job is executed on the SD machine, a sequence of CMS commands is executed that cause the MR to be assigned to the particular generic in the correct CMS instance. The developer can then officially edit the source files to effect the change.

(*ii*) When the developer is satisfied with changes, they are submitted for verification. The CMS status is set to *submitted* to prohibit any further editing for this MR, and a remote job is sent to the MR system on the SC machine. This job will cause the MR status for the particular generic and subsystem to change to *submitted.*

(*iii*) A remote job to reject an MR can be sent to CMS on an MR status change of under-test to assigned or submitted to assigned. In CMS, the MR would again be available for editing with a subsystem status of being-fixed.

These are the only MR/CMS interconnections necessary. All other control functions require only MR system actions.

### *5.4 Product construction standardization*

A major ingredient of the 3B20D Processor SDS is the standardization of the format and content of the makefiles used by build. The standards are enforced by allowing developers to create only a skeleton makefile and providing a makefile generator to produce the full makefile. The advantages accrued by standardized makefiles are the guaranteed correctness and completeness of object dependency lists and the ability to create cross-reference listings of dependencies to determine changed file impacts.

## VI. SUMMARY

Software development for the 3B20D Processor is administered with an important set of standards and a powerful set of tools to track development efforts and enforce these standards. The developers have at their disposal a flexible system for the generation of C programs and utilities to adequately prepare for debugging and efficiently construct object programs. The resulting 3B20D Processor software development system strikes a critical balance between offering freedom

and flexibility to programmers while managing and monitoring orderly software change procedures.

## REFERENCES


1. M. J. Rochkind, "The Source Code Control System," IEEE Trans. Software Eng., *SE-1* (December 1975), pp. 364–370.
2. J. R. Kane, R. E. Anderson, and P. S. McCabe, "3B20D Processor & DMERT Operating System: Overview, Architecture and Performance of DMERT," B.S.T.J., this issue.
3. H. D. Rovegno, "A Support Environment for MAC-8 Systems," B.S.T.J., *57*, No. 6 (July–August 1978), pp. 2251–64.
4. S. C. Johnson, "A Portable Compiler: Theory and Practice," Conf. Rec. Fifth Annual ACM Conference on Principles of Programming Languages, Tucson, Arizona, January 23, 1978, pp. 97–104.
5. S. C. Johnson, and M. E. Lesk, "Language Development Tools," B.S.T.J., *57*, No. 6 (July–August 1978), pp. 2155–77.
6. R. Sethi and J. D. Ullman, "The Generation of Optimal Code for Arithmetic Expressions," J. ACM, *17*, No. 4 (October 1970), pp. 715–28.
7. A. V. Aho and J. C. Johnson, "Optimal Code Generation for Expression Trees" J. ACM, *23*, No. 3 (July 1976), pp. 488–501.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# Overview, Architecture and Performance of DMERT

By J. R. KANE, R. E. ANDERSON, and P. S. McCABE

(Manuscript received March 22, 1982)

*A process-oriented operating system, the Duplex Multiple Environment Real Time (DMERT) operating system, was designed for the 3B20D Processor and offers both real-time and time-shared operation, with emphasis on high availability. The design objectives and architecture of the DMERT operating system and an explanation of how the system performance is characterized are presented. A companion article describes in depth the DMERT operating system design.*

## I. INTRODUCTION

The direct predecessor of the 3B20D, the 3A Processor, included a real-time monitor known as the Extended Operating System (EOS).[1] Experience with EOS demonstrated that applications could develop their software with less effort and that synergy resulted between the hardware and software developers. Hence, the hardware is optimized to support the software and the software, in turn, uses the hardware in the most effective manner. The success of EOS led to the decision to support 3B20D applications with a more extensive operating system than EOS. The operating system that resulted is known as the *D*uplex *M*ultiple *E*nvironment *R*eal *T*ime (DMERT) operating system.

The basic architecture of DMERT is based on an earlier system named MERT,[2] which was derived from the *UNIX** operating system.[3] Both the *UNIX* and the MERT operating systems were origi-

* Trademark of Bell Laboratories.

nally developed to execute on commercial equipment; today *UNIX* operating systems are used widely for time-sharing on a variety of computers. The "D" in "DMERT" reflects one of the characteristics that distinguishes it from the previous two operating systems, namely that DMERT is designed to execute on a *duplex* 3B20 Processor. Thus, the DMERT architecture draws upon concepts from EOS, MERT, and *UNIX* operating systems.

The applications using the 3B20D Processor have been described in detail in a previous article.[4] Notice that while different, they have several common characteristics. First, a major component of the application is software. Second, the major mission of this software is real-time oriented with response times as short as several milliseconds. Third, each application has a need for continuous operation 24 hours a day, 7 days a week and hence stringent processor availability requirements. Fourth and finally, each application is to be operated over a long period of time, which requires extensive software for monitoring and reporting on system status as well as changing and upgrading the system while it is in operation.

This paper describes the development objectives of DMERT, which were chosen to satisfy the above application characteristics. The operating system architecture used to achieve these objectives is next described. Finally, this article describes the performance characterization of DMERT. The design details of DMERT are presented in depth in the next article[5] in this Journal.

## II. DMERT DESIGN OBJECTIVES

### *2.1 Support multiple real-time applications*

It is necessary for the DMERT operating system to support many applications, each with different real-time demands. Some applications include data bases that need many disk jobs serviced quickly. Others control telecommunication equipment requiring rapid response to an event such as an interrupt and dedicated processing capacity for an interval thereafter. To satisfy these diverse needs, a design objective was established to provide modularity in the operating system to allow a high degree of application tailoring.

### *2.2 Improve application development productivity*

The real-time applications of the 3B20D Processor often have major software components that are not time critical. The "rule of thumb" stating that 80 percent of the time is spent on 20 percent of the software generally applies to these applications. Hence, a design objective of DMERT was to support a feature-rich operating system environment for the non-time-critical software while retaining real-time responsiveness for the rest.

To increase productivity of the developers, an objective of efficiently supporting a programming language at a level substantially higher than assembly language was established. (See the previous article[6] for further details.)

To allow technology upgrading of the 3B20D Processor without impacting the application software, an objective of shielding application programmers from hardware implementation details, such as memory protection[7] and I/O devices,[8,9] was established.

### *2.3 Error tolerant design*

To meet the reliability objectives of the 3B20D Processor, it is necessary to support software packages for error checking and recovery. Some of these are described in subsequent articles.[10–12] To reduce the complexity of both the operational and recovery components of the system, a design objective was established to separate recovery software from the core of the system.

An objective of incorporating extensive internal consistency and integrity checks within all software components was established to ensure that critical software modules protected themselves from errors in other parts of the system.

## III. DMERT ARCHITECTURE

### *3.1 Processes*

One of the basic design goals for DMERT was to build modular and independent processes, each having localized data known only to itself. Hence, the notion of a process is fundamental to the DMERT architecture, which is essentially composed of a kernel and a collection of cooperating, concurrent processes. The following sections define what a process is, how processes communicate with each other, and how they are used by DMERT applications.

### *3.2 Definition of a process*

A process is a collection of related, logical segments (programs and data) that can be brought into memory to form an executable entity. A segment is the basic memory entity in DMERT. A segment is composed of 1 to 64 pages, each 512 32-bit words in length. Segments can grow dynamically in increments of a page. A process consists of at least three segments: code or text, a stack used for temporary data, and a special type of data segment called a process control block (pcb). The pcb segment contains unique information that identifies the process to the operating system. This information includes the process number, type of process, priority, and address space qualifiers that define the virtual address space for a process. Each process has its own virtual address space of up to 128 segments. These virtual addresses

are mapped to physical addresses by 3B20D hardware under the control of the DMERT operating system.

A process can be dynamically created to perform a set of functions and then to gracefully terminate itself when its task is finished. Processes that continuously perform work remain "alive" at all times; however, they may sleep or be inactive until an interrupt fires. These features allow main memory to hold only the processes necessary, at a given point in time, to support the application.

### *3.3 Process types*

DMERT has four basic types of processes: kernel, kernel process, supervisor, and user. DMERT may be viewed as a hierarchy of virtual machines, where successive levels put additional restrictions on access rights and further remove the programmer from the details of the physical machine. However, the higher levels may take advantage of services provided by the lower levels. In general, the higher the level, the more services available to the application programmer; the lower the level, the more real-time-efficient the program execution. This level structuring of virtual machines permits DMERT to manage real-time applications, while at the same time providing the flexibility of a time-sharing system for background tasks. This approach avoids contention for system resources with the high-priority tasks and simplifies the implementation effort for lower priority tasks.

#### *3.3.1 Kernel*

The DMERT kernel provides the most primitive virtual machine. The kernel handles hardware interrupts, timer interrupts, and operating system traps. In all cases, the kernel saves the state of the interrupted process, provides whatever service is requested, and then restores the state of the interrupted process. The kernel services are fairly primitive but they execute efficiently.

Also part of the DMERT kernel are special processes that provide scheduling, memory management, and other services. For example, the memory manager and the scheduler are two of the special processes in DMERT. The memory manager loads processes into main memory, selects segments to be swapped out to disk when additional main memory is required, and provides routines that may be called by the kernel. The scheduler controls the execution of time-shared processes, that is, supervisor and supervisor-user processes. Special processes behave as kernel processes, except that they do not have their own virtual address space, but rather reside in the kernel's address space. These special processes communicate with the kernel through function calls instead of operating system traps, and they have access to global system data.

### 3.3.2 Kernel processes

Kernel processes comprise the next virtual machine layer in DMERT. They are completely interrupt-driven and are designed to provide time-critical processing in a real-time environment. Kernel processes have their own virtual address space. However, they share the kernel's stack and the kernel's message buffer segment. Swapping is not necessary with kernel processes because their segments are locked in memory to ensure rapid response to events such as interrupts. The various peripheral device drivers and the file manager, which implements a hierarchical file system, are examples of kernel processes.

At process build time, kernel processes are set up to share the operating system's stack and message buffers. This design was chosen for quick access to arguments of operating system traps and fast message communications between processes. Since kernel processes use the kernel's stack, they must run until they complete their task and then return control to the kernel.

### 3.3.3 Supervisor and user processes

Supervisor processes form the third layer of virtual machine. These processes can use all the services provided by the kernel and kernel processes. Supervisor processes provide time-sharing services that can be considered background tasks. They share the real time of the processor with each other according to priorities administered by the scheduler, which is a special process. In general, supervisor segments are not locked in memory, but can be swapped out. Thus, supervisor processes take much longer to dispatch than either special or kernel processes.

Supervisor processes can be designed to run "stand-alone" or they may be used to implement a fourth virtual machine layer called user processes. The DMERT process manager is a supervisor process that does not support a user level. However, the *UNIX* supervisor provides a user environment identical to that seen by a *UNIX* program. This is done through code at the supervisor level that calls upon the services of lower virtual machine layers. DMERT can simultaneously support multiple supervisors, each supporting its own user processes. It should be noted that while DMERT treats a supervisor/user combination as a single process with a dual address space, both levels exist conceptually. Also, supervisor-level code executes more efficiently than user-level code because a supervisor has direct access to the lower level primitives, while the user interface to these primitives is coordinated by the user's supervisor.

### 3.3.4 Interprocess communication

DMERT provides a rich set of interprocess communication and

synchronization mechanisms including messages, events, interprocess traps, and shared memory. These interprocess communication primitives are fundamental to the DMERT structure. Most of the system services are requested by an exchange of events and messages between a requesting process and either a system process or the kernel.

*3.3.4.1 Messages.* Processes are in general separate and distinct entities. Two processes working together on a task must be able to exchange information. To satisfy this need messages may be sent from any level process to any other level process. These messages can be up to several hundred bytes long. The sender need only know the target process number and a pre-agreed format of the message. An optional acknowledgment capability is provided so the sender can synchronize actions with the receiver.

*3.3.4.2 Events.* Communications between processes may occur using an event mechanism. An event is a single bit that is set by DMERT or a process and can be interrogated by the receiving process. Presently, 32 bits are available of which the DMERT operating system reserves 16 bits for its use. Thus, two or more processes can communicate internal states using events.

*3.3.4.3 Interprocess traps.* A mechanism exists in DMERT to allow a lower-level process to support a higher-level process. A user-level process may trap to a supporting supervisor and a supervisor may trap to a kernel process. Trapping implies a transfer of control from one process to another with the passing of input parameters to the target process. The lower-level process returns status and control back to the trapping process after it has completed the required support work.

*3.3.4.4 Shared memory.* Processes are built with a view of their own virtual address space and in general cannot access any other process's process's address space. This affords protection; however, sharing large amounts of data is difficult. Cooperating processes that must exchange information at rates higher than those supported by messages or events can share segments. A shared segment is a part of the virtual address space of several processes simultaneously. The application must control access to the shared data.

### *3.4 Multiple environment support of applications*

DMERT simultaneously supports both a real-time and a time-sharing philosophy. Kernel and kernel processes operate in a real-time environment and have first call on the available real time of the 3B20D Processor. The remaining time is shared among supervisor and user processes.

Most telecommunication applications build their own virtual machine or "operating system" as a kernel process that can respond to

the time-critical stimuli characteristic of telecommunications. This kernel-process approach also allows fine tuning of the telecommunications operating system, independent of the DMERT operating system. In at least one case, the application specific virtual machine is a kernel process that runs in emulation mode. Being a microcoded machine, the 3B20D Processor can efficiently execute another machine's instruction set. By using the emulation mode, existing debugged application code, including operating systems, can be carried forward to the 3B20D Processor and DMERT with little additional software development effort.

Some applications spread their functions over the existing DMERT virtual machines. For example, the time-critical functions related to disk and data link usage are implemented as kernel processes, and the administrative and postprocessing functions are implemented as supervisor and user processes. An application of this type is normally used as a data base management system and/or a data communications system. Generally, an application fine tunes its system by moving processes to different execution levels within the virtual machines.

## IV. PERFORMANCE

The 3B20D/DMERT system is capable of providing computing services in a stand-alone mode; however, usually it is utilized by surrounding it with application hardware, firmware, and software. The application hardware may include additional units identical to those already a part of the processor complex (e.g., additional memory, disk, data links), or it may include hardware unique to the application system (e.g., bus controllers, or time- and space-division switches). The application software frequently includes drivers, schedulers, and fault-recovery facilities, as well as the more usual "application programs." As a result of this diversity of software interfaces to DMERT, the performance modeling and measuring of the application system requires an extensive performance characterization of DMERT, rather than the more traditional benchmark approach used in general-purpose, computer-performance evaluation. The following sections describe the approach taken and the type of performance data made available to DMERT applications.

### *4.1 Performance characterization*

Since the application software may interface DMERT at all levels of the virtual machine (hardware, firmware, and the software levels of kernel, kernel process, supervisor, and user process), the performance characterization of the system must include data for all of these levels. In addition, the application can make use of a variety of DMERT system resources such as memory, peripheral devices, message buffers,

etc, and hence these resources must be accounted for in the performance characterization as well.

The 3B20D performance characterization addresses four areas: the operating system, input/output, DMERT services, and DMERT overhead. Each of these areas will be discussed in more detail, and the sum of these areas covers all significant aspects of DMERT performance, as well as providing some models for the application developers to use in the analysis of the application system performance.

#### 4.1.1 Operating system characterization

The goal of the performance characterization of the operating system was to identify the cost in resources for every service available at every level. The predominant service interface in DMERT is the Operating System Trap (OST), and hence every significant OST in DMERT was characterized with respect to its central processing unit (CPU)-time at the various modes and execution levels available in DMERT.

The kernel, supervisor, and user OSTs cover a broad range of DMERT services:

(*i*) *Timing:* clock reading and setting, single and repetitive timeouts, and process sleeping requests.

(*ii*) *Memory management:* locking and unlocking of memory segments, growing and shrinking of memory segments, and swapping of memory segments.

(*iii*) *Scheduling and interrupting:* protecting against interrupts (critical region), priority changing, and fielding of interrupts.

(*iv*) *Interprocess communication:* interprocess message sending and receiving, and sending and fielding of interprocess events.

(*v*) *Process switching and other functions:* switching from one process to another, changing the execution level of a kernel process, creating another process, faulting another process, routing and rerouting of standard inputs and outputs.

#### 4.1.2 Input/output system

The input/output (I/O) system of DMERT is characterized from an *internal* point of view; that is, each of the kinds of I/O services are characterized with respect to their primary resource consumption. The various I/O services are all measured with respect to the CPU-time consumed for each transaction and the maximum throughput rate based upon the elapsed time for each transaction.

A basic set of operations can be performed on most I/O devices: open, close, read, and write. Seeking is a disk-only service, and rewinding is a tape-only service. Except for data links, read and write operations can be invoked in several ways, depending on the physical organization of the data on the device. Most file operations can be

invoked directly or through the file manager. The devices supported by DMERT include disks, tapes, terminals, and data links. Finally, the disk I/O services can be invoked in several modes: normal (synchronous, buffered I/O), asynchronous (which allows the invoking process to continue running while the I/O request is being serviced), physical (no buffering constraints, or services) and synchronous writes (which guarantee an immediate write to the disk, rather than the potential delayed write possible in the normal mode).

Most of these I/O services are available to kernel, supervisor, and user processes via separate OSTs. Each OST is characterized with respect to CPU-time and maximum device throughput for each of the applicable cases.

#### *4.1.3 DMERT applications services*

Application software may utilize high-level services from DMERT as well as the more primitive OST-invoked services. The Craft Interface[12] is an example of this general category covered by the term DMERT services. The Craft Interface system provides an extensive and sophisticated set of terminal-oriented facilities that are used by both DMERT and the application software. Additional examples of DMERT services are the diagnostic and audit facilities that are a part of DMERT, but also may be invoked by application software.

Of these DMERT services, the Craft Interface has been characterized for performance owing to its importance to early DMERT applications. The performance characterization chosen was to measure the CPU-time usage of the three most important application services: the Program Documentation Standard (PDS) Shell, the Control and Display Administrator, and the Output Spooler. Each of these services was measured with a range of appropriate job sizes.

#### *4.1.4 DMERT overhead*

The final category in the performance characterization of DMERT is the system overhead. While system overhead is not invoked explicitly as a service, it provides the essential services of the operating system. There are several types of system overhead:

(*i*) *Functional:* provides for the capability of a multi-environment, real-time operating system by handling the timing-based facilities for interrupt servicing, scheduling, craft terminal polling, and data link polling.

(*ii*) *Sanity:* provides for sanity and overload monitoring by resetting hardware sanity timers, monitoring DMERT and application sanity timers, and checking for process lock-out conditions.

(*iii*) *Preventive maintenance:* provides for routine preventive maintenance exercising and running routine software audits.

(*iv*) *Fault maintenance:* provides for fault detection, location and recovery, including removing faulty units from service, and testing and restoring replacement units. This overhead is incurred only when a fault occurs.

(*v*) *Services:* provides for other non-electable services not covered by the previously described types, such as Craft Interface services and Plant Measurements services invoked in providing the processor system.

The DMERT operating system overhead is characterized in terms of CPU time and is expressed as a percentage. The first two types of overhead (functional and sanity) constitute the "continuous" overhead seen by the application, and cannot be controlled or throttled by the application. This component of the overhead is less than 5 percent for DMERT. The preventive maintenance overhead can be controlled in two ways: routine software audits can be throttled so that their peak resource usage is limited to a desired value, and the routine diagnostic exercising can be scheduled during times of light load. The fault maintenance overhead is measured as a single-fault, worst-case scenario for the fault detection, isolation, and recovery, as well as the testing and restoral, of the repaired unit. The total resource usage is averaged over the specified two-hour repair interval. The services overhead includes the normal administrative activities necessary to maintain and administer the processor complex. The total system overhead for functional, sanity, preventive, and fault maintenance and services is less than 15 percent for DMERT.

### *4.2 Performance measurement techniques*

The key resource in the 3B20D/DMERT system, and in the applications systems built upon it, is CPU time. It is shared among many processes, both DMERT and application, in four execution modes, at 16 execution levels and 256 priority levels. The sharing is interrupt driven, with preemption from processes at higher priorities and at higher execution levels. While it is relatively simple to measure CPU time in an overall sense, it is a very complex job to measure the CPU time used by a particular process, function, or service. To solve this problem, the DMERT kernel was instrumented. Finally, hardware monitors were used to measure the performance of the processor complex, and also to verify the correctness and accuracy of the software performance measurement instrumentation.

#### *4.2.1 Kernel instrumentation*

The DMERT kernel was extended to provide detailed accounting of the CPU time usage (by execution mode and level) for the system as a whole, and also for each process. To keep the overhead low enough

to allow the instrumentation to be a permanent part of DMERT, a statistical sampling approach was used. Each 10 milliseconds the kernel records the process currently executing, together with the execution mode and level, assigning the previous 10 milliseconds to that process. Over a reasonable period of time (seconds or more) these statistical sampling results will converge arbitrarily close to the actual CPU time usage.

## V. SUMMARY

An overview has been given for the objectives of DMERT based on its goal of providing a high-reliability, real-time processor system for telecommunications applications. This overview has indicated some of the development objectives of DMERT and has given the approach for the performance characterization of the whole system. Also included is a description of the various kinds of operating system overhead, including measured values for DMERT, and a description of the performance measurement instrumentation within the DMERT kernel.

## VI. ACKNOWLEDGMENTS

Many members of Bell Laboratories' staff contributed to the successful introduction of DMERT. Key contributors to the work described in this article and not mentioned as authors or references were: S. F. Ho, P. C. Kao, T. K. Liu, T. M. Raleigh, E. P. Schan, D. S. Trushin, and S. M. Udupa.

## REFERENCES

1. C. H. Elmendorf, "Meeting High Standards with the Extended Operation System," *Bell Lab Rec.*, (March 1980), pp. 97–103.
2. H. Lycklama and D. L. Bayer, "The MERT Operating System," B.S.T.J., *57*, No. 6., Part 2 (July 1978), pp. 2049–86.
3. D. Ritchie and K. Thompson, "The *UNIX* Time Sharing System," B.S.T.J., *57*, No. 6, Part 2 (July 1978), pp. 1905–29.
4. R. W. Mitze and W. C. Schwartz, "The 3B20D Processor & DMERT Operating System: 3B20D Processor and DMERT As a Base for ESS Applications," B.S.T.J., this issue.
5. M. E. Grzelakowksi, J. H. Campbell, and M. R. Dubman, "The 3B20D Processor & DMERT Operating System," B.S.T.J., this issue.
6. B. R. Rowland and R. J. Welsch, "The 3B20D Processor & DMERT Operating System: Software Development System," B.S.T.J., this issue.
7. M. W. Rolund, J. T. Beckett, and D. A. Harms, "The 3B20D Processor & DMERT Operating System: 3B20D Central Processing Unit," B.S.T.J., this issue.
8. R. E. Haglund and L. D. Peterson, "The 3B20D Processor & DMERT Operating System: 3B20D File Memory Systems," B.S.T.J., this issue.
9. A. H. Budlong and F. W. Wendland, "The 3B20D Processor & DMERT Operating System: 3B20D Input/Output System," B.S.T.J., this issue.
10. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "The 3B20D Processor & DMERT Operating System: Fault Detection and Recovery," B.S.T.J., this issue.
11. J. L. Quinn and F. M. Goetz, "The 3B20D Processor & DMERT Operating System: Diagnostics," B.S.T.J., this issue.
12. M. E. Barton and D. A. Schmitt, "The 3B20D Processor & DMERT Operating System: Craft Interface," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# DMERT Operating System

By M. E. GRZELAKOWSKI, J. H. CAMPBELL, and M. R. DUBMAN

(Manuscript received March 17, 1982)

*General operating system services in Duplex Multiple Environment Real Time (DMERT) are provided by a kernel and a set of cooperating processes. These services support a multitude of process-oriented features that include a rich set of interprocess communication mechanisms and a sophisticated collection of memory manipulation primitives. The operating system provides processes with two scheduling alternatives and various creation and termination options. The cooperating processes include input/output drivers and device handlers that enable processes to communicate with a variety of peripheral devices. Another set of processes simulates a* UNIX™ *operating system and file system.*

## I. INTRODUCTION

This article describes the design of the DMERT operating system nucleus. The reader should be familiar with the basic architecture of Duplex Multiple Environment Real Time (DMERT), which is described in a companion paper.[1] The operating system nucleus is composed of: the kernel, which supports interprocess communication mechanisms, the system clock, and interrupts; the special processes, which perform memory management and scheduling; two supervisor processes, which handle process management and the *UNIX** operating system environment; and three kernel processes that control communication with peripherals and the file system.

In general, DMERT applications view the operating system nucleus

* Trademark of Bell Laboratories.

as one entity rather than a set of cooperating processes. Thus, the interrelationships between these processes are hidden from them. This article describes the design of DMERT's nucleus from the user's perspective; that is, it is feature oriented rather than process oriented. After finishing this paper, however, the reader should be familiar with both the feature set and the internal design of the cooperating processes.

In order for a process to use any of the operating system's features, it must interface to the operating system with Operating System Traps (OSTs) or Interprocess Communication (IPC) mechanisms. OSTs were discussed in Ref. 1, and Section II describes IPCs. Sections III, IV, and V define the image and life cycle of processes by describing how memory is handled, the scheduling options, and how processes are created and terminated. Section VI summarizes the characteristics of a special type of process, namely a user process. Section VII discusses the input/output (I/O) and disk interfaces, and Section VIII describes DMERT's file systems.

## II. INTERPROCESS COMMUNICATION

As stated in the previous paper, DMERT provides an extensive set of interprocess communication and synchronization mechanisms, including messages, events, process ports, faults, interprocess traps, first in-first outs (FIFOs), and shared memory. These interprocess communication mechanisms are described below.

### *2.1 Events*

An event is a one-bit piece of information that may be sent, received, and recognized by a process. DMERT has 32 event bits, several of which are reserved for special meanings by DMERT. Those not reserved may be used in any manner agreed upon by a set of cooperating processes. DMERT provides several event-type OSTs that enable one process to send an event to another process or to a class of processes.

Processes have multiple entry points for start, event, OST, and fault entries. The purpose of the event entry is to allow the system to direct the process's attention to an event that has just occurred. Thus, when any of the process's event bits are set, control is passed to its event entry.

Additional control over events is given to a kernel process. It has the ability to mask and reenable the bits that comprise its event flag word. This gives a kernel process the capability to more precisely control its flow by only allowing certain communications to occur.

To provide timely response to events, the process receiving the event is interrupted and the state of its activities is saved by the kernel. After

the event is acted on, the process is restored to its original state and continues from the point where it was interrupted.

### 2.2 Messages

Messages are the primary method of communication between cooperating processes in the operating system. The DMERT kernel provides a variety of OSTs for allocating, sending, receiving, canceling, auditing, and freeing messages for individual processes.

A message consists of a fixed-size header followed by a variable-size block of data, called the message body. The header contains routing directions, status indicators, and other administrative information. The message body's contents and structure are determined by the cooperating processes using the message.

A reserved event is used to notify a process that it has received a message. Hence, when a process receives a message, control is passed to its event entry. Waiting messages are queued in a message buffer segment in the kernel's address space and are directly accessible by kernel processes. However, kernel processes do not manipulate messages directly without first dequeueing them by using one of the message dequeueing OSTs. Supervisor processes do not have direct access to the message buffer segment. Messages must be copied into and out of this segment from and to the supervisor's own space.

### 2.3 Ports

Communications by messages require the knowledge of the target process's Process Identifier (PID). Since processes can be created and terminated dynamically, PIDs also are generated dynamically and a process typically knows only its own PID. Process ports permit processes to communicate with each other without knowing each other's PID. A process port is a globally known "device" to which a process may attach itself for receiving messages. Other processes may communicate with a process connected to a port by sending a message to that port rather than to a specific PID. Thus, process ports permit unrelated processes to communicate with each other.

### 2.4 Faults

Faults are very similar to events. A fault is a one-bit piece of information that may be sent, received, and recognized by a process. DMERT has 32 fault bits, some of which are reserved for special meanings by DMERT. When a process's fault bits are set, control is passed to its fault entry.

Faults are used to inform processes of error and overload conditions and take precedence over events. If a process has both events and faults pending, control will be passed to its fault entry first.

### 2.5 Interprocess traps

As mentioned in Ref. 1, an OST is generally used to transfer control to the kernel to perform some service on behalf of the requesting process. As such, an OST does not constitute a form of interprocess communication. An interprocess trap is a generalized notion of the OST, where a process can trap to some other process rather than the kernel to have some service performed. Because this interprocess trap can communicate service requests or other data between two processes, it is a form of IPC.

Interprocess traps are more restrictive than other IPC mechanisms. A process can trap to another process only if the former process's execution level is lower than the latter's. Control is passed to the trapped process's OST entry.

### 2.6 FIFOs

Unrelated processes can use FIFOs for character stream communication. Conceptually, a FIFO is a queue of characters; characters flow through in a first in-first out order (hence, the name). The queue is fixed in length and may become filled if writers write faster than readers read. When such a condition occurs, further writes are prevented. Similarly, reads are prevented when the queue is empty.

Each FIFO has a name; that is, each is a special file in the file system (described in Section VIII). This naming characteristic distinguishes it from a pipe,[2] because only processes spawned by a common ancestor can communicate using a pipe. Naming allows any process that knows a FIFO's name to use it, regardless of its ancestry.

Any process level may use this mechanism. The FIFO is a suitable replacement for messages where communication is a character stream.

### 2.7 Shared memory

A segment is the basic unit of shared memory. It can be shared between different processes, or between multiple instances of the same process. A segment also can be shared among any number of arbitrary processes by assigning a global name to it and allowing any process to access it that has been given the global name.

Since it is completely managed by the cooperating processes, shared memory is the form of interprocess communication most susceptible to error, but it is also the most efficient. Within the operating system, shared segments are used wherever reliability will not be sacrificed and where real-time response is paramount.

## III. MEMORY MANAGEMENT

DMERT memory management centers around the segment. A segment is a set of logically related pages. A page is 2048 bytes of

contiguous main memory that always begins on addresses that are multiples of 2048. A segment is a collection of pages that do not have to be physically contiguous. All segments in the system are unique, that is, no page can belong to more than one segment. (Reference 3 contains a detailed description of pages, segments, and memory translation.)

### *3.1 Segment attributes*

Segments are created in main memory by the operating system on demand and disappear when they are no longer needed. When the initial image of a process is created, its segments are loaded into main memory by the memory manager. Although a segment of a user or supervisor process can be swapped out to a swap area reserved on the disk if additional main memory is needed for a process of higher priority, the segment remains known to the system until it is not required by any process.

A segment is identified internally by its Segment Identifier (SID), which is the virtual address of its entry in the kernel's segment description table. A segment of a supervisor process also can be referenced by a segment number that indexes into the process's segment list. The segment list specifies the segments that are known to the process, including those that are not necessarily in the current virtual address space of the process.

A process has segments that may be private or shared by other processes. Text segments are shared when possible to avoid duplication of commonly used functions. Data segments are usually private since most processes want sole access to their own data. However, a data segment can be shared with another process for implementing interprocess communication (see Section 2.7) and can have a global name. Segment sharing increases the efficiency of memory management since a process needing to use a shareable segment already in main memory need only attach the segment to its virtual address space, thus reducing segment swapping.

A segment can be defined as executable, readable, writable, or some combination thereof. Two different processes sharing a segment can have different access rights to it. So, for example, an "owner" of the segment could have read and write access, while another "user" of the segment would only be allowed to read it.

A segment also can be designated as swappable or nonswappable. If nonswappable, the segment will not be swapped from the main memory to disk memory. If it is imperative that the segment not be moved around in main memory as, for example, when I/O is being done in the segment, it can be locked in main memory. In addition, a segment can be blocked (that is, made unavailable) to other processes while it is being initialized.

Finally, segments can be either active or inactive. If inactive, the segment is not presently a part of the address space for each process that owns it.

### 3.2 Process images

The executable image of a process consists of a set of segments. Originally, a kernel process had its segments completely specified at link time. Later enhancements allow kernel processes to dynamically add or drop segments using messages to the memory manager. A supervisor process can use OST calls to create or destroy segments, make copies of segments, request named segments, attach or detach segments to its address space, grow or shrink a segment, lock a segment in main memory for I/O, or make a segment swappable or nonswappable.

A kernel process starts with at least four segments: the Kernel Process Control Block (KPCB) segment, the stack segment, the system message buffer segment, and the process's text segment. The KPCB segment contains the process-dependent information. The stack and system message buffer segments are shared with all other kernel processes and the DMERT kernel while the process's text segment is shared only with other invocations of the same kernel process. (Most kernel processes have only one invocation active at a time.)

A supervisor process starts with at least three segments: the Process Control Block (PCB) segment, which contains the system's tables for the process; the stack segment; and the text segment. The process's data may be combined with its stack into a single segment or it may be placed in a separate segment. The access modes for the PCB and text segment are read-only and read-execute, respectively. The process may never write its PCB. The stack segment of a supervisor process is never shared, while a text segment is usually shareable between multiple invocations of the same process.

## IV. SCHEDULING

DMERT simultaneously supports both a real-time and a time-sharing environment. The kernel and kernel processes operate in a real-time environment and have first call on the available time of the 3B20D processor. The remaining time is shared among special, supervisor, and user processes.

### 4.1 Real time

Any process that must satisfy a real-time requirement should be a kernel process. DMERT maintains a process hierarchy based on 16 execution levels. A kernel process can belong to levels 3 through 15. (Levels 0 through 2 are reserved for time-sharing environment.)

DMERT bases its real-time allocation strategy on three concepts: execution levels, round robin scheduling, and preemption. DMERT dispatches processes at the highest execution level first. For example, a process belonging to execution level 15 is dispatched before a process belonging to level 14, which is dispatched before a process at level 13, and so on. For each execution level, DMERT maintains a list of waiting processes. When a process requests servicing, it is added to the appropriate list in a round robin fashion. As the operating system descends the hierarchy, it dispatches all the waiting processes at each level.

Generally, a kernel process executes until it exits. However, if another kernel process at a higher execution level is awakened, DMERT preempts the executing process. Upon completion of the preempting process, if no other higher level processes are awakened, the operating system resumes the suspended process.

DMERT's management of real time is straightforward and adds only a minimal amount of overhead. In fact, preempting one kernel process and dispatching another takes only about 320 microseconds. At the same time, applications are allowed to assign their own process's execution levels, which customizes their control and distribution of real time. This approach has proved to be quite flexible and permits a variety of applications.

#### *4.2 Time sharing*

As stated previously, the portion of real time not utilized by the kernel or kernel processes is time shared among supervisor and user processes. Processes supporting the time-sharing environment, such as the scheduler and the memory manager, are special kernel processes that reside at execution level 2. These processes are at the bottom of the real-time hierarchy and gain control of the processor only after all other real-time work is completed.

Supervisor and user processes normally execute at level 0. Level 1 is reserved for when they need to lock each other out. Within execution level 0, the scheduling hierarchy of supervisor and user processes is based on priority. The major difference between priority in the time-sharing environment and execution levels in the real-time environment is that DMERT adjusts priorities dynamically, whereas execution levels are fixed.

Priority adjustment is based on two factors: state and age. A time-sharing process can be in one of three states: sleeping, waiting, or executing. As a process enters a new state, it begins to age (from zero) until it changes state. Processes in different states age at different rates.

DMERT uses age and state to adjust the process's scheduling

priority. In particular, the priority of a sleeping or executing process is reduced as it ages and the priority of a waiting process is increased as it ages. The result is that a low-priority process waiting to run will eventually move above higher priority processes that have been executing.

DMERT selects the time-sharing process to execute by searching down a priority list until it finds a process waiting to run. The selected process continues to run until it changes state or is preempted by a kernel process. It cannot be preempted by a higher priority time-sharing process. A time-sharing process can leave the executing state for one of the following reasons: (*i*) it roadblocks (enters the sleeping state); or (*ii*) it uses up its allotted time slice and is returned to the waiting state; or (*iii*) it terminates. When a process times out, it is chained to the end of the list of waiting processes at the appropriate priority. Then, DMERT searches the priority list to find the next time-sharing process that is waiting to execute.

## V. PROCESS CREATION AND TERMINATION

The DMERT operating system supports the dynamic creation and termination of kernel, supervisor, and user processes. A process can be created on demand from a process load file maintained in secondary memory and can be terminated and recreated at will. Process creating and termination can be requested via messages, OSTs, or commands executed at a terminal.

### *5.1 Process creation*

The load file of a process is generated by means of the process loader, which is part of the software development system.[4] A process load file contains the text and data segments that comprise the initial image of the process. It also specifies a variety of basic process attributes, such as the execution level, stack size, instruction privileges, entry points, and, for time-shared scheduling, priority and time slice.

The creation of a process involves the identification of each of its segments in the kernel memory management tables, the formation of its initial image in main memory, and the identification of the process itself in the operating system scheduling tables. Start-up of a created process is generally accomplished by sending it a special initialization event, in which case control is passed to the process's event entry. In the case of supervisor and user processes, control is passed to the process's start entry.

#### *5.1.1 Process creation mechanisms*

There are three distinct mechanisms for creating a process. The system bootstrap mechanism creates a set of processes that constitute

the nucleus of the operating system and are essential to its operation. These include a process, called the process manager, whose principal function is the creation of other processes from process load files. The process manager can be instructed to create processes via the mechanism, and of sending it a message containing the name of the process load file and other relevant information. The third mechanism is the execution of a "fork" system call by a user process that results in the creation of a copy of the process executing the fork. The copy, called a child process, can subsequently execute another file via the "exec" system call, i.e., exchange its text and data for that of another process load file.

The creation of processes during system bootstrap is carried out by the kernel in a very efficient manner utilizing full access to the memory management tables and functions. The process manager is implemented as a swappable supervisor process that calls upon the services of the memory manager through OSTs and the file manager through messages. The fork mechanism is carried out within the process executing the fork and avoids the overhead associated with loading in the process manager.

### *5.1.2 Process creation features*

Certain features are provided by the process creation mechanism. These include a distinction between single-copy and multiple-copy processes, the sharing of segments, and the use of shared libraries.

Single-copy processes are not replicated in main memory by successive creations. Instead, the operating system increments a usage count, which is decremented whenever the process is terminated. The process remains in the system so long as its usage count is at least 1. Device handlers that are repeatedly opened and closed are typically single-copy processes.

Successive creation requests for a multiple-copy process results in processes with distinct PIDs that have independent existences. As mentioned earlier, the text segment is usually shared among the invocations of a multiple-copy process. Data segments are typically private.

Processes that have a parent-child relationship may conveniently share segments as part of the creation process. The process load files of the parent and child processes define the segments that are shareable. When the parent process requests the creation of the child via a message to the process manager, it identifies the segments to be shared and specifies the access permissions for the child process. A process can share segments with any number of different child processes.

Shared libraries consist of segments of text and data that can be incorporated into any number of processes. A shared library is formed

by the process loader, and its process load file is maintained in secondary memory. If a process wants to use a shared library, the load file of the process need only contain the name of the library. The segments of the library are loaded into the process when the process is created unless they have already been loaded to satisfy the creation of another process. Symbolic references are resolved when the process load file is formed. The use of shared libraries decreases the size of the text segments in the process load file.

### *5.2 Process termination*

Any process running under DMERT may elect to terminate itself or may be terminated by request of another process or the kernel. Exceptional cases are allowed; that is, the operating system has the capability of declaring processes to be essential or nonterminable. In particular, the processes that provide the basic operating system services are essential and cannot be terminated short of another system initialization. A kernel or supervisor process may be declared to be nonterminable as an option of the process loader. For such processes termination requests are always denied.

The operating system supports a variety of mechanisms for requesting process termination, as described below. However, regardless of the specific request mechanism used, the operating system takes the same basic actions.

#### *5.2.1 Termination actions*

All requests to terminate a process eventually result in a message being sent to the system scheduler specifying the PID of the process to be terminated. The termination request will result in the removal of the process from the system, unless it is a single-copy process with a usage count greater than 1, in which case the usage count is decremented. Removal of the process entails deleting its entry in the scheduling and segment tables. Shared segments that are needed by other processes for execution are retained in the system. The other segments are removed from the memory management tables and any associated secondary memory uses for swapping are freed up.

The operating system also takes some actions on behalf of the terminating process to free up other system resources that may have been allocated to it. Messages still queued up for the process are deleted or returned to the sender with a special acknowledgment. The process is detached from system ports and interrupts. For a supervisor process, the operating system issues close requests for any files it may have in the open state.

The message sent to the scheduler to terminate a process may optionally request a memory dump of the process. In this case, the

operating system produces a file in secondary memory similar to a process load file, which includes a copy of each of the process's segments at the time of termination. Such dumps are often useful for debugging purposes. Another option frequently exercised to coordinate the interactions of processes is to have the parent process notified when one of its child processes is terminated.

#### *5.2.2. Process termination request mechanisms*

A process may send a message to the system scheduler to terminate itself or another process. In certain situations, the operating system will terminate a process automatically. For example, a process that is faulted but that does not have a fault entry is terminated by the kernel.

A user process typically executes an exit system call to terminate itself. The exit system call is converted into a terminate message to the scheduler.

The operating system provides various OSTs to terminate a group of processes. One OST can be used by supervisor or kernel processes to terminate all processes having the same Utility Identifier (UID). (The UID is an identifier of a process that is selected at process load time and is administered to be unique to each process load file). This OST terminates all invocations of a multiple-copy process, since multiple invocations have the same UID. Another OST can be used by a kernel process to terminate all processes belonging to a specified process class. DMERT supports the optional grouping of processes into one of several classes, with each class chosen at load time. The capability of terminating all processes in a given class is useful in overload control schemes. Another termination OST available to kernel processes is specifically designed to handle system message buffer overload. This OST terminates processes that are using more than a specified percentage of the system message buffer resources. The actions of this OST also can be restricted to processes with a specified execution level or process class.

Several commands are provided to permit process termination from a terminal. One command will force the termination of any process with a specified PID and also will cause a memory dump of the process to be created. Another command is provided to terminate all processes with a specified UID. It also has the special feature of clearing the names of all shared segments used by the processes. This feature is used in conjunction with the updating of a process.[5]

## VI. SIMULATED *UNIX* OPERATING SYSTEM

The time-sharing environment supported by DMERT simulates a *UNIX* operating system[2] implemented via a supervisor process called the *UNIX* Supervisor Process (USP). The USP supports standard

*UNIX* software including system calls from C programs, file operations, process communication through pipes, and interpretation of terminal commands through the "shell" process.

Processes controlled by the USP are called user processes. The USP partitions its address space into a user area and a supervisor area and appears as a single process to the DMERT kernel. Thus, the user and supervisor are physically combined into the same process, having the same PID, scheduling priority, PCB, etc. Each time a user process forks, another USP is formed.

The role of the USP is to supply services to its user portion. It accomplishes this through supervisor OST calls and through communication with other DMERT processes. For example, file system capabilities are provided by the USP sending the appropriate messages to the DMERT file manager process.

The availability of a simulated *UNIX* operating system in DMERT allows *UNIX* programs from other processors to execute on the 3B20D Processor. DMERT provides some capabilities to user processes not currently supported by the standard *UNIX* operating system. These include asynchronous I/O directly to or from the user's address space and memory management of user process segments. In addition, there are a number of file system capabilities, such as contiguous files, that are provided through the DMERT file management facilities discussed in Section VIII. User processes also have access to the DMERT IPCs such as messages and events.

DMERT's memory management capabilities allow a user process to manipulate and share portions of its address space on a segment basis. In particular, a user process can create a new segment in its address space and can specify the virtual address of the segment. It can acquire an existing and named segment into its address space and also remove segments from its address space. Segments listed in a user process's PCB can be activated or deactivated through OSTs to the USP. OSTs permit it to share up to three segments with a process it creates via the DMERT process creation functions described earlier.

## VII. I/O FACILITIES

The operating system supports communication with peripheral devices through a set of drivers and device handlers. These drivers isolate most processes from the details of the peripheral system, and they ensure efficient use of the peripheral devices by scheduling access to them on an equitable basis.

The architecture of the I/O software closely resembles the I/O hardware architecture.[5] The I/O Processor (IOP) driver and the device handlers manage the IOPs, the Peripheral Controllers (PC), and the

Peripheral Controller Subdevices (PCSDs). The disk driver manages and controls the disk file controllers and the disks.

### *7.1 Input/output processor driver*

The IOP driver is a kernel process that administers all IOP transactions in the 3B20D DMERT system. The driver is responsible for normal I/O activities fault recognition and recovery, configuration management, and diagnostic access.

The IOP, from a software standpoint, can be visualized as a three-level structure (see Fig. 1). The "front-end" Peripheral Interface Controller (PIC) controls up to sixteen peripheral controllers (PCs), and

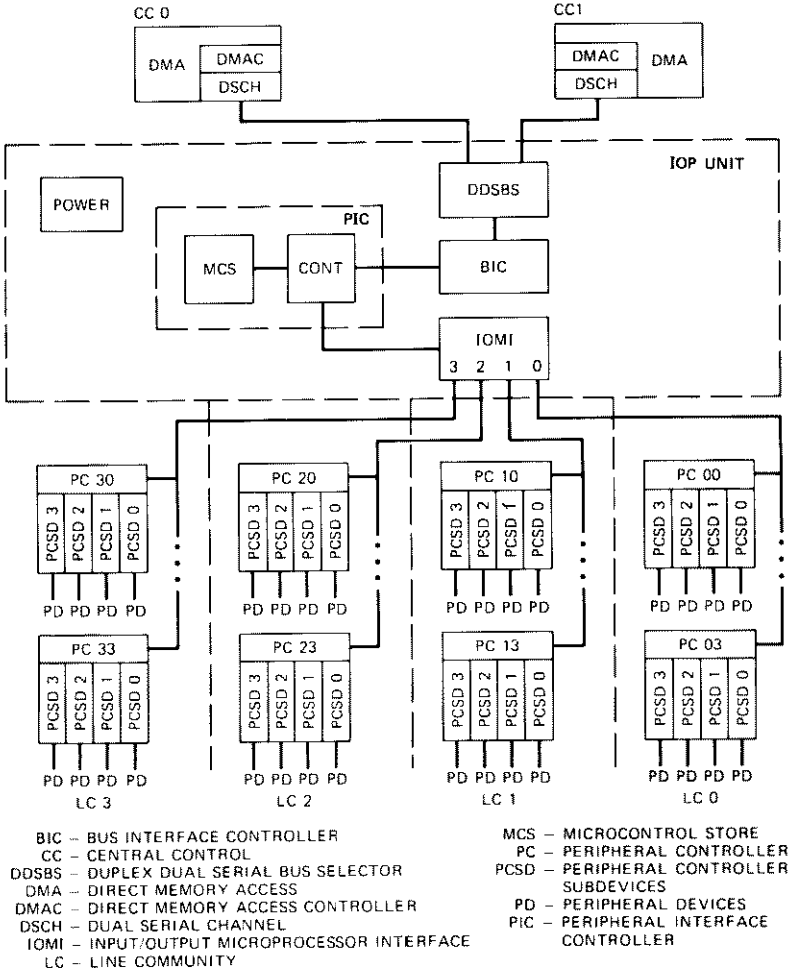



Fig. 1—Input/output processor.

each PC controls up to four subdevices (PCSDs). The subdevice provides the interface to the end device, such as magnetic tape unit, teletypewriter (TTY), data link, etc. Four PCs are combined to form a PC "community" with each PC community having a separate power supply.

Each element in the IOP (PIC, PC, and PCSD) has a corresponding element in the driver called a handler and corresponding unit control and option blocks in the Equipment Configuration Data (ECD) data base.[6] The handler for the PCSD is referred to as the device handler. A handler for a PC is called the generic PC handler or application PC handler. The handler for the PIC is called the generic PIC handler. The term generic implies that the handler is capable of performing all required handler functions for more than one PC type (that is, TTY, magnetic tape, data link, etc.).

Handlers are collections of C-language functions that have well-defined interfaces with the driver and are responsible for carrying out all maintenance (excluding diagnostics), recovery, and normal mode operations for their respective elements in the IOP. Handlers contain the necessary specialized logic to deal with a given unit type.

The handlers' service routines, input routines, control routines, and associated libraries form a single IOP driver process (IODRV).

IODRV can be subdivided into the following functional areas:

(*i*) Common service routines: routines that are frequently called from numerous points within IODRV and the handlers.

(*ii*) Configuration control: routines that maintain proper configuration of IOP units (inverted tree structure).

(*iii*) Input routines: routines that process primary inputs from the DMERT operating system and pass them off to IODRV configuration control or handlers.

(*iv*) Application and generic handlers: the operational interface between the user and physical device (magnetic tape, terminals, etc.).

(*v*) Maintenance handler: the diagnostic interface to the IOP units.

(*vi*) Archive libraries: system routines used by IODRV and other kernel-level processes.

Normal mode activities are carried out through the input routines, common service routines, and the operational handlers. All communications within the driver are through function calls.

I/O messages enter IODRV as message events and pass through the message input routines. Similarly, operating system traps enter IODRV at the OST entry point and pass through the OST input routines. Typically, I/O messages and OSTs contain a Logical Device Identification Number (LDIN) that identifies a logical (or virtual) device with which a user wishes to do I/O. IODRV maps the LDIN to

one or more physical devices. Once a physical device is identified, the IODRV can identify the corresponding handler via the ECD and pass control to it.

Completion reports, or responses, are deposited in the IODRV response queue by the IOP. If responses have been added to the response queue within a certain batching interval, the IOP will interrupt the 3B20D, causing IODRV to be entered. IODRV pops each queued response and, based on the PC and PCSD identifier in the response, maps to a physical unit, and passes control to the handler.

Maintenance and recovery activities are coordinated through IODRV configuration control routines, which, in turn, call on the handlers at appropriate points in time to carry out specialized maintenance operations at the subdevice level. Diagnostics for the PIC and PC are handled exclusively by the maintenance handler.

#### *7.1.1 Handler applications*

Peripheral devices supported by the IOP include TTY terminals, Maintenance TTY (MTTY) terminals, magnetic tape drives, data links, and the Scanner and Signal Distributor (SCSD). Interfaces to these devices are provided by IODRV and device handlers. IODRV is responsible for initializing units upon bootstrap and removing and restoring units upon manual requests or faults. Each handler-peripheral device combination determines the interface mechanism and the set of features to be supported. The following sections give a brief description of the facilities supported by each peripheral device type.

#### *7.1.2 Terminal devices*

The Craft Interface Handler (CIH) provides access to terminal devices. The CIH communicates with two types of controllers: Maintenance Terminal Controllers (MTTYCs) and Terminal Controllers (TTYCs). MTTYCs support four subdevices: a Maintenance Terminal (MTTY); a Receive-Only Printer (ROP); a Switching Control Center (SCC) interface; and an Emergency Action Interface (EAI).[7] TTYCs support terminals (TTY).

The MTTY, TTY, and ROP are known as terminal devices. The SCC and EAI devices are not terminal devices and are accessed via other handlers (see below). All standard terminal operations supported by the *UNIX* operating system are available to TTY devices, including read, write, open, and close requests, which are supported through a message interface. The ROP does not support reads.

#### *7.1.3 Data links*

The Communication Protocol Handler (CPH) provides access to synchronous data links. The CPH is designed to communicate with

two types of peripheral controllers: (*i*) the MTTY controller, and (*ii*) the synchronous data-link controller. In the case of the MTTY, access is available only to the SCC peripheral controller subdevice, which supports synchronous data link communication between the 3B20D and an SCC office using the BX.25 protocol.

The synchronous data link controller supports the BX.25 link layer (level 2) communication protocol and the Digital Data Communication Message Protocol (DDCMP) through the use of different versions of peripheral controller software. The CPH software supports two access methods: link-layer protocol access (level-two-only access) and BX.25 packet-level (level 3) protocol access. The use of a link-layer protocol (BX.25 and DDCMP) assures the integrity of data transmissions on a physical link. The use of a packet-layer protocol (BX.25) allows the added capability of multiplexing multiple users on a physical link. Flow control procedures also are used on both protocol layers.

In addition to the the two different access methods, the handler supports both a simplex and a duplex link configuration. In the duplex configuration, two physical links make up a logical communication path between the 3B20D and another system. The CPH automatically routes data through the currently active physical link. Link switching is done automatically when the active link fails.

#### *7.1.4 Magnetic tape drives*

The magnetic tape peripheral controller handles up to four 9-track 800 or 1600 bits per inch (bpi) tape drives. The magnetic tape handler provides the interface to this controller and supports open, read, write, seek, and close requests through a message interface. Seeks are not supported for write operations.

#### *7.1.5 Scanner and signal distributor*

Administration and control of the Scan and Signal Distributor (SCSD) points currently involves two DMERT kernel processes: the SCSD administrator and the SCSD handler. The latter is an integral part of the I/O driver process. The primary function of the SCSD software is to provide an interface enabling client processes to manipulate distribution points and receive information about the state of the scan points (i.e., autonomous scan state transition and directed scan reports). The SCSD administrator allows a client process to identify SCSD points by logical or physical addresses; logical addressing allows applications to code software independently of physical cabling.

The SCSD handler translates messages from the administrator into SCSD controller commands and receives responses from the controller

and forwards these responses to the administrator through a message interface.

#### 7.1.6 Direct user interface

For some applications the current method of communication with the peripheral controller subdevices through IODRV is not efficient enough to meet their needs. Therefore, the Direct User Interface (DUI) exists to expedite data transfers between an application process and a specialized 56-KB BX.25 data-link controller.

The DUI handler is an integral part of IODRV. In the normal mode of operation, the only functions of the DUI handler are to set up and clean up the DUI table, which is in a common area of memory and is used for passing commands and status information between the application process and the peripheral controller. Using the DUI table, jobs are passed directly to the peripheral controller by the application process without any intervention from IODRV.

A secondary function of the handler is to administer the fault recovery strategy for the peripheral controller subdevice. If the subdevice has to be removed or restarted, the handler will tear down the DUI table and send a message to the application process.

### 7.2 Disk driver

The disk driver is a kernel process that handles all normal disk I/O and all maintenance disk I/O. Only system initialization I/O bypasses the disk driver and transfers information directly from the system boot device to main memory. The disk subsystem consists of the disk driver, the Disk File Controller (DFC), and the Moving Head Disk (MHD) drives. The 3B20D supports a maximum of eight DFCs, each having up to eight MHDs. The DFC and MHDs are described in Ref. 8.

MHDs may be used in a simplexed or duplexed configuration. In simplex mode the MHD stands alone. Should a file become damaged it will be irretrievably lost. In duplex mode two MHDs are maintained such that each is an exact copy of the other. Should one disk fail the other can be used in simplex mode.

#### 7.2.1 Operational characteristics

The disk driver handles open, close, read, and write message requests. Open and close messages are passed from the file manager (see Section VII) to the disk driver, while read and write requests may be sent directly from any process or routed through the file manager. Before the kernel attaches the read or write message to the disk driver's message queue, it verifies that the segment is locked in main

memory (see Section 3.1). It also verifies that the I/O transfer is within the bounds of the segment.

When the disk driver processes the I/O message, it translates the LDIN contained in the message to one or more physical devices. The request is then placed in one of three circular job submit queues in main store associated with the DFC for each specified physical device.

The three types of disk job queues are high-priority, base-priority, and special. Special commands sent by maintenance processes or originated in the disk driver are immediately executed by the DFC from the special job queue. High-priority jobs can be sent by any client process and will be processed by the DFC before base-priority jobs. All other jobs are placed in the base-priority queue.

Whenever the DFC completes a job requested by the driver, it returns a response indicating the outcome of the job. All job responses are placed in a single main store response queue, regardless of the priority of the original job. The DFC generates an interrupt to the driver after each response is added to the queue. The driver only clears the interrupt after processing the last entry in the response queue.

The disk driver handles job responses each time it is entered at its interrupt entry. The job response indicates the status and identity of the job being reported. If the job was successful, the driver sends a successful job completion acknowledgment to the client using the same message buffer that requested the I/O. In writing to duplexed disks, the driver guarantees that both disks were written successfully before acknowledging the job. In reading from duplexed disks, the driver reads from a single disk, alternating disks between requests.

If a job failed, the driver determines whether the device should be removed from service or if it should retry the job.

When the driver wishes to retry a failed job, it sets up a retry request in the main store retry queue. The format of an entry in the retry queue is the same as that of a job in any of the other queues. After writing the entry in the queue the driver wakes up the DFC with a programmed I/O command. The DFC then takes this job, even if the other submit queues contain work. When the driver handles the response from the retry request, it knows the queue is available for reuse.

#### *7.2.2 Reliability characteristics*

The disk driver also has a message interface for maintenance commands. Once the device (MHD or DFC) is taken out of service by the disk driver, the device can be reserved for maintenance access and the disk driver provides the maintenance client processes unlimited access to the device. During maintenance, specific areas of a MHD can be read or written by bypassing many of the operational checks performed

on normal I/O requests. This allows the creation of a disk and the system update of a disk with a new software generic.[6]

### VIII. FILE SYSTEM

All accesses to the file system are done through the file manager, a DMERT kernel process. In addition to maintaining file system security and integrity, the file manager translates read and write requests within the file system to physical I/O requests on the disk.

The DMERT file system is similar to the file system provided by the *UNIX* operating system and features a hierarchical structure, byte-oriented files, and uniform access to files, directories, and periphery. In addition to regular files, which are scattered throughout the disk and can grow dynamically, DMERT also provides contiguous and extent files, which are contiguous on disk but have limits on their growth. Contiguous and extent files are optimum for data base and object files, where large, fast I/O transfers are needed. For field update, DMERT provides a "windowless move" facility, which automatically moves an updated object file over the old one, thus eliminating any possibility that the file be used or the system initialized while the file is in an inconsistent state.

To meet DMERT's reliability requirements, DMERT file systems are crash resistant. In particular, a crash does not jeopardize file system integrity, the file systems do not need manual repair, and they are available within seconds after a crash.

The file manager uses two techniques to ensure crash resistance. First, it orders all writes to disk to maintain a consistent file system state. To create, link, or write a file, the ordering is:

(*i*) Write the data blocks
(*ii*) Write the indirect i-node blocks
(*iii*) Write the i-node*
(*iv*) Write the directory entry, if necessary.

To unlink or truncate a file, the ordering is:

(*i*) Write the cleared directory entry, if necessary.
(*ii*) Write the cleared i-node.
(*iii*) Free the blocks.

Second, to ensure that no block is allocated to more than one file, the file manager rebuilds a file system's free-block list before it is used following a crash. Doing this for the 50-000 block, 2048-i-node *root* file system adds about 10 seconds to DMERT's boot procedure.

These two techniques are sufficient to ensure crash resistance, and we have found no problems with these in the field.

---

* An i-node describes a file and contains its block addresses. An indirect i-node block extends the i-node and contains more block addresses.

## IX. SUMMARY

This article has described the DMERT nucleus, which consists of the kernel, the special processes, the I/O drivers and file manager, the process manager, and the *UNIX* supervisor. The major services provided by this nucleus include a multitude of interprocess communication mechanisms, a sophisticated set of memory allocation features, both real-time and time-shared scheduling, dynamic process creation and termination, a simulated *UNIX* environment's communication with terminals, magnetic tape drives, data links and disks, and powerful real-time and time-shared file system capabilities. The operating system has been continually evolving since DMERT was conceived, and is expected to continue to evolve over the next few years. This article has described the first official version of DMERT, which entered service in the Bell System during September 1981.

## X. ACKNOWLEDGMENTS

We thank C. J. Antonelli, J. Q. Arnold, T. P. Bishop, R. W. Fish, N. A. Martellotto, R. R. Snead, P. J. Stankus, R. M. Venzon, and R. E. Yuknavech for their contributions to this document. Special thanks go to J. J. Wallace for his contributions to the file manager section.

## REFERENCES

1. J. R. Kane, R. E. Anderson, and P. S. McCabe, "The 3B20D Processor & DMERT Operating System: Overview, Architecture, and Performance of DMERT," B.S.T.J., this issue.
2. D. Ritchie and K. Thompson, "The *UNIX* Time-Sharing System," B.S.T.J., *57*, No. 6, Part 2 (July–August 1978), pp. 1905–29.
3. I. K. Hetherington and P. Kusulas, "The 3B20D Processor & DMERT Operating System: 3B20D Memory Systems," B.S.T.J., this issue.
4. B. R. Rowland and R. J. Welsch, "The 3B20D Processor & DMERT Operating System: Software Development System," B.S.T.J., this issue.
5. A. H. Budlong and F. W. Wendland, "The 3B20D Processor & DMERT Operating System: 3B20D Input/Output System," B.S.T.J., this issue.
6. R. H. Yacobellis, J. H. Miller, B. G. Niedfeldt, and S. S. Weber, "The 3B20D Processor & DMERT Operating System: Field Administration Subsystems," B.S.T.J., this issue.
7. M. E. Barton and D. A. Schmitt, "The 3B20D Processor & DMERT Operating System: Craft Interface," B.S.T.J., this issue.
8. R. E. Haglund and L. D. Peterson, "The 3B20D Processor & DMERT Operating System: 3B20D File Memory Systems," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# Field Administration Subsystems

By R. H. YACOBELLIS, J. H. MILLER, B. G. NIEDFELDT, and S. S. WEBER

(Manuscript received March 10, 1982)

*This article describes the field administration facilities of the Duplex Multiple Environment Real Time (DMERT) operating system, as provided on the 3B20D Processor. These facilities are: Recent Change/Verify, the subsystem that allows manipulation of office-dependent configuration information; Field Update, the software change mechanism; and System Update, the component used to install a new generic program in an office. The article also includes information on how these capabilities fit into the overall scheme of field support in an in-service office environment.*

## I. INTRODUCTION

An integral part of high-reliability applications of the Duplex Multiple Environment Real Time (DMERT) operating system is the administration of system hardware information and of software. This includes both the initial delivery of the system as well as subsequent upgrades. In DMERT[1] there are three commonly used capabilities to apply, track, and administer such changes. These are Recent Change/Verify, Field Update, and System Update. They are listed in this order according to decreasing frequency of field use and increasing impact (typically) on the overall system. Each of these capabilities is designed to permit display of some aspect of the current status of the system, to change that status in a simplified and highly reliable way, and to either reverse such changes or make them permanently a part of the system. This article discusses each in turn, and provides examples of their use. Each capability may form the base for an application-dependent version of its function. These functions are discussed briefly in the rest of this introduction.

The 3B20D Recent Change/Verify (RC/V) system provides the ability to change and manipulate various aspects of office-dependent information. This capability is focused on the system hardware and software configuration and is based on the Low-Level Access (LLA) Data Base System, whose operation is normally hidden from field-site administrators. RC/V is used manually or automatically to verify and change the hardware and software components known to the system, and the ways in which they are interconnected.

Field Update is used to correct problems in the operation or functionality of the system. Field Update is the official fix mechanism for DMERT. Rapidly installed emergency fixes, as well as more routine trouble corrections, may be installed into the software or other files in DMERT via Field Update.

Finally, System Update, also known as Generic Update, changes a major portion of the entire DMERT or application generic program. In doing so, System Update may write over old generic information or provide a completely restructured generic program image. Typically, a new generic release will involve a new structure for RC/V information as well, so RC/V may be involved with such an update. The following sections provide more details on these fundamental administrative capabilities of DMERT.

## II. RECENT CHANGE/VERIFY—LOW-LEVEL ACCESS DATA BASE SYSTEMS

The 3B20D/DMERT System has provided a data base management capability as part of the DMERT operating system. Built upon a Low Level Access (LLA) data base system are the Equipment Configuration Data Base (ECD), System Generation Data Base (SG), and the 3B Recent Change/Verify (RC/V) and Data Base Evolution Systems. This section describes these systems and their relationship to the field administration environment.

### *2.1 Low-Level Access Data Base System*

The Low-Level Access Data Base System organizes and manipulates data in a C-language environment. The name low level implies that the system places minimal restrictions on its users: decisions about data organization and retrieval are left to the application. LLA trades user convenience for greater flexibility in data base design and performance tuning.

LLA gives the user latitude in defining both data units and data models and provides a powerful set of primitives to access the data. System characteristics include:

(i) Data definition via a hierarchy of abstract types

(ii) Specification of data mapping from the data base to the user's buffers

(iii) Ability to select various access methods, i.e., logical organization of subsets of data

(iv) Data access through a library of functions

(v) Isolation of operating system dependencies in a small number of program modules.

Figure 1 gives a simplified schematic of the operation of an LLA application.

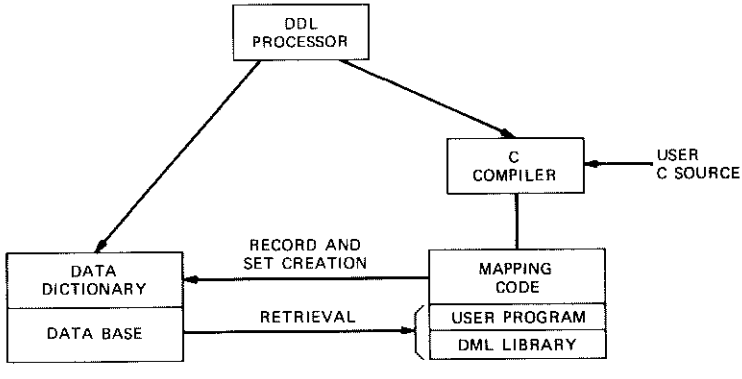
#### *2.1.1 Data definition*

The Data Definition Language (DDL) is used to define the "shapes" of records, the LLA data type for retrieval and storage. It also allows user-defined "views" of the data base via data mapping, and the specification of data models by associating records with access methods. The recognizer for the DDL, the Data Definition Language Processor (DDLP), has many C-compatible features, such as common syntax for preprocessor lines, comments, identifiers, constants, and type definitions. The DDLP generates C code to implement data mapping and C definitions, and a data dictionary to describe data types.

#### *2.1.2 Data manipulation*

The Data Manipulation Language (DML) is a library of functions that perform actions on instances of the data types defined by the DDL. The DML provides the following facilities:

(i) Creation and deletion of instances of data types




DDL – DATA DEFINITION LANGUAGE
DML – DATA MANIPULATION LANGUAGE

Fig. 1—Low-level access application.

(ii) Retrieval and update of existing instances of data types
(iii) Gathering of information about existing data instances.

These categories exist for instances of data bases, sets, and records. Generally, the lifetime of an instance of a data type starts with creation, proceeds through several retrievals and updates, and ends with deletion.

LLA is not used directly by a field administrator. Instead, the creators of various LLA data bases, be they 3B20D/DMERT system programmers or 3B20D application designers, provide appropriate higher-level access to their particular LLA data base application.

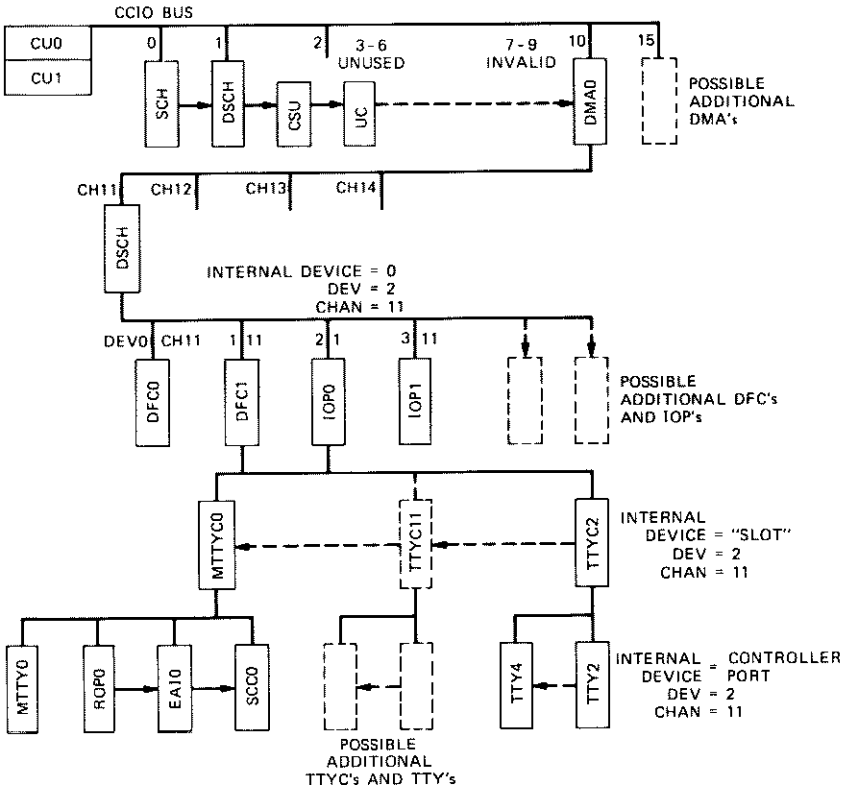
### *2.2 3B20D Data Base Recent Change and Data Base Evolution Systems*

#### *2.2.1 3B20D data bases*

The 3B20D/DMERT operating system has two major LLA databases. The Equipment Configuration Data Base (ECD) describes the processor and peripheral hardware configuration, while the System Generation (SG) Data Base describes the system parameters, boot processes, and disk image and ECD administration information. The concept of a data base was adopted to eliminate redundant device information, provide a unified approach to handling and accessing that information, and provide easy methods for generating and changing it.

Records in the ECD data base represent the hardware devices in the 3B20D Processor system, such as the Control Unit (CU) and Input/Output Processor (IOP), and are logically linked in a manner analogous to the physical linkages (see Fig. 2). In addition, records are provided to organize physical devices as logical devices and to maintain error counts for each physical device. To provide rapid access, the ECD is always kept in main memory.

The information in the ECD and SG data bases is used by several classes of users. The DMERT operating system, itself, forms one set of using processes and includes the device drivers, processor and peripheral diagnostics, and processor and peripheral fault-recovery programs. The second class of users of these data bases is the human user, whether that person be a Bell Laboratories' application designer adding new peripherals to the ECD or an operating company craft preparing to add more memory to an on-line 3B20D in the field. Two types of access have been provided for these two classes of users: The DMERT operating system processes access the ECD through a collection of LLA primitives that provide rapid access to those specific items required, for example, by the device drivers. Human users utilize the Recent Change/Verify system, which provides a forms-oriented input, via a cathode ray tube (CRT) terminal. The user may create, change, delete, or merely review the forms. Error and consistency checking is provided at the time of initial entry and before storage into the data base.




CCIO – CENTRAL CONTROL I/O BUS
CH – CHANNEL
CSU – CACHE STORE UNIT
CU – CONTROL UNIT
DEV – DEVICE
DFC – DISK FILE CONTROLLER
DMA – DIRECT MEMORY ACCESS
DSCH – DUAL SERIAL CHANNEL
EAI – EMERGENCY ACTION INTERFACE
IOP – INPUT/OUTPUT PROCESSOR
MTTY – MAINTENANCE TERMINAL
MTTYC – MAINTENANCE TERMINAL CONTROLLER
ROP – RECEIVE ONLY PRINTER
SCC – SWITCHING CONTROL CENTER
SCH – SERIAL CHANNEL
TTY – TERMINAL
TTYC – TERMINAL CONTROLLER
UC – UTILITY CIRCUIT

Fig. 2—Prototype 3B20D configuration.

### 2.2.2 3B20D Recent Change/Verify

The Recent Change/Verify system is built upon the LLA data base management system and utilizes the LLA primitives for accessing and managing its two DMERT data bases. There are three basic components of 3B20D RC/V (see Fig. 3). The first is the front-end form processing system. This component is known as the On-line Data Integrity (ODIN*) subsystem. ODIN allows the various forms to be specified through a series of CRT screen mask definitions and for each

* ODIN is a product of Western Electric Company.

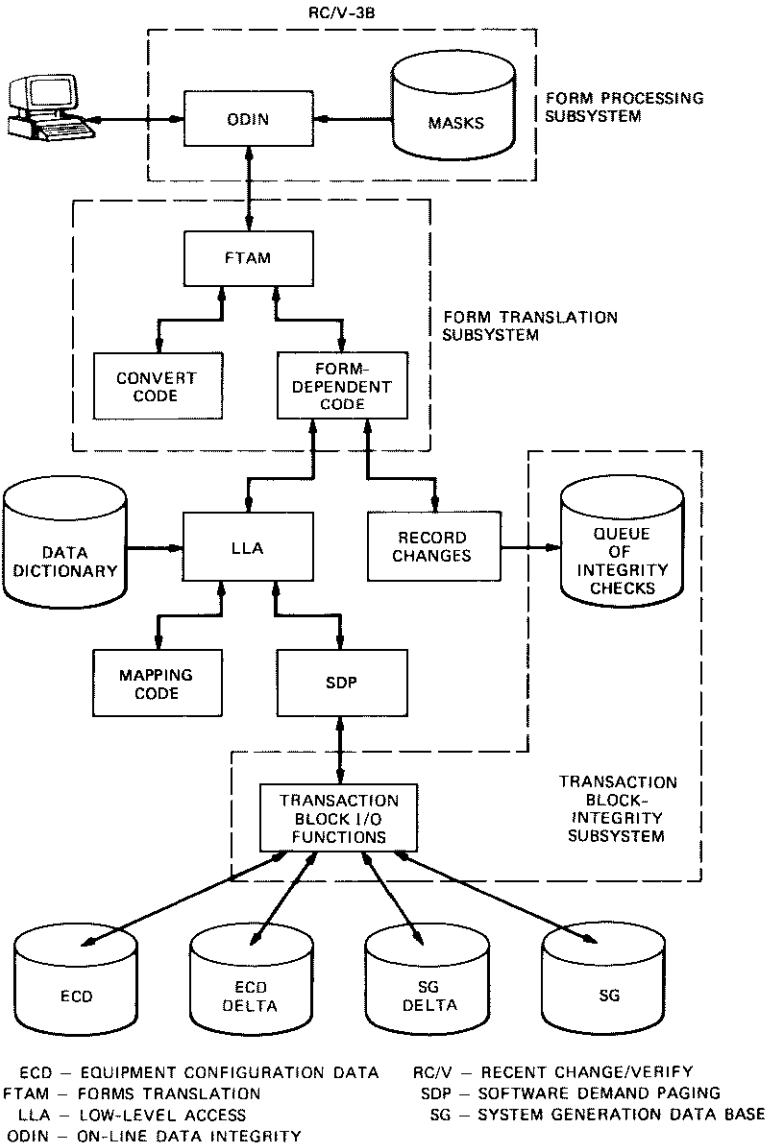



Fig. 3—Components of Recent Change/Verify.

of these definitions to contain certain syntactic information to be checked upon entry. For the ECD/SG data bases there are 36 different form types, each of which has an associated mask definition. Most forms are either ECD or SG forms, but there are a few that are directives for the RC/V or Evolution systems. For each form type some error checking is provided. The second fundamental component

of RC/V is the Form Translation and Mapping subsystem. This takes the output of ODIN and transforms it into LLA record definitions and access functions. Then the LLA functions are used to actually manipulate the data in the ECD and SG data bases. The third component is the transaction block-integrity check subsystem. This provides a mechanism for checking consistency between forms. RC/V has implemented the concept of a "transaction." Two special forms delimit a transaction. Upon processing a transaction-end form, RC/V invokes the integrity checks as well as linking the new information into the data base.

As we stated earlier, the ECD that describes the running 3B20D is always in main memory; however, there is also a copy on the disk. In order for a change to be made permanent it must be applied to the disk as well as the memory version. To maintain the integrity of the ECD, changes are soaked on the memory version (test state) before they are applied to the disk version (active). A special form has been provided to perform this final step of activating changes to the disk copy of the data base. Upon processing of this form, RC/V copies the main memory copy of the ECD to the disk. To facilitate error checking and correction, a journal file of all transactions is kept on-line and can be printed on the Receive-Only Printer (ROP) at the request of the office craft. Also, an error log file is maintained and a periodic audit of the ECD structures is performed.

#### *2.2.3 Data Base Evolution System*

Because the release of a new 3B20D/DMERT generic is anticipated to be associated with changes to the ECD or SG forms or the LLA primitives, a system for transforming these data bases has been provided. The Data Base Evolution system (DBEVOL) allows this transformation to occur in a regular and uniform manner without special programs needing to be written. DBEVOL allows old data to be restructured, new data fields to be added to existing forms, and old data to be deleted or changed. DBEVOL also provides semantic hook functions that allow applications to tailor some specific information before completing the data base evolution.

DBEVOL has two types of steps. The first set is characterized as pre-processing. Here a translation data base (also an LLA data base) is built on a host support processor. The inputs are the old and new form specifications (as used by RC/V) and a specification of the changes in Form Translation Language. These inputs are supplied with the new DMERT generic program. If semantic hook functions are required by the application they are also an input to the final translation data base. A translation data base matching the required changes in the standard DMERT ECD is also released with new DMERT generics.

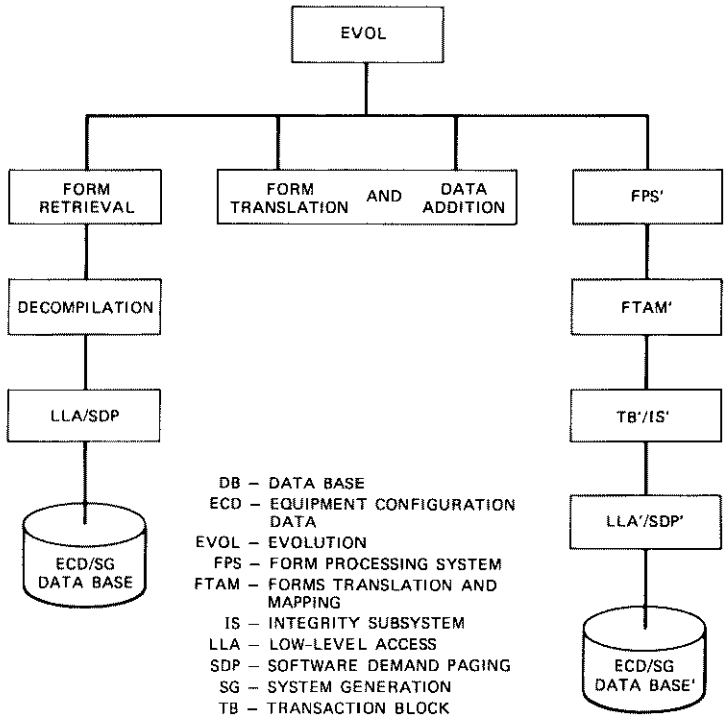



Fig. 4—Evolution of 3B20D/DMERT Data Base Management System.

The second set of actions are run-time steps that produce a new, evolved ECD/SG data base pair (see Fig. 4). The first step is a dump of the old ECD using the "old" existing generic RC/V. This is produced using one of the special forms provided by the RC/V system. Then this snapshot of the old data base is translated into a snapshot of the new data base. The "new" RC/V is then used to load the new data bases into the proper LLA format for the 3B20D.

DBEVOL runs on both the support processor and the 3B20D giving the using applications considerable flexibility in choosing a strategy for performing data base evolution. The evolved data base is actually put in place on the running 3B20D during the generic update scenario described below.

## III. FIELD UPDATE

Field Update, which is typically called "overwriting" in traditional Electronic Switching Systems (ESSs), is the problem correction mechanism for DMERT. While overwriting usually applies specifically to program bugs, Field Update may be used to correct any file on the 3B20D disk. Such files may contain human-readable text or binary

tables, for example. (In DMERT, files are structured like a *UNIX** operating system file system.[2]) Field Update must perform this updating without disturbing call processing or other critical system functions. Since operating systems do not normally support this style of updating, some difficult technological problems had to be overcome in designing and implementing Field Update. Some of these problems and their solutions are described below, followed by a more general discussion of the overall structure and use of Field Update.

### 3.1 Problems and solutions

Like most modern operating systems, DMERT supports the concept of a process, which is a collection of tightly coupled executable programs. Programs are in turn broken down into units that perform specific activities, called functions. Processes can communicate with each other, generally at "arms-length," and are normally protected from each other by DMERT software and the 3B20D hardware and microcode. Since Field Update runs as a cooperating set of processes within DMERT, some highly specialized operating system interfaces were required to break through this protection. Furthermore, the real-time critical processes in DMERT or its applications must run continuously [they are termed "non-killable" (NK)], so that they are always available to process events quickly. The running process images of such processes must be accessible and changeable in main memory, again via special operating system functions.

Since a process is a collection of functions, the C-language[3] function was chosen as the unit of update. The implementation of field update specified that there be a single reference point for each changed function, so as not to require changes everywhere such a function was involved. To solve this, the concept of a Transfer Vector (TV) used in ESSs was implemented within a process image. Figure 5 is an example of a simplified process image showing this. In Fig. 5, the TV area contains a list of the addresses of the process's functions. When a change is made to function f, the new version f′ is written into a special "patch" area provided with the process, and the particular address in the TV area is switched to point to f′ (see Fig. 6). This solution also allows the fix to be backed out by changing the address in the TV back to its original value. When the fix has been tested and is ready to apply permanently, the space occupied by f can be made available for future fixes (Fig. 7). While this concept is simple, introducing TVs to DMERT had operating system implications down to the microcode level. With TVs, the impact of introducing a new or changed function

* Trademark of Bell Laboratories.

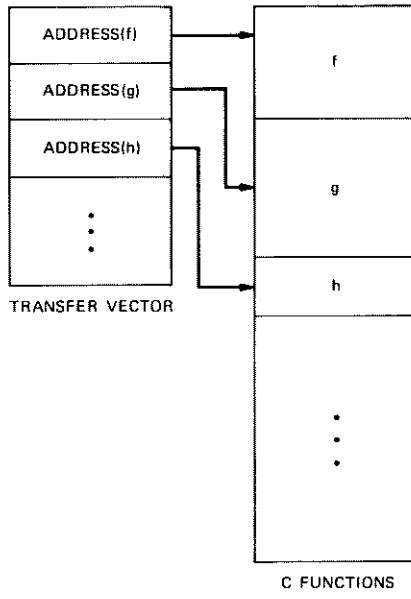



Fig. 5—Simplified DMERT process image.

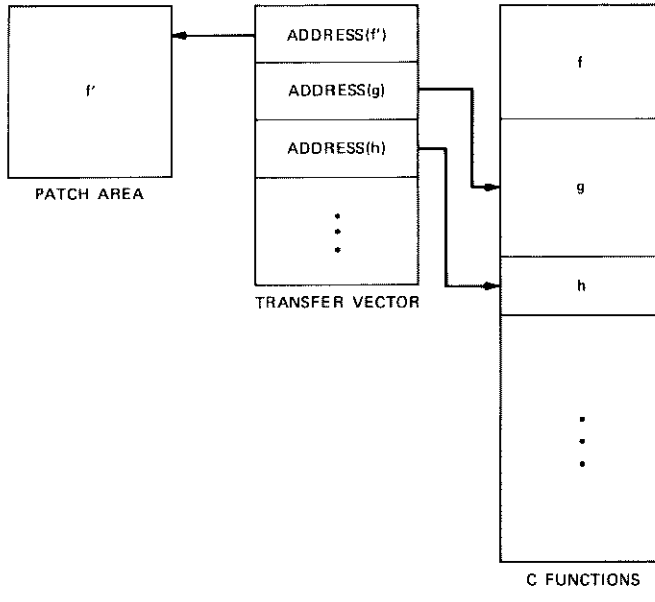



Fig. 6—Function f replaced by function f'.

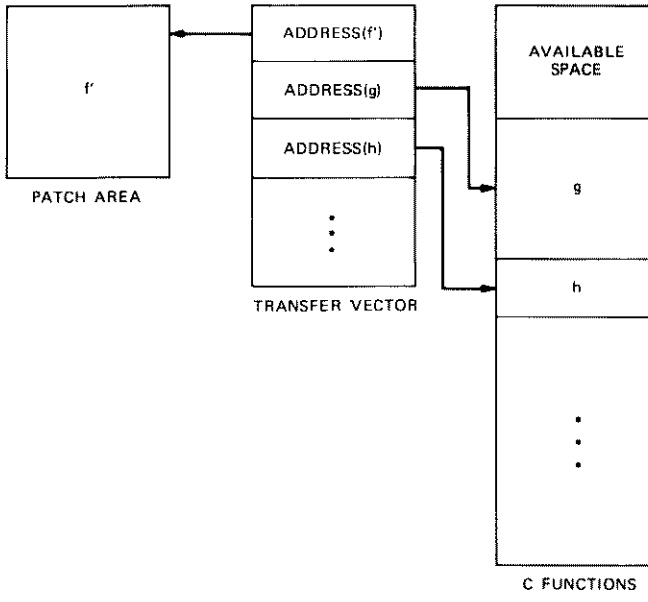



Fig. 7—Reclaiming the space occupied by function f.

has been restricted to a small, well-defined area of the process, making this activity inherently more reliable.

Traditional operating systems do not have the ability to change a critical function or process while the system is running. Since DMERT is derived from such an operating system, many challenges were encountered in providing the field update capability. Some specific areas included:

(i) The ability to change a file both instantaneously and in a temporary way. This is used in updating both non-killable processes and more routine processes that can be terminated and restarted;

(ii) Retention of sufficient symbolic information to properly update the 3B20D disk-resident versions of processes ("pfiles");

(iii) The ability to update C functions even though the old versions of the functions had been suspended while field update was running;

(iv) The ability to change data contents or the structure of data used by a continually running process;

(v) The ability to coordinate changes to functions within a process.

### *3.2 The use of field update*

Field Update is an end-to-end concept within DMERT; that is, it is involved with the development, distribution, installation, and tracking of changes. When a process is first introduced into DMERT, or when its subsystem architecture changes, the process developer must com-

municate its characteristics to personnel who administer the DMERT source programs. The developer also must create a script of commands to be executed at a field site, which will be used to install, back out of, or make permanent a fix to the process. Generally, this will be simple to do because there are categories of existing process scripts, and new processes will fit into an existing category (or a simple modification to one will suffice). Once these steps are taken, the developer can depend upon the DMERT administrative system[4] and specific Field Update change development commands to remember these details. This approach standardizes the development of fixes so that each is handled the same way, as opposed to being a unique activity. The primary advantage comes when an emergency fix must be created quickly without the extra burden of collecting procedural information.

When a developer has created a fix and tested it, the standard change development mechanisms produce a package called a Broadcast Warning Message (BWM), which is used to transmit and install the fix (see Fig. 8). System Test personnel use this package to test the

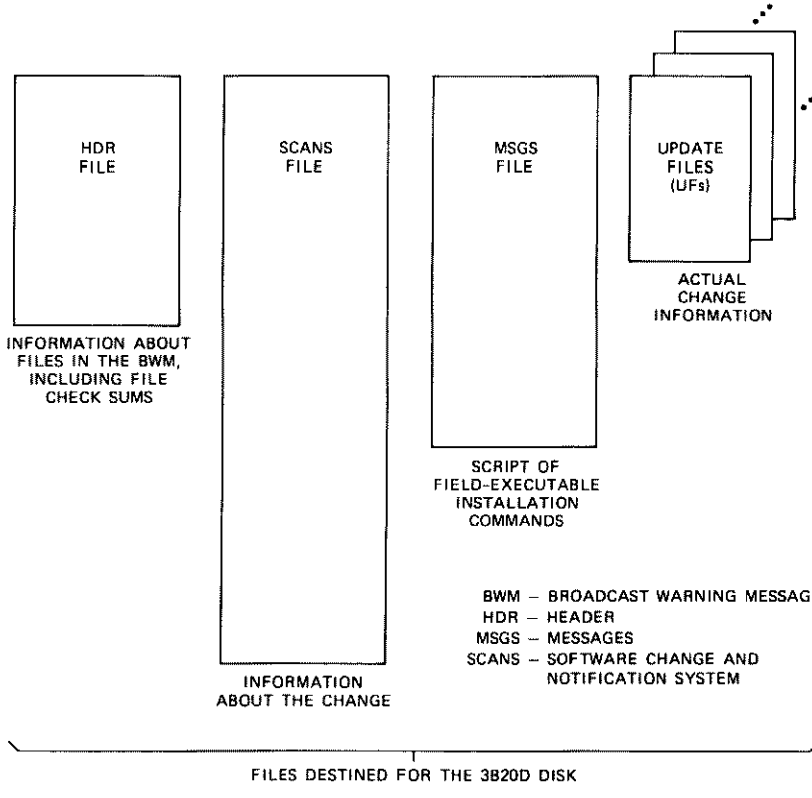



Fig. 8—Structure of a broadcast warning message.

field updatability of the fix as well as its impact on the system in the same way it will be installed at a field site (see the article on "System Integration and Test" in this issue of the Journal). When testing is completed, the fix can be packaged together with other fixes via automated tools into an official BWM for delivery to application project personnel, who will intermix it with application-specific BWMs and send it on. During this packaging, the particular order of installation of specific fixes is indicated both within and across BWMs.

A BWM consists of a set of files in a *UNIX* operating system directory, and can be transmitted via magnetic tape to a site. The Bell System is standardizing on the Software Change and Notification System (SCANS-II) as the official change distribution network, and the files in a DMERT BWM are also compatible with SCANS-II. DMERT also provides file reception software for use with SCANS-II. Typically, personnel at a Switching Control Center (SCC) will interrogate SCANS-II, recognize that a change is pending for one of their associated field sites, and initiate transmission of the change to the field site.

Once a change reaches a field site, it is stored in a staging area on disk until it is manually installed. The developer-produced script of commands is sent as part of the BWM (see Fig. 8), and is used by office personnel to install the change. With a short sequence of DMERT Field Update commands, the fixes can be:

(i) Installed
(ii) Tested
(iii) Backed out or made permanently a part of the system.

While a fix is being installed, an internal system error will result in automatically backing it out; once it is soaking in a temporary state, it may be backed out manually, or automatically if the system undergoes a major recovery action.

Each field site maintains an on-line log of all Field Update activity since the last System Update (see Section IV). This may be used to verify the current state of the office as far as installed BWMs are concerned, and is used each time a new change is installed to guarantee proper sequencing of changes. Other Field Update-related utility programs in DMERT can be used to print out a C function-to-process address map, and to verify that the main memory (executing) copy of a process matches its image on the 3B20D disk (see Section 3.1).

By the facilities mentioned above, Field Update allows fix creation in a style compatible with normal program development, prepackaging of developer-approved installation scripts, fix coordination both within and across BWMs, automated delivery and installation mechanisms, and detailed change tracking. These capabilities make Field Update a truly end-to-end DMERT change mechanism.

### 3.3 Field update example

Let us presume for this example that a problem has been found in the DMERT disk driver program, whose pfile is called dkdrv.o in directory /bootfiles. The developer has constructed a fix and tested it, and further system impact testing has verified it. The fix is given a DMERT official BWM name of BWM82-0028 (the first two digits are the year, and the last four a sequence number), and is passed to personnel in an application of DMERT, who approve it and send it out as application BWM, BWM82-0037. Once the fix has arrived at a field site, it is installed via the commands shown in Fig. 9. The descriptions below explain the commands:

(i) Request a printout of change information that field update has logged against process dkdrv.o.

(ii) Prepare the site to receive the BWM. After SCANS-II receives a command to send the BWM (not shown), it is transmitted automatically to the site with data error detection and positive reporting.

(iii) Install the fix into the system.

(iv) Test the fix (coupled, perhaps, with manual actions).

(v) Make the change permanent and remove the BWM files from the system. In this particular case the DMERT boot image is rebuilt as part of making the fix permanent, because the changed process is one of the system boot processes.

(vi) Once again display the change status of dkdrv.o.

(vii) Print a map of C functions and their addresses for drdrv.o.

(viii) Reclaim the space occupied by old versions of C functions in dkdrv.o.

The installation command mentioned above causes an entire set of commands to be executed, those in the "install" section of the script originally provided by the developer. An example of that script is shown in Fig. 10, which shows the Messages (MSGS) file for BWM 82-0037.

```
(i)    UPD:DISPLAY; FN "/bootfiles/dkdrv.o"!
(ii)   IN:REMOTE:START!
       VFY:BWM: 82-0037!
(iii)  UPD:BWMNO 82-0037!
       UPD:EXEC 82-0037: CMD APPLY!
(iv)   UPD:EXEC 82-0037; CMD SOAK!
(v)    UPD:EXEC 82-0037; CMD OFFICIAL!
       CLR:BWM:ALL!
(vi)   UPD:DISPLAY; FN "/bootfiles/dkdrv.o"!
(vii)  UPD:TRC; FN "/bootfiles/dkdrv.o" : ALL!
(viii) UPD:AUD!
```

Fig. 9—Commands to Receive and Incorporate BWM 82-0037.

```
  APPLY.
                MRs: d8200002; DMERT BWM82-0028
UPD:UPNM BWM82-0037;FN"/bootfiles/dkdrv.o":UF"/etc/bwm/82-0037/one.m"!
  SOAK.
                The fix(es) should soak for at least 1 days 00 hours 00 minutes.
                It will be apparent that the fix(es) have been applied:
                When no disk restore failures occur,
                commands to soak the fix appear here.
  BKOUT.
                If the fix results in the need to reboot the system, the fix will
                have been backed out automatically. If the fix does not result
                in a reboot but otherwise does not work correctly, it can be backed
                out by entering the command [s]:
UPD:BKOUT;UPNM BWM82-0037!
  OFFICIAL.
UPD:UPNM BWM82-0037;OFC!
                This will update the bootfile APPDMRT.
```

Fig. 10—MSGS file for BWM 82-0037.

## IV. SYSTEM UPDATE

DMERT System Update provides a safe, reliable mechanism for field personnel to introduce new versions of DMERT and application software into 3B20D/DMERT systems, while minimizing service disruption. System Update differs from Field Update in the magnitude of the program and data changes being installed. Normally, a system update will replace all the software in the system with the release of a new generic program, which is a complete reissue of DMERT and/or application software and/or data. For this reason, system updates always include a memory reinitialization with a full bootstrap (reinitialization of all processes and data from disk). Only the contents of protected application segments, special memory areas where application systems may retain critical information, are retained across the boot. Since a system update includes a reinitialization, only the version of the software on the 3B20D disk is updated. The main memory images of system processes will then be re-read from the disk during the bootstrap. This section describes how this disk updating is done within DMERT, and gives an overview of the overall System Update process.

### *4.1 System Update concepts*

The DMERT System Update Program (SUPR) provides a way to replace the entire contents of the 3B20D disk with a new version of those contents from a magnetic tape. SUPR deals with masses of data, and changes the disk contents section by section rather than file by file or logical data base updates. These sections are called *partitions*. To do this, SUPR takes advantage of the fact that the 3B20D disks are duplexed for reliability, writing the new system information onto

only one of a pair of disks. This is the *off-line disk method* of system updating. It derives its name from the fact that one of a pair of disks must first be removed from active service (taken off-line) before writing the new system onto it. With the off-line disk method the amount of redundant disk information is kept to a minimum during the update, and the disk structure may be completely changed. There is some increase in system vulnerability during the time that the disks are not running in duplex mode.

Certain aspects of the system update procedure have caused unique requirements and changes within DMERT. The key to the off-line disk method is protecting both generic programs from being overwritten during the update procedure. Since these generics reside on duplex disk mates, an off-line disk must never be restored to service. (The restore process includes a copy from the on-line to off-line disk.) The attributes of the "off-line" device state in the ECD were expanded to provide this capability. After a bootstrap on a new generic disk image, the disk copy of the old generic must similarly be marked off-line, and hence protected from restorals. This was accomplished by having each generic's ECD record the disks containing the *other* generic as off-line.

It was also necessary to be able to access partitions on an off-line disk, in order to read or write partitions on an off-line disk, to transfer files from the old generic to the new generic, and to perform recent changes on the new generic ECD (for example, in marking old generic disks as off-line). This was done by having the disk driver program access the Volume Table of Contents (VTOC)—the directory of the disk's contents—on the off-line disk during the update process. This is a special case, since the VTOC on an off-line disk may be different from that of its mate disk, or may not even be sane. When updating multiple disks, SUPR uses a special disk identifier added to the VTOC to ensure that the disk image being written corresponds to the information on that disk. As another safeguard, System Update uses checksums (special numbers computed from the data in a file) on the generic tape to check the new generic data for damage before writing it to the disk.

### *4.2 System update scenario*

SUPR provides a complete update scenario, including a means to reverse the update and re-establish the original system. Because of the major impact on the application during a system update, the complete update procedure is broken down into several distinct steps, and allows the craft to choose the best time to begin each successive step of the update. The update may be canceled at any step of the procedure. Application-dependent processing may be introduced at any step.

Under favorable conditions only the forward steps of SUPR would be used, resulting in a successful update. These steps are:

(i) Enter new generic—Read all the new generic data onto the off-line system disk.

(ii) Proceed with new generic—Make final preparations prior to booting the system from the new generic.

(iii) Boot from new generic—Manually boot the system using the new generic.

(iv) Commit to new generic—Complete propagation of the new generic into the system after the soak period by removing all aspects of the old generic.

If the new generic does not work as expected, the craft would not commit to it, but would start a backout procedure to return to the original system.

SUPR also provides a convenient mechanism to allow application-dependent processing at each step of the update procedure. This is accomplished by transferring control to an application process that can perform whatever actions are appropriate. The types of actions most likely to be done as part of the application processing would be to transfer data (files, data bases, office-dependent information) from the old generic to the new generic or to save call registers and billing information in protected application segments prior to suspending call processing and booting from the new generic.

## V. SUMMARY

This article has dealt with the subsystems of DMERT that administer changes to system data. Recent Change/Verify is used to change system configuration data and its underlying data base, Field Update allows "bug fixes" and logical file changes, and System Update will install an entirely new version of the operating system. These subsystems were described and examples given of their use. In each case DMERT provides change application, testing, and rejection or acceptance capabilities in a context very similar to that of typical operating systems, but in a highly reliable way.

## REFERENCES


1. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman, "The 3B20D Processor & DMERT Operating System: DMERT Operating System," B.S.T.J., this issue.
2. D. M. Richie and K. Thompson, "The UNIX Time-Sharing System," B.S.T.J., 57, No. 6, Part 2 (July-August 1978), pp. 1905-29.
3. B. W. Kerninghan and D. M. Ritchie, *The C Programming Language*, Englewood Cliffs, N. J.: Prentice-Hall, 1978.
4. B. R. Rowland and R. J. Welsch, "The 3B20D Processor & DMERT Operating System: Software Development System," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# 3B20 Field Utilities

By G. P. ELDREDGE and J. G. CHEVALIER

(Manuscript received March 10, 1982)

*The term "field utilities" describes a number of tools used by telephone company craft and support staff as well as Western Electric and Bell Laboratories field support personnel for trouble-clearing and routine maintenance activities on the 3B20D/DMERT system. This complementary set of tools provides debugging coverage for the system regardless of load or system functionality. In addition, it deals with the challenges and complexities posed by the concepts of parallel processing, virtual addressing, and swapping. This article describes the various field utilities and discusses their capabilities.*

## I. INTRODUCTION

The term "field utilities" includes a number of tools used by telephone company, Western Electric, and Bell Laboratories support personnel to perform trouble-clearing and routine maintenance activities. Currently, software debugging and investigation tools include the Field Test Set (FTS), the Generic Access Package (GRASP), and IBROWSE, an interactive tool used to "browse" through the contents of main memory. In unusual cases, a Micro-Level Test Set (MLTS) may be used in a troubleshooting mode. The Program Documentation Standard (PDS) Field Maintenance Commands are a collection of tools used to perform more routine operational maintenance on the operating system. Each of these capabilities will be described in this article.

## II. TROUBLESHOOTING AIDS

The nature of large, evolving software projects is such that, despite multiple levels of testing by developers, integration teams, system test groups, and field site acceptance teams, some software "bugs" escape

Table I—Comparison of 3B20D/DMERT debugging tools

| Attribute/Tool | FTS | GRASP | IBROWSE | MLTS |
|---|---|---|---|---|
| Interference | None | Small, self-regulated | Small, not regulated | Extreme |
| Scope of capabilities | Medium | High | Low | Medium |
| Debugging level | Assembly | Assembly | Assembly, source | Microcode, assembly |
| Limitations | Limited on kernel | No special processes or kernel | No breakpoints, no trace | Difficult with supervisor or user processes, no data breakpoints |
| Language | C-like | PDS, MML | ADB-Like | Terse |
| Target users | Bell Labs, WE | Operating Co., Bell Labs, WE | Bell Labs, WE | Bell Labs, WE |
| Target software needed | None | DMERT | DMERT | Microcode |
| Support processor software needed | *UNIX* Operating System (FTS) | None | None | None |
| Hardware needed | FTS, DUC, terminal | UC or DUC (optional) | Terminal | MLTS, terminal |
| Theater of use | Limping or loaded field site | Running, nonoverloaded field site | Running, nonoverloaded field site, off-line | Lab, dead field site |

detection and are included in field releases of software. In the real-time systems used in switching, the bug may be so subtle that it may surface only under equipment configurations, telephone user actions, and/or traffic loads not easily reproduced in a system laboratory environment. System debugging tools must be available in a field site carrying live traffic to solve these problems when they arise.

The 3B20D/Duplex Multiple Environment Real Time (DMERT) operating system employs advanced computer technologies that require equally sophisticated tools to isolate errors. Parallel, time-sliced execution of processes, virtual addressing, and swapping all contribute to the need for a variety and diversity of system debugging tools. Table I is a comparative summary of these various troubleshooting tools available to field sites.

### *2.1 Field test set*

In rare cases, a system problem could occur that leaves the system

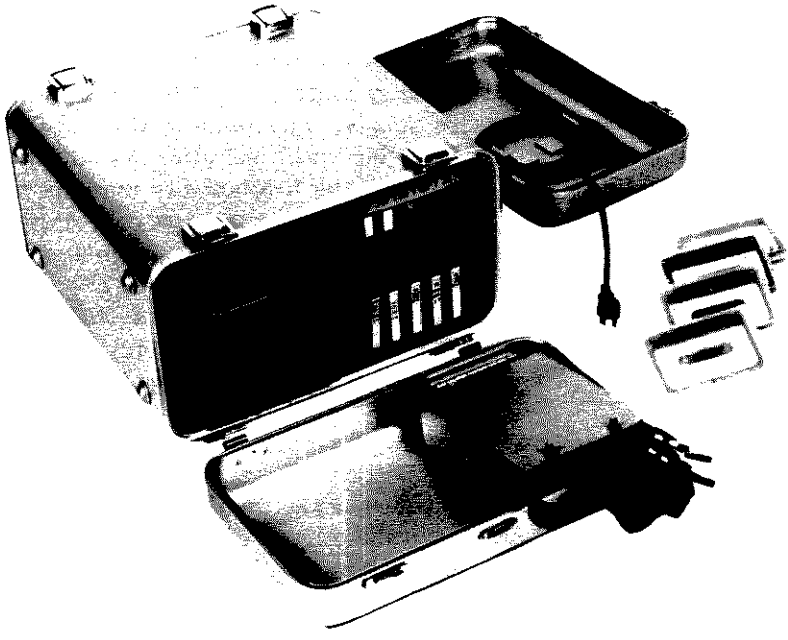


Fig. 1—Field test set.

functionally inoperative. In other cases, the traffic level may be so high that system overload mechanisms become active when unexpected results in the system indicate a software error. In either case, on-line utility systems, which assume basic functionality and nonoverload conditions, are not appropriate to isolate the problem. The Field Test Set (FTS) was designed specifically to meet this need in the field. It is strictly a monitoring device and therefore does not affect processor performance or rely on system operability. This non-interfering characteristic is extremely important when maintenance personnel are trying to isolate problems at a field site carrying a heavy traffic load.

The FTS is a small, portable unit (see Fig. 1) that is easily transported and connected to the 3B20D Processor through the Dual-Access Utility Circuit (DUC). The DUC contains hardware matchers and a 2048 entry trace memory and provides access to the processor for the FTS and GRASP (see Section 2.2). The external FTS unit connects to the DUC through an eight-foot cable. The FTS intelligence is contained in this external unit that includes a microprocessor with memory management, one megabyte of random-access memory (RAM), and a cassette transport. User access is provided through a local or remote terminal with phone access provided by the FTS.

The *UNIX** operating system was chosen as the FTS operating system. There are many advantages to using an operating system on the FTS and in particular the *UNIX* system. The FTS resident software was developed and tested as individual modules written in the high-level C language. This substantially reduced the software development time and effort. Also, the *UNIX* operating system commands provide substantial portions of the functionality required for the FTS software. Although the *UNIX* system requires disk storage for its file system, a disk system was not considered rugged enough for portability. Therefore, a "virtual disk" is supported as part of system memory. The *UNIX* operating system is booted into the system from cassette tape by resident erasable programmable read-only memory (EPROM) software. The EPROM also contains the unit's self-diagnostic software.

The FTS/DUC system supports a rich variety of trace and data-matching options. The lowest level trace, a so-called transfer trace, records program addresses of all transfers executed by a program or a range within a program. An intermediate-level function trace records program function call/return sequences. At a higher level, a record may be kept each time a different process begins execution. Multiple trace modes can be active simultaneously. Information is recorded into the trace memory under control of a variety of sophisticated matcher circuits. Masking capability is provided so that a matcher can look for a particular value of a single bit or groups of bits as well as word values. Matchers are included for address, address range, data, access type (e.g., read, write, or read/write) and process ID matching. When a matcher or a combination of matchers is triggered, a signal is produced that causes a "snap" of information into the trace memory. The matchers and matcher combinations allow very selective trace memory recording. This reduces both the size of the trace memory required and the amount of post-processing necessary to interpret the trace data.

The trace memory is operated in either a pre-trace or a post-trace mode. In the former case, the trace memory records information until it receives a stop trigger. The trace data represent program flow leading up to a particular event. In the post-trace mode, the trace memory starts recording upon receiving a trigger and stops when the trace memory is full. This provides a history of program flow after a particular event.

The DMERT operating system software is predominately written in the high-level C language. C enables the programmer to work with function-level rather than machine-level operations. To support this,

* Trademark of Bell Laboratories.

the FTS includes matching and tracing capabilities for software functions and process IDs. Function tracing records the program address and the data parameters passed on the stack to a selectable function or range of functions. Process ID matching and tracing becomes necessary in a virtual memory machine since processes are dynamically relocatable in physical memory. Processes are assigned unique ID values when they are created. The active process ID value is presented to the FTS process ID matchers and trace memory. These matchers, combined with the virtual address matchers, permit matching and tracing on virtual rather than physical addresses.

Since the FTS is an external system, it is the appropriate choice when problems must be investigated in code that has tight timing constraints or in a system that is heavily loaded. Its most attractive features are its excellent trace facility and the fact that the FTS operates in a mode that does not interfere with 3B20D operation. Although it was not designed to access machine registers or write memory, the FTS is a powerful tool in the hands of support personnel to isolate difficult system problems.

### *2.2 Generic access package*

The concept of an on-line software debugging mechanism in real-time machines is not new.[1] Software problems may occur when the system is functional and processing traffic in a non-overload environment. Such problems can be solved in the 3B20D by use of the Generic Access Package (GRASP).

GRASP is an on-site tool for software debugging. Since it supports an interface to the DUC, GRASP provides a set of trace and data-access trap functions similar to those provided by the FTS. In addition, it provides the capability to place multiple breakpoints in code, to print the contents of memory and many machine registers, and (with some restrictions) to write memory and registers regardless of whether the DUC is available or operating correctly. GRASP has a self-regulating mechanism designed to prevent itself from taking too much real time and thereby interfering with traffic processing or driving the system into overload.

Since GRASP is "just another process" running on the machine, the design presents some unique challenges. GRASP needs to be able to identify the target process, assure that it is in main memory, and be able to gain access to its address space.

A logical process is specified by a logical tag (called a "utility ID") that is compiled into the process. All incarnations of a logical process will have the same tag since they all originate from the same object file on the disk. The tag is stored in system tables when the process is brought up and is available throughout the life of the process.

Upon a request from GRASP, the operating system searches the tables, prepares a list of real process tags (called "process IDs") for processes whose utility IDs match GRASP's request, and sends the list to GRASP. Translation between the utility ID, which is known to the craftperson, and the process ID, which is known to the operating system, is thus accomplished.

GRASP relies on cooperation with the target process to be informed when the target process is in main memory. All processes that GRASP may need to monitor must have two function calls compiled into the code, which form the run-time communication mechanism with GRASP. One is placed in the process's initial entry routine; the other is placed to execute "on demand" by GRASP.

After a process has been selected, it is forced into main memory through cooperation with the process. GRASP sends an agreed-upon event to the process; its only response to that event is to call the associated library function. That function identifies the process and notifies GRASP that it is in main memory.

Access to the target address space is then accomplished by using address translation hardware called Address Translation Buffers (ATBs). The Program Status Word (PSW) for each process is constructed to be able to handle two address spaces at one time. The identity of the address translation buffers being used by a particular process are included in that process's PSW. Instructions are provided in the instruction set to indicate which of the two address spaces to use. In addition, a special breakpoint instruction has been provided. When the breakpoint is executed by the target process, GRASP's PSW is modified so that GRASP is given access to the address space in which the breakpoint fired. This presupposes that GRASP and the target process are using different address translation buffers; that assumption is enforced by the operating system.

GRASP is especially useful when multiple breakpoints are needed (GRASP can handle up to 20), when breakpoints must be planted in several processes simultaneously, where register information is needed, or when investigation must be done remotely from a central maintenance facility.

### *2.3 IBROWSE*

Neither the Field Test Set nor GRASP provides a mechanism to examine the kernel address space. IBROWSE, an interactive tool used only by Bell Laboratories and Western Electric support personnel, can be used to peruse the address space of any DMERT process in main memory; it fills the need to be able to view the operating system tables and message buffers in the kernel address space. IBROWSE also can be used on an off-line support processor to analyze tape dumps of main memory taken at field sites.

IBROWSE can display the contents of virtual or physical memory in a user-specified format. The user can direct that the raw machine data be represented as any combination of null-terminated strings, characters, or one-byte, two-byte (short), or four-byte (long) data types in octal, decimal, or hexadecimal format. This flexibility to specify the translation of raw data is immensely helpful when viewing DMERT data structures. IBROWSE supports the concepts of current address, next address, and current format, which are useful in displaying consecutive memory locations. It can view any process in memory, from kernel through kernel processes, supervisors, and user processes. It has the ability to search forward or backward for a specified data pattern, in either virtual or physical addressing modes. IBROWSE also supports a user-defined macro facility and I/O redirection.

The main strengths of IBROWSE are its ability to view the address space of any process in main memory and its capability to analyze data from an off-line Control Unit (CU). Since use of IBROWSE requires relatively detailed knowledge of DMERT, its users are intended to be specialized Bell Laboratories or Western Electric support personnel; for that reason, no attempt has been made to make IBROWSE part of the official DMERT release. Each time the support teams need it, they load it into the target machine.

### *2.4 Micro-level test set*

Should a problem result in a "dead" system or one that is continually attempting automatic recovery actions and is unable to start the operating system, the Micro-Level Test Set (MLTS) is used. The MLTS is a low-level test system aimed primarily at hardware register and microcode access. It consists of an interface circuit that plugs into the 3B20D like any other board and an external control circuit. Since the MLTS is equipped with an RS232 interface and a 212A data set, it may be configured with a terminal on-site or may be operated from a remote location.

The MLTS is the only field utility tool that provides read/write access to all internal hardware and firmware registers and is the only one that facilitates access to the processor's microcode. The MLTS provides microcode breakpoints, can read and write microstore and main store locations, and can read and write machine registers that are not accessible to other troubleshooting tools. Although its primary use is in a laboratory environment, there are infrequent cases where such capabilities are required to solve problems during field tests.

## III. OPERATIONAL UTILITIES

Since DMERT supports a hierarchical file system as well as the concept of processes, some types of problems must be dealt with and

resolved at the process or file-system level. For example, a process may be running when it should not be or the file system may contain some transient files that should have been cleared. The *UNIX* operating system itself provides many utilities for process control and file system maintenance; these same capabilities are needed in the Program Documentation Standard (PDS) syntax for Electronic Switching System (ESS) applications.

PDS field maintenance commands can be described in three categories:

(*i*) File system manipulation and maintenance

(*ii*) Process control

(*iii*) Magnetic tape operations that are support-processor compatible.

PDS commands are provided to allow the craft or support person to determine what files exist on the disk and what their access permissions are; the craft may alter the access permissions, add new files, or remove existing files. A basic text editor is provided to facilitate creation or modification of ASCII files. In addition, tools are provided to start a process, stop a process, and to determine what processes are known to the system.

Although these utilities do not fall into the class of "debugging" tools, they nevertheless provide a window into the system at a high level that is very useful to solve certain types of system problems.

## IV. SUMMARY

Because of its architecture and technology, the 3B20D/DMERT system presents a number of challenges to those who must isolate problems in a running system in the field. Problems may be caused by hardware failures, software deficiencies, microcode errors, or operational overloads and inconsistencies. A set of tools has been developed to isolate problems that may occur in the field. Together, these utilities provide a continuum of system trouble identification capabilities for the 3B20D/DMERT system in the field.

## V. ACKNOWLEDGMENTS

The authors acknowledge the contributions of Messrs. R. H. Holt, J. P. Kehn, G. A. Moore, J. D. Peterson, and D. J. Thompson, and Ms. C. A. Toman for their inputs, critiques, and reviews.

## REFERENCES

1. G. F. Clement, P. S. Fuss, R. J. Griffith, R. C. Lee, and R. D. Royer, "1A Processor: Control, Administrative, and Utility Software," B.S.T.J., *56*, No. 2 (February 1977), pp. 237–54.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating Systems:*

# Fault Detection and Recovery

By R. C. HANSEN, R. W. PETERSON, and N. O. WHITTINGTON

(Manuscript received March 18, 1982)

*The 3B20D Processor is designed to be a high-availability system for utilization in electronic switching systems. This high availability translates into the development of numerous features and capabilities for the 3B20D that distinguish it from other processors. The reliability objectives for the processor are described and related to the subsystems that have been developed to meet each objective. This article discusses processor and peripheral fault recovery, system integrity, and other software subsystems that provide the high availability and maintainability for the processor.*

## I. INTRODUCTION

The 3B20D Processor has extensive maintenance subsystems associated with it and is designed to meet the high-availability standards of Bell System electronic switching systems. This implies that the processor must perform within an objective of not more than two minutes downtime per service year when used in an electronic switching application. The many subsystems that have been developed to provide the high-availability capability are described in this article. In particular, software and hardware fault recovery are discussed along with the microcode assists for the recovery.

Much evolution has taken place in recovery architectures for electronic switching systems.[1,2] Earlier processor systems used extensive hardware-matching algorithms that resulted in intricate software recovery.[3,4] More recent hardware technologies have enabled the cost-effective design of processor systems with unique fault-detection capabilities.[1,5,6] These capabilities have led to much simpler recovery software. This article describes the detection mechanisms for the 3B20D and the software maintenance architecture.

## II. SYSTEM RELIABILITY REQUIREMENTS

The reliability objective for the 3B20D Processor system, as with other similar systems, is to keep the overall system unavailability—i.e., the time that the system cannot be utilized by operational (call processing) functions—below 2.0 minutes per year.[7] In keeping with the ESS processor tradition, the total system downtime is allocated to four general categories: hardware, software, recovery, and procedural.

The processor has 0.4 minute per year allocated to malfunctions in the system hardware. Like other highly reliable systems, the 3B20D is equipped with redundant hardware units. Thus, failures must occur in both redundant units before the system is unable to establish a working configuration. In the case of simultaneous failures in both units, until one is repaired and system integrity is reestablished, the system is considered unavailable. This portion of the overall system downtime is a function of the failure rates of the various components (FIT rate), the system architecture, and the mean time to repair (MTTR). The hardware reliability model for the 3B20D Processor within a given application is dependent on the hardware configuration used and the maintenance technique used (this determines the repair time).

The processor has 0.3 minute per year allocated to malfunctions in the processor operational software. This is a classification of software faults that can render the system features inoperative. This allocation includes cases such as software faults that require a bootstrap to recover the system. As in the case of other high-availability systems, the 3B20D/DMERT system has a design objective of having no software failures the system cannot recover from. To help recover the system against software failures, DMERT has three levels of defenses that attempt to recover the system from such faults: hardware protection, system integrity monitor, and audits. The 3B20D Processor has several levels of hardware protection that detect the sanity of the system software. The system integrity monitor in the DMERT system has an elaborate scheme of software and hardware sanity timers as well as overload detectors that protect the system against software "resource hogs." DMERT audits include all of the explicit audits in the system as well as the defensive checks built into the common processor software. The intent of the audits is to help defend important processes against data mutilation.

The processor has 0.7 minute per year allocated to limitations in fault-recovery programs. These failures are classified by the inability of fault-recovery software to achieve a working configuration of the system due to some hardware failure condition even if a working state of the hardware is possible. These cases are characterized by the necessity for manual intervention to reestablish system integrity or by an automatic initialization to regain system integrity.

The 3B20D has a comprehensive fault-recovery scheme that attempts to recover the system from all foreseeable single hardware fault conditions. In several cases, recovery mechanisms are generated for multiple fault situations (e.g., memory failures) when that is considered to be a probable situation.

Finally, the processor has 0.6 minute per year allocated to procedural errors. This category covers cases where a craft person uses an improper maintenance procedure or follows a poorly designed procedure that results in a machine outage. The 3B20D is designed with a defensive craft interface using the PDS (Program Documentation Standards) and MML (Man Machine Language) languages.[8] The craft interface also includes emergency action and display-page capabilities that attempt to simplify the complexities of maintaining the 3B20D.

The system reliability requirements also include the various aspects of maintaining the 3B20D. These maintainability aspects include diagnostics, transient error analysis, emergency recovery procedures, routine maintenance procedures, growth and retrofit capabilities, system and process update capabilities, and field utilities. Diagnostics are provided to detect and assist the repair of classical hardware failures in the system. The diagnostic requirements include sufficient run-time performance so that a rapid repair can be carried out. Diagnostics provide greater than 90 percent fault detection.

The ability to repair circuitry exhibiting transient failures is provided through fault-recovery error reports. For example, data about transient memory faults is printed out to the craft and includes address and pack location where the error was detected. If that circuit pack continues to have a history of transient errors, the craft has sufficient information to effect a repair. Error analysis capabilities are provided on the 3B20D through the use of fault-recovery messages and error logs.

Emergency recovery procedures are provided to reconfigure the system when automatic recovery does not succeed. These capabilities allow the craft to repair the 3B20D in case of catastrophic failures. These procedures include use of the emergency action page, processor recovery message analysis, and dead-start diagnostics. Routine maintenance procedures are provided to keep the 3B20D in peak operating condition. Growth and retrofit procedures allow hardware additions and removals without affecting the system service. Finally, various utilities are provided with the DMERT system to locate system problems in field installations.

## III. GENERAL RELIABILITY AND MAINTENANCE ARCHITECTURE

In this section, we provide an overview of the 3B20D fault-recovery architecture that is described in further detail in later sections. Figure

1 illustrates the hardware architecture of the 3B20D. As is indicated in the figure, the processor system has very loose coupling between any of the mate subsystems. The memory to memory update coupling is provided to keep both active and standby memories identical. This allows the switching of processors without losing the integrity of the software running on the system.

The other coupling between the processors is through the maintenance channel. The maintenance channel provides two capabilities important to the integrity of the processor. First, it provides a control and communication bus for the purpose of diagnosing the off-line processor from the on-line processor. Second, it provides low-level maintenance control for fault-recovery programs so that a switch in processor activity can be carried out with no operational interference. In addition, other maintenance controls can be exerted over the channel to start an initialization sequence on the other processor or to stop execution on the other processor. One other coupling, the Dual Duplex Serial Bus Selector (DDSBS), allows either processor to talk to any peripheral controller. Thus, no matching techniques are utilized between major subsystems or peripherals in the 3B20D for the purposes of fault detection in the hardware. This means that unique fault-detection techniques are essential in each subsystem of the 3B20D.

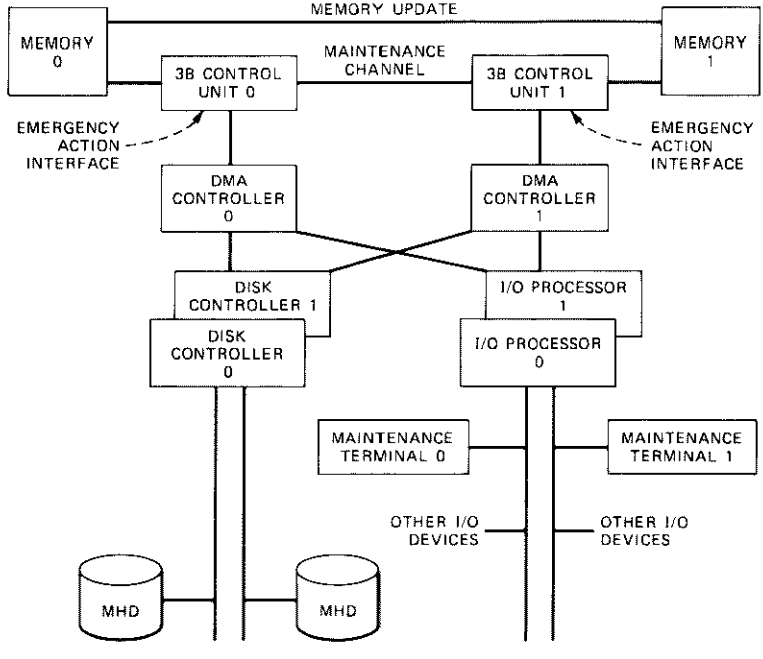



Fig. 1—The 3B20D system architecture.

To provide these detection algorithms, extensive use of local matching circuits, parity techniques on all buses, Hamming detection with single-bit error correction on the main store, cyclic redundancy codes on the disks, and numerous sanity timers throughout the control unit and peripherals are used as the primary fault-detection techniques. In addition, routine diagnostics are used to detect failures in the fault-detection hardware itself. Other routine sanity checks are used to ensure that peripheral subsystems are healthy. Finally, system-integrity checks catch certain subtle problems that are not caught by unique detectors.

### *3.1 Fault-recovery architecture*

When any of the unique detectors determine an error condition, an error interrupt (or error report in the case of certain peripherals) is registered in the processor. The most severe of these will result in automatic hardware sequences that switch the activity of the processors (hard switch). Less severe errors result in microinterrupts that enter microcode and software charged with recovery of the system.

The microcode and recovery software provides a layered approach to the recovery architecture. Figure 2 illustrates this architecture with microcode providing low-level access to the hardware while the recovery software provides the high-level control mechanisms and decision making. This technique has resulted in several hardware design modifications requiring minimal change to the recovery software.

Figure 3 illustrates the principal architecture and features provided by the recovery software. The bootstrap and initialization routines provide a fundamental set of microcode and software algorithms to control the processor initialization and recovery. These actions are stimulated by a Maintenance Restart Function (MRF), which repre-

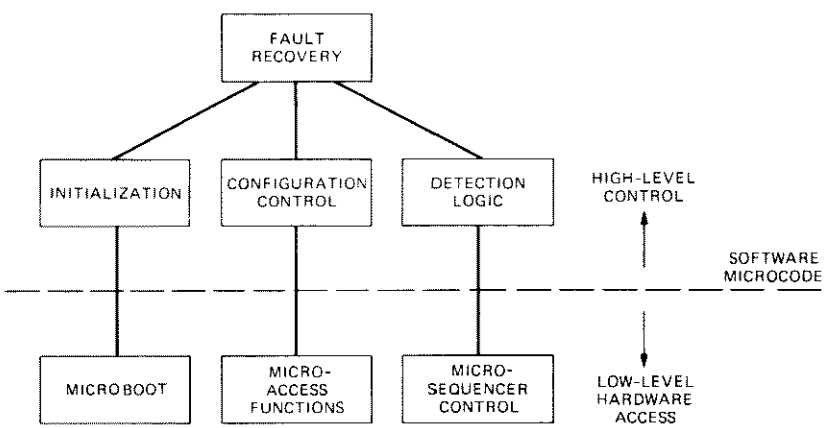



Fig. 2—Maintenance software structure.

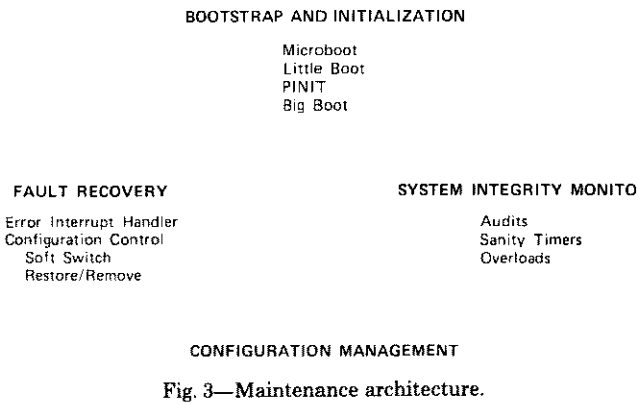



Fig. 3—Maintenance architecture.

sents the highest priority microinterrupt in the system. An MRF sequence can be stimulated from either hardware- or software-recovery sources.

The fault-recovery and system-integrity packages control fault detection and recovery for hardware and software, respectively. The Error Interrupt Handler (EIH) is the principal hardware fault-recovery controller. It receives all hardware interrupts and controls the recovery sequences that follow. The configuration-management program (CONFIG) determines if this particular error is exceeding predetermined frequency thresholds. If a threshold is exceeded, CONFIG requests a change in the configuration of the processor to a healthy state. Thus, CONFIG serves as an error-rate analysis package[10] in the 3B20D maintenance architecture for both hardware and software errors.

### *3.2 Software integrity architecture*

Software fault recovery is very similar in architecture to hardware fault recovery. Each major unit of software is expected to have associated with it error-detection mechanisms (defensive checks and audits), error thresholds (provided by the system-integrity monitor and CONFIG), and error-recovery mechanisms (failure returns, data correcting, audits, and initialization techniques). In addition, both SIM (System Integrity Monitor) and EIH oversee the proper execution of the process. SIM ensures that a process does not put itself into an infinite execution loop or excessively consume some system resource (e.g., message buffers). EIH, through the use of hardware and microcode detectors, ensures that processes do not try to access memory outside of defined limits or execute instructions that are not permitted to the process. Each process has initialization and recovery controls (analogous to hardware) so that a recovery can be effected. Figure 4 illustrates the software-recovery architecture.

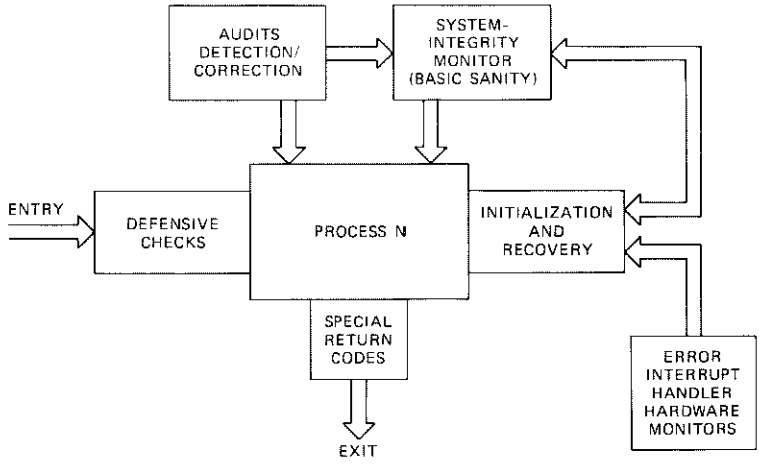



Fig. 4—Software fault-recovery architecture.

If recovery actions result in the removal of hardware units from the system, diagnostics[9] are dispatched automatically to analyze the specific problem. Audits represent the software counterpart for diagnostics with the exception that the routine interval is much shorter and correction is possible in the case of audits.

## IV. FAULT RECOVERY

In this section, we describe the fault-recovery strategies associated with the 3B20D Processor. In particular, we describe the fault recovery and initialization strategy along with the microcode assists required to carry out these functions. We also describe the manual control capabilities provided by the processor and software. These control mechanisms are termed emergency mode. Finally, we describe some of the software audit and integrity techniques in the DMERT operating system.

### *4.1 Fault recovery and system initialization*

Fault-recovery strategies are based on the fault-tolerant architecture of 3B20D. Major hardware units are fully duplicated. This duplication provides a high probability that a combination of operational units can be retained in the face of faults. The mate processors are only loosely coupled; interprocessor connections are limited to the maintenance channel and memory-update circuitry. This architecture forms the foundation of the hardware-recovery strategy employed in the 3B20D, namely to isolate an entire faulty processor as opposed to attempting fault resolution at the subunit level.

DMERT is a modular operating system that provides a wide range

of protection from various types of classical errors. Examples include write-protected memory areas, memory ranges that can be used only for text execution, and protected virtual address spaces. Thus much of the recovery from these types of errors is built into DMERT directly. Those overt recovery actions that are required are greatly simplified by the underlying architecture. Hard faults and other conditions requiring recovery actions are treated according to their severity. Fault categories that will be described individually are hard faults, thresholded faults, configuration faults, sanity time-outs, and software-requested recovery actions.

The 3B20D has built in self-checking circuitry designed to detect hard faults as soon as they occur. This circuitry simplifies recovery since early fault detection limits the possible damage done by the fault. Faults in this category indicate that the processor is no longer capable of proper operation and results in an immediate stop of the currently running processor and a switch to the standby processor. Since the standby processor does not match the active processor instruction by instruction, an initialization sequence is required to start execution properly.

Some types of faults and errors are not severe enough to justify an immediate stop and switch recovery action. Examples of errors of this kind are hardware faults detected in the standby processor memory and software errors such as write-protection violations. Another type of error in this category is hardware faults that are handled by self-correcting circuitry. Although most faults are detected by self checking, some units, such as main memories, have fault rates that justify self-correcting capabilities. Disks also are self correcting through the use of cyclic redundancy codes. All errors in this class are reported to the recovery system as error interrupts.

Recovery software classifies the interrupt by type, gathers and saves all available information about the interrupt, and reports the error to the system configuration-management package. If a particular software process is suspect as the cause of the interrupt, such as in a software-triggered event, the process that was running at the time of the interrupt is faulted and entered at its fault entry after a stable system configuration is guaranteed. The fault entry of a process contains recovery and initialization sequences that are special to the process involved.

All error interrupts are reported to configuration management. Errors are logged against the failing unit and error rates are compared to allowed error thresholds. If the affected threshold is exceeded, further action is required and is based on several factors. If the faulty unit is essential to the system and a mate unit is available, the faulty unit will be removed from service and scheduled for diagnostic testing.

If there is no available mate unit, the faulty unit will be initialized and returned to service since in the case of essential units it is better to have a faulty unit than no unit. Nonessential units are removed and scheduled for diagnostic testing whenever their error thresholds are exceeded.

Each processor has a sanity timer that will result in an initialization if it times out. The active processor maintains both its own and the standby sanity timer so that if the active processor is completely dead, an initialization of the standby processor will be triggered by a sanity timer time-out.

The system provides an Operating System Trap (OST) for use by software to request an initialization. This capability is used by critical system processes when they encounter errors that preclude performance of a critical system function. Initializations occur when an error or fault has been detected that cannot be recovered from without a change in hardware and/or software status. A stop and switch to the other processor may or may not be associated with any given initialization. All initializations include actions of varying severity depending on what is required to deal with various faults and errors.

The first event in the initialization sequence is a hard-wired transfer to a fixed location in the CU microstore where microcode makes a decision as to whether to bring this processor on-line or to switch to the other processor. If the current initialization is of level two or higher, the appropriate processes and data bases are loaded from disk. All available data about the initialization trigger is saved and a decision is made to bring this processor on-line or stop for the off-line initialization.

The DMERT kernel initialization or bootstrap routine is then called to restart system processes or to fault active processes as appropriate. The initialization is now complete and the system has returned to normal operation. If an initialization does not recover the system to an operational state, another and more severe initialization will be triggered automatically. Whether to escalate or not is controlled by the initialization interval. Any initialization that occurs during a window of time following the previous initialization will escalate to the next higher level. The length of the initialization interval is a system generation parameter that is established by the application. In addition to the DMERT initialization levels, provision is made for an application to specify between one and sixteen levels for each DMERT level. For example, if the application specifies two levels for DMERT level one, the normal execution of initialization levels would be (1,1), (1,2), (2,1), $\cdots$, where the first number indicates the DMERT level and the second number is the associated application level.

Data about various recovery actions taken by the system are sup-

plied to provide all possible information about what went wrong and to provide data that can be used by maintenance personnel to assist them in isolating difficult faults. Recovery data are provided in several forms. Each error interrupt is accompanied by a printout containing available information about the state of the processor when the interrupt occurred. A more difficult problem is presented by initializations. Since they are more severe than interrupts and in fact represent a discontinuity in processing, gathering and preserving error data is more difficult. Initializations, as well as interrupts, can occur at a rate much too fast for data to be printed. The solution is to save all pertinent data in a protected area of memory for printing after the system has recovered.

Various kinds of error data are not generally printed as a part of the standard system output but instead are saved in error files on the system disks. Examples of this kind of data are device-driver errors and failing-memory data. One of the more useful pieces of data output by the system are Processor Recovery Messages (PRM). These are low-level one-line messages that are printed in real time. The PRMs thus represent progress marks through the recovery sequences and are extremely useful in those cases where stability cannot be achieved or postmortem data cannot be gathered.

### *4.2 Special microcode for recovery*

A large fraction of the total amount of CU microcode is provided to aid in recovery. The bulk of this recovery microcode is in PROM because most functions are required in spite of the power history of the CU or its boot devices. Functions that are required even if the CU is not ready to execute its instruction set include microinterrupt processing, maintenance channel assists so that one processor can access the other processor and microcode to initialize hardware subsystems. Additional recovery microcode that resides in writable microstore (WMS), extends the processor's instruction set to provide convenient diagnostic and recovery software access instructions. When diagnostic performance requirements do not justify a special instruction, a microstore scratch area is available that can be loaded with arbitrary microsequences that can then be executed for special tests or functions. Before software can run, the WMS must have been loaded from disk. This happens initially as part of the processing of the MRF microinterrupt.

#### *4.2.1 MRF and microboot*

When a maintenance restart interrupt occurs, a long sequence of microsteps begins to establish system sanity. Both processors may be in their MRF sequence at the same time and each one may try to

become the active processor. The MRF code first makes decisions on whether to do an off-line initialization or an on-line initialization. If a processor determines that it has just powered up, it clears main store and does an off-line initialization unless forced on-line.

A number of tests are made on data in the system status register, SSR, to select one of four possible actions: microboot, tapeboot, processor initialization, or stop and switch. The simplest actions are to begin execution of a processor initialization program called PINIT or to stop and switch to the other processor. This is accomplished by sending a switch command over the maintenance channel to the other processor.

Tapeboot is a complex sequence of microcode that is only done when requested manually via the craft interface. Its function is to create a new system disk from tape. Using the tape device and disk device selected by the craft interface it initializes those I/O units and initializes the WMS from tape. A boot program, called load disk from tape, is read from tape into main store, and memory-management tables are created to allow it to run the hardware complex without the operating system. This program then reads the tape to make a DMERT disk image.

Microboot uses information on the DMERT disk to initialize the writable microstore and read in the first software boot program called little boot. To do this, it must first select the disk drive to use as a boot device. If the craft interface has forced either the primary or secondary boot device, it uses that device. Otherwise, microboot selects a disk drive based on the state of hardware control bits. Alternate boots will use alternate devices. Microcode is read from the disk and then copied to WMS. Finally, little boot is read from the boot partition and given control.

#### *4.2.2 Microaccess assists*

Although the MRF sequence is the most complex microcode recovery assist, both diagnostics and recovery software have special microcode. There are six maintenance channel assists in PROM. They are:

Write Main Store
Read Main Store
Write Writable Microstore
Read Microstore
Write Utility Circuit
Read Utility Circuit

In addition there is microcode in WMS to support a set of instructions provided for the diagnostic and recovery software. Diagnostics

have instructions to manipulate the maintenance channel and aid in I/O diagnoses. They also share instructions with recovery. These instructions include groups of instructions for:

On-Line Main Store Controller
Off-Line Main Store Controller
Maintenance Store Operations

Finally, both diagnostic and recovery software use privileged instructions (shared with the operating system) to read or write special registers. They also can activate unit initialization sequences that are used in the various parts of the MRF microcode.

### *4.3 Emergency modes*

Emergency mode on the 3B20D refers to the facilities and procedures provided to prevent the system from experiencing a total outage. For example, emergency facilities are applied when the system is unable to recover automatically. The most characteristic of these are: duplex failure of the control unit, duplex failure of the system disks, duplex failure of the essential I/O devices, failure of fault recovery to find a working configuration of hardware, software faults that will not allow the system to operate properly, errors that destroy the integrity of the disks, and software overwrites that introduce catastrophic errors into the software.

Emergency mode capabilities are built into the system to address these mechanisms that can fail the 3B20D as a system. The emergency action interface (EAI) on the 3B20D provides for manual initialization capabilities that can recover the system from several of the conditions mentioned above. This interface allows the craft to select a processor and disk configuration in a case where certain configurations may be leading to the problem. The interface also allows the craft to reconfigure the system to handle maintenance hardware failures. For example, the craft can inhibit error sources and sanity timers through EAI commands, thus allowing recovery from certain maintenance failures even though both processors are affected. The EAI also provides capabilities for craft initializations to deal with loss of subsystem capabilities.

The 3B20D also provides the craft with other manual capabilities through the port switch select, the disk power inverter select, and the unit power switches. These can be used to reconfigure the system to handle certain problems in the system. In rolling bootstrap conditions, the 3B20D outputs diagnostic information through processor recovery messages. This information provides a gross diagnostic capability in the event of a complete system outage.

Tape load boot is an emergency procedure provided for the situation where a system has destroyed its only valid copies on disk of the generic software. Here the site would have a tape copy of the generic and data base, and read the tapes into the disk via the EAI tape load boot facility.

The final backup repair procedure is the dead start diagnostics. Primarily used as an installation tool, the dead start diagnostics allow for the repair of a completely sick processor by using a remote host processor.

The craft interface provides the mechanism through which the status of the system can be determined, the configuration of the system's hardware or software can be changed, and special emergency actions can be taken during catastrophic failures of system component.[8] The emergency action interface (Fig. 5) allows the craft in the field to access the system during times when a major portion of the system is nonfunctional to the point where the normal capabilities of the craft interface cannot be used. The limited capabilities of the emergency action interface include forcing failing hardware off-line or on-line, notification of the status of critical system resources, and forcing a reinitialization of the system.

### *4.4 Software integrity*

Section III described the architecture of the software integrity system. In this section, we describe some of the specific audit and overload measures that have been included in the DMERT system.

The DMERT audit package verifies the validity of critical data structures. Most audits exist throughout the system within the processes that control the data to be audited. In some cases, several audits are invoked consecutively to form a sequenced mode audit. Most requests for running audits come from an audit control structure, i.e., the audit manager.

Audits in DMERT verify data, not functions. The basic types of auditable data are system resources and stable data. Though most of the auditable data in the operating system resides in the kernel, additional data resides in other critical processes, such as the file manager and device drivers. Smaller amounts of auditable data reside in supervisor processes, such as in the *UNIX** operating system and the process manager.

Some audits, scheduled on a regular basis, are known as routine audits; others, scheduled on request, are known as demand audits. A partial list of the DMERT audits follows:

---

* Trademark of Bell Laboratories.

```
LAB 2        3B/DMERT  2.0.5.5                    <D>        01/28/82 18:30:31

 SYS EMER    CRITICAL    MAJOR      MINOR      BLDG/PWR     BLD INH    CKT LIM   SYS NORM
 TRAFFIC     SYS INH     CU         CU PERPH   LINK

CMD: 52!-OK                                  _____ EMERGENCY ACTION PAGE _____

                                          MTTY__8
CU-0_                  FOFL               EAI-0_ ASW   PRM-0 E700 0100 0101 00C7 79 D6 0C
CU-1_ ACT RUN FONL          RCVRY         EAI-1_ ASW   PRM-1 EA21 DD00 8300 0BD0 79 DA 04
SCCS_

                  SET CLR        CU-0 CU-1          SET CLR

10 FONL-0         20 21 PRI-DISK__                  30 31 BACKUP-ROOT__      50 APPL
11 FONL-1         22 23 SEC-DISK_  SET  SET         32 33 MIN-CONFIG_ SET    51 INIT
12 FONL-ACT       24 25 INH-TIMER_ INH  INH         34 35 INH-HDW-CHK__      52 BOOT
13 CLR-FONL       26 27 PRM-TRAP__                  36 37 INH-SFT-CHK__      53 BOOT + ECD
                                                    38 39 INH-ERR-INT__      54 BOOT + MEM
14 CLR-EAI        28 PRM-DUMP                       40 41 INH-CACHE__        55 LDTAPE-0
15 CFT-INIT                                         42 43 APPL-PARAM__       56 LDTAPE-1
```

Fig. 5—Emergency action interface page.

(*i*) Message buffers—This audit finds and frees lost message buffers, i.e., messages that have been on a process's queue for extended periods of time.

(*ii*) Scheduler—This audit checks for linkage errors in the scheduler's ready and not-ready lists.

(*iii*) Memory manager—This audit recovers lost swap space and corrects any overlap of swap space.

(*iv*) File manager—This audit checks all internal file manager structures: task blocks, buffers, mount table, etc. The audit corrects the information and has the ability to back out an aborted task and free its resources.

(*v*) File system—This audit is demanded by the file manager whenever a file system is mounted read/write. It checks and corrects the file system's super block free list, and free-block bit map. This audit verifies the integrity of the mounted file systems concurrent with their use.

## V. MAINTAINABILITY

The maintainability of the 3B20D system is the second vital component that guarantees the overall high reliability required of the system. There are conditions where automatic recovery is unable to restore the system to a fully functioning state. This is where maintainability is critical to satisfying DMERT's high-reliability requirements. The basic premise of maintainability is to provide basic data-gathering and data-analysis mechanisms as well as the ability to act on the results of that analysis. These mechanisms must be able to collect and analyze diagnostic and debugging information from various hardware and software components within the system in order to isolate the error. These mechanisms must then allow the craft to control and modify the configuration of the system based on the diagnostic and debugging information collected. Furthermore, these mechanisms must yield their information as quickly as possible while disturbing the rest of the system as little as possible.

Maintainability comprises such areas as diagnostics, transient-error analysis, routine maintenance procedures, field utilities, and plant measurements. Once the error has been isolated and analyzed, the problem must then be corrected as quickly and benignly as possible. This procedure is termed updatability, and it includes such aspects as growth and retrofit for hardware, emergency fixes, function update, and system update for software. Maintainability is quite naturally partitioned into diagnostics (hardware) and the various field utilities (software).[11] However, central to the ability of the craft to maintain and control the 3B20D hardware and software is the ability to interface to the various maintenance facilities provided within the system. This

is one of the very important capabilities of the craft-interface system. The craft interface provides the craft and others with the means to request diagnostics, receive error-analysis reports, initiate emergency-recovery procedures, gather plant-measurements data, and exercise routine maintenance programs. In addition, the craft-interface system allows configuration control by providing access to growth and retrofit procedures, system- and function-update capabilities, emergency-fix facilities, and the various field utilities. This section discusses the capabilities of the subsystems, which provide basic maintainability of the DMERT system. Diagnostics are discussed in Ref. 9.

One component of the maintainability required of DMERT-based systems is the ability of these systems to accept hardware and software changes in a way that does not interfere with their primary tasks. In other words, a DMERT-based system must be able to accept changes without disturbing call processing, networking, or other critical functions. DMERT supports this through several aspects of updatability. The first is growth; the ability to add or remove hardware and related software components to the running system. Growth extends from physically connecting new equipment—such as memory boards—through informing the system of its existence, exercising it, logically connecting it into the system's configuration, and committing its use in the system. Other subsystems—such as a hardware and software fault recovery and diagnostics subsystem—then take over to ensure that the new system component continues to be sane and usable.

The second aspect of updatability is retrofit: the ability to replace hardware components in the system with similar components of a different vintage or with different capabilities or interface characteristics. Retrofit procedures may "de-grow" or remove old units and then grow or add new ones. They also may add the new units first and then perform a transition from the old units to the new. Thus, retrofit of units may involve extensive periods of time where old and new units coexist in the system. Retrofit may also involve substantial software changes to interact with new units and to recognize the existence of both old and new units.

The third component of updatability, field update,[12] deals exclusively with software and data file changes in DMERT. Such changes are done logically, on a file-by-file or functional level. Just as with growth and retrofit, field update can install or replace system programs or files, inform the system about them, logically connect them into the system, exercise them in that state, and then commit to or back out of them. Field update is intended primarily for installing fixes or small features that do not perturb the system's architecture.

The fourth updatability component, system update, allows program and data changes of much greater magnitude, up to complete software

system replacement. A bootstrap is required to install the changes for any system update. By using disk redundancy or backup copies of sections of DMERT's disks, system update can prepare a new, partial, or total version of the system on disk and then switch to it (and back, if necessary). Where field update performs a logical change of files, system update does a physical change of a set of partitions (file systems and/or file partitions).

## VI. SUMMARY

This article has described the basic architecture of the fault-recovery and system-integrity subsystem for the 3B20D Processor. These subsystems are tied into the maintainability aspects of the processor. All of the features provided are responses to the reliability objective of no more than two minutes downtime in each year of service. The features and architecture continue in the tradition of former high-availability processors.

## VII. ACKNOWLEDGMENTS

The authors thank D. G. Gilbert, G. T. Surratt, B. G. Niedfeldt, and D. J. Fitch for their assistance with various sections of this article.

## REFERENCES

1. R. C. Hansen, "System Reliability Strategies," Proc. Nat. Elec. Conf., *35* (1981), pp. 40–51.
2. P. D. Carestia and F. S. Hudson, "No. 4 ESS: Evolution of the Software Structure," B.S.T.J., *6*, No. 6 (July 1981), pp. 1167–1201.
3. R. W. Downing, J. S. Nowak, and L. S. Tuomenoksa, "No. 1 ESS Maintenance Plan," B.S.T.J., *43*, No. 5 (September 1964), pp. 1961–2019.
4. P. W. Bowman et al., "1A Processor: Maintenance Software," B.S.T.J., *56*, No. 2 (February 1977), pp. 255–87.
5. T. F. Storey, "Design of a Microprogram Control for a Processor in an Electronic Switching System," B.S.T.J., *55*, No. 2, (February 1976), pp. 183–232.
6. M. W. Rolund, J. T. Beckett, and D. A. Harms, "The 3B20D Processor & DMERT Operating System: 3B20D Central Processing Unit," B.S.T.J., this issue.
7. Jonas Butvila, "Reliability and Its Impact on System Design," Proc. Nat. Elec. Conf., *35* (1981), pp. 43–7.
8. M. E. Barton and D. A. Schmitt, "The 3B20D Processor & DMERT Operating System: Craft Interface," B.S.T.J., this issue.
9. J. L. Quinn and F. M. Goetz, "The 3B20D Processor & DMERT Operating System: Diagnostic Tests and Control Software," B.S.T.J., this issue.
10. M. M. Meyers, W. A. Routt, and K. W. Yoder, "No. 4 ESS Maintenance Software," B.S.T.J., *56*, No. 7 (September 1977), pp. 1139–67.
11. G. P. Eldredge, and J. G. Chevalier, "The 3B20D Processor & DMERT Operating System: Field Utilities," B.S.T.J., this issue.
12. R. H. Yacobellis, J. H. Miller, and B. G. Niedfeldt, "The 3B20D Processor & DMERT Operating System: Field Administration Subsystems," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating System:*

# Diagnostic Tests and Control Software

By J. L. QUINN, R. L. ENGRAM, and F. M. GOETZ

(Manuscript received March 10, 1982

*Comprehensive diagnostic tests and multifeatured control software designed for execution on several host processors help craft to quickly isolate faulty hardware anywhere in the 3B20D Processor. Besides meeting the requirements for Bell System switching systems, the 3B20D diagnostics provide a high degree of modularity and portability using an operating-system-based structure. The diagnostics are used in a wide range of development, production, and maintenance activities throughout the project life cycle. Many features of the system architecture and hardware are provided to allow thorough diagnosis in a time-shared noninterfering manner. Additional features are provided in the diagnostic control structure to extend the DMERT diagnostic capabilities to application systems based on the 3B20D Processor.*

## I. INTRODUCTION

Many of the diagnostic principles and features embodied in the 1A Processor[1] have been incorporated in the maintenance design for the 3B20D Processor. These design principles include: (*i*) use of a special-purpose test-design language that facilitates test interpretation; (*ii*) use of a table-driven control program approach; (*iii*) use of a common test data base covering all hardware versions of the 3B20D Processor; (*iv*) partitioning of diagnostic tests into phases associated with specific hardware functions; (*v*) control features allowing selective test execution and variable degrees of detail in outputted results; and, (*vi*) optional automatic trouble location. In the 3B20D Processor design, however, a more general diagnostic design approach was followed. This approach resulted in a more portable diagnostic control structure;

it allowed diagnostic execution in several environments: factory testing using a support processor, installation testing using a remotely located processor, and in-service, on-line testing of a standby mate processor.

## II. OBJECTIVES

As with earlier processor designs, the 3B20D Processor diagnostics must be effective and efficient in fault detection, provide consistent test results, protect the contents of memory, be noninterfering with normal system operation, allow automatic trouble location, and be easy to maintain and update. In addition to meeting these objectives, the 3B20D diagnostics were required to be:

(*i*) Portable—The diagnostic software must execute in several environments. Throughout this paper the execution environment is referred to as the host processor (or computer).

(*ii*) Flexible—The diagnostics would test multiple system configurations containing various vintages of circuits.

(*iii*) Modular—Standard control interfaces must accommodate differing test access facilities to the processor under test, input/output facilities, and DMERT application processes that are used to diagnose application-dependent hardware that interfaces to the 3B20D Processor.

(*iv*) DMERT compatible—Diagnostics must be integral with, rather than separate from, the operating system.[2,3]

To meet these design objectives, the diagnostic control structure was designed as an integral part of the operating system and had to support the evolutionary stages of development.

## III. DIAGNOSTIC ENVIRONMENTS

As shown in Fig. 1, the 3B20D Processor can be diagnosed from several execution environments. During the early phase of processor development, a local host computer was used to support hardware, software, and diagnostic design. This access arrangement continues to be used in factory testing. Later in the development, more efficient use was made of the host computer by providing access to a remote target 3B20D Processor over a dial-up telephone line. Ultimately, in the standard duplex-system configuration, the active control unit is capable of diagnosing its own peripheral controllers and the standby control unit. Each of these access arrangements is discussed below.

### *3.1 Local host diagnostics*

Figure 1a shows three local-host access arrangements. In the first, diagnostic programs executing in a host computer send test inputs and receive test results over a standard communications port to a Micro-level Test Set (MLTS). The MLTS connects directly to the 3B20D

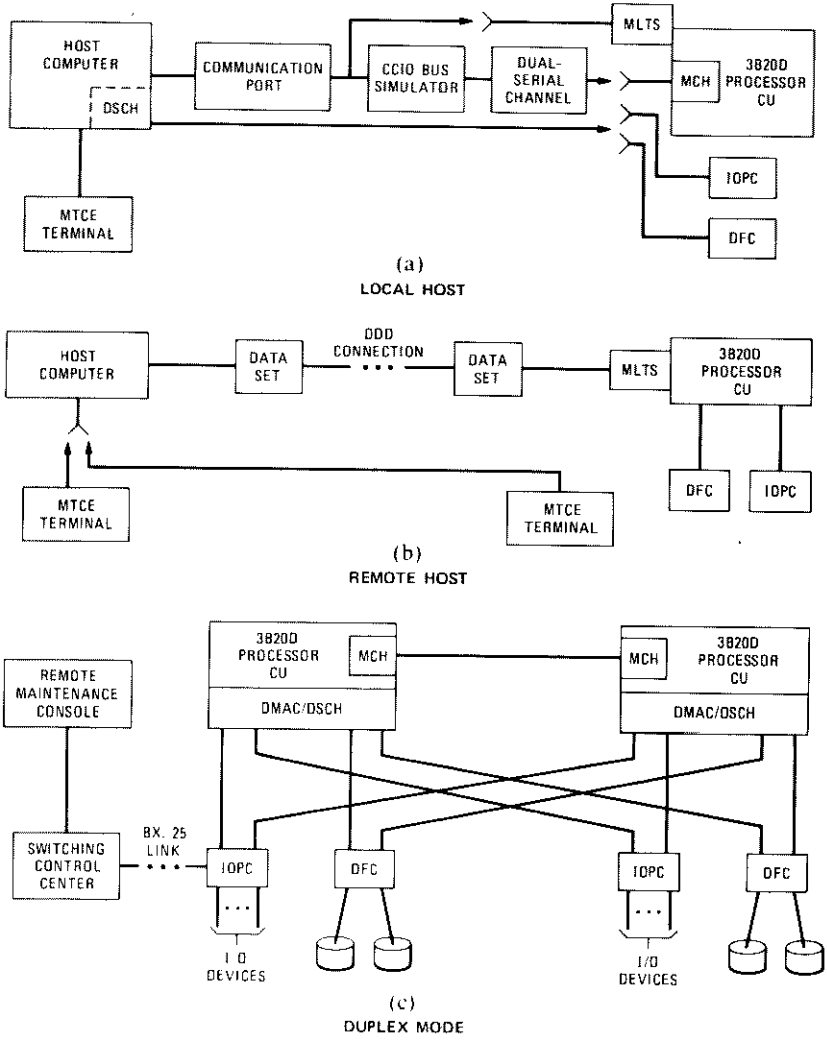



Fig. 1—The 3B20D Processor diagnostic environments.

control unit backplane, and provides complete access and control of the processor's microprogram control circuitry. For the second access path, a circuit was designed to simulate the Central Control Input/Output (CCIO) internal bus. The CCIO Bus Simulator (BS) is accessible using a standard communication input port. A Dual Serial Channel (DSCH) connected to the CCIO/BS can then communicate directly with a Maintenance Channel (MCH), the circuit designed for control-unit access. Like the MLTS the MCH can access the central control at a low level. However, only the MCH is used in the duplex configu-

ration (see Section 3.3); it communicates with either another MCH or a DSCH. As shown, the CCIO/BS–DSCH access path can also be used to diagnose the Input/Output Processor Controller (IOPC) and the Disk File Controller (DFC). Notice that when the local host is a 3B20D Processor, the path is from the DSCH of the host 3B20D Processor to the MCH, IOPC, or DFC of the target machine.

### *3.2 Remote host diagnostics*

The DSCH is designed to communicate over distances of approximately 100 feet. Remote-host (Fig. 1b) access arrangements can be used for diagnosing over longer distances. Using data sets and a telephone line, tests stored and executed on a remote computer can be applied through the MLTS to the control unit. Peripheral controllers (IOPC and DFC) can also be diagnosed by downloading tests into the control unit and executing them. Although remote-host diagnostics are useful in cases where a local host is unavailable, execution performance is limited by the transmission facilities used.

### *3.3 Duplex mode diagnostics*

The primary diagnostic execution environment is the 3B20D Duplex Processor (Fig. 1c). The active (on-line) processor acts as a local host for diagnosing the standby (off-line) processor. An MCH-to-MCH link provides the access path for testing the control unit. In the duplex mode, the DFC and IOPC are diagnosed from the on-line control unit using the operational interface path, a DSCH attached to the Direct Memory Access Controller (DMAC). Tests of the links from the off-line processor to the peripherals also can be run under the control of the active processor. As shown in Fig. 1c, the duplex system configuration also supports remote monitoring and control of diagnostics over a dedicated link to a Switching Control Center (SCC).

### *3.4 Multiple-target processors*

Although the target processor is always a 3B20D Processor, it can be of many types, versions, and sizes. The diagnostic control program accounts for these differences by referencing the Equipment Configuration Database (ECD). All information relevant to the particular diagnostic tests that should be applied to each hardware unit is contained in the ECD. This information includes the name of each hardware unit within a subsystem, subunits, and their logical interconnections, equipage options, and auxiliary information, such as channel address and baud rate. Whenever a circuit design is originated or updated, diagnostic tests are designed and appropriate ECD changes are specified.

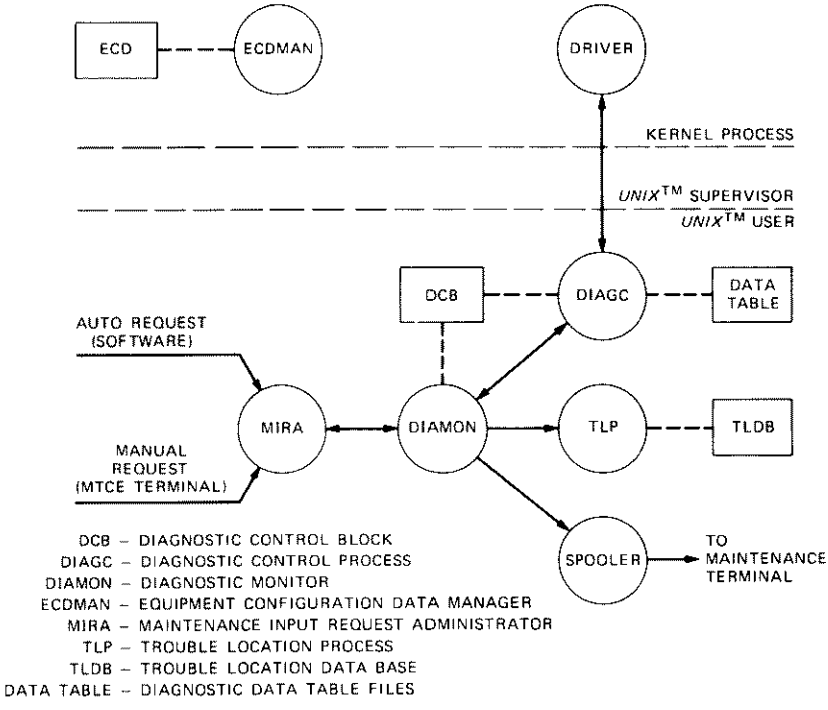



Fig. 2—Diagnostic control structure.

## IV. DIAGNOSTIC CONTROL STRUCTURE

The diagnostic control structure is depicted in Fig. 2. At the kernel process level are the modules that provide access to the ECD or drive the communication links previously discussed. The *UNIX** operating system supervisor resides at the supervisor level,[3] and provides a protected environment and operating system services for the higher-level processes. The modules operating under the *UNIX* operating system that pertain exclusively to diagnostics are: the Maintenance Input Request Administrator (MIRA), the Diagnostic Monitor (DIAMON), the Diagnostic Control process (DIAGC), and the Trouble Locating Process (TLP). Output messsages from the diagnostic structure are sent to the system spooler for printing. The first three of these modules are discussed below; the TLP is described in Section VII.

### *4.1 MIRA*

Scheduling and dispatching maintenance requests is the function of MIRA, the front-end process of the diagnostic structure. MIRA main-

* Trademark of Bell Laboratories.

tains a waiting queue and an active queue to administer each service request. Requests are serviced according to priority and resource availability; manual requests have higher priority than those initiated automatically. For each service request, MIRA spawns a DIAMON process and sends it a message. When the request is completed, DIAMON sends a message back to MIRA. Interfaces are provided in MIRA to administer routine exercise requests and inputs from the error-interrupt handler.[4]

Nine general types of request are handled by MIRA; they are described in Table I.

### *4.2 DIAMON*

Execution of each diagnostic is directed from start to finish by DIAMON. Specifically, DIAMON will:

(*i*) Initiate and control the actual diagnostic processes specified in a message from MIRA.

(*ii*) Communicate with the Equipment Configuration Database Manager (ECDMAN) and the appropriate device driver (the software control module for a particular hardware unit) to extract control data from the ECD and retrieve path names of related utility files.

(*iii*) Build the diagnostic control block containing all the data required by a diagnostic.

(*iv*) Spawn the appropriate diagnostic control process (DIAGC); separate processes are provided for the control unit and peripherals.

(*v*) Communicate diagnostic output to MIRA and the output spooler.

(*vi*) Spawn the remove and restore processes.

(*vii*) Interface with the TLP.

Table I—Description of diagnostic requests to MIRA

| Command | Description |
|---|---|
| Diagnose (DGN) | Diagnoses the unit specified in the request. |
| Remove (RMV) | Removes the specified unit from service. |
| Restore (RST) | Diagnoses the unit and restores it to service if all tests pass (ATP). |
| Restore Unconditional (RSTU) | Restores the unit to service without running the diagnostic. |
| Exercise (EX) | Starts the diagnostic in the interactive mode. This command allows stepping to a particular test, pausing, or looping over a diagnostic "phase" (ie., group of functionally related tests) segment. |
| Terminate (STOP) | Stops execution of a diagnostic. |
| Display (OP) | Displays status of queued requests in MIRA. |
| Inhibit (INH) | Inhibits diagnostic requests from other processes that automatically or routinely initiate diagnostics. |
| Allow (ALW) | Allows diagnostic sources, canceling any active INH request. |

### 4.3 DIAGC

DIAGC is a generic name that refers to a class of processes. The DIAGC is a unit or application-dependent module that controls execution of tests. Containing all unit-dependent task routines, DIAGC translates the interpretive diagnostics and provides the interface with DIAMON. A unit's diagnostic phase table (DPT) contains the name of a particular DIAGC process to be used in the diagnosis. DIAMON imposes no limit on the number of processes that can interface with it. The following functions are provided by DIAGC:

(*i*) Opens the diagnostic driver
(*ii*) Shares the buffer (DCB) provided by DIAMON
(*iii*) Initializes the raw data buffer
(*iv*) Executes the diagnostic
(*v*) Computes the test results
(*vi*) Provides interactive control if required
(*vii*) Provides an interface to DIAMON for test results and abnormal terminations (aborts).

### 4.4 Portability

All diagnostic control modules are written in the C language and execute in the *UNIX* operating system environment. This facilitates the porting of the control structure to other host processors that support C and *UNIX* operating system software. Variations of processor configuration and hardware vintage can be described in the ECD. DMERT application processes can provide additional DIAGCs and Data Tables to control diagnostic test execution for interfacing hardware. Several driver processes can be supported to allow diagnostics to be executed over standard communication ports, dual-serial channels, or maintenance channels.

## V. MAINTENANCE FEATURES

The combination of hardware-access circuits and modular-control programs, just discussed, provides the 3B20D Processor with considerable maintenance flexibility. Tests are selected according to the vintage of circuit under diagnosis. Displayed in Fig. 3 is a typical diagnostic input message in the PDS (Program Documentation Standards)[5] syntax, one of three command languages supported by DMERT. Requests can be made to diagnose an entire unit, a particular subunit, or all subunits in a specified community. Individual test phases or ranges of phases can be executed and the results printed with optional amounts of detail. Some diagnostic test phases—because of either their long execution time requirements or their dependency on other system hardware availability—are restricted to manual initiation. In-

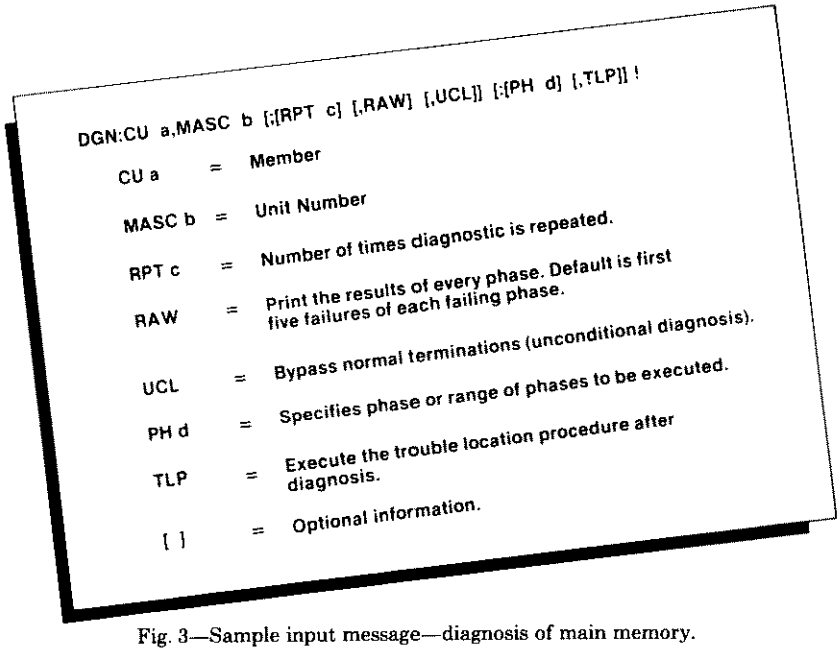



Fig. 3—Sample input message—diagnosis of main memory.

teractive features such as stepping, pausing, and looping are provided for facilitating difficult repairs. Units can be restored to service automatically if they pass all tests. Several host computer versions are supported along with application-dependent interfaces.

Diagnostics can be initiated either manually or automatically. Manual requests can be entered from either a local maintenance terminal or through a work-station terminal associated with a Switching Control Center System (SCCS) connected to the 3B20D Processor via a synchronous data link. Automatic requests originate from other software modules such as the error-interrupt handler, the routine exercise scheduler, or an application-software module.

## VI. DIAGNOSTIC TEST DESIGN

### *6.1 General*

The stringent availability requirements of Bell System applications using the 3B20D Processors had a significant impact on all aspects of system design. Diagnostic and maintenance engineers were actively involved in meeting these goals commencing with the initial architectural planning and requirements generation. Many hardware features are provided to monitor system integrity, to detect errors, to reconfigure the system, and to facilitate repair of the faulty equipment.[4,6] Although some of the features are for fault isolation during pack repairs, most are used at the system level to effect repair through pack

replacement. Diagnostics, the primary repair capability for the system, make extensive use of these hardware features for control and observation of the circuitry.

### *6.1.1 Circuit pack tests*

The initial factory testing of circuit packs uses test vectors, which can be applied at terminals of the pack connector in a commercially available computer-controlled test set. Most of the vectors are generated independently from system-level diagnostic tests. The packs are given additional tests in a 3B20D system test bed using diagnostics and some operational sequences.

### *6.1.2 System-level tests*

All 3B20D Processor diagnostics run under the DMERT operating system as user-level processes. To communicate with the unit being tested, the user-level process passes the test scenarios to a kernel process that interfaces to the hardware. Each of these kernel process drivers runs at its standard system priority level to perform the tests. If some time-critical tests are necessary, the priority level can be elevated to avoid interruption by other system processes.

Each of the diagnostic programs is structured to avoid any negative impact on the normal system functions. Special driver functions allow the drivers to handle error conditions generated by the diagnostic tests, thereby avoiding the normal error-handling routines. Since many fault conditions result in system errors, this capability is especially vital to allow thorough testing in the operating system environment.

In the Control Unit (CU) diagnostic, additional safeguards are implemented to assure proper handling of the system recovery and integrity hardware. Even with faults in the off-line CU integrity circuits, the system will maintain normal functionality during CU diagnostics.

The system diagnostics are organized on a unit basis, for example the Control Unit (CU), I/O processor (IOP) or Disk Controller (DFC). Each unit diagnostic is structured into test phases that pertain to a particular subunit. The phases are organized in a hierarchical fashion beginning with the more elemental operations and applying to the hard core of a subunit. Subsequent phases expand in complexity and in the totality of the circuitry exercised.

The user-level diagnostic processes, namely the control program and the test data tables, contain all of the information necessary for control and sequencing of tests. The control program has the interpretative routines for decoding each data table test statement. The program also can use various system configuration parameters, test results, and data table decision functions to modify program flow or to terminate the diagnostic.

#### 6.1.3 Maintenance channel access

As shown in Fig. 1c, the primary interconnection between the Control Units is the maintenance channel (MCH). This circuit is an enhanced version of the dual-serial channel, with special capabilities aimed at maintenance access and control of the off-line CU. It provides the ability to run, stop, load, clear, and step the CU. The MCH allows the active CU to read some off-line CU registers directly and others indirectly using microcoded sequences. Diagnostic-test programs can be loaded through the MCH to the off-line microstore or mainstore. The MCH is controlled by a DMERT kernel process driver that carries out the diagnostic test sequences.

### 6.2 Control unit diagnostics

The Control Unit (CU) has seven types of subunits: Central Control (CC), Main Memory Store (MAS), Store Address Translation (SAT), Direct Memory Access (DMA), I/O Channels (DSCH), Cache (CSU), and Utility Circuit (UC). The latter three are optionally equipped; the UC is normally used for program testing and is not further discussed herein. Some 3B20D Processor applications have special circuits that are part of the CU; diagnostics for them are concatenated to the CU diagnostic. A pictorial view of the CU hierarchy is shown in Fig. 4, which depicts the multiple levels of units as defined in the ECD.

#### 6.2.1 Central control tests

The first CU subunit tested is the CC, which contains the core of the CU. The CC diagnostics in turn test the maintenance channel, microcontrol logic and memory, registers, data-manipulation logic, memory access, timers, interrupts, I/O interface, error-control hard-

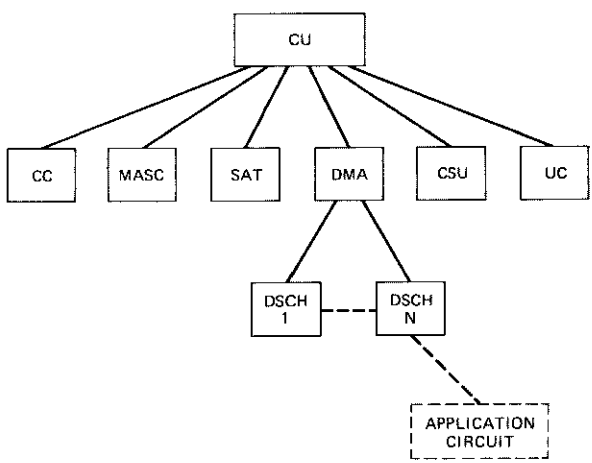



Fig. 4—Unit hierarchy of the control unit.

ware and integrity circuits. The testing uses a series of MCH operations for the basic tests. For the more complex test routines, down-loaded microcoded test programs are executed by the CU under test. The CC is extensively exercised in tests of the remaining subunits.

#### *6.2.2 Memory tests*

Initial testing checks out the basic memory-controller operations and the control and data paths from CC to the MAS. Tests are carried out on the error-detection and correction circuits in preparation for using them in array testing. Testing of the memory arrays (up to 16M bytes can be equipped) is with down-loaded microdiagnostic routines. Since the test-pattern programs are executing autonomously in the CU under test, all of its real time is used for testing. Whenever a hardware error is generated, control of the CU passes to a diagnostic error handler. The combination of self error detection and the micro-routines allows extensive pattern checking to be executed rapidly over the complete memory spectrum.

#### *6.2.3 Store address translation tests*

Functional tests are performed via MCH on all of the SAT control logic. The memory cells are then tested with various microcoded test patterns. The remainder of the tests, implemented as microdiagnostics, check out the multiplexor, compare logic, matchers, protection logic, and SAT to MAS interface.

#### *6.2.4 Cache tests*

The cache is comprised of a high-speed four-way-set associative memory and a 2K by 36-bit interrupt stack. The diagnostic performs extensive tests of the memory cells, matchers, and select logic. In addition to functional tests, a special diagnostic routine called the cache exerciser is used to stress, at high data rates, the cache interfaces to CC, MAS, and SAT. This kind of testing is effective at detecting marginal fault conditions.

#### *6.2.5 Direct memory access tests*

The DMA diagnostic checks out the CC-to-DMA communication and control paths, the internal DMA functionality and the DMA operations to MAS. Many of the tests are coded into ROM (Read Only Memory) contained in the DMA. The remainder consist of down-loaded microcoded tests and off-line, main-memory, resident test programs. The final sequence of tests verifies the DMA cache "handshaking" operations. It is noteworthy that in the DMA diagnostics, except for control and down loading through the MCH, all test sequences are executed completely by the CU under test.

#### 6.2.6 Channel tests

The channel diagnostic carries out the remainder of the testing of the CU's I/O capability. Basic tests are performed on the communication and handshaking of the CU to all in-service system peripherals. More exhaustive tests (demand diagnostics) can be specified by the maintenance personnel for troubleshooting more elusive problems. These diagnostic phases require that a peripheral unit be configured as a "helper unit" (specified in the diagnostic input message) to allow the CU to carry out peripheral operations at a high rate.

### 6.3 Peripheral unit diagnostics

The Disk File Controller (DFC) and Input Output Processor (IOP) are described in Refs. 7 and 8. The DFC can control up to eight Moving Head Disks (MHD) of various capacities, types and manufacturers. The Peripheral Controllers (PC), which are under control of the IOP, are special-purpose I/O units described in Ref. 8. The testing for the DFC and IOP, which share a common front end, is primarily carried out under control of the on-line CU. Since both of these are intelligent controllers, many of the specific tests can be executed autonomously. The peripheral diagnostics utilize the DMERT kernel process drivers to interface to the hardware. Throughout these diagnostics, extensive use is made of driver-maintenance orders and special handling of error conditions.

#### 6.3.1 IOP and DFC tests

The peripheral diagnostics use common-control programs IODIAG and DFDIAG that contain all the CU resident tests and control routines. Separate sets of data table and down-loaded microcode files are used for each unit diagnostic. The overall sequence of testing proceeds from CU/controller interface to complex internal controller operations. Most of the latter make use of the operational firmware in the controller to carry out the test sequences. The more complicated controller tests are part of the resident diagnostic firmware and are initiated by special-driver operations. At the successful conclusion of DFC or IOPC testing, the unit is restored to service to allow testing to proceed on MHD or PC circuits.

#### 6.3.2 Moving head disk tests

Relatively limited maintenance capabilities are provided in the MHD itself. Most of the testing is carried out by the firmware routines in the DFC. To provide an overall check on MHD performance, one cylinder is devoted to diagnostic testing of each read/write head. The error-detection/correction capabilities can also be checked using this

area. As each MHD is tested successfully it can be updated from its mate copy and restored to service.

#### *6.3.3 Peripheral-controller tests*

Each controller is microprocessor controlled and can carry out most of its diagnostics autonomously. Some of the tests are firmware resident in the PC's. The remainder of the diagnostic routines are downloaded into the PCs' RAM (Random Access Memory). Some types of PCs also can exercise the units they control, for example a tape transport, and report back the results for use by the maintenance personnel.

## VII. TROUBLE LOCATING PROCESS

If the diagnostic request specifies the TLP option, the TLP process is invoked at the completion of diagnostic testing. The process compares characteristics of the failures with a resident data base of fault signatures. In each data table, the designers have partitioned the tests into groups. Any test failure in a group will set a flag bit, called a key, which is permanently assigned to the group. The TLP search, based on the phase and key information, results in a rank-ordered list of closest signatures and, ultimately, into an ordered list of suspected faulty equipment. This approach makes the data base and process less sensitive than earlier methods to circuit or test changes and to marginal failures. The data base (TLDB) is generated off-line from the results of physically inserting faults into units in a test laboratory. Test engineers also can modify the TLDB directly by inserting information into the test data tables. Fig. 5 depicts a typical diagnostic output message from a faulty memory unit.

## VIII. EVALUATION

Although many diagnostic tests were generated with the aid of hardware logic simulators, many tests were developed manually. To assure that the diagnostics met the objective—at least 90 percent of the simulated faults detected—an extensive evaluation process was carried out. Using physical fault insertion at the DIP (Dual In-Line Package) terminals, many thousands of faults were inserted. This approach has provided timely and effective design feedback for diagnostic test and TLDB development.

## IX. CONCLUSION

In addition to providing a variety of test-control options, the 3B20D diagnostics were designed for multiple execution environments. As a result, the diagnostics have been useful throughout the development

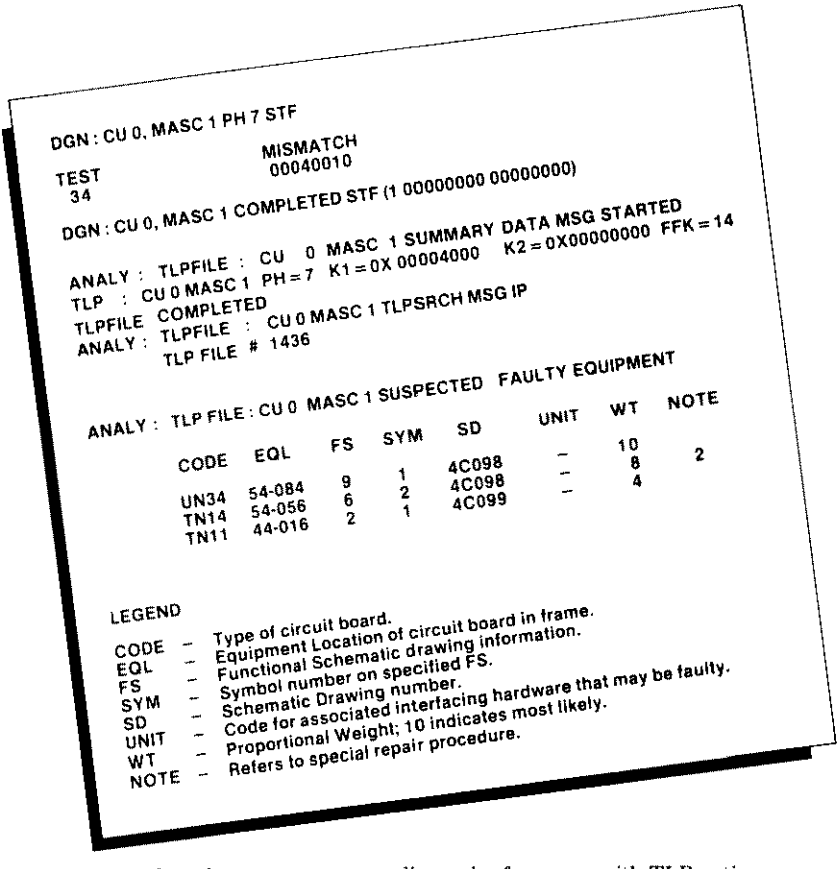



Fig. 5—Sample output message—diagnosis of memory with TLP option.

cycle and have supported the design laboratory, factory testing, installation, normal system operation, application interfaces, and field support. These diagnostics are the major tool for validating the 3B20D Processor hardware and for isolating any faults. The provision of a high degree of hardware self-checking, standby and active redundancy, self-diagnosis, microdiagnostics, and remote testing capability all have contributed to making the 3B20D Processor a high-availability real-time system. Coupled with the DMERT operating system, with its robust complement of features, the 3B20D Processor meets the needs of a wide variety of Bell System projects.

## X. ACKNOWLEDGMENTS

The authors would like to acknowledge the many designers and testers at Bell Laboratories and Western Electric whose combined efforts resulted in the diagnostic package described herein.

## REFERENCES


1. P. W. Bowman, M. R. Dubman, F. M. Goetz, R. F. Kranzman, E. H. Stredde, and R. J. Watters, "1A Processor: Maintenance Software," B.S.T.J., *56*, No. 2 (February 1977), pp. 255–88.
2. J. R. Kane, R. E. Anderson, and P. S. McCabe, "The 3B20D Processor & DMERT Operating System: Overview, Architecture, and Performance of DMERT," B.S.T.J., this issue.
3. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman, "The 3B20D Processor & DMERT Operating System: DMERT Operating System," B.S.T.J., this issue.
4. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "The 3B20D Processor & DMERT Operating System: Fault Detection and Recovery," B.S.T.J., this issue.
5. M. E. Barton and D. A. Schmitt, "The 3B20D Processor & DMERT Operating System: Craft Interface," B.S.T.J., this issue.
6. M. W. Rolund, J. T. Beckett, and D. A. Harms, "The 3B20D Processor & DMERT Operating System: Central Processing Unit," B.S.T.J., this issue.
7. R. E. Haglund and L. D. Peterson, "The 3B20D Processor & DMERT Operating System: 3B20D File Memory Systems," B.S.T.J., this issue.
8. A. H. Budlong and F. W. Wendland, "The 3B20D Processor & DMERT Operating System: Input/Output System," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

***The 3B20D Processor & DMERT Operating System:***

# 3B20D Craft Interface

By M. E. BARTON and D. A. SCHMITT

(Manuscript received March 10, 1982)

*The 3B20D craft interface package includes hardware, firmware, and software that enables telephone company craftspeople to obtain the status of and exert control over the system. Because this package consists of one or more standard keyboard-display terminals for human-machine interactions, it is flexible and can be adapted to a broad variety of applications. Furthermore, the use of standard terminals and data link protocols allows for inexpensive remote access with capabilities similar to local access capabilities. Finally, the use of video displays has made it possible to provide easy-to-use menus that guide the craftspeople through some of the complex control operations. This article describes the 3B20D craft interface capabilities and the internal architecture of the package.*

## I. INTRODUCTION

The "craft interface" is that part of the 3B20D Processor that enables people to obtain status information and exert control over the system. To those not involved in telephony, the word "craft" may seem odd. It has traditionally been used to refer to the people who work in and around telephone switching offices performing various maintenance functions on the equipment. In this article, the term is used somewhat liberally to mean any person who interacts with the 3B20D to perform administrative and maintenance functions.

The 3B20D's craft interface is a marked departure from previous systems developed at Bell Laboratories because it relies almost exclusively on video displays and keyboard controls instead of the key-lamp panels and teletypewriters usually found in the Master Control Center (MCC) of electronic switching systems. Status information is presented

visually as graphical displays and text messages on various terminals and printers. There is also a capability to provide audible status by connecting the 3B20D to an audible alarm circuit. System control is exerted primarily via a keyboard attached to the video display terminal, although the 3B20D also includes a separate power control panel for each major hardware unit.

Another important enhancement lies in the ability to access and control all aspects of the system from remote locations such as Switching Control Centers (SCCs). In the past, remote access was obtained by "piggy-backing" data links onto the typewriter terminals in the telephone office and by connecting a telemetry unit to the key-lamp control panel. The 3B20D has introduced a more "intelligent" data link using the CCITT X.25 communication protocol. This link can carry considerably more information and is less vulnerable to noise and other data communication failures. Furthermore, the use of the international standard message protocol (X.25) will standardize remote access to the 3B20D via packet switching networks.

This article first provides an overview of the 3B20D craft interface, primarily concentrating on how the system appears to the craftspeople. Then the internal architecture is described and the various 3B20D applications usages of the general facilities provided in the common system are explained.

## II. OVERVIEW

This section describes the 3B20D craft interface as it appears to the people who use it to administer and maintain the system.

### *2.1 Hardware*

The most frequently used parts of the craft interface are shown mounted in two equipment frames in Fig. 1. The left frame contains a "read-only printer" or ROP* on which all important status messages are logged. The right frame contains a keyboard-display terminal that is commonly referred to as the "maintenance CRT," or MCRT. Telephone switching applications of the 3B20D can choose either a frame-mounted or desk-mounted arrangement for the ROP and MCRT. A desk mounted version is shown in Fig. 1.

### *2.2 Text messages*

One way in which the 3B20D communicates with the craftspeople is via text messages. For example, when the message

* From the viewpoint of a programmer, it is a "write-only printer," since the programmer can only send (i.e., write) messages to it. However, the craftsperson cannot type on this device, and so from that viewpoint it is a "read-only printer."

```
DGN:CU 0;UCL!
```

is typed, the central processing unit 0 (CU 0) is diagnosed. The UCL keyword indicates that the diagnosis is "unconditional," which means that all tests will be run even if some of the early tests fail. When the diagnosis is complete, the CU diagnostic prints a text message such as:

```
DGN CU 0 COMPLETED ATP
```

This means that the diagnosis has been completed and all tests passed (ATP). For initial 3B20D applications, the text messages conform to the Bell System craft interface syntax, commonly known as the Program Documentation Standard (PDS) Language. However, all new switching systems developments will be adopting a craft interface language sanctioned by the International Telegraph and Telephone Consultative Committee (CCITT) under the name MML. Since PDS and MML are similar, and since the 3B20D is expected to enjoy broad use in international applications, the operating system was designed so that each application can easily choose the appropriate syntax.

Text messages are typed on the MCRT keyboard, and the response messages are displayed on the MCRT video display and/or printed on the ROP. The basic repertoire of messages available with the 3B20D covers a broad range of maintenance and administration activities. Each application can easily add its own messages to this repertoire.

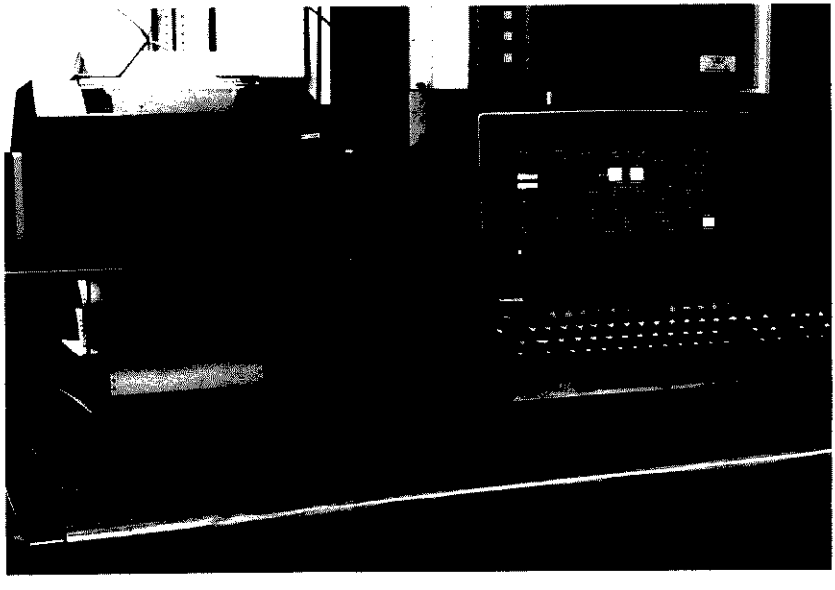


Fig. 1—Craft interface printer and terminal.

### 2.3 Control and display functions

As mentioned earlier, previous systems used key-lamp panels to display system status and to receive control signals from the craftspeople. The 3B20D uses the MCRT video screen and keyboard, as shown in Fig. 2, in place of such a panel. The upper part of the screen always contains a summary of important system indicators, including CRITICAL, MAJOR, and MINOR severity alarms and "type" alarms, such as CU and BLDG/PWR, which is the indicator for building power. The middle part can display a variety of "pages" that show system status in a graphical form. Finally, the lower part of the screen is used for text input and output.

The standard 3B20D software includes several display pages related to the common processor equipment, and each application can easily add its own pages. The "Common Processor Display Page" shown in Fig. 3 provides a diagram of the redundant components in the basic processor complex. At the left of the diagram is a "menu" listing the control operations that can be invoked when this page is displayed. To select a menu item, the craftsperson depresses the CMD/MSG key, which switches the craft interface from text message mode to command

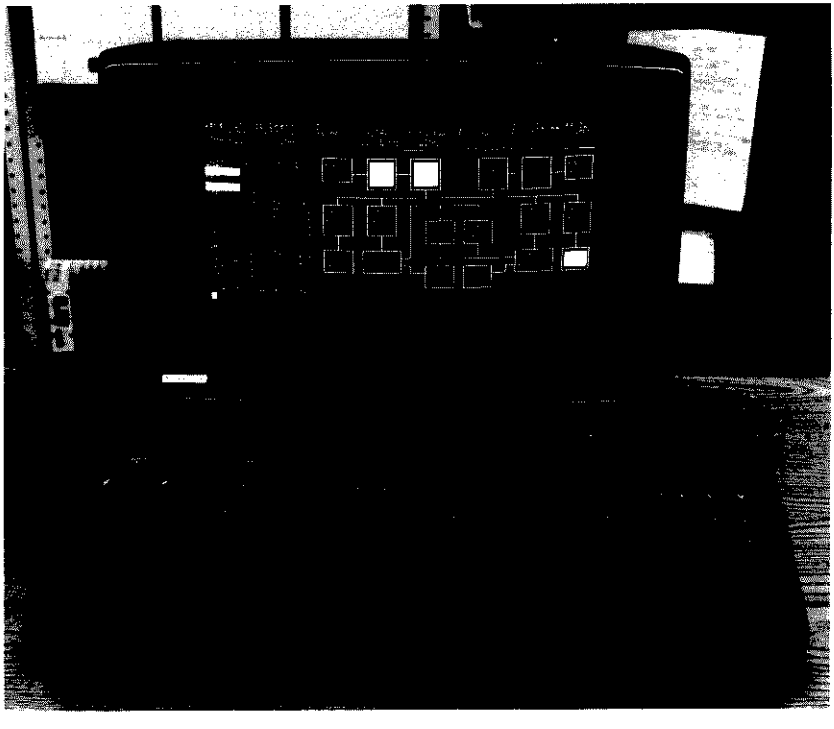


Fig. 2—Craft interface video screen and keyboard.

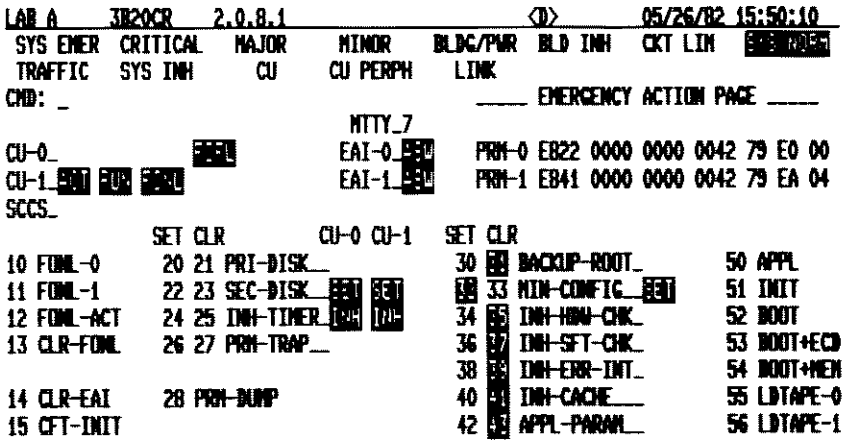



Fig. 3—Emergency action interface display page.

mode. Then the craftsperson types the menu item number, replacing "x" with a 0 or 1 where necessary. Finally, depression of the RETURN key (or the ! key) causes the command to be executed.

The craft interface stays in the command mode until the CMD/MSG key is depressed again. This key is one of four "special function keys." The ALM RLS key is used to retire audible and visual alarms. The EAI DISP key places the craft interface in the emergency action mode, which is described below. When in EAI mode, the NORM DISP key returns the screen to its previous display.

The Emergency Action Interface Page is different from other pages because it is directly controlled by a microprocessor in the MTTY controller (MTTYC) and, therefore, can be used even when the 3B20D software is not operating. As shown in Fig. 4, this page contains menu items that enable the craftsperson to re-initialize the system or to force the redundant units into a particular configuration. Typically, this page is used only when system sanity is suspect.

### *2.4 Remote access*

All capabilities of the craft interface except the power control panels can be accessed from a remote maintenance center via a dedicated data link that is attached to the MTTYC. The standard arrangement includes a primary and a backup link, both of which use the CCITT X.25 communication protocol. The remote site is usually a Switching Control Center (SCC) that contains a collection of computers and terminals that interface with these X.25 links and provide the SCC craftspeople with sophisticated analysis and maintenance tools.

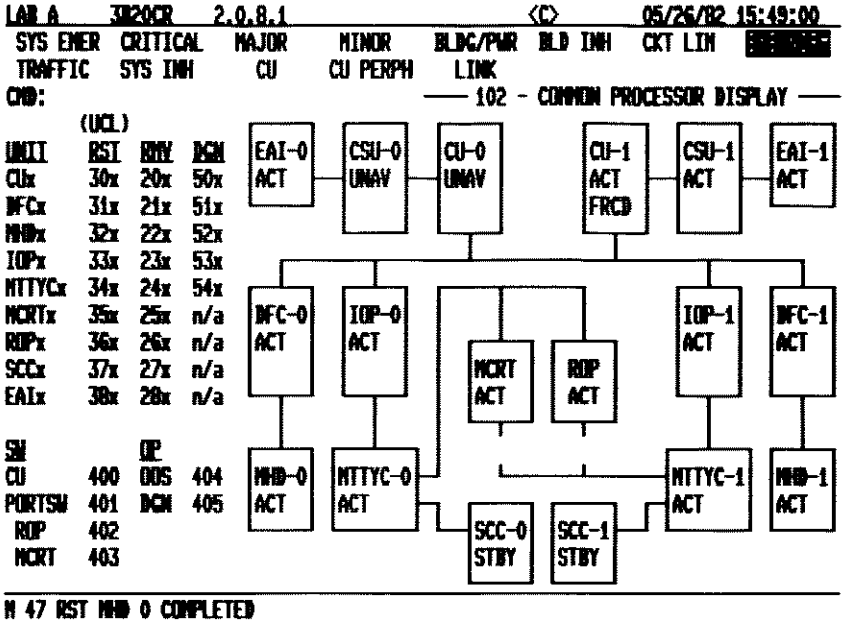



Fig. 4—Common processor display page.

## III. CRAFT INTERFACE ARCHITECTURE

This section discusses the hardware and software architecture of the 3B20D craft interface. Figure 5 shows the arrangement of hardware units pertinent to the craft interface, while Fig. 6 shows the software modules. The discussion of the hardware architecture that follows will cover the I/O Processor (IOP) driver and MTTYC handler software, as they are the fundamental parts of DMERT required to access the hardware.

### *3.1 Hardware architecture*

Referring to Fig. 5, one sees that each of the duplex processors is connected to both IOPs, and that each IOP supports up to sixteen peripheral controllers (PCs). Various PCs exist for terminals and printers, data links, tape units, etc. The IOP driver process, which is the part of DMERT responsible for communication with the IOPs, contains "handlers" that deal with the specialized functions of the PCs. The following handlers are pertinent to the craft interface: craft interface handler, X.25 handler, emergency action interface handler, general-purpose terminal handler, and alarm handler.

#### *3.1.1 Craft interface handler*

The MCRT, ROP, and X.25 links are attached to a PC known as the maintenance teletype controller, or MTTYC. The craft interface

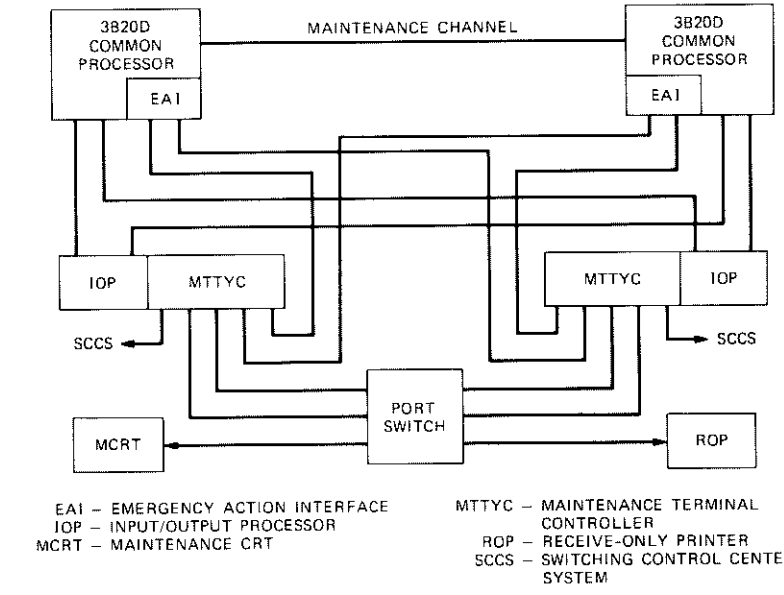



Fig. 5—Craft interface hardware overview.

handler controls the transfer of data to and from the peripheral devices associated with the MTTYC. The MCRT and ROP are administered directly by the MTTYC, while the X.25 links require the additional services of the X.25 handler described later. For each device or data link attached to the MTTYC, the handler supports all standard access operations of the *UNIX** operating system. In addition, this handler treats the single MCRT terminal as two "virtual terminals," with the upper part of the screen used for control/display functions and the lower part used for text messages as shown in Fig. 2. Each virtual terminal appears to the higher-level software as a separate device.

#### *3.1.2 X.25 handler*

The X.25 handler provides communication with a remote maintenance center via 1200 to 9600 bits per second synchronous data links using levels 2 and 3 of the CCITT X.25 protocol. Level 2, referred to as the link layer, provides link initialization, error control, and flow control on the physical data link and is implemented as firmware within the MTTYC. Level 3, referred to as the packet layer, multiplexes several independent data streams (logical channels) on the physical link and is implemented by the X.25 handler software. In effect, the X.25 handler makes the single MTTYC look like a multitude of independent communication channels.

* Trademark of Bell Laboratories.

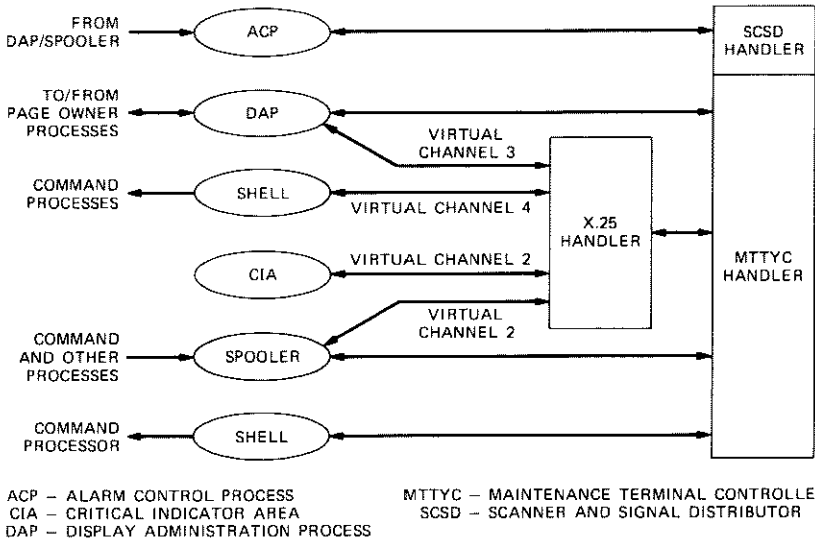



Fig. 6—Craft interface software architecture.

### *3.1.3 Emergency action interface handler*

The Emergency Action Interface (EAI) is a processor circuit pack that provides basic status information and manual control even under extreme circumstances. That is, the EAI circuit gives craftspeople limited access to the 3B20D regardless of DMERT software sanity. This access is in the form of the EAI page display shown in Fig. 4, which is controlled totally by the firmware in the MTTYC. The MTTYC interacts with the EAI circuit via the connection shown in Fig. 5 to acquire the status information for display on the MCRT. Also, when the craftsperson selects a menu item from the EAI display, the MCRT delivers the corresponding commands to the EAI circuit.

The emergency action interface handler only comes into play when DMERT is operating sanely. It has two major functions. First, it periodically "punches in" with the EAI circuit to indicate that the software is operating correctly. If the EAI (and, subsequently, the MTTYC) fails to receive this periodic signal, it will automatically initiate a system recovery operation to restore software sanity. Second, the EAI handler receives some non-emergency commands from the MTTYC via the EAI, including non-emergency initialization and reconfiguration signals.

### *3.1.4 General-purpose terminal handler*

The craft interface subsystem supports terminals other than the MCRT and ROP via the teletype controller, or TTYC. This controller

offers all MCRT and ROP functions except access to the EAI circuit. Its handler is similar to the craft interface handler used with the MTTYC and, in particular, offers the dual virtual terminal mode of operation needed to intermix control/display functions and text functions on a single terminal. This capability is typically used for dial-up monitoring of a system from a Western Electric or Bell Laboratories product support center.

#### *3.1.5 Alarm handler*

The scanner and signal distributor (SCSD) peripheral controller provides sense and control points that are tied into the system power controls and alarms. The SCSD handler detects sense point state changes and sends commands to change the states of control points. Higher-level software uses these capabilities to detect situations such as power removal, fuse operation, or thermal warnings and to respond by activating audible alarms or power shutdown circuits. The application can also tie into the SCSD and configure the higher-level software to detect and react to such things as building intrusion alarms.

### *3.2 Software architecture*

Figure 6 shows the modules that comprise the standard DMERT craft interface software subsystem. Already discussed were the device handlers that connect to the modules on the right of the figure. Each application usually adds its own modules that tie into the interfaces on the left. Also, many other parts of DMERT (e.g., the diagnostic subsystem) connect to the craft subsystem via these front-end interfaces. The modules in the middle of the figure are the "workhorses" of the craft subsystem and provide the internal interfaces used by DMERT programmers to interact with the craft personnel.

#### *3.2.1 Text input processing (shell)*

The Shell is the module that interfaces between the handlers and the processes that respond to text input commands. The term "shell" is borrowed from the *UNIX* operating system, and DMERT's craft shell operates in a manner similar to the other shell. That is, the DMERT shell reads an input line, parses it into a command verb and a list of "tokens," searches for the command process in the appropriate disk directory, creates the command process, and passes the list of tokens to it. The command process has access to a "shell library" that includes functions to do further parsing of the tokens.

The major difference between the DMERT shell and the other shell is that DMERT must parse commands that are typed in either the PDS or MML syntax, where as the *UNIX* operating system shell uses

a more general syntax for a broader variety of applications.* Another difference is that the PDS and MML languages include the notion of a "locking acknowledgment." That is, an input command process is required to give a two-character response (e.g., OK if the command is successful, PF if a printout follows) within a few seconds after the person types the message, and no other command can be typed until the acknowledgment appears. In the *UNIX* operating system, message acknowledgments are not required and command type-ahead is allowed. Therefore, the DMERT shell library includes functions that pass the acknowledgment back to the handler in order to unlock the terminal.

Referring again to Fig. 6, note that each text input channel has its own instance of the DMERT shell. This allows each channel to operate independently of the others, which means that several craftspeople can simultaneously interact with the system.

#### *3.2.2 Text output processing (spooler)*

The DMERT output spooler accepts text strings from higher-level processes and directs them to the appropriate output devices. [The term "spooler" is a computer science anachronism that comes from the days when information waiting to be printed was temporarily stored on reels (spools) of magnetic tape.] One might ask why the higher-level process doesn't write directly to the device (via the device's handler, of course). There are two reasons to avoid direct writing.

First, the PDS and MML languages require that each message be enclosed in an "envelope" that clearly delineates the message. This envelope generally contains a time stamp, a message priority/alarm indicator, and an end-of-message delimiter. The time stamp can be as simple as the number of minutes past the current hour, or it can be the complete date and time. The priority/alarm indicator shows whether the message is a result of a manual or an automatic action and whether the action being reported requires immediate attention. Finally, the end-of-message delimiter provides for automatic logging and browsing of messages by a computer in the remote maintenance center. Centralizing the generation of the output message envelope in the spooler simplifies the work that higher-level processes must do to produce text output. Also, changes or additions to the envelope can be introduced easily by modifying only the spooler.

The second reason for using the spooler approach is that many messages must be sent to several places. For example, the usual mode

* For DMERT applications that require the more general *UNIX* shell on terminals other than the MCRT, it is possible to configure the system in such a way that the *UNIX* shell is automatically activated on some or all general-purpose terminals. In other words, both the DMERT and the *UNIX* operating system shells are compatible with the general-purpose terminal handler described earlier.

of operation when a remote maintenance center is attached via the X.25 links is to send every output message to both the MCRT, ROP, and the remote center. However, if the ROP runs out of paper or if there are no craft personnel near the 3B20D, messages can be routed only to the remote center. To handle these and the diverse message routing situations that can arise, the spooler maintains a "map" showing where messages are to be routed based on their priority/alarm indicator and on a message type code that is received from the process that generated the message. The map also can be configured to route some message types to disk files instead of or in addition to printing them. This feature is useful for keeping a log of messages that are sometimes needed for problem analysis but that would overload the ROP or X.25 links if sent routinely. Input commands are provided to print the contents of these logging files when needed.

#### *3.2.3 Control/display processing*

The Display Administration Process (DAP) administers the upper part of the MCRT (and, possibly, other video terminals) containing the displays that replace the traditional key/lamp panels for 3B20D applications. DAP's fundamental purpose is to display "pages" from its repertoire and to accept commands listed on "menus" associated with the pages. Figure 3 shows the Common Processor Display Page, which is one of the standard pages delivered with DMERT. Typically, the majority of display pages are defined by the specific application processes.

For each page, there is a Page Description File (PDF) containing a pseudo-program that describes how the page should be "painted" on the video screen and what menu selections are allowed. PDFs are constructed like programs and compiled by a page description file generator (PDFGEN) program.

***3.2.3.1 Display functions.*** When DAP begins execution, no pages are active. Then, as the various parts of DMERT and the application are initialized, they send interprocess messages to DAP requesting that specific PDFs be loaded into main memory and activated as display pages. A maximum of 64 pages can be active at any one time. Other interprocess messages tell DAP which of the active pages to display on each video terminal.

Each page consists of up to 128 graphical constructs known as "indicators," a term reminiscent of the lighted indicators on a key/lamp panel. The process that initially informs DAP to activate a page becomes the owner of that page and it and other processes can subsequently inform DAP (via interprocess messages) to change the states of the indicators. For example, one popular type of indicator is the rectangle enclosing a few text items, such as the CU-0 box in Fig.

3. The CPDP page owner can send messages to DAP causing the phrase UNAV to change to OOS when control unit 0 (CU-0) changes from the unavailable state to the out-of-service state.

These state changes can be communicated in detail, for example, by sending a message to DAP specifying the characters OOS to replace UNAV. However, the usual method is to use state numbers instead of the actual characters. DAP has access to a table of 256 state entries specifying the standard text and video attributes associated with each of the 256 possible indicator states. The standard maintenance states, such as active, standby, and out-of-service, have predefined state numbers, and each application can define additional states for its own needs. The advantage of using state numbers is that the text and video attributes for each state can be centrally controlled. For example, one application could use the text ACT for the active state while another application used ACTIVE, and the only difference would be in the state table definition entries.

Video attributes were mentioned above in addition to the text that can be associated with an indicator. For the usual black-and-white terminals, DAP recognizes the "blink" attribute and the "reverse" attribute. The conventional use for the reverse attribute is to show that an indicator is, in some sense, active. In other words, a reversed indicator is similar to a lit lamp on a key/lamp panel. The blink attribute is used to draw attention to a situation that requires immediate action, just like a flashing lamp. In Fig. 3, the SYS NORM indicator is reversed to show that the system is operating normally. If a major alarm occurs, the MAJOR indicator will blink until the ALM RLS key is pressed.

DAP also includes the capability to deal with color terminals, which have a much richer set of attributes. For example, the MAJOR indicator could be displayed as white characters against a red background, while the MINOR indicator would be white against yellow. It is possible to define indicator states in the most general way for color terminals and then have the Equipment Configuration Database contain MTTYC or TTYC options that "downgrade" the color states for black-and-white terminals. In the example mentioned above, both the red and yellow backgrounds would be mapped into the reverse attribute.

***3.2.3.2 Control functions.*** In addition to displaying pages on the MCRT screen, DAP also can receive menu commands typed on the MCRT keyboard. These commands usually are referred to as "pokes" since they are similar to the action of poking a key on a key/lamp panel. As mentioned earlier, depression of the CMD/MSG key switches the terminal from the message mode to the command mode and vice-versa. When in the command mode, DAP displays a CMD: prompter towards the top of the screen, as shown in Fig. 3, and positions the cursor so that the typed characters appear in that line.

A command line consists of a number (usually 3 or 4 digits long) that optionally can be followed by some text characters. The craft interface handler, knowing that the terminal is in command mode, routes this input to DAP instead of to the SHELL. DAP examines the number to determine if it corresponds to a local or a global menu item. Local menu items are associated with the page(s) currently being displayed on the video screen. Global items are associated with any active page, even if the page is not currently displayed. In other words, a globally defined item will always be accepted and acted upon, even if its page is not being displayed.

If DAP successfully locates the menu item corresponding to the number that was typed, it usually sends an interprocess message to the owner of the page defining that menu item. This message contains the item number and the additional text characters typed, if any, as well as the originating terminal identification. The owner then takes whatever action is appropriate.

We used the word "usually" above because in some cases the response to a command is some simple action such as flipping to a new page on the display. In other cases, the PDF can specify a function to be executed by DAP upon receipt of the command, thereby bypassing the overhead of interprocess messages. One interesting aspect of this feature is the ability for DAP to translate a menu command into a text message to be passed to an instance of the SHELL, with the additional characters substituted in the message. This makes it possible for an application to design easy-to-use menus as an alternative to text message input, but to handle all terminal inputs internally as if they came through the SHELL.

A final note on commands has to do with locking acknowledgments. As with the SHELL, DAP requires a positive response to each command before another command can be accepted. For a command passed to a page owner via an interprocess message, the owner must send an acknowledgment message back to DAP within a certain time period or be abandoned. For commands handled via the function call approach, the function returns an acknowledgment code to DAP.

#### *3.2.4 Forms processing*

As described above, DAP is typically used for control/display functions related to maintenance activities such as configuration control. However, some applications require more general display functions than the indicator/menu approach appropriate for these maintenance scenarios. The craft subsystem provides facilities for entering textual information as part of displays.

A DAP page can be defined to contain input areas other than the standard command input line. When such a page is displayed, depression of the CMD/MSG key places the cursor at the first input area

instead of at the command line. The craftsperson can enter text into this area and/or move the cursor, using the terminal cursor control keys, to the next input area on the page. When the RETURN key is hit, DAP passes the typed information to the page owner via an interprocess message.

### 3.2.5 Alarm processing

The Alarm Control Process (ACP) is the part of the craft subsystem responsible for sounding audible alarms and displaying a summary of current system status at the top of the video screen. ACP is created during DMERT initialization and notifies DAP to activate the page known as the System Summary Area (SSA). It also attaches itself to the SCSD handler to gain access to the signal distributor points used to sound audible alarms. The plant measurements data base is automatically updated for severity-type alarm counts.

As the spooler and DAP receive messages from the higher-level processes, they check for situations requiring audible alarms and/or changes in the system status summary. For the spooler, alarm information is contained in the message prefix received from the higher-level process. For DAP, this information is derived from the indicator state data. Both cases result in messages being sent to ACP, which then operates the signal distributor points via the SCSD handler and/or sends DAP messages to change the states of indicators on the SSA page. Application processes also send messages directly to ACP for alarms.

Another message received by ACP from DAP is a notification that the ALM RLS key has been depressed. This causes ACP to reset the signal distributor point controlling the audible alarms. This key depression is also reported from the MTTYPC directly to the SCSD audible alarm retire scan point.

The Critical Indicator Area (CIA) process is closely related to ACP and DAP. Its function is to extract from the SSA page sixteen "critical indicators" of system status and periodically send them to the remote maintenance center via the X.25 link for alarming and display. The remote center can use this periodic "heartbeat" from the CIA process as one test that the system is operating sanely.

### 3.2.6 Common processor display page

Thus far we have described the general hardware and software modules that comprise the 3B20D's craft interface subsystem. DMERT also includes several processes that are, in effect, users of the craft subsystem. One of these is the process that owns the Common Processor Display (CPD) page that we have frequently used for examples (see Fig. 3). This Real-Time Status (RTS) process is created

as part of the DMERT initialization sequence and immediately sends messages to DAP to activate the CPD page. RTS also attaches itself to the Equipment Configuration Data Base (ECD), which it periodically examines to determine if any units shown on the CPD page have changed state. Spontaneous equipment configuration changes are reported to RTS through a library interface (CONFIG) from the various device handlers. In either case, appropriate messages are sent to DAP.

## IV. SUMMARY

The 3B20D/DMERT system has taken a significant departure from earlier switching processors in many areas, but perhaps none is so visible as the craft interface. The use of flexible video displays makes it possible to adapt the 3B20D to diverse applications quickly and economically. Also, the introduction of a reliable, secure, high-capacity data link for remote maintenance makes the 3B20D/DMERT system well suited for unattended operation, with resultant cost savings.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 1, January 1983
*Printed in U.S.A.*

## *The 3B20D Processor & DMERT Operating Systems:*

# Integration and System Test

By W. F. KLINKSIEK and H. L. MITCHELL

(Manuscript received March 18, 1982)

*This article describes the general approach that was taken in integrating and system testing the 3B20D Processor system. Since both the system hardware and software were developed simultaneously, the goals of the system test and integration plan naturally shifted emphasis and expanded their scope from achieving hardware stability to establishing software functionality and finally to demonstrating system stability. This article also overviews some of the project management techniques and procedures applied during the development of the 3B20D Processor.*

## I. INTRODUCTION

An important aspect of the development of any complex system such as the 3B20D Processor is the methodical integration and system testing during all phases of the development consistent with the experience gained from previous developments.[1–4] Since the hardware, software, and microcode were designed and developed simultaneously, the initial efforts focused primarily on the hardware and firmware using stand-alone exercise modules and system diagnostics run from a laboratory support processor. After the hardware reached sufficient stability, emphasis turned to functional testing of each major software subsystem and feature. Finally, as full functionality was achieved, the major thrust of testing focused on system integrity and reliability using the previously developed tests as a regression test package to assure no loss in functionality as problems were cleared. The development methodology is summarized in the relative timeline sequence chart shown in Fig. 1.

Also discussed in this article are some of the project management

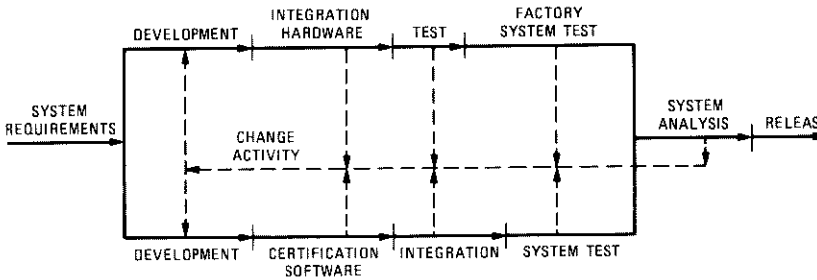



Fig. 1—Generalized development model.

techniques and administration tools used to control the changes and new features introduced into the system.

## II. EARLY HARDWARE/SOFTWARE INTEGRATION AND TEST STRATEGY

### *2.1 Objectives*

The objective of the initial integration and test effort on the prototype hardware was to verify basic instruction execution and memory access, establish full diagnostic capability of the hardware,[5] prove in peripheral access and functionality, and establish stable communication interfaces. In achieving these objectives, a stable software development environment was achieved for the major portion of the software development.

### *2.2 Stand-alone exercise modules*

The diagnostics were developed to initially run from the laboratory support processor in conjunction with the hardware development. This simultaneous development of the diagnostics and the processor hardware had the unique advantage of providing individual functional verification of each circuit pack or major unit before integration of the operational system was attempted, thereby saving much laboratory time ferreting out faulty hardware. The initial functional integration started with simplistic CPU test modules that afforded stand-alone verification of basic operation. Upon reaching acceptable functionality, stand-alone test modules were used to establish communication with the disk file controller and moving head disks.

#### *2.2.1 Central processing unit integration*

Two test modules were used extensively to integrate the early Central Processing Unit (CPU) hardware, firmware, and subsequent changes. The first test module was designed to test basic main-memory access and instruction execution with output to serial channel on the

Central Control Input/Output (CCIO) bus.[6] Loading this module from the laboratory support processor verified the communication link from the support processor to the 3B20D Processor. In addition, the execution of the module not only verified basic hardware functionality but also verified the data-link capability to a TTY via the serial channel. The second test module, in combination with a primitive version of the operating system, established two processes and cyclically sent messages between them. This capability not only tested more of the hardware features of the CPU, but also provided a means to verify stable operation over long periods of time. This test module was then expanded to verify memory update on the off-line Control Unit (CU) and "soft switch" capability between the duplex units.[7]

#### *2.2.2 File system integration*

Once basic operation of the CPU was verified, attention was pointed toward the file-system operation requiring integration of the Direct Memory Access (DMA) unit, the Disk File Controller (DFC) unit, and the Moving Head Disk (MHD).[8] Again a stand-alone test module, based on the disk driver software and the primitive operating system, was used for the integration of the hardware and firmware. Because of the large percentage of the hardware that had to be operational for successful execution of this test module, it became an invaluable tool not only for the integration of the preproduction hardware but also for Western Electric manufacturing, testing, and installation of early models of the 3B20D Processor in application system laboratories.

### *2.3 System software*

Once the hardware was integrated and verified to the limits of the stand-alone test modules, development of the operating system and system-initialization software proceeded rapidly, and the integration effort switched emphasis from strictly hardware to system software. The strategy was to incrementally integrate—from the primitive operating system—each new capability of the operating system and system-initialization software with the hardware until a fully cycling stand-alone basic processor system was achieved.

With the basic capability to initialize the system and cycle the operating system, integration proceeded to verify the 3B20D resident diagnostic control structure and diagnostics.

By this time, additional integration tests were necessary to more fully expand coverage of the system. Thus, a test process was developed that created disk read and write jobs with a variable number of these child processes specifiable up to the number of allowable Dispatch Control Table (DCT) entries. Because of the large percentage of the processor used by this test process and because of the controllable

activity, it became an invaluable regression-test vehicle for subsequent integration activities as well as a system stress test.

### 2.4 Results

The primary result of this early effort was the establishment of a stable hardware and operating software base for the development of the features.

## III. INTEGRATION AND SYSTEM TEST

The 3B20D system-level testing is actually divided into three distinct functional groups consisting of System Integration, System Test, and System Analysis. A brief historical review of the evolution of these groups is perhaps the best way to describe their respective functions. In early 1979 a decision was made to delegate the system testing of DMERT to Western Electric.[9] A Western Electric department was formed with the goal of taking over full responsibility for DMERT system testing by January 1, 1980. This transition actually took place about six months ahead of schedule in July 1979 and the system remains a Western Electric responsibility. The goals of the system testing group at that time were to release laboratory quality prereleases to DMERT applications to allow parallel application software development with the DMERT development. The system testing group also developed an extensive, documented set of tests that could be used not only to test the prereleases but would also serve as a base for testing all generic software releases in the years to follow.

Another group, the System Integration group, was responsible for planning and coordinating the building (compiling) of each DMERT release, getting the release installed and cycling in the 3B20D development labs, and assuring that basic functions worked. Once this was accomplished, responsibility for the detailed testing was turned over to the system testing group. Thus, the system testing group could concentrate more on actual testing and problem resolution and less on bringing up the internal loads.

### 3.1 Integration

System integration controls the flow of software changes from the time a developer completes a change through the release of that change to a customer. The specific areas of responsibility include:

(*i*) Load engineering and planning
(*ii*) Benchmark tracking and analysis
(*iii*) Integration testing
(*iv*) Release-letter generation
(*v*) Modification Request (MR) tracking and MR data base integrity.

### 3.1.1 Load engineering and load planning

For each DMERT release, an individual is assigned to be the load engineer. This individual serves as the focal point for all load-building activities. Specifically, the load engineer analyzes all changes planned for the release by the generic engineer, decides in what sequence changes should be taken, oversees the building of the load, and coordinates installation and integration testing of the load in the development labs.

Members of the integration team report to the load engineer who assures that all activities needed to deliver the load on schedule are assigned and completed. The load engineer with assistance from the integration team resolves daily problems and, as necessary, reschedules activities and people.

The load engineer in conjunction with the program administration staff coordinates the actual building of the load. The load engineer must thoroughly understand the mechanics of how the system is built, what software dependencies exist and how source code is controlled via the CMS/M2 system.[10]

### 3.1.2 Benchmark tracking and analysis

Each new generic feature or major software enhancement results in a set of benchmarks that identify the date at which major activities are scheduled for completion. Benchmarks serve a dual purpose: first, as a management tool for measuring how the project is doing relative to the plan; and second, as a planning aide for other people or groups identifying dependencies for other features, hardware availability, or lab installation.

Several tools have been used for identifying, tracking and reporting on feature benchmarks within the DMERT development organization.

### 3.1.3 Integration testing

One of the major objectives of the integration team is to assure that the load given to the system testing group is of sufficient quality to allow detailed functional testing. To verify that the system is of such quality, basic functional tests are run to assure that the major subsystems are operational. These include diagnostics, processor duplex operation, disk and I/O capabilities, and Recent Change and Verify.

### 3.1.4 Release-letter generation

Typically the applications that use the 3B20D Processor want the new DMERT software releases as soon as possible after the completion of system testing. This has presented a unique challenge to DMERT development management: the need to get releases, complete with essential documentation, to a number of different customers within one day of the completion of system test.

One vehicle used to supply necessary timely documentation to the customers is the release letter. This letter has evolved into a rather detailed document covering:

(*i*) Support processor installation procedures
(*ii*) 3B20D installation procedures
(*iii*) List of all file names
(*iv*) List of all changed files
(*v*) List of all required data base changes
(*vi*) MR descriptions for all MRs resolved in the release
(*vii*) MR exceptions list.

Of particular importance is the MR exceptions list. The intent of this list is to communicate to the customers known problems that exist in the release and, when available, action to be taken if it is observed on their machines. This communication mechanism saves many hours that applications personnel would spend analyzing problems already identified by the DMERT organization.

To assure timely distribution of this letter, all sections are put on a support computer and support programs are executed to assemble them into a document that is available on the day of the software release.

#### *3.1.5 Detailed MR tracking and data base integrity*

The integration team also was chartered to establish the integrity of MR data base, to produce accurate and timely reports, and to respond promptly to high-priority problems. Weekly audits of the entire data base are performed to assure that MRs do not remain in a transient state for an unreasonable length of time.

### *3.2 System test*

The primary objective of the 3B20D System Test group is to test the DMERT system on the 3B20D Processor in order to validate that all advertised features and capabilities perform according to their documented requirements. System tests are designed to test all the functional capabilities of the processor and its hardware both in no-load and stress environments.

In the two and one half years since its inception, the System Test group has developed a complete system testing package containing over 700 test cases. As new features are developed, test cases are developed and each feature is thoroughly tested. Test cases are documented and in many cases processes are written to automatically execute the tests. Once a feature is released for customer use, a subset

of the defined test cases is included as part of an on-going regression-test package.

A concept of certification testing was established to identify problems early in the development cycle. This allowed more problems to be debugged and fixed before release and resulted in a more stable system testing environment and higher ultimate product quality.

Certification testing requires the developers to build in the official environment[10] and to demonstrate the proper execution of their new code to a system tester before it can be delivered to the integration team. The system tester has an option to request particular tests to be run with the new code and thus certify that the software to be submitted has passed some basic tests and can be approved for further processing. Software not passing certification is rejected and the developers have to correct the deficiencies and schedule a follow-up certification test.

### *3.3 System analysis effort*

The System Analysis Group (SAG) effort was planned as an extension to Integration and System Testing. As its objectives, SAG was to perform tests aimed at measuring the performance and reliability of the 3B20D as a system. A separate development laboratory was constructed with the primary intention of simulating and functioning as a field site. Since this was the only 3B20D laboratory planned to run for long periods of time without rebooting, many problems of a periodic or long-term nature were first observed there.

SAG members approached the stability aspect of the job by first defining measurable metrics. Objectives were defined based on the measured system reliability. The SAG team then identified and investigated problems that impacted system reliability and reported the effects on system stability once the problems were resolved.

Stability data was collected during weekend testing. The tests involved running a controlled-load package containing system exercise processes for specified periods of time, usually several days. These tests were generally run unattended to evaluate hands-off machine performance. All messages to the Read Only Printer (ROP) were stored on disk, dumped at the end of the test and analyzed using a program developed for this purpose.

Three sets of objectives were defined for data analysis: a long-term objective for system reliability; a cut objective that identified satisfactory stability levels for first application at in-service offices; and the objective of establishing concern thresholds. Any data above the concern threshold was clearly unacceptable for even initial in-service machines. Data lying in the area between the cut objective and the

concern thresholds needed additional understanding in order to make a go/no-go decision on cutover.

An example of one of the metrics used to track stability is shown in Fig. 2 for ten releases of DMERT prior to the first machine cutover in September 1981.

## IV. FACTORY SYSTEM TEST

Factory System Tests (FST) and Quality Assurance (QA) tests are the final hardware tests run at the Western Electric Company manufacturing plants to assure that a quality hardware product is delivered to the customer.

### *4.1 Objectives*

The objectives of FST and QA are to test the hardware functionality and interconnections of fully assembled systems to assure that the processors as built meet design intent. These extensive tests assure the highest possible quality in the product when shipped to the customer.

### *4.2 FST test strategy*

Instead of developing special test software for the FST, the actual DMERT operating system is enhanced with additional exercise processes to form the Factory/Installation Software Test (FIST) package. The testing is divided into two phases: the normal operation phase and the stressed operation phase. These tests apply to all hardware delivered by the factory including the system as ordered, the comple-

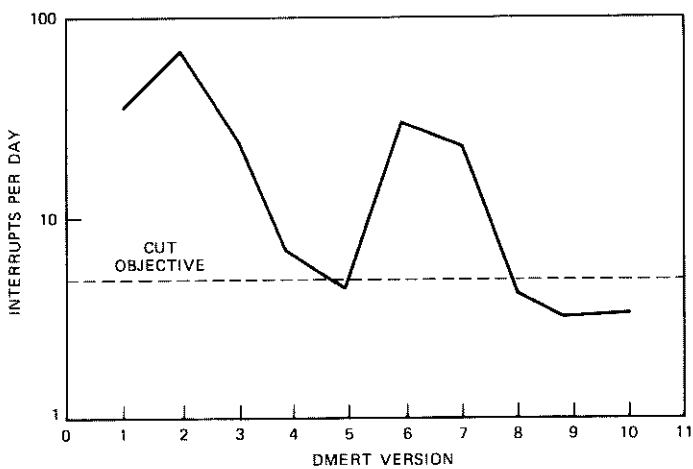



Fig. 2—Interrupt incidence history.

ment of spare circuit packs, growth units and circuit packs, and repaired product.

#### *4.2.1 Normal operational tests*

The normal operational tests are designed to verify the functionality, interconnections, and basic maintenance operations associated with each unit under normal operating conditions. Included in these tests are the activation of system initializations under all possible minimum configurations using the power switch and the craft interface terminal. The tests then assure functionality of all units under simulated maintenance conditions by removing and restoring each unit using both the power switch and the craft-interface terminal. During this test the system must remain operational. The next phase of testing requires the running and passing of all diagnostics for each unit within the system. Finally a series of special exercise processes are used to simulate actual operation of the disks, tape units, TTY and other data link controllers, and a CU soft-switch process for duplex capability verification.

#### *4.2.2 Stressed environmental operational tests*

The 3B20D Processor is designed to operate under a wide range of temperature and battery conditions. To assure that the system meets the design intent to operate under these conditions, two additional test environments are imposed on the machine before shipment.

***4.2.2.1 Low voltage.*** The power converters are stressed most under conditions of low-input voltage; thus, the system must pass all the tests prescribed above at an input voltage of $-43.75 \pm 0.05$ volts. This voltage is 91 percent of the nominal $-48$ volts.

***4.2.2.2 High temperature.*** High-temperature operation of the 3B20D Processor is critical to avoid outages during commercial power or mechanical failures that result in the loss of building air-conditioning systems. The system tests prescribed above must pass in a system that has been operating at a stable elevated temperature of $49°C \pm 1°C$ for a period of at least four hours.

### *4.3 QA testing*

In addition to the factory system test on all systems, additional tests are rerun under the auspices of the Bell Laboratories quality assurance organization and the Western Electric quality review organization to assure that statistical quality control limits are not exceeded, thus maintaining a high level of quality for the customers.

### 4.4 Result

A major milestone was achieved in March 1980, when the first field shipment to the Traffic Service Position System (TSPS) site in San Antonio, Texas, was not only delivered on schedule, but passed the complete battery of factory system tests.

## V. ADMINISTRATION

In this section, a brief overview of some of the important aspects of project-management and project-control techniques are presented.

### 5.1 Change authorization

From the beginning of the project, the hardware design was under very tight controls. All changes or feature additions had to be approved by a management-change committee with representation from Bell Laboratories and Western Electric. This committee provided both a forum to review designs and design changes and to discern the economic impact of each change. This committee then established a joint subcommittee, called the Engineering Support Group, to schedule and track each change from design through manufacture and ultimately to the installation into the various system development laboratories.

Software change control was less tightly controlled during the initial development and relied heavily on the software development supervisors responsible for each subsystem. Once the software was delivered to the application more stringent controls were introduced. At that point, feature content, overall coordination, and generic scheduling are the responsibility of the Generic Engineer and the Project Manager.

### 5.2 Application interfaces

To assure that the 3B20D Processor system meets the needs of the variety of Bell System applications, a group was established to act as the single focal point for the applications for all feature requests and MRs.

#### 5.2.1 Feature content

To establish feature content of the system, the Application Interface group, in concert with the applications, developed a prioritized list of feature requests and enhancements for the Project Manager and the Generic Engineer to review. Thus, a final list of features and enhancements was established taking into account customer needs, schedules, and resource limitations.

#### 5.2.2 Modification requests

Initially the Application Interface group also acted as a clearinghouse to prioritize, from the users point of view, the problems that

they discovered as the generic matured. This list, in conjunction with internally generated MRs, formed the basis for the Generic Engineer to approve MRs to be fixed for inclusion in the generic. Once an MR was approved, the Load Engineer tracked its progress through development, integration, system test, and release.

Once the first generic was cut into service, a committee was established with representation from applications, DMERT development, generic engineering, system test, load engineering, and field support. This committee's function was to tightly control and adjudicate all software changes so as to assure that field service was not adversely affected and that real service problems were quickly attended to and delivered on a timely basis.

### *5.3 Project-tracking tools*

A finite-state MR control mechanism was put into place to track and record changes in the status of MRs during the development cycle.[10] From this data base, various reports were automatically generated for use by all organizations associated with the project. This central source of project-status information was an essential ingredient to the determination of areas of concern so that action could be taken, as well as a repository of all schedule information relating to MRs. This capability formed the nucleus of the automated project-management tools.

## VI. CONCLUSION

The 3B20D Processor is operating effectively in the field since the first cutover in September 1981. The rapid field buildup during the first six months (24 machines cut into service) could not have been possible if all parts of the system were not of the highest quality and designed for high reliability. Much of the success of the project is attributed to the extensive testing both by the DMERT development organization, Western Electric organizations and application organizations during each step of the system's introduction.

## VII. ACKNOWLEDGMENTS

The authors are reporting on the work of many processor system and application personnel in Bell Laboratories and Western Electric. We wish to acknowledge all of their efforts and their dedication, which made the project so successful. In particular, the authors acknowledge the following individuals for assisting in the preparation of this text: R. J. Colby, J. M. Field, K. A. Giesting, D. I. Sandel, P. J. Stankus, R. R. Stozek, and D. S. Trushin.

## REFERENCES


1. H. A. Hilsinger, K. D. Mozingo, C. F. Starnes, and G. A. Van Dine, "1A Processor: Testing and Integration," B.S.T.J., *56* (February 1977), p. 289.
2. C. Haugk, S. H. Tsiang, and L. Zimmerman, "System Testing of the No. 1 Electronic Switching System," B.S.T.J., *43* (September 1964), p. 2575.
3. D. R. Barney, P. K. Giloth, and H. G. Kienzle, "No. 1 ESS ADF: System Testing and Early Field Experience," B.S.T.J., *49* (December 1970), p. 2975.
4. B. P. Donohue, III and J. F. McDonald, "SAFEGUARD Data-Processing System: Processor-System Testing and the System Exerciser," B.S.T.J. (1975), SAFEGUARD Supplement, p. S111.
5. J. L. Quinn, R. L. Engram and F. M. Goetz, "The 3B20D Processor & DMERT Operating System: Diagnostic Tests and Control Software," B.S.T.J., this issue.
6. M. W. Rolund, J. T. Beckett, and D. A. Harms, "The 3B20D Processor & DMERT Operating System: 3B20D Central Processing Unit," B.S.T.J., this issue.
7. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "The 3B20D Processor & DMERT Operating System: Fault Detection and Recovery," B.S.T.J., this issue.
8. R. E. Haglund and L. D. Peterson, "The 3B20D Processor & DMERT Operating System: 3B20D File Memory Systems," B.S.T.J., this issue.
9. M. E. Grzelakowski, J. H. Campbell, and M. R. Dubman, "The 3B20D Processor & DMERT Operating System: DMERT Operating System," B.S.T.J., issue.
10. B. R. Rowland and R. J. Welsch, "The 3B20D Processor & DMERT Operating System: Software Development System," B.S.T.J., this issue.

## ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ACHI | application channel interface |
| ACP | alarm control process |
| AJT | active job table |
| ALU | arithmetic logic unit |
| ALW | allow |
| API | attached processor interface |
| APS | Attached Processor System |
| ATB | address translation buffer |
| ATP | all tests passed |
| BGB | bidirectional gating bus |
| BIC | bus interface controller |
| BLDG PWR | building power |
| BPI | bits per inch |
| BS | bus simulator |
| BWM | broadcast warning message |
| CAD | computer-aided design |
| CC | central control |
| CCIO | central control input/output |
| CCIS | common channel interoffice signaling |
| CCITT | International Telegraph and Telephone Consultative Committee |
| CH | channel |
| CHAN | channel |
| CIA | critical indicator area |
| CIH | craft interface handler |
| CMS | Change Management System |
| CONFIG | configuration management program |
| CPD | common processor display |
| CPH | communication protocol handler |
| CPU | central processing unit |
| CRT | cathode ray tube |
| CSU | cache store unit |
| CU | control unit |
| DAP | display administration process |
| DATA TABLE | diagnostic data table files |
| DBEVOL | Data Base Evolution System |
| DBS | duplex bus selector |
| DCB | diagnostic control block |
| DCT | dispatch control table |
| DDCMP | digital data communication message protocol |
| DDL | data definition language |

| | |
|---|---|
| DDLP | data definition language processor |
| DDSBS | duplex dual-serial bus selector |
| DEV | device |
| DFC | disk file controller |
| DFDIAG | disk file diagnose |
| DFI | disk file inverter |
| DGN | diagnose |
| DIAGC | diagnostic control |
| DIAMON | diagnostic monitor |
| DIO | DMA I/O bus |
| DIP | dual in-line package |
| DMA | direct memory access |
| DMAC | direct memory access controller |
| DMERT | Duplex Multiple Environment Real Time |
| DML | data manipulation language |
| DMU | data manipulation unit |
| DPT | diagnostic phase table |
| DRAM | dynamic random access memory |
| DSCH | dual serial channel |
| DST | destination |
| DUC | dual-access utility circuit |
| DUI | direct user interface |
| EAI | emergency action interface |
| ECC | error correction code |
| ECD | equipment configuration data or data base |
| ECDMAN | equipment configuration database manager |
| EIH | error interrupt handler |
| EOS | extended operating system |
| EPROM | erasable programmable read-only memory |
| ER | error register |
| ESS | Electronic Switching System |
| EX | exercise |
| FIFO | first in-first out |
| FIST | factory/installation software test |
| FPS | form processing system |
| FST | factory system tests |
| FTAM | forms translation and mapping |
| FTS | field test set |
| GRASP | generic access package |
| IB | instruction buffer |
| INH | inhibit |
| IODRV | IOP driver process |
| IOP | input/output processor |
| IOPC | input/output processor controller |

| | |
|---|---|
| IPC | interprocess communication |
| IPS | inches per second |
| KPCB | kernel process control block |
| LC | line controller |
| MSGS | messages |
| MSI | medium-scale integration |
| MTC | maintenance |
| MTTR | mean time to repair |
| MTTY | maintenance TTY |
| MTTYC | maintenance terminal controller |
| MU | mask unit |
| MUX | multiplexor |
| NCP | network control point |
| NK | non-killable |
| NRZ | non-return to zero |
| ODIN | on-line data integrity |
| OOS | out of service |
| OP | display |
| OST | operating system trap |
| PA | program address |
| PC | peripheral controller |
| PCB | process control block |
| PCSD | peripheral controller subdevice |
| PCU | power control unit |
| PD | peripheral device |
| PDF | page description file |
| PDFGEN | PDF generator |
| PDS | program documentation standard |
| PE | phase encoded |
| PFC | peripheral frame control |
| PIC | peripheral interface controller |
| PID | process identifier |
| PINIT | processor initialization program |
| PRM | processor recovery message |
| PROM | programmable read-only memory |
| PSBR | primary segment base register |
| PSDC | parallel serial data interface |
| PSI | peripheral system interface |
| PSW | program status word |
| QA | quality assurance |
| RAM | random access memory |
| RC/V | recent change/verify |
| RFI | radio frequency interference |
| RMU | rotate mask unit |

| | |
|---|---|
| RMV | remove |
| ROP | read-only printer |
| ROP | receive-only printer |
| RST | restore |
| RSTU | restore unconditional |
| RTS | real-time status |
| RU | rotate unit |
| SAC | store address control/controller |
| SAG | system analysis group |
| SAR | store address register |
| SAT | store address translator |
| SBR | segment base register |
| SC/SD | scanner/signal distributor |
| SC | software control |
| SCANS | Software Change and Notification System |
| SCC | switching control center |
| SCCS | Source Code Control System |
| SCCS | Switching Control Center System |
| SCH | serial channel |
| SCM | store complete signal |
| SCR | silicon controlled rectifier |
| SCR | store control register |
| SCSD | scanner and signal distributor |
| SD | software development |
| SDC | store data controller |
| SDP | software demand paging |
| SDR | store data register |
| SDS | Software Development System |
| SG | system generation data base |
| SGO | store go signal |
| SGS | Software Generating System |
| SID | segment identifier |
| SIM | system integrity monitor |
| SIR | store instruction register |
| SMD | storage module drive |
| SP | software production |
| SRC | source |
| SREG | special registers |
| SSA | system summary area |
| SSBR | secondary segment base register |
| SSI | small-scale integration |
| SSR | system status register |
| STOP | terminate |
| SUPR | system update program |

| | |
|---|---|
| SYSGEN | system generator/generation |
| TB/IS | transaction block and integrity subsystem |
| TLDB | trouble locating data base |
| TLP | trouble locating process |
| TSPS | Traffic Service Position System |
| TTL | transistor-transistor logic |
| TTY | teletypewriter or terminal |
| TTYC | terminal controller |
| TUS | Test Utility System |
| TV | transfer vector |
| UC | utility circuit |
| UID | utility identifier |
| USP | *UNIX* supervisor process |
| VLSI | very large-scale integration |
| VOH | voltage output high |
| VOL | voltage output low |
| VTOC | volume table of contents |
| WCS | writeable control store |
| WMS | writeable microstore |
| YACC | yet another compiler compiler |

## CONTRIBUTORS TO THIS ISSUE

**Roy E. Anderson,** B.S.E.E., 1970, University of Illinois; M.S.E.E., 1972, University of Maryland; Bell Laboratories, 1970—. Mr. Anderson initially worked in the field of computer-controlled electrical measurement equipment for transmission facilities. His interest in computers led him into the area of designing real-time operating systems. Today, he is involved with developing a distributed digital electronic switch.

**Marshall E. Barton,** B.A., 1962, M.S. (Mathematics), 1964, Miami University; Bell Laboratories, 1964—. Mr. Barton initially was responsible for the development of support software for No. 2 ESS and No. 3 ESS. Since 1979 he has been associated with 3B20D DMERT, most recently as Supervisor of the Craft Interface Design Group.

**J. T. Beckett,** B.S. (Engineering), 1961, Harvey Mudd College; M.S.E.E., 1964, Ph.D. (Electrical Engineering), 1967, Case Western Reserve University; Bell Laboratories, 1967—. Part time lecturer, Electrical Engineering, Illinois Inst. of Tech., 1968–1975. Responsible for microcode on 3A processor and for the development of the 3B20D microcode and microcode tools. He also has been responsible for the development of software debugging tools. He is currently Supervisor of the No. 5 ESS Project Management Tool Development Group. Member, ACM, IEEE.

**Harry. L. Bosco,** B.S.E.E., 1972, Monmouth College, M.S.E.E., 1974, Polytechnic Institute of Brooklyn; Bell Laboratories, 1965—. Mr. Bosco has worked in data communications and hardware design for No. 4 ESS. He was promoted to Supervisor of the Network Design Group for No. 5 ESS in 1977 and to Head of the No. 5 ESS Line Interface and Peripheral Circuits Department in 1980. In 1981, he became Head of the No. 5 ESS Product Management Department, responsible for the generic planning, system architecture, and management of the No. 5 ESS product line. Member, Sigma Pi Sigma, Eta Kappa Nu.

**John M. Brown,** B.S.M.E., 1967, M.S. (M.E. & I.E.), 1969, University of Michigan; Bell Laboratories, 1969—. Mr. Brown started his career with Bell Laboratories by working on the development of 1A Technology apparatus. He has worked on the development of 1A Processor semiconductor stores and supervised the physical design of the 3B20D Processor. In 1980, he was appointed Head of a department responsible for the circuit design of the 3B20D Processor Control Frame and for physical design of the 3B Processor family.

**A. H. Budlong,** B.E.E., M.S. (Physics), Marquette University, 1950; Member of the teaching staff of the Department of Physics, Marquette University, 1948–1952; Bell Laboratories, 1953—. Since joining Bell Laboratories, Mr. Budlong has been engaged in exploratory development of electronic switching circuits and has conducted a group in charge of switching training at Bell Laboratories. He was involved in the development of the No. 1 electronic switching system in the areas of trunk and service circuits, and automatic message recording equipment, and in the design of high-speed communication buses and power facilities for the 1A Processor. He has been in charge of a group developing magnetic tape file systems, printers, and miscellaneous peripheral circuits for the 3B20D Processors. Mr. Budlong is the coauthor of a book entitled *Electronic Switching Theory and Circuits.* He is the Dean of the Undergraduate School and also a Professor of Electronic Engineering at the Midwest College of Engineering. Member, Sigma Pi Sigma and Pi Mu Epsilon.

**J. H. Campbell,** B.S. (Mathematics), 1964, Pittsburg State University; M.S., 1969, Ph.D, 1974 (Computer Science), Iowa State University; Bell Laboratories, 1974—. Since joining Bell Laboratories, Mr. Campbell has worked on the Extended Operating System (EOS) project and on the 3B20D/DMERT project doing operating systems development. Currently, he supervises a group engaged in the development and support of I/O drivers for real-time applications. Member, ACM, Sigma Xi.

**J. G. Chevalier,** B.E.E., 1951, Ohio State University; Bell Laboratories, 1956—. Mr. Chevalier has worked on a printed wiring board processes, applications for both military and telephone projects, connector design, studies of contact finishes, and the physical design and packaging of electronic equipment for switching systems. He presently supervises a group responsible for the development of packaging technologies for future versions of the 3B20D Processor.

**Noel X. DeLessio,** B.S.E.E., 1960, M.S.E.E., 1961, Ph.D. (Electrical Engineering), 1966, Polytechnic Institute of Brooklyn; Bell Laboratories, 1966—. Mr. DeLessio worked on SAFEGUARD system design and supervised guidance design for the SPRINT missile system. Subsequently, he supervised the Exploratory Development Group of the Operator System Laboratory and is currently Head of that Laboratory's Processor Applications Department.

**M. R. Dubman,** A.B., 1957, Princeton University; M.S., 1959, Massachusetts Institute of Technology; Ph.D., 1970, University of California-Los Angeles; Bell Laboratories, 1970—. From 1959 to 1970, Mr. Dubman was engaged in the development of statistical analysis techniques used in testing of Saturn/Apollo rocket engines for Rockwell International. At Bell Laboratories he has been involved in the development of software for the 1A and 3B20D Processors. Presently, he is Supervisor of a group responsible for the 3B20D system laboratories.

**Gary P. Eldredge,** B.S.E.E., 1971, M.S.E.E., 1971, Brigham Young University; Bell Laboratories, 1972—. Mr. Eldredge has designed a number of programs for the 1A Processor to assist debugging and updating software in operational field sites. Since 1977, he has been involved in the development, administration, and system testing of DMERT software. He recently was responsible for a group that designs utilities that are used in field sites to test, monitor, and maintain various parts of the software. He currently is Supervisor of the 3B20D fault recovery group. Member, Tau Beta Pi, Phi Kappa Phi, Sigma Xi.

**Robert L. Engram,** B.S.E.E., 1969, Howard University; M.S.E.E., 1973, Stanford University; Motorola, Portable Products Division, 1969–72; Bell Laboratories, 1972—. Prior to joining Bell Laboratories Mr. Engram was involved in design and development of analog and digital circuits for paging and terminal equipment. He received a patent for his work inhibiting shock falsing in a two-tone sequential decoder. Upon joining Bell Laboratories, Mr. Engram was initially involved in exploratory development of digital circuits and systems including AP3 and VSS. In 1977, he became Supervisor of Test Facilities for TSPS and later supervised a group performing Software Integration and Testing. He joined the 3B20D/DMERT Project in 1980 as Head of the Processor System Integrity Department and presently heads the Operating System Development Department. Member, IEEE, Tau Beta Pi, National Technical Association.

**Rudolph J. Frank,** B.S.E.E., 1966, Seattle University, M.S.E.E., 1968, Ph.D. (Electrical Engineering), 1971, Oregon State University; M.S. (Business Management), 1981, Stanford University; Pacific Northwest Bell, 1964–1966; Bell Laboratories, 1971—. At Pacific Northwest Bell, Mr. Frank was an electronics data processing supervisor in the comptroller's division. At Bell Laboratories he has worked on new feature planning and exploratory development in the Traffic Service Position System laboratory. In 1975, he was designated Bell

Laboratories Visiting Professor of Electrical Engineering at Southern University (Baton Rouge, La). Mr. Frank became Supervisor of the No. 4 ESS Network Management Control Group in 1976 and has managed several large software development projects. In 1980 he was awarded a Sloan fellowship in the School of Business at Stanford University. Mr. Frank is now Head of the Toll Digital Maintenance Planning and Development Department at Bell Laboratories, Member, IEEE, Eta Kappu Nu.

**Alan W. Fulton,** B.S.E.E., 1966, University of Arizona; M.S.E.E., 1967, Ph.D. (Electrical Engineering), 1971, Stanford University; Bell Laboratories, 1966—. Mr. Fulton has been involved in the design of processors for electronic switching systems. He currently is Head of the Processor Technology Department. This department is responsible for developing and integrating the silicon, packaging, and computer aided design tools required for processors. Member, Tau Beta Pi, Sigma Xi.

**Lee E. Gallaher**, B.S.E.E., 1951, M.S.E.E., 1956, Case Western Reserve University; Instructor in Electrical Engineering at Case Western Reserve University, 1952–1955; Bell Laboratories, 1955—. Mr. Gallaher has worked on memory systems for ESS including the Flying Spot Store, the Twister Memory and integrated circuit memories. He was promoted to Department Head in 1965 and was responsible for the design of the switch units for the No. 101 ESS and circuit designs for the No. 2 ESS and the No. 3 ESS. More recently he has been responsible for the architecture and development of the 3B20D processor. Member, IEEE, Tau Beta Pi, Sigma Xi, Eta Kappa Nu.

**Frank M. Goetz,** B.E.E., 1953, Manhattan College; M.S. (Mathematics), 1960, New York University; Bell Laboratories, 1953—. Mr. Goetz has worked on logic design, software development, and system design for electromechanical and electronic switching systems. He received five patents in the areas of electronic counters, error-correction circuitry, signal detection, and processor system design. Since 1962, he has been a Supervisor responsible for various aspects of electronic switching system and processor maintenance. At present, he is responsible for maintenance design of the 3B20S Processor. Member, IEEE, Eta Kappa Nu.

**Maureen Grzelakowski,** B.S., 1976, M.S., 1978 (Computer Science), Northwestern University; Bell Laboratories, 1977—. Ms. Grzelakowski was initially responsible for designing software development

environments for ESS projects. In 1980 she participated in the development of the DMERT nucleus. Since then she has supervised the Operating System Architecture and Planning, and Generic 2 Engineering Groups.

**Ronald E. Haglund,** B.S.E.E., 1960, M.S.E.E., 1963, and Ph.D. (Electrical Engineering), 1969, Iowa State University; Bell Laboratories, 1970—. Mr. Haglund initially worked on file store design for the No. 1A Processor. In 1979 Mr. Haglund became Supervisor of the 3B20D File Store Group with responsibility for the file stores for the 3B20D and other processors. He currently is Supervisor of the Peripheral Technology Group with responsibility for all 3B peripheral devices.

**Robert C. Hansen,** B.S.E.E., 1966, and M.S.E.E., 1969, Michigan State University; Ph.D. (Control Science), 1973, University of Minnesota; Bell Laboratories, 1973—. Mr. Hansen has worked on maintenance software for No. 4 ESS and the 3B20D. He is presently Supervisor of the Requirements and Architecture Group, concerned with feature evolution of the 3B20D. Member, IEEE, Tau Beta Pi.

**D. A. Harms,** B.S.E.E., 1963, M.S.E.E., 1965, University of Minnesota; Bell Laboratories, 1965—. Mr. Harms has been involved with the design of processors since 1967. Since 1970, he has supervised groups responsible for the design of memory subsystems, microstores, and peripherals for the No. 2 ESS, No. 2A ESS, No. 3 ESS, 3A Processor, and the 3B20D Processor. He is currently Supervisor of the Common Processor Circuits group responsible for the design of a diagnostic processor for 3B20S and for circuits incorporating the *Bellmac*™-32A microprocessor in single board computers. Member, Eta Kappa Nu, Phi Theta Kappa.

**Irvine K. Hetherington,** B.S.E.E., 1966, Lehigh University; M.S.E.E., 1967, Stanford University; Bell Laboratories, 1972—. Mr. Hetherington initially was associated with exploratory development of integrated circuit analog switching networks for telecommunications applications. From 1972 to 1976, he was involved with the design of the microcontrol store and main memory system of the 3A Processor. Since 1977, Mr. Hetherington has had several design and supervisory responsibilities associated with the CPU and memory systems of the 3B20D Processor. He currently supervises the 3B20D Processor Data Communications Peripherals Group. Member, Tau Beta Pi, Eta Kappa Nu.

**J. Richard Kane,** B.S.E.E., 1968, University of Pittsburgh; M.S.E.E., 1970, Northwestern University; Ph.D. (Computer Science), 1973, Northwestern University; Bell Laboratories, 1968—. Mr. Kane initially worked in the areas of fault detection and system recovery for No. 4 ESS and later on testing tools. He system tested the first No. 4 ESS. Mr. Kane then supervised groups responsible for the development of the DMERT operating system, testing tools for the 3B20D Processor and developing local area networks. He currently supervises a group planning the design and development of new processors.

**W. F. Klinksiek,** B.S.M.E., 1966, M.S.M.E., 1967, and Ph.D. (Mechanical Engineering), 1971, Virginia Polytechnic Institute and State University; Bell Laboratories, 1971—. Mr. Klinksiek has worked on the physical design and packaging of the 1A Processor and the 3B20D Processor specializing in thermal and power engineering. He has supervised the development of packaging technology and computer automated design systems and managed software administration for the 3B20D. Recently, he was a Supervisor responsible for generic engineering, application interface, and application engineering for the 3B20D Processor System. Currently, he is Head of Processor Systems Customer Support Department. Member, Phi Kappa Phi, Tau Beta Pi, ASME.

**S. H. Kulpa,** B.S.M.E. 1972, Michigan Technological University; M.S.M.E., Stanford University; Bell Laboratories, 1973—. Mr. Kulpa has been involved with the design and development of the packaging technology for the 3B20D Processor. He currently is Supervisor of a physical design group.

**Peter Kusulas,** B.S.E.E., 1963, M.S.E.E., 1965, Rutgers University; Bell Laboratories 1963—. Mr. Kusulas has worked on the development of magnetic and semiconductor memory systems for the 101 ESS, No. 2 ESS, and the 3A and 3B20D Processors. Since 1980 he has supervised several groups engaged in the development of the *Bellmac*™ microprocessor module and systems based upon it.

**N. A. Martellotto,** B.E.E., and B.S., Applied Mathematics, 1957, Georgia Institute of Technology; M.E.E., 1959, New York University; M.B.A., 1970, University of Chicago; Bell Laboratories, 1957—. Starting with the Bell System Data Processor project in 1957, where he did logic design and programming, Mr. Martelloto has been involved with computers and software development throughout his Bell Laboratories

career. Early projects include EPBX and No. 1 ESS. Mr. Martellotto holds a patent related to the basic notion of ESS generic programs. In 1966 he became Department Head of the Indian Hill Computation Center (IHCC). In 1976, he resumed design and development of ESS software development support programs and other related work. In late 1979, Mr. Martelloto became DMERT project manager and for the next two years was involved with all aspects of the project, from operating system development to field support. He is now Head of the Software Development Systems Department. Member, IEEE, Tau Beta Pi, Eta Kappa Nu.

**Peter S. McCabe,** B.S. (Engineering), 1956, Trinity College; B.S.E.E., 1957, Rensselaer Polytechnic Institute; Bell Laboratories, 1957—. Mr. McCabe has worked on memories for Nike-Zeus, digital circuits for the 101 EPBX, software design for the UNICOM, AUTOVON, four-wire No. 1 ESS, and No. 4 ESS projects. Mr. McCabe supervised support software system development and program administration development and operation for No. 4 ESS. Since 1980 Mr. McCabe has supervised the performance measurements group for the DMERT project. Member, Tau Beta Pi, Eta Kappa Nu, Sigma Pi Sigma; associate member, Sigma Xi.

**JoAnne H. Miller,** B.S. (Mathematics), 1967, University of Michigan; M.S. (Computer Science), 1976, University of Colorado; GTE-Sylvania, 1968–1970; University of Colorado-Institute of Behavioral Science, 1971–1976; Bell Laboratories, 1976—. Prior to joining Bell Laboratories Ms. Miller was involved in system analysis and real-time programming for missile control and interactive computer graphic applications. Upon joining Bell Laboratories Ms. Miller was initially involved in No. 1 ESS software restructure design. Since early 1979 she has supervised groups responsible for the system testing and delivery of microprocessor software development systems, and design and development of several components of 3B20D/DMERT System, including the Recent Change/Verify System and equipment configuration data base. Currently, she supervises a group responsible for the field deployment and support of the 3B20D/DMERT system. Member, IEEE, ACM.

**H. L. Mitchell,** B.S.E.E., 1970, Rensselaer Polytechnic Institute; M.B.A., 1981, Illinois Benedictine College; Western Electric, 1970—. Mr. Mitchell's original assignment was to work with Bell Laboratories on system testing of the Safeguard MSR system. He has worked on

No. 4 ESS software development and testing and on 3B20D/DMERT system testing, system integration, and 3B20D Field Support. He is currently Department Chief, Call Processing and Residential Feature Development—1/1A ESS.

**Robert W. Mitze,** B.S. (Mathematics), 1969, California Institute of Technology; M.S. (Mathematics), M.S. (Computer Science), 1976, University of Wisconsin; Bell Laboratories, 1976—. Mr. Mitze has worked in the development of software engineering environments for the C language, project management of the DMERT operating system development, and distributed architecture specification for special-purpose applications. Since March 1981, he has been Head of the Advanced Operating System Development Department.

**B. G. Niedfeldt,** B.S.E.E., 1959, University of Maryland; M.S.E.E., 1961, New York University; Bell Laboratories, 1959–1962; Bellcomm, Inc., 1962–1970; Bell Laboratories, 1970—. Mr. Niedfeldt was engaged in exploratory development of high-speed data terminals. While at Bellcomm, he participated in the evaluation of the guidance and navigation aspects of the Apollo lunar landing missions. He then joined the Safeguard project, working in the multiprocessor operating system area. In 1974, he joined the No. 4 ESS team and, among other things, was responsible for the first generic (4E0) system test and release. In 1977, he joined the 3B20D/DMERT software development team, and is presently Supervisor of the Operating System, Software Integrity and Data Base Systems Group. Member, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, and Omicion Deta Kappa.

**Leland D. Peterson,** B.S.M.E., 1971, Illinois Institute of Technology; M.S.M.E., 1974, Northwestern University; Bell Laboratories, 1966—. Mr. Peterson has been involved with physical design of the 1A Processor, Voice Storage System (VSS), 1A Attached Processor System, and the 3B20D family of processors. He has participated in exploratory physical design and in the development of new hardware packaging technology. Currently he is Supervisor of the Large Processor Physical Design Group.

**Ralph W. Peterson,** B.S. (Physics), 1961, Wayne State University; M.S.E.E., 1963, New York University; Bell Laboratories, 1961—. Mr. Peterson has worked on software development on several electronic switching systems, including extensive work on software development tools. He presently is Supervisor of the Processor Microcode Group,

responsible for the development of microcode for 3B Processors. Member, ACM, Eta Kappa Nu, Tau Beta Pi.

**John L. Quinn,** B.S.E., 1959, Stevens Institute of Technology; M.S.E.E., 1961, New York University; Bell Laboratories, 1959—. Mr. Quinn worked on the design and system testing of the No. 1 ESS. He has been Supervisor of groups concerned with ESS system testing and development of maintenance software. For the 1A Processor project he was responsible for the processor diagnostics and for CPU logic/fault simulation. During the development of the 3B20D Processor he was responsible for the Processor Control Unit diagnostics and for the diagnostic control programs. Member, IEEE, Eta Kappa Nu.

**Michael W. Rolund,** B.S.E.E., 1961, Cooper Union; M.S.E.E., 1963, New York University; Bell Laboratories, 1961—. Mr. Rolund has been involved with the development of memory systems and processors including No. 1 ESS, the 1A Processor, and the 3B processor family. He currently is Head of the Processor Design Department responsible for the 3B20D and its evolution. Member, IEEE, Tau Beta Pi.

**Bruce R. Rowland,** B.S., 1973, Michigan State University; M.S., 1975, and Ph.D. (Computer Science), 1977, University of Wisconsin–Madison; Bell Laboratories, 1977—. Mr. Rowland began work in the area of languages and compilers, where he helped to extend the C programming language and coordinated the development of compilation tools for four processors, including the 3B20D Processor. He currently is supervising a group planning the development of 3B Processor networking and doing exploratory work in computer system design.

**Jack M. Scanlon**, B.S. (Applied Science), 1964, University of Toronto; M.S.E.E., 1965, Cornell University; Bell Laboratories 1965—. Mr. Scanlon joined Bell Laboratories in 1965, and initially worked on fault-diagnosis, resolution, and recovery techniques for the Bell System's No. 1 ESS. He also worked on a missile flight simulator for the Safeguard project. In 1968, he became Supervisor of the No. 1 ESS Call Program Group, responsible for the development of new customer call-features. In 1971, he joined the initial development team for No. 4 ESS with responsibility for system design and call-handling software. In 1974, he was promoted to Head of the No. 4 ESS System Design and Operations Department, where he was responsible for design of new capabilities for No. 4 ESS in domestic and international

applications, and for exploratory work on new software techniques. In 1977, he became Director of a Laboratory where he was responsible for the exploration of a secure voice/data communications system for the government. In June 1979, he was appointed Executive Director, Processor and Common Software Systems Division. In November 1982 he was appointed to the newly created position of Vice President, Processors, Western Electric. This work involves development of processors, microprocessors, *UNIX** operating systems, and programming languages and tools for Bell System applications. Mr. Scanlon has been granted four patents in processor design and electronic switching system design. He has published numerous articles on processor design for real-time, time-shared systems, electronic switching systems design, and software development techniques. Member, Computer Science Board of the National Academy of Sciences, IEEE.

**D. A. Schmitt,** B.S.E.E., 1965, St. Louis University; M.S. (Mathematics), 1968, Stevens Institute of Technology; Southwestern Bell, 1962–1965; Bell Laboratories, 1965–1969 and 1973—. At Bell Laboratories, Mr. Schmitt has been involved in the TSPS, No. 3 ESS, and VSS projects. He currently is in charge of the DMERT Operating System Architecture Department. Member, Eta Kappa Nu, Pi Mu Epsilon, Alpha Sigma Nu.

**W. C. Schwartz,** B.S.E.E., 1966, Purdue University; M.S.E.E., 1967, University of Michigan; Bell Laboratories, 1966—. Mr. Schwartz has worked on No. 4 ESS call-processing and fault-recovery software subsystems, software testing methodologies, and logic analysis and simulation systems. More recently, he was the generic engineer and application interface for DMERT. He currently is Supervisor of the Operating System Group in the Processor and Operating Systems Development Laboratory. Member, IEEE, Tau Beta Pi, Sigma Xi.

**Wing N. Toy,** B.S.E.E., 1950, and M.S.E.E., 1952, University of Illinois, and Ph.D. (Electrical Engineering), 1969, University of Pennsylvania; Bell Laboratories, 1952—. Mr. Toy has been involved in the design of highly reliable processors for the Bell System electronic switching systems and other telephone-related applications for the past 25 years. He was on the Faculty of the Computer Science Division of Electrical Engineering at the University of California, Berkeley, as a Visiting MacKay Lecturer during the 1973–1974 academic year. Mr.

* Trademark of Bell Laboratories.

Toy holds 19 U.S. patents and is co-author of two books, *Mini/Microcomputer Hardware Design* and *Microprogrammed Control and Reliable Design of Small Computers*. He is currently Supervisor of the Voice/Data Technology Group. Fellow, IEEE.

**Susan S. Weber,** B.S. (Mathematics and Computer Science), 1977, Iowa State University; M.S. (Computer Science), 1978, Purdue University; Bell Laboratories, 1977—. Ms. Weber initially designed software development systems with her primary emphasis on source administration and object generation. Since becoming involved with DMERT, she has worked on both system update and field update and has participated in generic engineering and application interface issues. Currently, she is the Supervisor of the Field Update Group in the Operating System Development Department.

**R. J. Welsch,** B.S. (Mathematics), 1967, Marquette University; M.S. (Computer Science), 1972, Northwestern University; Bell Laboratories, 1967–1968; U.S. Army, 1968–1970; Bell Laboratories, 1970—. At Bell Laboratories, Mr. Welsch has had experience with No. 1 ESS/ADF, No. 4 ESS, and systems engineering for electromechanical switching systems. From 1974 to 1978, he was a member of the Program Administration and Support Program Group of the Operator Systems Laboratory (TSPS). From 1978 to 1980, Mr. Welsch was involved with developing software development systems utilizing PDP 11/70's running the *UNIX* operating system as front-end processors to larger IBM main frames. Since 1980, he has been involved with 3B20D/DMERT software development and administration. Mr. Welsch is currently Supervisor of the DMERT Administrative Software Systems Group.

**F. W. Wendland,** B.S.E.E., 1965, M.S.E.E., 1966, Cornell University; Bell Laboratories, 1966—. Mr. Wendland has been involved in the development of electromagnetic compatibility facilities and standards for ESS systems, automatic test facilities for ESS processor subsystems, 1A and 3B20D Processors, and I/O system architecture and design. He currently supervises a group working on the architecture and design of networking hardware for the 3B processor line. Member, IEEE.

**Neil O. Whittington,** B.S. (Physics), 1965, M.S. (Physics), 1970, Illinois State University; Bell Laboratories, 1970—. Mr. Whittington has worked on the development of software for the 1A Processor and the 3B20D Processor. He is presently Head of the Application Support Department, concerned with DMERT generic planning and application interfaces with the 3B20D. Member, IEEE.

**Robert M. Wolfe,** B.S.E.E., 1952, University of Louisville; M.S.E.E., 1957, Columbia University; Bell Laboratories, 1952—. From 1952 to 1962, Mr. Wolfe worked on applied research on ferromagnetic and ferroelectric devices and on data collection and data transmission systems. In 1962, he joined the Switching System Development area, where he headed the development of the Automatic Intercept System, the AMA Recording System, the Service Evaluation System, and the Network Control Point System. He is currently Head, Network Control Point Department. Member, ACM, IEEE.

**Robert H. Yacobellis,** B.S. (Mathematics), 1967, Carnegie-Mellon University; M.S., 1969, Ph.D., 1973 (Information Sciences), University of Chicago; Bell Laboratories, 1967—. Mr. Yacobellis worked initially on No. 1 ESS 4-wire AUTOVON (Government Switching). From 1969 through 1977 he was in No. 4 ESS working on overload control and later on program administration. In 1978 he became Supervisor of a group providing common loaders and simulators to ESS projects, and in 1980 became responsible for the 3B20D Field Update Group. He currently is in charge of documentation and software quality for the 3B20D. Member, Tau Beta Pi, ACM.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 February 1983 Number 2, Part 1

Copyright © 1983 American Telephone and Telegraph Company. Printed in U.S.A.

# Digital Communications Over Fading Radio Channels

By G. J. FOSCHINI and J. SALZ

(Manuscript received August 19, 1982)

*A major contribution to system outage in a terrestrial digital radio channel is deep fading of the frequency transfer characteristic, which in addition to causing a precipitous drop in received signal-to-noise ratio (s/n) also causes signal dispersion that can result in severe intersymbol interference. Because the temporal variation of the channel is slow compared to the signaling rate, the information theoretic channel capacity and the "Efficiency Index" in bits/cycle—a figure-of-merit we use for the communication techniques considered—can be viewed as random processes. Starting from an established mathematical model characterizing fading channels (derived from extensive measurements), we estimate the probability distribution of channel capacity and the distributions of efficiency indices for different communications techniques. The repertoire of communication methods considered involves quadrature amplitude modulation with adaptive linear and decision feedback equalization, and maximum likelihood sequence estimation. For specific outage objectives the maximum number of bits per cycle achievable by each technique is estimated. The sensitivity of the distributions to bit-error-rate objective and unfaded s/n is assessed. For certain desired operating points the efficacy of adaptive equalization is demonstrated. There are some operating points where adaptive equalization alone is not adequate and therefore space diversity should be considered. An estimate of the effect of frequency diversity is also included.*

## I. INTRODUCTION AND SUMMARY

Fading of terrestrial digital radio channels owing to multipath reception is a prime cause of system outage. For a specific hop a mathematical model of these fades has been developed by W. D. Rummler[1,2] from extensive measurements of the channel frequency power transfer characteristic over time. The radio channel has a time-varying frequency characteristic, with additive Gaussian noise; however, the temporal variations are sufficiently slow in comparison to the data symbol rate that the characteristics can be represented as a random ensemble of static frequency power transfer functions. The presence of additive noise implies that each member of the ensemble is limited to a maximum rate of transmission of data, depending on the communication method. For each specific communication technique, the stochastic nature of the channel makes it meaningful to consider the probability distribution of data rates that can be supported at a certain bit-error-rate objective.

The purpose of this paper is to explore the relative performance of various communication techniques employing quadrature amplitude modulation (QAM), distinguished by the type of equalization method used. These techniques include variants of linear equalization, decision feedback equalization, and maximum likelihood sequence estimation (MLSE). For these methods a unified set of Chernoff bounds on the probability of error is obtained. Given a communication method, a channel impulse response, an error-rate objective, a received unfaded channel s/n, a channel bandwidth, and a signaling rate we use the Chernoff bounds to estimate the maximum number of bits per cycle of bandwidth (not necessarily integer-valued) for which the constraints are met. By computing the maximum number of bits per cycle supported by each member of a large representative population of channels, we obtain the cumulative probability distribution function. One can use the cumulative distribution curve to determine the probability of outage at a prescribed bit rate.

The information theory bound on the number of bits per cycle attainable is also derived. In a sequence of plots we compare the different schemes with each other and with the information theory limit.

If $F(r)$ is the probability distribution function of data rates associated with a communication method and we set an outage objective, $\xi$, then the value $r_\xi$ for which $F(r_\xi) = \xi$ represents the maximum data rate at which it is possible to transmit and meet the outage objective. We present and discuss these distributions in the context of desired long-haul and short-haul outage objectives and rates associated with the digital hierarchy constraints. The efficacy of adaptive equalization is established. The advantage of decision feedback and MLSE over

optimum linear equalization is not very substantial. There are some desired operating points for which space diversity should be considered.

For a fair comparison of different communication techniques, the transmitter filter shape must be optimized for a fixed transmitter power. We found the performance to be insensitive to whether or not the transmitter filter is optimized and we provide a theoretical guideline to indicate when this optimization becomes significant.

Our results indicate that optimized equalizer structures yield data rates only a few bits/cycle below channel capacity. It therefore appears that higher dimensional constellations[3] spanning two to four symbol intervals could go a long way toward obtaining that which can be expected practically. Although we did not analyze higher dimensional signal design or optimize the constellation in QAM, it is reasonable to expect that these techniques can offer at most an equivalent few dB increase in s/n. Another method of achieving coding gain "of the order of 3-4 dB" is described in Ref. 4. Moreover, the real limitations on the selection of signal points in a practical system will most likely arise from the nonlinear operation of radio frequency (RF) power amplifiers rather than from s/n limitations.

We argue the merits of adaptive transversal equalization and provide numerical support for our claims. This is not to say that fixed or even adjustable bump and/or slope equalizers in the frequency domain could not provide adequate performance in some cases. However, fluctuating (and sometimes nonminimum) phase distortion associated with fading and other linear filters admits robust and stable compensation via adaptive transversal filters. These structures with adjustable taps can automatically equalize any phase characteristic without noise enhancement and therefore are natural candidates in these applications, especially at a high number of levels where even small amounts of phase distortion can degrade system performance.

Our analysis was carried out with ideal models and an infinite number of taps. The actual number of taps needed in any application would be determined from experiments and/or more detailed analysis.

## II. THE EQUALIZED QAM SYSTEM—IDEALIZED MODEL

The use of equalizers to mitigate the effects of intersymbol interference and noise in voiceband data transmission is by now standard practice. We are thus naturally led to consider the application of these techniques in digital data transmission over the radio channel where slowly varying, frequency-selective fading is the predominant impairment. Here we review and derive the applicable mathematical theory that will be used in the sequel to evaluate the system performance indices.

To focus on basics and avoid extensive numerical analysis, we consider idealized equalizer models represented as transversal filters with an infinite number of taps. Tap adjustment algorithms are well established and our formulas are derived under the assumption that the taps have converged to their optimum values.

Our analyses are based on the digital communications model depicted in Fig. 1. To appreciate the applicability and generality of this baseband model to digital radio communications, we observe that, for any bandpass linear channel, the output waveform, when the input is any linearly modulated data wave, can be represented as

$$s(t) = \mathrm{Re}\left\{\sum_n \tilde{a}_n \tilde{h}(t - nT + t_0)\exp[i(2\pi f_0 t + \theta)]\right\},$$

where $\mathrm{Re}\{\cdot\}$ stands for the "real part." The data symbols $\{\tilde{a}_n\}$ transmitted at $T$-second intervals, are statistically independent and, in two-dimensional modulation systems such as QAM, they assume complex values. The overall equivalent baseband impulse response, $\tilde{h}(t)$, is also complex-valued. The real part represents the in-phase response, while the imaginary part is the quadrature component. The frequency, $f_0$, is the carrier frequency, $\theta$ is the carrier phase, and, $t_0$ is the timing phase. Ideal demodulation with a known carrier frequency $f_0$ and carrier phase $\theta$ implies a translation of the received bandpass signal to baseband. The real part of the resulting complex signal represents the in-phase modulation, while the imaginary part is the quadrature modulation. This then is the rationale, in addition to economics of notation, for using the complex baseband model depicted in Fig. 1.

We restrict our treatment to ideal Nyquist systems with no excess bandwidth. This permits less cumbersome calculations without loss of physical insights. We also derive our formulas by assuming flat transmitting filters and prove later that in-band optimum shaping yields imperceptible additional benefits. Also neglected is adjacent channel interference, as ideal bandlimiting eliminates this problem.

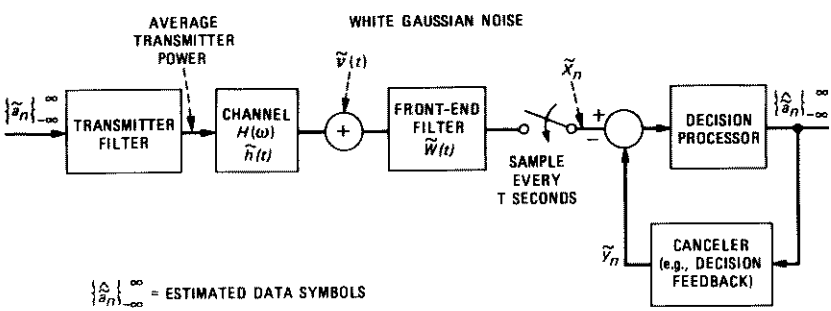



Fig. 1—Complex baseband model for QAM data transmission.

We now return to Fig. 1 and discuss the various functions and notations indicated. Without loss of generality we assume that the complex data symbols, $\{\tilde{a}_n = a_n + ib_n\}_{-\infty}^{\infty}$, take on values on a set of positive and negative odd integers with equal probability. Accordingly,

$$E\tilde{a}_n = 0 \quad \text{and} \quad E|\tilde{a}_n|^2 = 2\frac{L^2 - 1}{3},$$

where $E(\cdot)$ denotes mathematical expectation and $L$ (even) is the maximum number of data levels assumed by $a_n$ and $b_n$. Thus, in QAM $L^2$ data points are available for conveying information and the source therefore generates

$$R = \frac{\log_2 L^2}{T} \text{ bits/sec.} \tag{1}$$

For a given channel bandwidth, $\mathscr{W}$, the efficiency index is defined as

$$I = R/\mathscr{W} = \frac{2\log_2 L}{\mathscr{W}T} \text{ bits/cycle.} \tag{2}$$

As we shall see, the relationship among $P_e$- probability of error, s/n, $\mathscr{W}$, $T$ and $H(\omega)$-channel frequency characteristics is rather complicated. The determination of the relationship for different communication techniques is our chief task in the sequel.

From a mathematical point of view, the fading radio channel is characterized by a slowly varying linear distorting filter whose base-band equivalent complex impulse response is the Fourier transform of the transfer function $H(\omega)$, shifted to zero frequency:

$$\tilde{h}(t) = h_1(t) + ih_2(t) = \int_{-2\pi\mathscr{W}}^{2\pi\mathscr{W}} H(\omega)e^{i\omega t}\frac{d\omega}{2\pi}.$$

At the receiver the added complex noise process, $\tilde{v}(t) = v_1(t) + iv_2(t)$, is assumed to be white Gaussian with $v_1(t)$ independent of $v_2(t)$ and each possessing a double-sided spectral density, $N_0/2$. So,

$$\begin{aligned} E|\tilde{v}(t)|^2 &= Ev_1^2(t) + Ev_2^2(t) \\ &= N_0\delta(0), \end{aligned}$$

where $\delta(\tau)$ is the Dirac delta function. The average transmitted signal power, $P_0$, for a flat transmitting filter can easily be calculated. However, for our purposes a more relevant quantity is the received, unfaded signal power

$$P = K^2P_0 = 2\frac{L^2 - 1}{3}\frac{K^2}{T^2},$$

where $K$ is a constant that includes the effects of amplifiers, antennas,

and the unfaded channel loss. Also, the added average noise power in the Nyquist band, $\mathcal{W} = 1/2T$, is

$$P_v = \frac{N_0}{T}.$$

Thus the unfaded received $s/n$, a most important system parameter, is

$$s/n = \rho = 2\frac{L^2 - 1}{3}\frac{K^2}{N_0}\frac{1}{T}. \tag{3}$$

The receiver structures under consideration consist of a perfect demodulator followed by a front-end filter possessing the complex impulse response $\tilde{W}(t)$, a sampler, a decision device, and a canceler. The design of an optimum receiver entails the selection of $\tilde{W}(t)$ and the canceler for a particular channel characteristic. Since the channel characteristics are usually unknown to the receiver, these components must be determined adaptively.

To understand the function of the canceler, consider the signal sample at the output of filter $\tilde{W}(t)$,

$$\tilde{x}_n = \sum_{k=-\infty}^{\infty} \tilde{r}_k \tilde{a}_{n-k} + \tilde{z}_n, \quad -\infty \leq n \leq \infty, \tag{4}$$

where $\tilde{r}_k = \tilde{r}(kT + t_0)$ is the overall complex-sampled system impulse response and $\tilde{z}_n$ is the Gaussian noise output sample. Ideally the canceler strives to synthesize the value

$$\tilde{y}_n = \sum_{k \in S} \tilde{r}_k \tilde{a}_{n-k} \tag{5}$$

and to subtract it from (4) where the set of integers $S$ is defined as $\{k \in S: k = -N_1 \cdots -1, 1 \cdots N_2\}$. The canceler's ability to synthesize these values presumes that some past ($k = 1, \cdots, N_2$) and/or future ($k = -1, \cdots, -N_1$) transmitted data symbols are perfectly detected and, moreover, that the set of complex numbers, $\tilde{r}_k$, are adaptively estimated.

The front-end filter, $\tilde{W}(t)$, is usually determined adaptively by minimizing the mean-squared error (MSE) between the sample, $\tilde{x}_n - \tilde{y}_n$, and the expected data symbol $\tilde{a}_n$:

$$MSE[N_1, N_2, \tilde{W}(t)] = E|\tilde{x}_n - \tilde{y}_n - \tilde{a}_n|^2, \tag{6}$$

and the optimum filter, $\tilde{W}(t)$, is chosen to achieve

$$(MSE)_0 = \min_{\tilde{W}(t)} MSE[N_1, N_2, \tilde{W}(t)] = MSE[N_1, N_2, \tilde{W}_0(t)]. \tag{7}$$

Since (6) is a quadratic functional of $\tilde{W}(t)$, a unique minimum can always be found. It is standard to represent the linear filter $\tilde{W}(t)$ by a

transversal structure and in practice the search for the minimum is accomplished by varying the taps of this filter until a minimum of the time average of the squared error is found. Clearly, to realize such a minimization procedure, estimates of the transmitted data symbols must be used.

## III. SYSTEM PERFORMANCE—GENERAL

To get at the efficiency index of a system, the error rate as a function of data rate for any choice of the canceler set, $\{S\}$, and front-end filter, $\tilde{W}(t)$, must be explicitly expressed. Unfortunately, exact relationships are not mathematically tractable for the simplest of systems and so we must employ upperbounds. Fortunately, for the systems under consideration, it is possible to obtain exponentially tight inequalities.

With this approach in mind, note that after perfect cancellation, the decision variable, from (4) and (5) becomes

$$\begin{aligned} s_n &= \tilde{x}_n - \tilde{y}_n \\ &= \tilde{r}_0 \tilde{a}_n + \sum_{k \notin J} \tilde{r}_k \tilde{a}_{n-k} + \tilde{z}_n, \end{aligned} \tag{8}$$

where now the set $J$ is $S \cup 0$, $\{k \in J: k = -N_1 \cdots 0 \cdots N_1\}$. Decisions in QAM are made on the real part of $s_n$ and, separately, on the imaginary part of $s_n$. Simple calculations give

$$Re(s_n) = \mu_0 a_n - \upsilon_0 b_n + \sum_{k \notin J} (\mu_k a_{n-k} - \upsilon_k b_{n-k}) + z_{n1},$$

and

$$Im(s_n) = \mu_0 b_n + \upsilon_0 a_n + \sum_{k \notin J} (\mu_k b_{n-k} + \upsilon_k a_{n-k}) + z_{n2}, \tag{9}$$

where

$$\tilde{r}_k = \mu_k + i\upsilon_k,$$

and

$$\tilde{z}_k = z_{k1} + iz_{k2} = \int_{-2\pi W}^{2\pi W} \tilde{v}(t) \tilde{W}(t) dt.$$

For an $L$-level system, slicing levels are placed at $0 \pm 2\mu_0 \cdots \pm \mu_0(L-2)$ and compared with the received samples $\mathrm{Re}(s_n)$ and $Im(s_n)$. An error occurs whenever the noise plus intersymbol interference (in-phase and quadrature) exceed in magnitude the distance from the transmitted level to the nearest decision threshold, $\mu_0$. However, the outside two levels can be in error in one direction only.

Now denote the event of an error committed in the "real" rail by $E_r$

and in the "imaginary" rail by $E_i$. Then the probability of system error, $P_e$, is the probability of either (or both) $E_r$ or $E_i$ occurring,

$$P_e = P(E_r \cup E_i) \leq P(E_r) + P(E_i), \tag{10}$$

where

$$P(E_r) = \left(1 - \frac{1}{L}\right) \Pr\left[\left|z_{n1} - \sum_{k \notin J} (\mu_k a_{n-k} - v_k b_{n-k}) + v_0 b_n\right| \geq \mu_0\right]$$

and

$$P(E_i) = \left(1 - \frac{1}{L}\right) \Pr\left[\left|z_{n2} - \sum_{k \notin J} (\mu_k G_{n-k} + v_k a_{n-k}) - v_0 a_0\right| \geq \mu_0\right]. \tag{11}$$

Because of symmetry, $P(E_r) = P(E_i) = P(E)$, and so we only need to upperbound $P(E)$.

We adopt a bounding procedure introduced by B. Saltzberg[5] to analyze the error rate in an unequalized baseband system. We have extended Saltzberg's approach to our systems and it can be shown that

$$P(E; A, B, \delta) \leq 2 \exp\left\{ \frac{\left[\mu_0 - (L-1)\left(\sum_{k \in A} |\mu_k| + |v_k| + \delta v_0\right)\right]^2}{2\left\{\sigma_{z_1}^2 + \frac{L^2 - 1}{3}\left[\sum_{k \in B} \mu_k^2 + v_k^2 + (1-\delta)v_0^2\right]\right\}} \right\}. \tag{12}$$

The set of integers $A$ and $B$ form a partition on the set of integers not included in $J$. That is,

$$A \cup B = \Omega = \{k: k \notin J\}$$

and

$$A \cap B = \phi.$$

The variable $\delta = 1$ *or* $0$, and

$$\sigma_{z_1}^2 = \frac{N_0}{2} \int_{-\infty}^{\infty} |\tilde{W}(t)|^2 dt.$$

The sharpest upperbound is obtained by minimizing (12) with respect to the sets $A$, $B$, and $\delta$. Algorithms for carrying out this minimization can be devised readily.

## IV. SYSTEM PERFORMANCE

### 4.1 Discussion

Equation (12) is a rather general upperbound on the error rate for any passband linear data transmission system and it will now be specialized to include the effects of the different choices of equalizers.Before proceeding with the detailed numerical analysis, we need to make a connection between the mean-squared error (MSE), which is minimized by equalizers, and the system probability of error, which, ideally, should be the quantity minimized.

A straightforward but tedious approach for getting at the error rate might be to first determine the filter, $\tilde{W}(t)$, which minimizes the MSE for any particular equalization scheme, calculate the overall resulting impulse response, and then use eq. (12) to upperbound the error rate. This approach can be circumvented by exploiting the explicit relationship between the minimum MSE and the value of the overall impulse response at $t = t_0$ when the optimum filter, $\tilde{W}_0(t)$, is used.

The optimum structure of the minimum mean-squared error receiver can be shown to consist[6] of a matched filter in cascade with a transversal filter combined with a linear intersymbol interference canceler. The implication of this structure is that the resulting overall system transfer function is a real function of frequency. Or, the complex-sampled impulse response, $\{\mu_k + i\upsilon_k\}_{-\infty}^{\infty}$, must be a real number at $k = 0$, which results in $\upsilon_0 = 0$. This follows from the Fourier Transform representation of $\tilde{r}(t)$, from which we see that at $t = 0$ the integrand is real and nonnegative. Indeed the overall phase characteristic has been removed by the matched filter (without enhancing the noise*). Using the fact that $\upsilon_0 = 0$ and careful numerical analysis of the available channel characteristics, our calculations showed that for all practical purposes the bound (12) becomes

$$P(E, S) \leq 2 \exp\left\{\frac{-\mu_0^2}{2\left[\sigma_{z_1}^2 + \sigma^2(L) \sum_{k \notin S} (\mu_k^2 + \upsilon_k^2)\right]}\right\}, \tag{13}$$

where we set $\sigma^2(L) = \dfrac{L^2 - 1}{3}$.

As will become apparent, the argument of the exponential function in (13) can be directly related to the minimum mean-squared error.

---

* A fractionally spaced ($T/2$) transversal filter can automatically synthesize any matched filter and thus eliminate phase distortion and also compensate for timing phase (see Ref. 7.)

Towards this end we recall a well-known[6] result that states that the best achievable MSE has the simple representation,

$$(MSE)_0 = 2\sigma^2(L)(1 - \mu_0), \tag{14}$$

where $\mu_0$ is the sample at $t = t_0$ at the output of the optimum filter. Also, when the optimum filter, $\tilde{W}_0(t)$, is used, a straightforward calculation of the resulting MSE gives

$$(MSE)_0 = 2\sigma^2(L)(1 - \mu_0)^2 + 2\sigma^2(L) \sum_{k \neq S} (\mu_k^2 + v_k^2) + 2\sigma_{z_1}^2. \tag{15}$$

Relationships (14) and (15) make it possible to write (13) as

$$P(E; S) \leq 2 \exp\left\{ -\frac{1}{(MSE)_0} \left[ 1 - \frac{(MSE)_0}{2\sigma^2(L)} \right]^2 \right\} \sim 2\exp -\frac{1}{(MSE_0)} \quad \text{for} \quad N_0 \to 0, \tag{16}$$

relating error rate and minimum MSE. This is an extremely useful inequality since $(MSE_0)$ as a function of channel characteristics is often explicitly known for different equalizer structures.

It is also interesting to note (this has been pointed out before[8]) that the filter, $\tilde{W}(t)$, that minimizes MSE also minimizes the upperbound on $P_e$. This is true because the same quadratic functionals in $\tilde{W}(t)$ are involved in the optimization of both expressions.

We are now in a position to specialize our formulas to the various equalizer structures under investigation.

The six examples that follow do not require the knowledge of channel phase characteristics to compute performance. Implicit in each of these schemes is the complete removal of phase distortion, which can be accomplished without noise enhancement. Only a magnitude characterization of the channel transfer response was available at the time the work reported here was done. While departure from flatness of the magnitude fundamentally affects performance, theoretically, departure of phase from linear has no effect on attainable performance. Therefore, the lack of phase characterization of the channels was not an obstacle to our study. However, a complex characterization of the channel would be useful in determining the minimum number of required taps in the designs of the equalizers.

### 4.2 Pure phase equalization

In this particular equalizer, $N_1 = N_2 = 0$ (where $N_1$ and $N_2$ are the lengths of the precursive and postcursive cancelers, respectively). We choose $\tilde{W}(t)$ so that

$$W(\omega) = e^{-i\phi(\omega)}, \, |\omega| \leq \frac{\pi}{T}$$

$$= 0, \, |\omega| > \frac{\pi}{T},$$

where $W(\omega)$ is the Fourier transform of $\tilde{W}(t)$ and $\phi(\omega)$ is the channel phase characteristic. For this choice of filter, only the magnitude of the channel transfer function enters into the computation of the bound, as shown in eq. (13).

Using the well-known Poisson sum formula along with some algebra, it is possible to write (13) more explicitly, i.e.,

$$P_e \leq 2 \exp\left\{ - \frac{\rho}{2\sigma^2(L)} \frac{\langle H \rangle^2}{1 + \rho \, \langle (H - \langle H \rangle)^2 \rangle} \right\}, \tag{17}$$

where we used the shorthand notation

$$\langle \cdot \rangle = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} [\cdot] d\omega$$

and $H$ is used in place of $|H(\omega)|$.

#### 4.3 Linear equalization

Here again $N_1 = N_2$ (no canceler) and $\tilde{W}(t)$ is chosen to minimize (6). The expression for the optimum MSE in this case has been shown to be[9]

$$(MSE)_0 = 2\sigma^2(L) < \frac{1}{1 + \rho H^2} >. \tag{18}$$

This formula is directly used in (16) to calculate the upperbound on error rate:

$$P_e < 2 \exp\left[ - \frac{1}{2\sigma^2(L)} \left( < \frac{1}{1 + \rho H^2} > \right)^{-1} \right].$$

#### 4.4 Inverse equalization

In this case $N_1 = N_2 = 0$ (no cancellation) we choose $\tilde{W}(t)$ to be the inverse of the channel frequency characteristics,

$$W(\omega) = H^{-1}(\omega), \, |\omega| \leq \frac{\pi}{T}$$

$$= 0, \, |\omega| > \frac{\pi}{T}.$$

Here the channel is clearly perfectly equalized so that intersymbol interference is completely eliminated; the penalty is increased output

noise power. For this simple scheme, it is possible to express the error rate exactly but for reasons of uniformity we use the upperbound

$$P_e \leq 2 \exp - \left\{ \frac{\rho}{2\sigma^2(L)} \frac{1}{\frac{1}{<\overline{H^2}>}} \right\}. \tag{19}$$

### 4.5 Decision feedback

For this equalization system $N_1 = 0$ and $N_2 = \infty$ and again we choose $\tilde{W}(t)$ to minimize (6). In this type of equalizer, the causal or postcursor intersymbol interference is entirely eliminated and an expression for the optimum MSE is known,[10,11] as shown below.

$$(MSE)_0 = \sigma^2(L)\exp\{-\langle \ln[1 + \rho H^2] \rangle\}. \tag{20}$$

This is used in (16) to express an upperbound on error rate.

### 4.6 The ideal equalizer

In this utopian scheme the precursor and postcursor cancelers become infinite, $N_1 = N_2 = \infty$, so that all the intersymbol interference is eliminated. In this ideal situation we obtain the very best possible result, namely, the matched filter bound, which is a lower bound on $P_e$. This scheme assumes that it is possible to detect each pulse $\tilde{a}_k \tilde{r}(t - nT)$ optimally by a matched filter without incurring interference from all other pulses. The filter, $\tilde{W}(t)$, in this case is chosen to be matched to the channel characteristic, i.e.,

$$W(\omega) = H^*(\omega), \, |\omega| \leq \frac{\pi}{T}$$

$$= 0, \, |\omega| > \frac{\pi}{T},$$

where * denotes the complex conjugate.

For this idealization the upperbound on error rate is simply,

$$P_e \leq 2 \exp\left\{ - \frac{\rho}{2\sigma^2(L)} \langle H^2 \rangle \right\}. \tag{21}$$

No other detection scheme can do better. In the next section we use these formulas to calculate the various efficiency indices.

Before concluding this section, we remark that there is one more easy case and one extremely difficult case that might be considered as candidates for making comparisons. Suppose that no filtering, other than out-of-band elimination of noise, were performed at the receiver. What performance can one expect? While we cannot answer this question exactly because channel phase characteristics are unavailable

at this writing, we would expect performance to be worse than removing the phase characteristic entirely—a situation we will examine.

The second approach, which is a very difficult one to analyze, involves the use of a finite-state Viterbi decoder. Nevertheless, we will report a bound on the performance of this processor. Specifically, the performance of an infinite canceler (the matched filter) is superior to the maximum likelihood (Viterbi) decoder. As shown later, for the channels considered, the matched filter bound is close in performance to decision feedback. Consequently, we shall see that the performance of maximum likelihood sequence estimation is tightly bracketed because it is superior to decision feedback.

### *4.7 Information theory bound on communications efficiency index*

In this section we discuss a formula for the maximum number of bits per cycle that can be attained for a given $H(\omega)$. If $H(\omega)$ were constant in frequency the formula for the efficiency index in bits per cycle would be simply

$$I = \log_2(1 + \rho|H|^2).$$

It is reasonable to expect that if $H(\omega)$ is frequency-dependent, the maximum efficiency index would be

$$I = \frac{1}{\Omega}\int \log_2(1 + \rho|H(\omega)|^2)d\omega, \tag{22}$$

where the integral is over a frequency band of size $\Omega = 2\pi W$. Indeed this is the case. To outline a derivation, we note first that A. Kolmogorov has generalized Shannon's notion of capacity to provide a very fundamental definition that gives a useful starting point for developing capacity formulas in nonstandard situations such as the one at hand.[12] M. S. Pinsker[13] was able to derive from the Kolmogorov approach a formula for the amount of information in a stationary Gaussian process about another stationary Gaussian process related to it. Specifically, if $S_x(\omega)$ and $S_y(\omega)$ are the power spectral densities of the processes and $S_{xy}(\omega)$ is the cross-spectral density, the formula for the amount of information is

$$-\int \ln\left(1 - \frac{|S_{xy}(\omega)|^2}{S_x(\omega)S_y(\omega)}\right)d\omega.$$

If we require the transmitter output to be Gaussian, then since the additive noise is Gaussian, Pinsker's formula can be applied to the case where $x$ is the transmitted process and $y$ is the received process to obtain (22). Requiring the transmitter output to be Gaussian is really not a limitation since, when the additive noise is Gaussian, one can prove that capacity is attainable with a Gaussian transmitter output

using the methods discussed in Refs. 14 through 16. Thus, (22) gives the efficiency index formula.

References 16 and 17 also provide approaches to establishing (22).

### 4.8 Information theory limit on index when transmitter is optimized

So far we have treated $\rho$, eq. (3), as a constant. In this section we set the stage for exploring the advisability of optimizing the output power spectral density to maximize the efficiency index. Since we are now allowing in-band shaping of the transmitter filter frequency characteristic, we will consider $\rho$ as a function of $\omega$ and write $\bar{\rho}$ to denote the previously considered case where $\rho$ is constant over the band.

Although the analysis in this section is focused on the information theory limit on the efficiency index, the decision feedback index involves the identical functional form and so our analysis will be applicable to decision feedback as well.

We will compare the previously discussed index

$$I(\text{flat}) = \frac{1}{\Omega} \int \log_2(1 + \bar{\rho}\,|H(\omega)|^2) d\omega$$

with

$$I(\text{opt}) = \frac{1}{\Omega} \int \log_2(1 + \rho_0(\omega)\,|H(\omega)|^2) d\omega,$$

where $\rho_0(\omega)$ is the function maximizing $I$ under a constraint on $\int \rho(\omega)\, d\omega$, the received signal-to-noise ratio in the absence of fading. This constraint is equivalent to a constraint on the transmitter output power, since in the absence of fading the channel has a flat loss characteristic.

This optimization problem is known[17] and yields easily to the calculus of variations. The solution is called "water pouring." The name stems from the graphical interpretation that if $\bar{\rho}\Omega$ is the constraint on $\int \rho(\omega)\, d\omega$, the optimum $\rho(\omega)$, which we denote by $\rho_0(\omega)$, is obtained by forming a vessel with base $|H(\omega)|^{-2}$ and vertical sides at the band edges. One pours "water," that is, area, of amount $\bar{\rho}\Omega$ into the vessel and $\rho_0(\omega)$ is given by the depth of the water at $\omega$. It is clear that this construction obeys the constraint. Generally, if $\bar{\rho}$ is sufficiently small, the poured water will not touch both of the vertical sides of the vessel. In such situations, it would be advantageous to limit the transmitted power to a frequency band less than Nyquist. In our case, however, the unfaded signal-to-noise ratio is so great that, for the simulated channels, the water level always meets both vertical sides. In other words, the transmitted power always occupies the full Nyquist band. Thus, $\rho_0(\omega) = A - |H(\omega)|^{-2}$, where $A$ is chosen so that $\int \rho_0(\omega) d\omega = \bar{\rho}\Omega$, that is, $A\Omega - \int |H(\omega)|^{-2} d\omega = \bar{\rho}\Omega$ or

$$\rho_0(\omega) = \bar{\rho} + \frac{1}{\Omega}\int |H(v)|^{-2}dv - |H(\omega)|^{-2}.$$

Therefore,

$$\frac{I(\text{flat})}{I(\text{opt})} = \frac{\int \log_2(1 + \bar{\rho}|H(\omega)|^2)d\omega}{\int \log_2\left[\frac{|H(\omega)|^2}{\Omega}\int |H(v)|^{-2}dv + \bar{\rho}|H(\omega)|^2\right]d\omega}.$$

We expand the logarithm for $\bar{\rho}$ large to find an asymptotic representation. We get, after a cumbersome derivation,

$$\frac{I(\text{flat})}{I(\text{opt})} = 1 - \frac{1}{2\bar{\rho}^2\log_2\bar{\rho}}\left(1 + \frac{\int \log_2|H|^2 d\omega}{\Omega \log_2\bar{\rho}}\right)^{-1} \cdot\left[\frac{\int \frac{1}{|H|^4}d\omega}{\Omega} - \left(\frac{\int \frac{1}{|H|^2}d\omega}{\Omega}\right)^2\right] + O\left(\frac{1}{\bar{\rho}^3\log_2\bar{\rho}}\right). \quad (23)$$

The last multiplier in the perturbation expression represents the variance associated with a random sampling of a specific $|H(\omega)|^{-2}$. This multiplier is zero if $|H(\omega)|^{-2}$ is a constant. We note, for later use, that for $\bar{\rho} = 10^{6.3}$, (i.e., a 63-dB s/n in the absence of fading) we have

$$\left(\frac{1}{2\bar{\rho}^2\log_2\bar{\rho}}\right) < 10^{-14}.$$

#### *4.9 Communication efficiency index*

The relationships in eqs. (17) through (21) are unifying expressions for the error rate for the five equalization cases. From Section 1.0 we have $I = 2\log_2 L$ and $\sigma^2(L) = [(L^2 - 1)/3]$. These two equations in conjunction with the $P_e$ formulas enable us to determine $I$ as a function of the $P_e$ objective, $\rho$, the channel, and the equalization scheme. Specifically,

$$I = \log_2\left[\frac{G(H, \rho)}{\ln(P_e/2)} + 1\right],$$

where $G(H, \rho)$ is a function that depends on the communication method. With the channel response considered to be a random function, $I$ is a random variable, and we can determine its probability distribution function for each communication scheme. The quantities $\rho$ and $P_e$ are parameters of the distribution.

## V. MODEL FOR THE FADING CHANNEL

Now we describe the mathematical model for frequency-selective fading owing to multipath reception. This mathematical model for the random functions $|H(\omega)|^2$ is due to W. D. Rummler[1,2] and is based on measurements of a 26.4-mile hop between Palmetto and Atlanta, Georgia. The measurements of frequency-selective fades were made on a 25.3-MHz channel situated in the 6-GHz band during the heavy fading month of June (1977).

Rummler's model uses a two-ray representation of the signal, which was quite adequate for fitting the experimental records. (The model is not necessarily intended to depict the underlying physical mechanism for a fade. The true mechanism could involve a much more complex ray combination.) Also, it is not possible to deduce the phase characteristic associated with any particular amplitude characteristic. It has been experimentally determined that this kind of a channel cannot always be viewed as minimum phase.[18]

In the model, the $|H(\omega)|^2$ functions are 68-degree sections of scaled, displaced cosine waves. Specifically, conditional on a fade occurring

$$|H(\omega)|^2 = a^2|1 + b^2 - 2b\cos(\omega\tau + \theta)|,$$

where:

(*i*) $b = 1/10^{B/20} > 0$ with $B$ an exponential random variable with mean 3.8.

(*ii*) The parameter, $a$, is a log normal random variable with dependence on the parameter, $b$. Specifically, $a = 1/10^{A/20}$, where $A$ is normal with a mean of $24.6(B^4 + 500)/(B^4 + 800)$dB and a standard deviation of 5 dB.

(*iii*) The phase, $\theta$, is independent of $a$ and $b$ and has a constant density on each section $|\theta| > \pi/2$ and $|\theta| < \pi/2$ with $P\{|\theta| < \pi/2\} = 5 \cdot P\{|\theta| > \pi/2\}$.

(*iv*) The scale factor $\tau$ is a constant = 6.31 nanoseconds.

In the model, the channel is in the faded state for 8060 seconds in a normal heavy fading month. Thus the channel can be viewed as being in one of two states where:

$$P\{\text{unfaded state}\} = 0.99689$$
$$P\{\text{faded state (Rummler model operative)}\} = 0.00311.$$

In what follows we employ this model to estimate the outage distributions for various communication methods. The model should be regarded as a working assumption valuable in gaining initial insight into the potential of the communication techniques we consider. However, we emphasize that more measurements may be required to refine Rummler's model to accommodate different geographical situations and wider bandwidths than 25 MHz, and to sharpen the accu-

racy of the representation of the more severe fades that are of major concern in what follows.

## VI. OUTAGE OBJECTIVES AND SOME PROSPECTIVE INDEX VALUES

From the proposed performance objectives for the digital transmission network,[19] we have that the round trip system availability objective is 99.98 percent. So the probability of outage is 0.0002 round trip or 0.0001 one way. The 0.0001 breaks down as 0.00005 for fading, 0.000025 for equipment failure, and 0.000025 for maintenance and plant errors. Thus, for a 4000-mile system composed of 156 hops, each with a nominal length of 25.6 miles, we get a per hop outage probability of $3.2 \times 10^{-7}$ for fading. This corresponds to about 10 seconds of outage per year. If we assume the year is composed of three heavy fading months and nine months with no fading, we obtain that the probability of outage in a heavy fading month is $1.28 \times 10^{-6}$. For a 250-mile short-haul system, the outage objective is 16 times less stringent on a hop, namely, $2.05 \times 10^{-5}$.

For the purpose of discussion we shall later consider the possibility of accommodating two DS-3 digital signals in a 20-, 30-, and 40-MHz channel. Each DS-3 signal corresponds to 672 64 kb/s voice circuits, so that two DS-3 signals correspond to about 90 Mb/s. Thus, for 20-, 30-, and 40-MHz channels we require 4.5, 3, and 2.25 bits per cycle, respectively. We will use $10^{-4}$ as the probability of bit-error threshold for registering outage. The sensitivity to this threshold will also be analyzed.

## VII. COMPUTER PROGRAM

A comprehensive FORTRAN program was written to compute and display outage distributions. The program is composed of three main segments.

The first segment simulates the power transfer characteristic for the channel in the faded state. It uses a PORT routine to generate random numbers uniformly distributed on [0, 1] and functions of these are evaluated to produce the random variables with the three densities underlying Rummler's model. The variables $A$ and $B$ are appropriately correlated. The random channel characteristics are then computed and a file containing them is produced. The file contains 25,000 characteristics.

The second segment calculates the efficiency index for each channel and then computes the probability distribution function of the indexes. Various options and parameters can be chosen in exercising this stage. These include:

(*i*) Method of communication (i.e., type of linear equalization, decision feedback equalization, MLSE, and the information theoretic optimum processing)

(*ii*) Transmitter spectrum, i.e., optimization of the transmitter spectral density for a given average power constraint or flat power spectral density

(*iii*) Probability of bit-error objective

(*iv*) Unfaded signal-to-noise ratio at the receiver input

(*v*) Channel bandwith.

Option (*ii*) is only available for the decision feedback and the information theoretic optimum communication schemes. Extending the option to the other schemes seemed inadvisable because of the closeness of the results, as will be seen later.

The number 25,000 was found through computational experience to stabilize the density tail in the range of interest and yet not be wasteful of computer resources. Since the number 25,000 is very close to the number of experimental records of fade characteristics, we could have worked from original experimental data. We elected to work with the Rummler model since it is weighted to track the worse fades, which are our interest here, and since the model is widely accepted.

The final segment of the computer program provides labeled plots of the outage distribution functions. It uses the graphic package DISSPLA.

## VIII. PRINCIPAL RESULTS

### *8.1 Preparatory remarks*

For the purpose of presenting our principal results we will need the following notation for the outage distribution functions:

$F_{PH}$: phase distortion removed

$F_{LIN}$: optimum linear equalization

$F_{DF}$: postcursive intersymbol interference (ISI) removed

$F_{MF}$: all ISI removed (matched filter bound)

$F_{IT}$: information theory limit (Shannon).

The efficiency index distributions were computed for 30-MHz channels. Strictly speaking, the notion of an index in bits per cycle is imprecise in that $F(I)$ (the probability distribution of bits per cycle) would change if calculated at 20 MHz or at 40 MHz. However, by calculation we established that, for the purposes of the discussion that follows, treating $F(I)$ as invariant over the 20- to 40-MHz range of bandwidths is an adequate approximation.

In the actual development of systems of the kind we have idealized, much more detailed performance analysis is required than that reported here. One important aspect we have not considered is the effect of excess bandwidth associated with practical filter designs with rolloff factors other than zero. To get a preliminary idea of the effect of rolloff of an amount $\alpha$ on the communication efficiency indexes, we would simply scale the distributions abscissas by an amount $1/(1 + \alpha)$.

Equivalently, we would inflate the desired number of bits per cycle by $1 + \alpha$ before going to the curves. Thus, in considering the accommodation of two DS-3 signals, with $\alpha = 1/3$, we would inflate the 4.5-, 3-, and 2.25-bit per cycle values corresponding to 20-, 30-, and 40-MHz channels by 1.333 to obtain 6, 4, and 3 bits per cycle. We would then consult the derived curves at these values to obtain the outage probabilities. For the purpose of discussion in the section that follows, we use these three inflated bits per cycle values along with the long-haul and short-haul objectives of $1.3 \times 10^{-6}$ and $2.1 \times 10^{-5}$, respectively, given in Section VI. Subsequently, an alternative means of accounting for $\alpha$ will be given.

### 8.2 The graphs

The most striking features of the outage distribution functions $F(I; P_e, \rho)$ are exhibited in Fig. 2. The beneficial effects of adaptive equalization are apparent. The three equalization schemes yield roughly similar results; however, as one looks more closely at the extreme outage tail, $F_{LIN}$, $F_{DF}$, and $F_{MF}$ begin to depart from each other. $F_{IT}$ is displaced over two bits per cycle to the right of $F_{MF}$, while

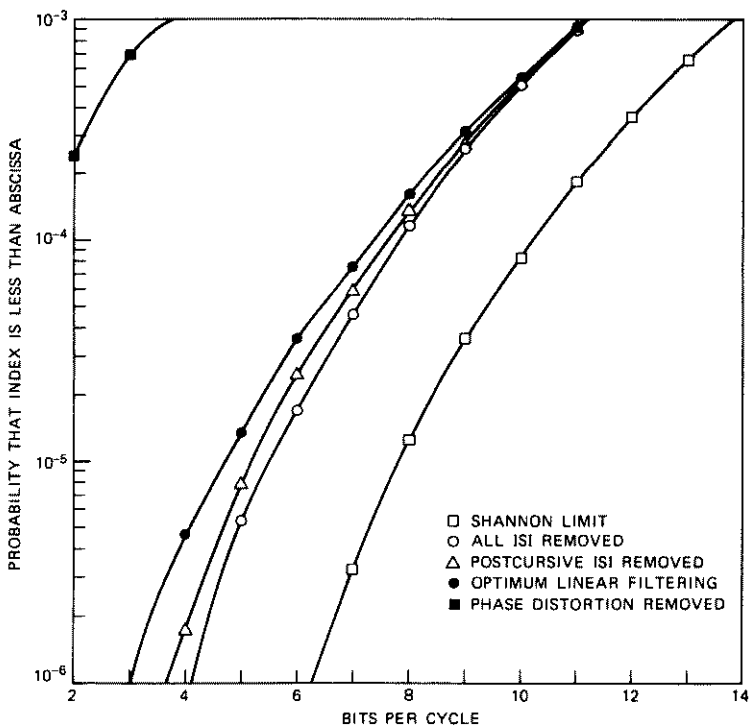



Fig. 2—Comparison of index distribution tails at 63-dB s/n, p.e. $< 10^{-4}$ (p.e. does not apply to the Shannon limit curve).

$F_{PH}$ is very substantially to the left of $F_{LIN}$. For the 40-MHz band (3 bits/cycle), both outage objectives are met with linear equalization. For the 30-MHz band (4 bits/cycle) and the long-haul objective, linear and decision feedback equalization are not adequate and some coding or use of maximum likelihood sequence estimation are possible solutions. However, it may be practical to overcome the shortfall by some other means, such as improving the amplifier noise figure. For the 20-MHz channel (6 bits/cycle) the long-haul objective is not met. Also, this efficiency is so close to the information theory limit that any attempt to achieve it by coding may be ill-advised because of complexity. On the other hand, with some moderate coding the short-haul objective for 20-MHz channels should be attainable. For the other two bands, short-haul objectives are roundly met.

The plot for the equalizer that inverts the channel is not shown, as it is not perceptibly different from that for the optimum linear equalizer. This is expected since the optimum linear filter is essentially inverting the channel at the high signal-to-noise ratios we are considering.

Figures 3, 4, and 5 show the sensitivity of $F(I, P_e, \rho)$ to $P_e$. The

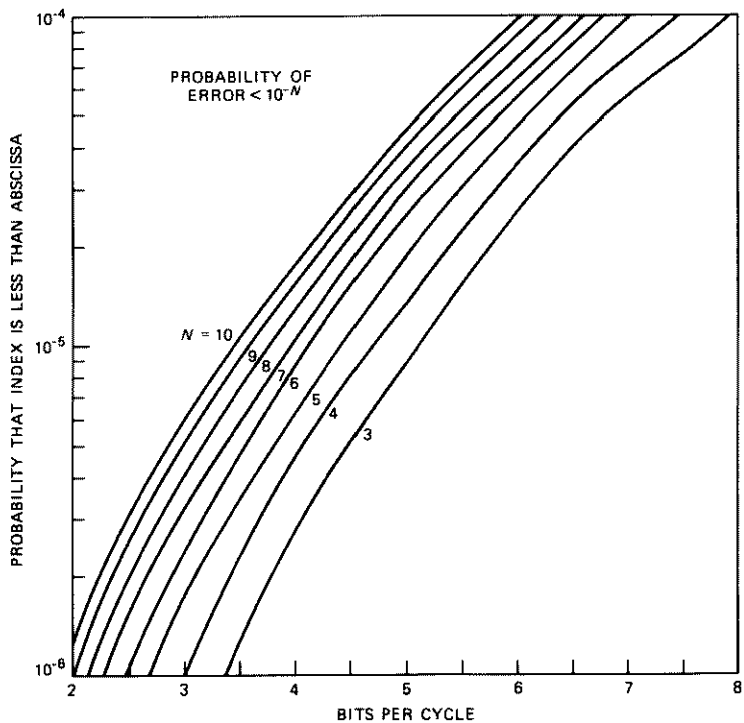



Fig. 3—Index distribution tail sensitivity to probability of error for linear equalization at 63-dB s/n.

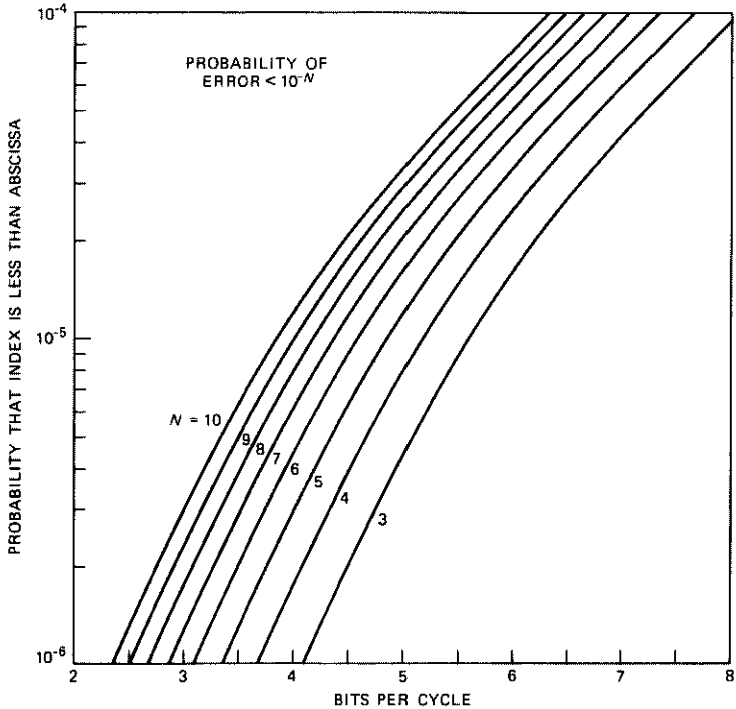



Fig. 4—Index distribution tail sensitivity to probability of error for decision feedback at 63-dB s/n.

sensitivity is especially small at the low error rates needed for data (as opposed to voice) transmission. An asymptotic analysis shows that, for large $\rho$, the curves translate to the left in accordance with a $\log_2 N$ shift, where $10^{-N}$ is the $P_e$ objective. This insensitivity is an illustration of the well-known result[20] that once a pulse code modulation (PCM) operating point is achieved it takes a very small improvement to make the error rate an order of magnitude smaller. In fact, if at some operating data rate the probability of error turns out to be $10^{-5} - 10^{-6}$, one should be able to design an error-correcting code of small redundancy and moderate complexity that could improve the error rate by several orders of magnitude.

Figures 6 through 9 illustrate the sensitivity to signal-to-noise ratio. The translation in all cases is roughly 1/3-bit/cycle/dB. Note the curves for the Shannon limit have an ordinate range of 4 to 10 bits per cycle, while the others range from 2 to 8 bits per cycle. No sensitivity for $F_{PH}$ is given since, unlike all the other distributions, there is negligible improvement as $\rho$ increases. This is because, as $\rho$ increases, the effect of intersymbol interference (ISI) remains and nothing is being done to mitigate it. In the other four cases, ISI tends to be

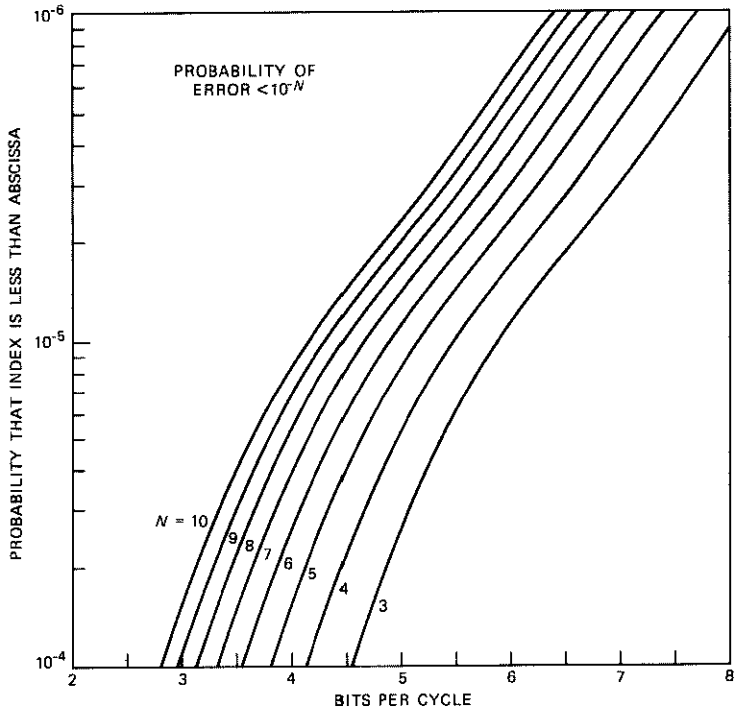



Fig. 5—Index distribution tail sensitivity to probability of error for the case of all ISI removed.

eliminated as $\rho$ increases (the channels can be perfectly equalized without an inordinate amount of noise enhancement).

In Section 8.1 we mentioned the $(1 + \alpha)^{-1}$ scaling as a method of accounting for rolloff. This assumed $\mathscr{W} = 1/T$ so that $(1 + \alpha)\mathscr{W}$ is the actual bandwidth. Suppose instead that the real bandwidth is fixed at $\mathscr{W}$ but the data rate is slowed by an amount $(1 + \alpha)$, leaving the average transmitter power and $N_o$ constant. Then the true s/n is increased by $10 \log_{10}(1 + \alpha)$. From this alternative perspective the suggested $(1 + \alpha)^{-1}$ scaling would be supplemented by a shift to the right of the probability distribution function tail by approximately $(10/3)\log_{10}(1 + \alpha)$ bits per cycle. Whether in estimating the effect of rolloff one takes the perspective that the symbol rate or the bandwidth is fixed is a matter of convenience.

Next, we consider optimization of the transmitter power spectral density. There is a practical question as to whether such an optimization could be achieved since the fade characteristic, which is first determined at the receiver, would need to be relayed back to the transmitter in time to be useful. However, the question is academic since we demonstrated that, even if an optimized transmitter could be

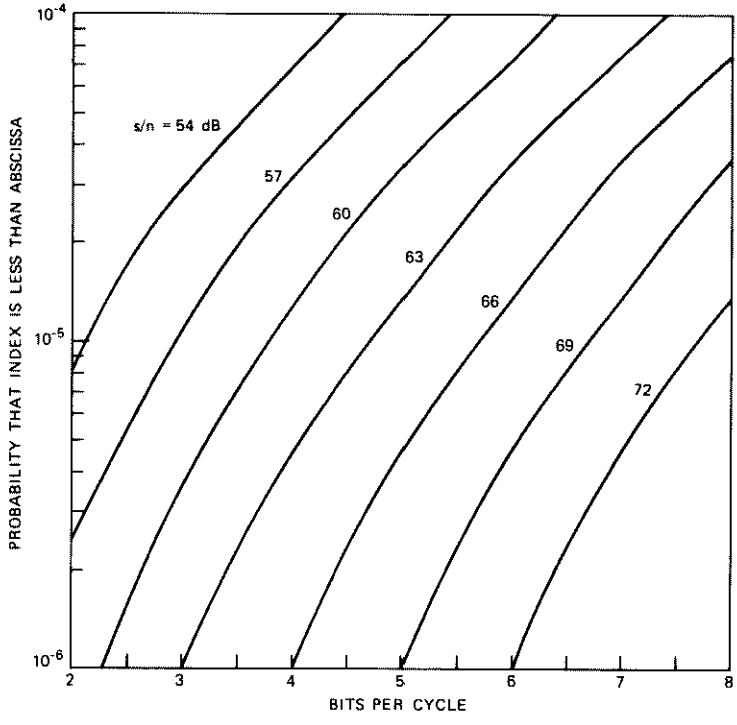



Fig. 6—Index distribution tail sensitivity to s/n for linear equalization, p.e. $< 10^{-4}$.

adapted in real time, the performance benefit would be negligible.

To understand why it is not worthwhile to optimize the transmitted power spectral density, we first consider the information theory limit that was anlayzed in Section 4.7. Our detailed numerical work has shown that a plot of the tail of the index distribution for the information theory limit under the assumption of an optimized transmitter would be imperceptibly different from the $F_{DF}$ tail plotted in Fig. 2. This closeness of the two distributions follows from the fact that, for the severest fades in our data base of 25,000 channels, $|H|^{-2}$ is of the order of $10^{-6}$ and the terms involving $|H|^{-4}$ ($\sim 10^{-12}$) in the perturbation expression, eq. (23), are not enough to overcome the $10^{-14}$ multiplier.

Since the decision feedback index has the same form as (22), we can also conclude that the distribution tail corresponding to decision feedback would not be significantly altered if the optimum transmitter were used.

The decision feedback index and information theory limit on the index give imperceptible benefits when the power spectral density is optimized; therefore, it seems extremely unlikely that there is any

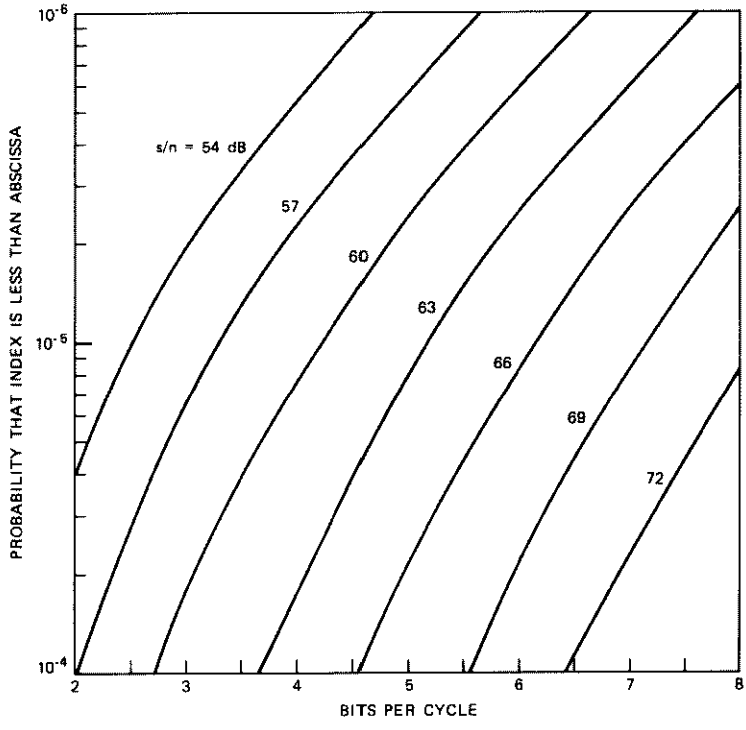



Fig. 7—Index distribution tail sensitivity to s/n for decision feedback, p.e. $< 10^{-4}$.

worthwhile benefit associated with optimizing the transmitter in the other cases.

## IX. INITIAL ESTIMATE OF THE EFFECT OF FREQUENCY DIVERSITY

In implementations, digital radio systems are often protected with frequency diversity. In such systems impairments such as fading and equipment outages prompt the switching of communication traffic to a protection channel situated at a different frequency. The notation $mxn$ means that $m$ protection channels back up $n$ working channels. So long as a protection channel is not occupied by an impaired channel, or is not itself impaired, it is available for temporary use in any of the $n$ working systems. Some illustrations are $2 \times 10$ and $1 \times 11$ at 4 GHz, while at 6 GHz $2 \times 6$ and $1 \times 7$ are examples.

For FM systems the factor expressing the improvement in outage associated with frequency diversity is given by the expression $100/DG \cdot f_0 H^2$ in eq. (24) [corresponding to (34) and (35) in Ref. 21]. The parameter $f_0$ represents the carrier frequency in gigahertz and $D$ denotes the path distance measured in miles. The parameter $G$ depends

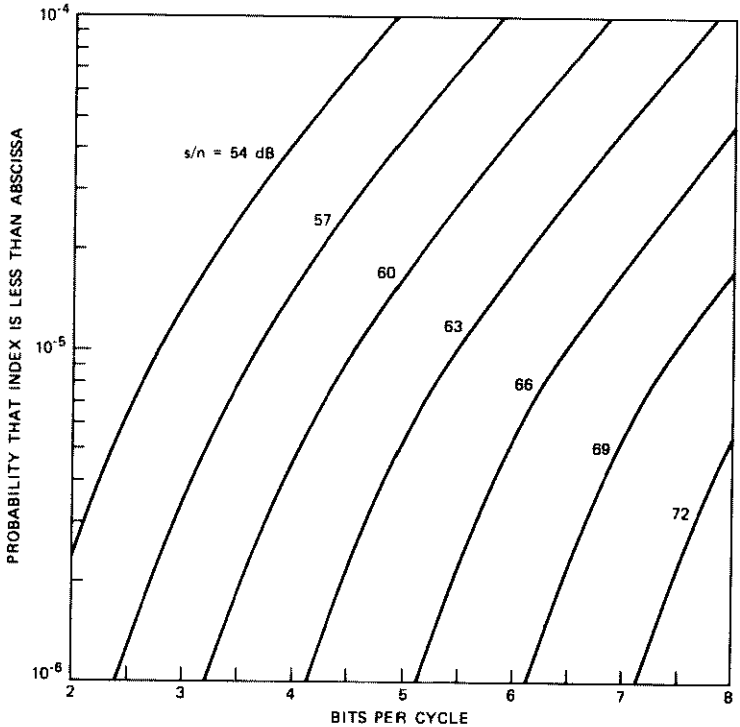



Fig. 8—Index distribution tail sensitivity to s/n for the case of all ISI removed, p.e. $< 10^{-4}$.

on the details of the frequency protection. $G$ incorporates combinatorial effects corresponding to using $m$ channels to back up $n$ as well as empirical expressions involving the individual frequencies of channels involved. The term $H$ is commonly expressed in decibels as $-20 \log H$ and is called fade margin. The fade margin is the smallest loss relative to the unfaded received signal at which the system fails. The notation $H$ for the voltage level agrees with the previous use of $H$ in this paper so long as the channel has a flat characteristic.

As pointed out in Ref. 22 the notation of a flat fade margin is considered meaningless in digital radio systems since the frequency-selective aspect of the fade characteristic appears necessary to describe performance of a channel. As of this writing we are not aware of any method in the extant literature for extending our results to include the effect of frequency diversity. However, for the special case of optimum linear equalization there is a way to introduce an equivalent flat fade margin so as to enable the use of (24) in making a (preliminary) estimate of the diversity effect. The estimation method was discovered in the course of generalizing the computer program to compute index

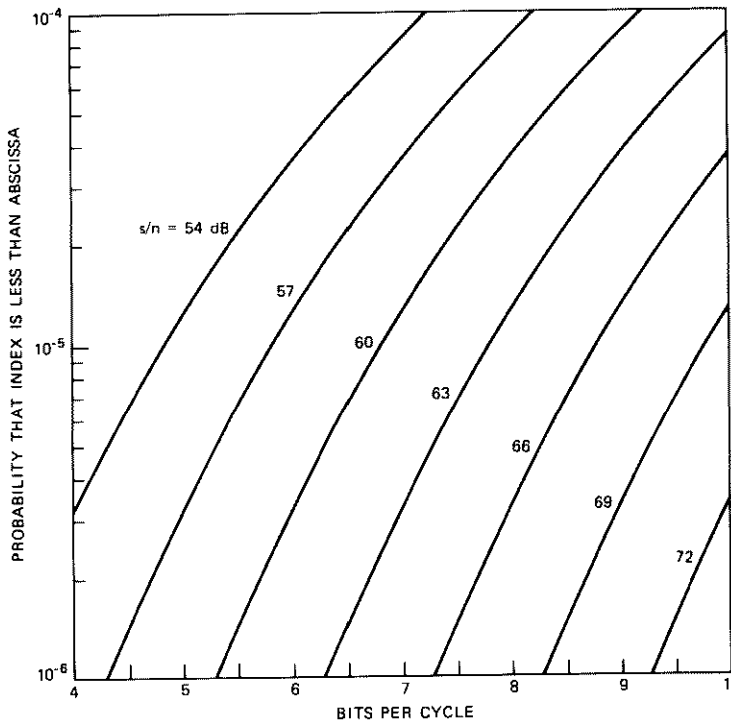



Fig. 9—Distribution tail sensitivity to s/n for the Shannon limit.

distributions for arbitrary bandwidths. The generalization was needed to investigate—and, as it turned out, to substantiate—the bandwidth insensitivity of the distribution in the 20- to 40-MHz range. The generalized program was also exercised for bandwidths an order of magnitude smaller and it was observed that the $F_{LIN}$ distribution changed imperceptibly.* So the $F_{LIN}$ tail in the range of interest here can be correctly obtained by using the univariate samples $|H(0)|^2$. Treating (18) as an equality and solving for $|H(0)|^2$ and then substituting in (24) gives an estimate of the improvement owing to frequency diversity.

For an illustration refer to Fig. 2 for which s/n = 63 dB and $Pe < 10^{-4}$. We see that with linear equalization, at the long-haul outage objective of $1.3 \times 10^{-6}$, 3.2 bits per cycle can be supported and the corresponding number for the short-haul objective is 5.3 bits per cycle. Using (24) and the $G$ values from Ref. 21 we show in Table I the

* This is not true of the individual values of $I$ and no claim is made for invariance of $F_{DF}$, $F_{MF}$, or $F_{IT}$.

Table I—Estimates of improved bits/cycle indices using frequency diversity

| | Channel at 4 GHz | | Channel at 6 GHz | |
|---|---|---|---|---|
| Short Haul | (1 × 11) | 7.1 | (1 × 7) | 6.5 |
| | (2 × 10) | 7.8 | (2 × 6) | 7.1 |
| Long Haul | (1 × 11) | 5.3 | (1 × 7) | 4.6 |
| | (2 × 10) | 5.8 | (2 × 6) | 5.3 |

following estimates of improved indices when frequency diversity is employed.

## X. ACKNOWLEDGMENT

Discussions with N. Amitay, A. Gieger, L. J. Greenstein, V. K. Prabhu, B. G. King, G. Vannucci, A. Vigants, C. B. Woodworth, W. D. Rummler and Y. S. Yeh were valuable in the course of the work reported here.

## XI. POSTSCRIPT

Application of multilevel QAM in the radio channels might be inhibited by the amplitude (AM-AM) and (AM-PM) nonlinearities present in RF power amplifiers. A method for solving this problem without sacrificing amplifier power efficiency will be described in a forthcoming paper.[23]

## REFERENCES

1. W. D. Rummler, "A New Selective Fading Model: Application to Propagation Data," B.S.T.J., *58,* No. 5 (May–June 1979), pp. 1032–71.
2. W. D. Rummler, "More on the Multipath Fading Channel Model," IEEE Trans. Commun., *COM-29,* No. 3 (March 1981), pp. 346–52.
3. A. Gersho and V. B. Lawrence, unpublished work.
4. G. Ungerbock, "Channel Coding with Multilevel/Phase Signals," IEEE Trans. Inform. Theory, *IT-28,* No. 1 (January 1982), pp. 55–67.
5. B. R. Saltzberg, "Intersymbol Interference Error Bounds with Application to Ideal Bandlimited Signaling," IEEE Trans. Inform. Theory, *IT-14,* No. 4 (July 1968), pp. 563–8.
6. M. S. Mueller and J. Salz, "A Unified Theory of Data Aided Equalization," B.S.T.J., *60,* No. 9 (November 1981), pp. 2023–38.
7. R. D. Gitlin and S. B. Weinstein, "Fractionally Spaced Equalization: An Improved Digital Transversal Equalizer," B.S.T.J., *60,* No. 2 (February 1981), pp. 275–96.
8. R. D. Gitlin and J. E. Mazo, "Comparison of Some Cost Functions for Automatic Equalization," IEEE Trans. Commun., *COM-21,* No. 3 (March 1973), pp. 233–7.
9. T. Berger and D. W. Tufts, "Optimal Pulse Amplitude Modulation, Part I, Transmitter-Receiver Design and Bounds from Information Theory," IEEE Trans. Inform. Theory, *IT-13,* No. 2 (April 1967), pp. 196–208.
10. J. Salz, "Optimum Mean-Square Decision Feedback Equalization," B.S.T.J., *52,* No. 8 (October 1973), pp. 1341–73.
11. D. D. Falconer and G. J. Foschini, "Theory of Minimum Mean-Square-Error QAM Systems Employing Decision Feedback Equalization," B.S.T.J., *52,* No. 10 (December 1973), pp. 1821–48.
12. S. P. Lloyd, "On a Measure of Stochastic Dependence," Theory of Probability and Its Application, No. 7, 1962, pp. 312–22.

13. M. S. Pinsker, "A Quantity of Information of a Gaussian Random Stationary Process, Contained in a Second Process Connected with it in a Stationary Manner," Doklady Akad. Nank S.S.S.R., New Series, *99*, 1954, pp. 213–16.
14. R. K. Mueller and G. J. Foschini, "The Capacity of Linear Channels with Additive Gaussian Noise," B.S.T.J. *49*, No. 1 (January 1970), pp. 81–94.
15. J. L. Holsinger, "Digital Communication over Fixed Time-Continuous Channels with Memory-with Special Application to Telephone Channels," Lincoln Laboratory, Technical Report 366, Lexington, MA, October 1964.
16. J. B. Thomas, "Statistical Communication Theory," New York: John Wiley and Sons, 1969, Chapter 8.
17. R. M. Fano, "Transmission of Information," New York: M.I.T. Press and John Wiley and Sons, Inc., 1961.
18. B. G. King, unpublished work.
19. M. A. Rezny and J. S. Wu, unpublished work.
20. B. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of PCM," Proc. IEEE (November 1948), pp. 1324–31.
21. A. Vigants and M. V. Pursley, "Transmission Unavailability of Frequency—Protected Microwave FM Radio Systems Caused by Multipath Fading," B.S.T.J., *58*, No. 8 (October 1979), pp. 1779–96.
22. A. Gieger and W. T. Barnett, "Effects of Multipath Propagation on Digital Radio," IEEE Trans. Commun., *COM*-29, No. 9 (September 1981), pp. 1345–52.
23. A. A. M. Saleh and J. Salz, "Adaptive Linearization of Power Amplifier Nonlinearity in Digital Radio Systems," B.S.T.J., *62*, No. 4 (April 1983).

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 2, February 1983
*Printed in U.S.A.*

# Chromatic Dispersion Measurements in Single-Mode Fibers Using Picosecond InGaAsP Injection Lasers in the 1.2- to 1.5-μm Spectral Region

By C. LIN, A. R. TYNES, A. TOMITA, P. L. LIU and D. L. PHILEN

(Manuscript received September 23, 1982)

*We describe the use of picosecond InGaAsP injection lasers for measuring chromatic dispersion in single-mode fibers in the 1.2- to 1.5-μm spectral region. Injection lasers at various wavelengths and a single-mode fiber-to-fiber switch are used in the pulse delay and pulse-broadening measurements. The simplicity and the compactness make the setup useful for field measurements and quality control.*

## I. INTRODUCTION

Modal and chromatic dispersion measurements in multimode and single-mode fibers provide important information about the bandwidth limitations in optical fiber transmission. A near-infrared, fiber Raman laser with subnanosecond pulses in the 1- to 1.7-μm region can be used for dispersion measurement in both multimode and single-mode fibers.[1-3] While this infrared-fiber, Raman-laser-based measurement system has been very useful and widely adopted, it has its limitations in terms of time resolution (by the mode-locked laser pulsewidth, ~140 ps) and is not suitable for field measurements because of its large size.

In this paper we describe a simpler dispersion measurement system based on picosecond InGaAsP injection lasers and ultrafast InGaAs p-i-n detectors. This is similar to the measurement system we used for measuring high-bandwidth multimode fibers,[4] except now in order to study the chromatic dispersion in single-mode fibers by pulse delay measurements, we need to use injection lasers at different wavelengths in the 1.3-μm spectral region. Besides being more compact, this system also has a better time resolution than the fiber Raman laser setup.

## II. EXPERIMENTAL APPROACH

The experimental approach is straightforward. Figure 1 shows the setup in which several InGaAsP injection lasers with wavelengths in the 1.2- to 1.5-$\mu$m spectral region are used for pulse delay and pulse-broadening measurements in single-mode fibers. A picosecond electrical pulser is used to drive one or two injection lasers at a time to obtain optical pulses of 30 to 80 ps in duration. The laser pulses coming out of single-mode fiber pigtails are selected by a low-loss, single-mode, fiber-to-fiber switch[5] and sent through the test fiber, the output of which is detected with a pigtailed InGaAs fast-pin photodiode.[6] The optical pulsewidth and wavelength-dependent pulse delay information can be stored in the digital oscilloscope for further processing.

The InGaAsP injection lasers used are supplied by Lasertron Co. and have their center wavelengths near 1.21, 1.26, 1.315, 1.335, and 1.525 $\mu$m when pulsed to give short optical pulses. The technique of generating short optical pulses is that of gain-switching,[7,8] which manifests itself as controlled relaxation oscillation[9] and requires proper adjustment of the pumping level to give a single, short, optical pulse. Either the short electrical pulses from a comb generator[7] and step-recovery-diode circuit[10] or a high radio frequency (RF) sinusoidal drive[11] can be used as the pump, except it has been shown that for low repetition frequencies (a few hundred MHz or less), short electrical pulse pumping results in shorter optical pulses.[9] The laser spectral width is typically ~7 nm owing to multi-longitudinal-mode oscillation in the short-pulse generation.[7,8]

Figures 2a and b illustrate a typical measurement result obtained with a 6-km-long single-mode test fiber. The laser pulses at 1.335 and 1.315 $\mu$m are selected with the single-mode fiber switch and are sent

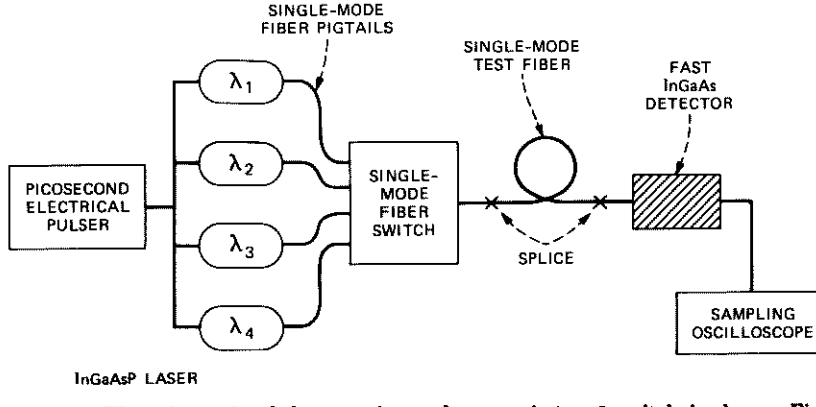



Fig. 1—The schematic of the experimental setup. A 4 × 1 switch is shown. Five injection lasers are actually used in our preliminary measurements. The fifth laser has a connectorized fiber pigtail for direct connection to the test fiber.

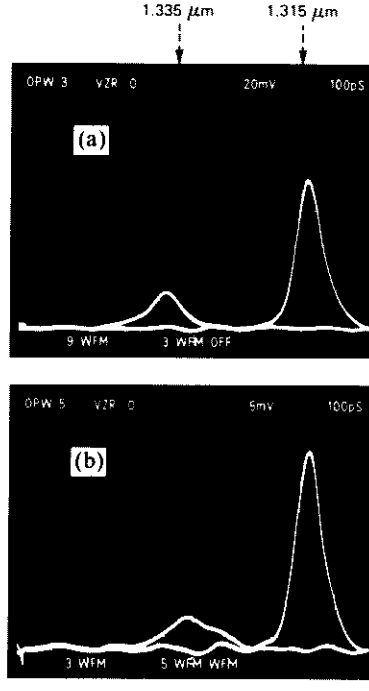



Fig. 2—The pulse pair at 1.335 μm and 1.315 μm (a) before and (b) after the 6-km-long single-mode fiber.

through the test fiber. The pulses before and after the fiber as recorded and stored are shown in Figs. 2a and b, respectively. These are both doubly exposed oscilloscope pictures showing the relative pulse delay change and pulsewidth change owing to dispersion in the test fiber. Note that the pulse at 1.315 μm being near the minimum dispersion wavelength, $\lambda_0$, experiences no broadening, while the pulse at 1.335 μm shows considerable broadening owing to chromatic dispersion in the fiber. The display is adjusted to show the relative delay difference between the two optical pulses before and after the test fiber. The change in the delay difference ($\Delta\tau$) is ~60 ps, with the 1.335-μm pulse experiencing more delay because the 1.315-μm pulse lies much closer to the minimum dispersion wavelength, $\lambda_0$. Figures 3a and b show similar results for the pulse pair at 1.21 and 1.315 μm. The 1.21-μm pulse has experienced more delay and broadening than does the 1.315-μm pulse. In time, the 1.21-μm pulse falls behind the 1.315-μm pulse after the 6-km fiber, even though it is ahead of the latter by more than 2 ns. The relative delay change $\Delta\tau$ is 3.1 ns.

Similar wavelength-dependent pulse-delay change and pulse-broadening results are obtained for the pulses at 1.26 and 1.525 μm with respect to 1.315 μm. The experimental results $\Delta\tau(\lambda)$ (in ns/km) are

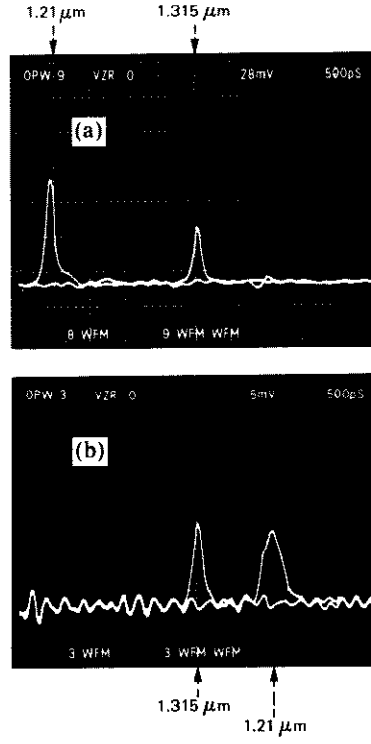



Fig. 3—The pulse pair at 1.21 μm and 1.315 μm (a) before and (b) after the 6-km-long single-mode fiber.

plotted as dots in Fig. 4a. The solid line is a third-order, Chebyshev polynomial fit ($F(\lambda) = a + b\lambda + c\lambda^2 + d\lambda^3$) on an *HP*-85 computer. The derivative of this fitted polynomial is a second-order polynomial whose root (zero-crossing point) gives $\lambda_0$, the minimum chromatic dispersion wavelength. Figure 4b plots the obtained chromatic dispersion $M(\lambda)$ in ps/nm-km. The obtained $\lambda_0$ is ~1.314 μm ± 0.004 μm, in reasonable agreement with the 1.309-μm value measured with the fiber Raman laser setup.

## III. DISCUSSION AND SUMMARY

Compared with the fiber Raman laser setup, the combination of the picosecond injection lasers and the single-mode fiber switch provides a simple, compact, measurement setup for single-mode fiber chromatic dispersion measurements. In addition, the setup has a better time resolution owing to the short pulse duration and the jitter-free characteristics. While it has a lower dynamic range than the fiber Raman laser, with its 10- to 15-dB dynamic range, typical single-mode

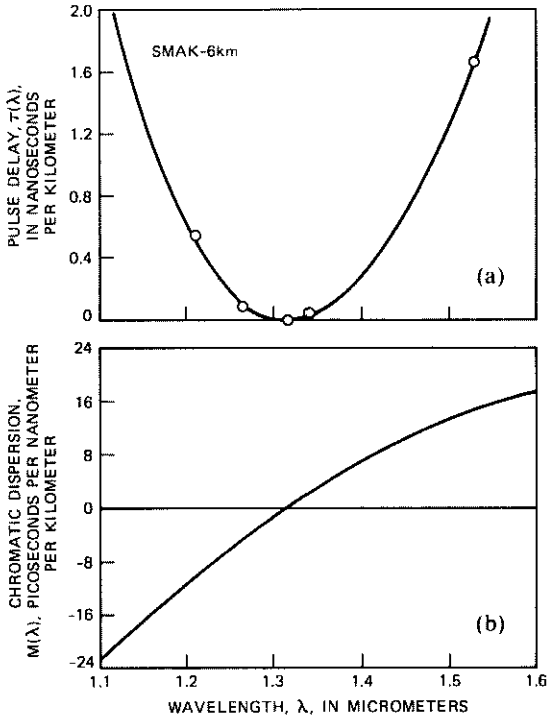



Fig. 4—(a) The time delay vs. wavelength data (dots) with respect to the reference wavelength 1.315 μm. The solid curve is the fitted third-order polynomial. (b) Chromatic dispersion obtained by differentiating the fitted polynomial of (a).

fiber lengths of 5 to 10 km can be measured over the 1.2- to 1.5-μm spectral range, and more than 20 km can be measured over the low-loss region close to 1.3 and 1.55 μm. In practice, the discrete wavelength coverage limits the wavelength resolution. For studying fibers with new or unconventional dispersion characteristics over a wide spectral range, more injection lasers are needed (e.g., 10) for more wavelength coverage. In single-mode fibers for which the design phase has passed and a specific design concerning chromatic dispersion is established, this setup is useful for quality control and field testing by measuring the pulse delay and pulse broadening at a number of selected wavelengths.

## IV. ACKNOWLEDGMENTS

We would like to thank R. Knerr and S. Kaufman for supplying the single-mode fiber switch, W. C. Young and L. Curtis for the single-mode fiber connectors, and P. F. Glodis for providing the single-mode fibers.

## REFERENCES


1. L. G. Cohen, P. Kaiser, and C. Lin, "Experimental techniques for evaluation of fiber transmission loss and dispersion," Proc. IEEE, *68* (October 1980), pp. 1203–9.
2. W. F. Love, "Bandwidth spectral characterization of $G_eO_2$-$P_2O_5$-$SiO_2$ multimode optical waveguides," Digest of 6th European Conference on Optical Communication, York, U.K., 1980, pp. 113–15.
3. M. Horiguchi, Y. Ohmori, and H. Takata, "Profile dispersion characteristics in high bandwidth optical fibers," Appl. Opt., *19* (September 1980), pp. 3159–69.
4. C. Lin, P. L. Liu, T. P. Lee, C. A. Burrus, F. T. Stone, and A. J. Ritger, "Measuring high bandwidth fibers in the 1.3 $\mu$m region with picosecond InGaAsP injection lasers and ultra fast InGaAs detectors," Electron. Lett., *17* (June 1981), pp. 438–40.
5. W. C. Young and L. Curtis, "Cascaded multipole switches for single-mode and multimode optical fibers," Electron. Lett., *17* (August 1981), pp. 571–3.
6. T. P. Lee, C. A. Burrus, K. Ogawa, and A. G. Dentai, "Very-high-speed back-illuminated InGaAs/InP pin punch-through photodiodes," Electron. Lett., *17* (June 1981), pp. 431–2.
7. C. Lin, P. L. Liu, T. C. Damen, D. J. Eilenberger, and R. L. Hartman, "Simple picosecond pulse generation scheme for injection lasers," Electron. Lett., *16* (July 1980), pp. 600–2.
8. P. L. Liu, C. Lin, I. P. Kaminow, and J. J. Hsieh, "Picosecond pulse generation from InGaAsP lasers at 1.25 and 1.3 $\mu$m by electrical pulse pumping," IEEE J. Quantum Electron., *QE-17* (May 1981), pp. 671–4.
9. C. Lin, unpublished work.
10. P. T. Torphammer and S. T. Eng, "Picosecond pulse generation in semiconductor lasers using resonance oscillation," Electron. Lett., *16* (July 1980), pp. 587–9.
11. H. Ito, H. Yokoyama, S. Murata, and H. Inaba, "Generation of picosecond optical pulses with highly rf modulated AlGaAs DH laser," IEEE J. Quantum Electron., *QE-17* (May 1981), pp. 663–70.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 2, February 1983
*Printed in U.S.A.*

# High-Frequency Impedance of Proton-Bombarded Injection Lasers

By B. W. HAKKI, W. R. HOLBROOK, and C. A. GAW

(Manuscript received August 11, 1982)

*Experimental and theoretical results are given of an investigation of the capacitance behavior with frequency of GaAs injection lasers. It is shown that, for shallow-bombarded stripe geometry lasers, the zero-bias capacitance decreases rapidly beyond a certain frequency. This is interpreted in terms of confinement of the low-level radio frequency current under the stripe at high frequencies. Comparison of the experimental results with the analytically derived expressions provides a measure of the material resistivity adjoining the active region. This inferred material resistivity is shown to be in good agreement with results obtained by more direct measurements. Finally, the general conclusions are also applicable to other optoelectronic devices operating at high frequency, such as light-emitting diodes.*

## I. INTRODUCTION

Injection lasers are inherently suited for high-frequency (>50 MHz) applications.[1,2] In the Bell System, the commonly used GaAs double-heterostructure (DH) stripe geometry laser is obtained by either shallow- or deep-proton bombardment.[3,4] For deep bombardment, the active stripe is well defined electrically by the high-resistivity material that confines it. In the case of shallow bombardment, as well as oxide stripe geometry lasers, current can flow laterally, beyond the stripe, through the conductive cladding layer.[5–8]

Previously, the impedances of several laser geometries were measured at high frequencies, and the results interpreted in terms of phenomenological equivalent circuits.[9–13] In this paper, we will restrict the analysis and measurements to zero-biased shallow-bombarded laser diodes, for simplicity. We will show that, for shallow-bombarded stripe geometry lasers, the low-level (radio frequency) RF current at

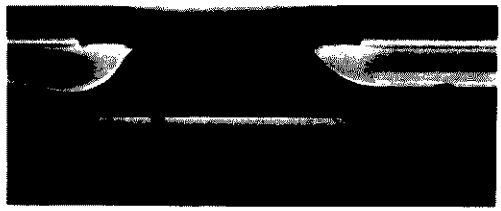
high frequencies is confined under the stripe. The interpretation of measurements in terms of the fundamentally derived analytical expressions provides a useful measure of relevant material properties. Finally, although the main emphasis is on stripe geometry lasers, the analysis is applicable to other optoelectronic devices. Thus, the approach can be applied to predict the small-signal, unbiased impedance of light-emitting diodes (LEDs) and photodetectors operating at high frequencies.

## II. EXPERIMENT

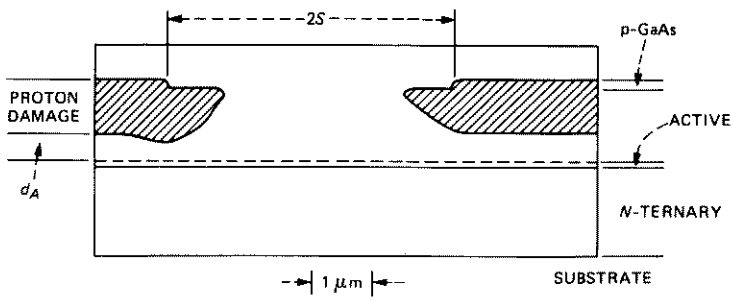
The injection laser is a standard GaAs DH grown by either liquid phase epitaxy (LPE) or molecular beam epitaxy (MBE) technology.[14,15] It consists of 2-$\mu$m-thick N-$Ga_{1-x}Al_xAs$ cladding, 0.1- to 0.2-$\mu$m $Ga_{1-y}Al_yAs$ active, 1.5-$\mu$m P-$Ga_{1-x}Al_xAs$ cladding, and, finally, 0.5-$\mu$m p-GaAs contact layer. Shallow-proton bombardment is used to define stripe widths of either 5- or 10-$\mu$m dimension. For these measurements, it is essential to obtain an accurate measure of the stripe width, 2S, and the distance of separation, $d_A$, between the active region and the semi-insulating damaged region. This is done by etching the mirror facets in a dilute A/B solution.[16] An example of such an etched mirror obtained by a scanning electron microscope (SEM) is shown in Fig. 1a. Figure 1b is a schematic of the various layers. The distance, $d_A$, is between the unetched region and the active layer. It is assumed that the etch stops abruptly where the conductivity changes from p-type to semi-insulating. (In general, this is a fair assumption, although it may not be accurate to better than 0.1 $\mu$m.)

Every diode is subjected to dc current-voltage and ac impedance tests. The dc measurement determines rectification properties, and current-voltage dependence over the range between $10^{-12}$ to $10^{-2}$ amperes. The ac impedance test used RF signal levels, at or below 50 mV, performed on unbiased diodes. At those RF signal levels, the diode conduction current is less than $10^{-11}$ amperes. The laser impedance is measured at zero bias over a frequency range extending from $10^4$ to $5 \times 10^7$ Hz. Over this frequency range the capacitive current is typically larger than $10^{-6}$ amperes. Hence, in the present impedance measurements, the junction capacitive current is well over $10^5$ times greater than the diode conduction current. Therefore, for our purposes, the junction can be assumed to be a capacitance with negligible rectified current flow.

Usually, two capacitance bridges are used at different frequency ranges, with an overlap in frequency to ensure accuracy. These impedance bridges measure the equivalent parallel capacitance and conductance of the diode. As an example, Fig. 2 shows the results of two such measurements and two experimental values when results were



(a)




(b)

Fig. 1—Mirror of shallow-bombarded stripe geometry DH GaAs layer after etching in dilute A/B etch. (a) Photomicrograph of the etched mirror as obtained by an SEM. In this case, the p-active layer has also been etched. (b) Layer schematic of the same mirror.

obtained on two different instruments. In Fig. 2a, the zero bias capacitance of diode A14 (grown by LPE) is constant as a function of frequency up to 1 MHz. Above 1 MHz, the equivalent parallel circuit capacitance decreases rapidly with an increase in frequency. The same general behavior is observed on another diode from a different slice. The results of diode B22 (grown by MBE) are shown in Fig. 2b. The zero bias capacitance decreases abruptly for frequencies in excess of $10^5$ Hz. We also find that, for both diodes, the equivalent parallel circuit conductances $G$ increase with frequency $f$ according to the relation $G \propto f^n$, where $n$ is between 1.25 and 1.5. However, accurate measurements of conductance are more difficult to obtain than those of capacitance.

The results of Fig. 2 show the abrupt drop in capacitance at frequencies that may differ by as much as an order of magnitude. Furthermore, some devices exhibit a more gradual reduction in the capacitance-frequency behavior.[17] Figure 3 shows an example of such a soft roll-off in the measured capacitance. The solid curve is the modified analysis, which accounts for asymmetry and metallization capacitance. The dashed theoretical curve is for the ideal symmetric

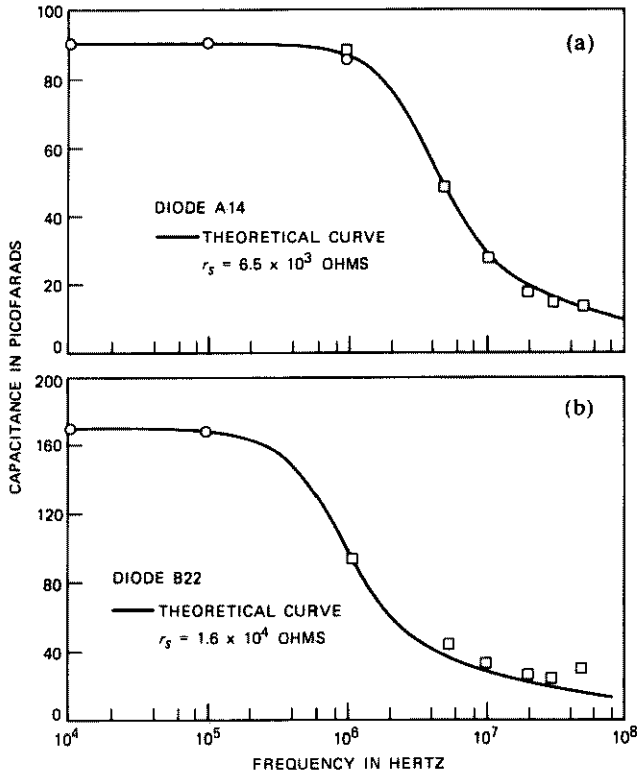



Fig. 2—Capacitance dependence on frequency for (a) diode A14 and (b) diode B22. The solid curves are obtained from analytical expressions. In (a) and (b), $d_A$ is 0.6 and 0.21 μm, respectively.

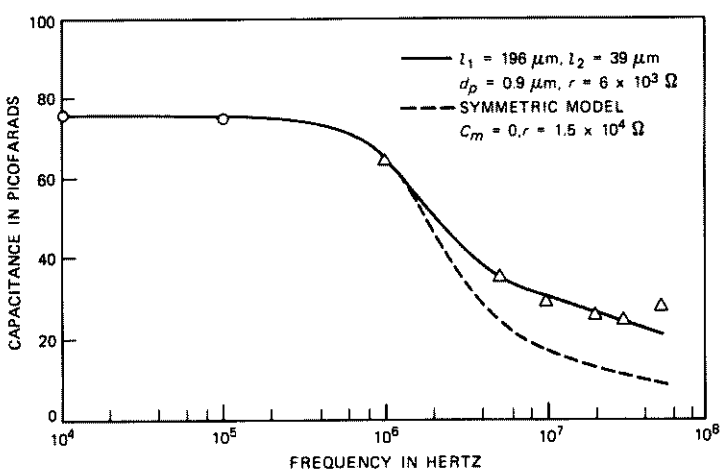



Fig. 3—Capacitance dependence on frequency for a severely eccentric stripe geometry device.

model. The distance between the stripe and the two edges, $l_1$ and $l_2$, respectively, as well as the thickness of the damaged region, $d_p$, are obtained from SEM measurements. Here, the high-frequency capacitance does not decrease as rapidly as that for the cases shown in Fig. 2.

## III. ANALYSIS

### *3.1 The ideal symmetric case*

Consider first the simple and ideal case of a stripe centered within the chip. In addition, assume that the proton-damaged region is sufficiently thick (greater than 2.5 micrometers), so that the only relevant capacitance is the junction capacitance. The RF current flows through the diode as shown schematically in Fig. 4a. There is a direct stripe capacitance, $C_s$, defined by the width $2S$ and length $L$ of the stripe. The RF current flows through the adjoining cladding layer, which, together with the junction, forms a transmission line. This transmission line is shown schematically in Fig. 4b. It consists of a distributed resistance, $r$, and a distributed capacitance, $c$. The analysis of such a transmission line is straightforward. The cladding sheet resistance, $r_s$, is equal to $\rho/d_A$, where $\rho$ is the average material resistivity. The capacitance per unit area is $c_a$ given by $C_0/A$, where $C_0$ is the low-frequency total diode capacitance and $A$ is the total diode area. In the equivalent distributed parameter circuit shown in Fig. 4b, the resistance per unit length $r$ is equal to $r_s/L$. Similarly, the capacitance per unit length $c$ is equal to $c_aL$. At any point $x$, the current and voltage are given, respectively, by

$$i(x, t) = I(x)e^{j\omega t} \tag{1}$$

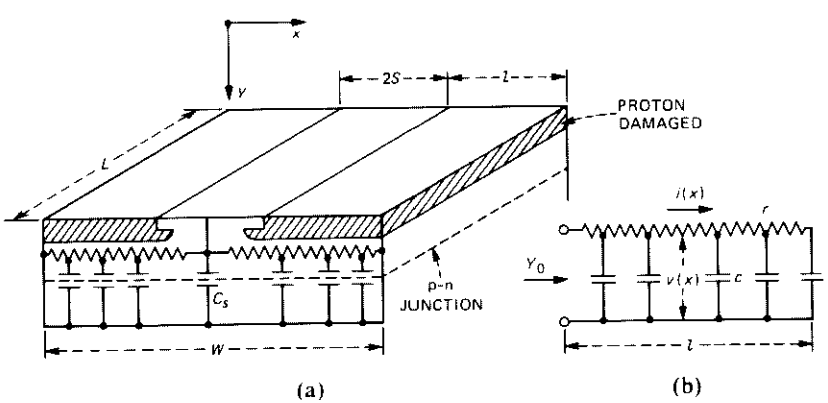



Fig. 4—Schematic of shallow-proton-bombarded, ideally symmetric, stripe geometry laser and equivalent circuit. (a) The various layers and the electrical elements. (b) The distributed parameter transmission line underneath the proton-bombarded region.

and

$$v(x, t) = V(x)e^{j\omega t}. \tag{2}$$

The spatial parts of the current and voltage are related by the equations

$$\frac{dV}{dx} = -rI \tag{3}$$

and

$$\frac{dI}{dx} = -j\omega cV. \tag{4}$$

These are almost standard transmission line equations that have to be solved with the boundary condition $I(\ell) = 0$, where $\ell = (W/2)\text{-}S$. This open-circuit boundary condition is appropriate when surface leakage is negligible. The input admittance, $Y_0$, is given by

$$Y_0 = G_0 + jB_0, \tag{5}$$

where

$$G_0 = \frac{\phi}{2r\ell}\frac{\sinh\phi - \sin\phi}{\cosh\phi + \cos\phi}, \tag{6a}$$

$$B_0 = \frac{\phi}{2r\ell}\frac{\sinh\phi + \sin\phi}{\cosh\phi + \cos\phi} \tag{6b}$$

and

$$\phi = \ell\sqrt{2\omega rc}. \tag{6c}$$

From the susceptance given in (6b), an equivalent distributed parallel capacitance, $C_d$, is obtained in the form

$$\frac{C_d}{C_0'} = \frac{1}{\phi}\frac{\sinh\phi + \sin\phi}{\cosh\phi + \cos\phi}, \tag{7}$$

where $C_0' = c_a\ell L$ is the low-frequency capacitance of the junction whose area is $\ell L$. A plot of $C_d/C_0'$ is shown in Fig. 5 as a function of $\phi$. It is seen that the equivalent distributed parallel capacitor value remains equal to the low-frequency value as long as $\phi < 1$. However, for $\phi > 1$, the capacitance decreases at the rate of $1/\phi$. This decrease at high frequency is due to the fact that as the frequency increases, a relatively larger fraction of the reactive current is shunted by the distributed capacitance at small distances. At sufficiently high frequencies most of the current is bypassed at small distance by the capacitance.

The choice of a parallel circuit, unfortunately, does not provide a straightforward connection between conductance and the material

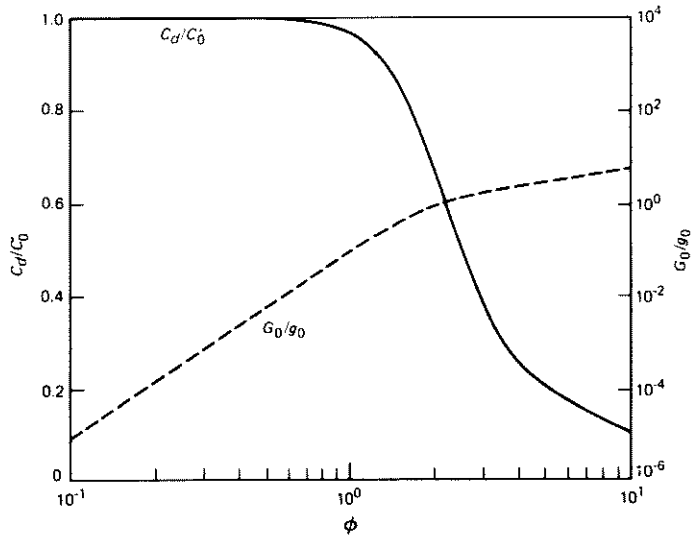



Fig. 5—Equivalent parallel circuit capacitance and conductance for the distributed transmission line of Fig. 4b. The variation is in terms of the normalized parameter, $\phi$, defined in the text.

resistivity. Nevertheless, let $g_0 = Ld_A/\rho\ell$, which is the dc conductance between $x = 0$ and $\ell$. Then, $G_0/g_0$ is plotted as a function of $\phi$ in Fig. 5. It is seen that the normalized conductance increases as $\phi^4$. Since $\phi$ varies as $f^{1/2}$, it follows that the conductance increases as $f^2$, which is somewhat faster than the experimentally observed variation.

The above results can be applied to the ideally symmetric stripe geometry configuration of Fig. 4a. The total equivalent parallel capacitor of the diode $C_p$, comprising the direct stripe and distributed capacitances, respectively, is given by

$$C_p = C_0\left[\frac{A_s}{A} + \left(1 - \frac{A_s}{A}\right)\frac{\sinh\phi + \sin\phi}{\phi(\cosh\phi + \cos\phi)}\right], \tag{8}$$

where $A_s = 2SL$ is the area of the stripe, $C_0$ is the total capacitance of the diode at low frequency, and the rest of the parameters are as defined previously.

Figure 2a shows a plot of eq. (8), with a sheet resistance chosen at a value of $6.5 \times 10^3$ ohms. The stripe width is 5 $\mu$m, the diode width is 250 $\mu$m, and the length is 380 $\mu$m. The stripe width is obtained by etching the mirror, as in Fig. 1. The only adjustable parameter in the curve of Fig. 2a is the sheet resistance. A similar theoretical curve is plotted in Fig. 2b for diode B22. Here, the stripe width is 10 $\mu$m, and the sheet resistance is $1.6 \times 10^4$ ohms. It is seen that the capacitance in Fig. 2b starts to decrease at a lower frequency because of the higher

sheet resistance. The high sheet resistance in diode B22 is due to the smaller distance of separation between the damaged region and the active layer.

### *3.2 Asymmetric stripe with metallization capacitance*

Two deviations from the earlier symmetric model are observed in some devices. First, the stripe can be eccentric. Second, the proton-damaged layer can be as thin as 0.5 micrometers. These two factors cause the high-frequency behavior to deviate from the ideal model. This has been noted in independent measurements by C. W. Thompson et al.[17]

The metallization can contribute a significant additional capacitance if the metallization is separated from the conducting semiconductor by a thin insulating layer. Such a condition can exist in oxide stripe geometry lasers as well as in some proton-bombarded lasers. However, this capacitance has unique characteristics, since it is not operative at low frequencies, where the electrodes are virtually connected. But at high frequencies, charge storage between the metal and the semiconductor can take place. A schematic of the various layers is shown in Fig. 6a. The proton-damaged layer thickness is $d_p$. This semi-insulating layer can, at high frequencies, contribute a metallization capacitance $C_m$ given approximately for GaAs by

$$C_m \approx 10/d_p \qquad pF, \tag{9}$$

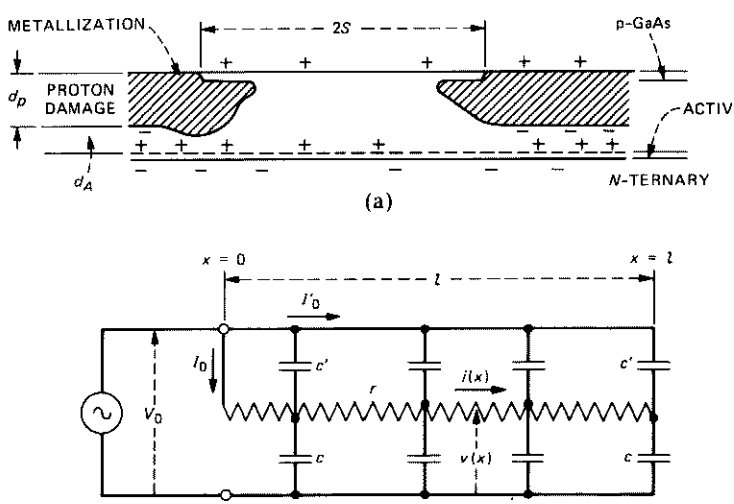



Fig. 6—Schematic of stripe geometry with thin damaged layer. (a) Charge buildup. (b) Equivalent transmission line circuit. The metallization capacitance results from charge storage across the proton-bombarded region.

where $d_p$ is measured in micrometers, and the chip is 250 × 380 μm, which is typical for our diodes. For oxide-defined stripe geometry lasers, the metallization capacitance is associated with the oxide layer itself.

The metallization capacitance is not effective at low frequencies. This is obvious from Fig. 6a, where conduction through the stripe region bypasses the charge across the damaged region. The equivalent transmission line circuit is shown schematically in Fig. 6b.[18] The circuit parameters are as defined previously, and solution of the transmission line equations follows the usual procedures. The total current branches into two components, $I_0$ and $I_0'$, respectively. The two admittances are given by

$$G_0' = \frac{\omega c c' \ell}{(c + c')} \left[ \frac{\sinh \phi - \sin \phi}{\phi(\cosh \phi + \cos \phi)} \right] \tag{10a}$$

$$G_0 = \omega c \ell \left[ \frac{\sinh \phi - \sin \phi}{\phi(\cosh \phi + \cos \phi)} \right] \tag{10b}$$

$$B_0' = \frac{\omega c c' \ell}{(c + c')} \left[ 1 - \frac{\sinh \phi + \sin \phi}{\phi(\cosh \phi + \cos \phi)} \right] \tag{10c}$$

$$B_0 = \omega c \ell \frac{(\sinh \phi + \sin \phi)}{\phi(\cosh \phi + \cos \phi)}, \tag{10d}$$

where

$$\phi = \ell \sqrt{2\omega r(c + c')},$$

$$c' = C_m / W, \tag{10e}$$

and $W$ is the total width of the chip.

Since the two admittances are in parallel, the equivalent parallel circuit capacitance is

$$C = \frac{C_0}{C_0 + C_m} \left[ C_0 \frac{\sinh \phi + \sin \phi}{\phi(\cosh \phi + \cos \phi)} + C_m \right], \tag{11}$$

where $C_0$ is the low-frequency junction capacitance, and $C_m$ is the metallization capacitance, as given in (9). It is clear from (11) that at low frequency, the $\phi$ function goes to one, and the capacitance becomes equal to the low-frequency junction capacitance. On the other hand, at high frequency, the $\phi$ function goes to zero, and the capacitance becomes equal to the metallization and the junction capacitances in series.

When the stripe is located off-center, the capacitance can be calculated in a straightforward manner. The total chip capacitance becomes

$$C_T = C_0 \left\{ \frac{A_s}{A} + \left( \frac{1 - A_s/A}{C_0 + C_m} \right) \left[ \frac{C_0}{(\ell_1 + \ell_2)} (\ell_1 F_1 + \ell_2 F_2) + C_m \right] \right\}, \tag{12}$$

where

$$F_i = \frac{\sinh \phi_i + \sin \phi_i}{\phi_i(\cosh \phi_i + \cos \phi_i)}, \tag{13}$$

$$\phi_i = \ell_i \sqrt{2\omega r(c + c')}, \tag{14}$$

$$\ell_1 + \ell_2 + 2S = W, \tag{15}$$

$\ell_1$ and $\ell_2$ are the lengths of the two distributed line sections, respectively, $2S$ is the stripe width, $A_s$ is the total area of the stripe, and $A$ is the area of the diode.

An example of a case of extreme asymmetry is shown in Fig. 3. Measurement of eccentricity indicates that the stripe is displaced by $\approx 70$ $\mu$m from its center position. The measured thickness of the proton-damaged region is 0.9 $\mu$m, which results in a metallization capacitance of 11 $pF$. When the expression for the asymmetric case, as given in (12), is applied to the results of Fig. 3, the agreement between theory and experiment is adequate for a sheet resistance of $6 \times 10^3 \Omega$. On the other hand, the symmetric model, shown as a dashed curve in Fig. 3, cannot fit the data over the extended frequency range. In addition, the symmetric model overestimates the sheet resistance.

### 3.3 Material resistivity

Having obtained a value of sheet resistance and layer thickness, $d_A$, the material resistivity can be derived. However, care should be exercised in accounting for proton damage spread into the conductive layer.

If $y = 0$ is the point at the edge of the damaged region where the local carrier concentration is (mathematically) zero, then for $y > 0$, the carrier concentration does not recover abruptly as a step function. Instead, the carrier concentration actually recovers as an error function of distance.[19] For our purposes, this recovery can be represented approximately by the relation

$$p(y) = p_0(1 - e^{-y/y_0}), \tag{16}$$

where $p_0$ is the background-free hole concentration, and $y_0$ determines the rate at which carrier removal decreases with distance.

For current flow parallel to the junction plane, the average resistivity $\rho$ over a thickness $d_A$ is

$$\rho = \rho_0 / \left\{ 1 - \frac{y_0}{d_A} \left[ 1 - \exp(-d_A/y_0) \right] \right\}, \tag{17}$$

where $\rho_0$ is the resistivity of the undamaged material. From published data of proton-bombarded GaAs, it would seem that the value of $y_0$ is $0.13 \mp .03$ $\mu$m.[19–21]

The above equations can now be applied to the results of Fig. 2. For diode A14 we obtain $\rho = 0.39$ Ω-cm; and $\rho_0 = 0.3$ Ω-cm. For diode B22, we obtain $\rho = 0.34$ Ω-cm, and $\rho_0 = 0.17$ Ω-cm. Several diodes from these slices were measured. The results were averaged and given in Table I. The average resistivity in the cladding, adjacent to the active region, in slice *A* is $0.28 \mp 0.09$ Ω-cm. Direct measurement of resistivity in this material by J. L. Zilko gives a value of 0.21 Ω-cm.[22] Hence, to within statistical diode-to-diode variation, the agreement is adequate. The resistivity of the cladding in slice B, given in Table I, is $0.18 \mp 0.02$ Ω-cm. These resistivities can also be used to estimate the free-hole concentration. For the typical 40-percent aluminum in the ternary, the hole mobility should be about 100 $cm^2/Vs$.[23,24] The deduced-hole concentrations for slices *A* and *B* are listed in Table I. To check these deduced values with direct-hole concentration measurements, a third slice *C* was evaluated. The hole concentration, in slice *C* in the cladding, as deduced from the capacitance-frequency measurement, was $(2.1 \mp 0.6) \times 10^{17}$ $cm^{-3}$. On the other hand, hole-concentration measurements on this slice, by the automatic feedback method,[21] gave values of $(1.9 \mp 0.4) \times 10^{17}$ $cm^{-3}$.[25] Hence, again the capacitance-frequency method gave values consistent with those obtained by other methods.

Stripe geometry lasers with deep-proton bombardment were also evaluated. In these devices the proton damage extends beyond the active region. Here, as opposed to the results for shallow bombardment, the capacitance remains small and nearly invariant with frequency. This again lends support to the model of RF current confinement in shallow-bombarded stripe geometry lasers.

## IV. CONCLUSION

Experiments conducted on shallow proton-bombarded stripe geometry lasers show that the capacitance decreases abruptly above a certain frequency. These results are interpreted in terms of a distributed parameter transmission line model. It is shown that the frequency beyond which the capacitance decreases depends on: the sheet resistance of the material above the active region, capacitance per unit area,

Table I—Material evaluation obtained from capacitance frequency measurements

| | Slice Average | | | | |
|---|---|---|---|---|---|
| Slice | $r_s$ ohms | $\rho$ ohms-cm | $d_A$ μm | $\rho_0$ ohm-cm | $p_0$ $cm^{-3}$ |
| *A* | $(6 \mp 2) \times 10^3$ | $0.37 \mp 0.12$ | 0.6 | $0.28 \mp 0.09$ | $(2.5 \mp 0.9) \times 10^{17}$ |
| *B* | $(1.5 \mp 0.1) \times 10^4$ | $0.36 \mp 0.03$ | $0.22 \mp 0.02$ | $0.18 \mp 0.02$ | $(3.4 \mp 0.4) \times 10^{17}$ |

and the dimension of the diode. These analytical expressions, when applied to experimental results, allow one to obtain an estimate of the material resistivity adjoining the active region. The deduced values of resistivity and doping concentration are in good agreement with the values obtained by other methods.

Although the analytical results are applied to laser structures, they are also applicable to other optoelectronic devices. For instance, the same general conclusions can also be drawn about the small signal impedance of LEDs operating at high frequency. For configurations other than the simple stripe geometry, the spatial distribution of RF current obviously changes. However, the qualitative results are the same.

## V. ACKNOWLEDGMENT

The authors are indebted to A. E. Bakanowski, H. E. Elder, E. I. Gordon, and C. W. Thompson, Jr., for useful discussions, and to A. Y. Cho and M. C. Tamargo for providing the laser material. The authors are also grateful to J. L. Zilko for providing independent resistivity measurements, and to C. G. Bergey and E. W. Bonato for detailed measurements.

## REFERENCES

1. S. E. Miller, E. A. J. Marcatili, and Tingye Li, "Research Toward Optical-Fiber Transmission Systems—Part I," Proc. IEEE, *61* (December 1973), pp. 1703–26.
2. S. E. Miller, Tingye Li, and E. A. J. Marcatili, "Research Toward Optical-Fiber Transmission Systems—Part II," Proc. IEEE, *61* (December 1973), pp. 1726–51.
3. J. C. Dyment, L. A. D'Asaro, J. C. North, B. I. Miller, and J. E. Ripper, "Proton-Bombardment Formation of Stripe Geometry Heterostructure Lasers for 300°K CW Operation," Proc. IEEE, *60* (June 1972), pp. 726–28.
4. B. W. Hakki, "Carrier and Gain Spatial Profiles in GaAs Stripe Geometry Lasers," J. Appl. Phys., *44* (November 1973), pp. 5021–28.
5. B. W. Hakki, "GaAs Double Heterostructure Lasing Behavior Along the Junction Plane," J. Appl. Phys., *46* (January 1975), pp. 292–302.
6. W. B. Joyce, "Current Crowding and Carrier Confinement in Double Heterostructure Lasers," J. Appl. Phys., *51* (May 1980), pp. 2394–2401.
7. H. Yonezu, I. Sakuma, K. Kobayashi, T. Kamejima, M. Unno, and Y. Nannichi, "A GaAs/$Al_xGa_{1-x}As$ Double Heterostructure Planar Laser," Jpn. J. Appl. Phys., *12* (October 1973), pp. 1585–92.
8. W. P. Dumke, "Current Thresholds in Stripe-Contact Injection Lasers," Solid State Electron., *16* (November 1973), pp. 1279–98.
9. F. K. Reinhart, "Reverse-Biased Gallium Phosphide Diodes as High-Frequency Light Modulators," J. Appl. Phys., *39* (June 1968), pp. 3426–34.
10. K. Aiki, M. Nakamura, T. Kuroda, K. Umeda, R. Ito, N. Chinone, and M. Maeda, "Transverse Mode Stabilized (AlGa)As Injection Lasers with Channeled-Substrate-Planar Structure," IEEE J. Quantum Electron., *14* (February 1978), pp. 89–94.
11. K. Furuya, Y. Suematsu, and T. Hong, "Reduction of Resonance-Like Peak in Direct Modulation Due to Carrier Diffusion in Injection Laser," Appl. Optics, *17* (June 1978), pp. 1949–52.
12. H. Kuwahara, Y. Daido, and H. Furuta, "Measurements of Impedance of DH Semiconductor Lasers," Proc. IEEE, *65* (September 1977), pp. 1412–13.
13. J. M. Dumant, Y. Guillausseau, and M. Monerie, "Small Signal Modulation of DH Laser Diodes: Effect of the Junction Capacitance," Optics Commun., *33* (February 1980), pp. 188–92.

14. L. R. Dawson, "Near-Equilibrium LPE Growth of GaAs-$Ga_{1-x}Al_xAs$ Double Heterostructures," J. Cryst. Growth, *27* (December 1974), pp. 86–96.
15. A. Y. Cho and H. C. Casey, Jr., "GaAs-$Al_xGa_{1-x}As$ Double-Heterostructure Lasers Prepared by Molecular Beam Epitaxy," Appl. Phys. Letters, *25* (September 1974), pp. 288–90.
16. J. C. Campbell, S. M. Abott, and A. G. Dentai, "A Comparison of 'Normal' Lasers and Lasers Exhibiting Light Jumps," J. Appl. Phys., *51* (August 1980), pp. 4010–13.
17. C. W. Thompson, Jr., H. E. Elder, and A. E. Bakanowski, private communication.
18. A. E. Bakanowski, private communication.
19. H. Matsumura, and K. Stephens, "Electrical Measurement of the Lateral Spread of the Proton Isolation Layer in GaAs," J. Appl. Phys., *48* (July 1977), pp. 2779–83.
20. B. R. Pruniaux, J. C. North, and G. L. Miller, "Compensation of n-Type GaAs by Proton Bombardment," Proc. 2nd Int. Conf. on Ion Implantation, edited by I. Ruge and J. Graul, New York: Springer-Verlag, 1971, pp. 212–27.
21. G. L. Miller, "A Feedback Method for Investigating Carrier Distributions in Semiconductors," IEEE Trans. Electron. Dev., *ED-19* (October 1972), pp. 1103–8.
22. J. L. Zilko, and P. J. Anthony, "Conductivity Profiling of GaAs/GaAlAs Multilayer Structures," J. Electrochem. Soc., *128* (April 1981), pp. 871–74.
23. S. Zukotynski, S. Sumski, M. B. Panish, and H. C. Casey, Jr., "Electrical Properties of Ge-Doped p-type $Al_xGa_{1-x}As$," J. Appl. Phys., *50* (September 1979), pp. 5795–99.
24. A. J. SpringThorpe, F. D. King, and A. Becke, "Te and Ge-Doping Studies in $Ga_{1-x}Al_xAs$," J. Electron. Materials, *4* (February 1975), pp. 101–18.
25. B. W. Hakki, "p-GaAs/P-$Ga_{1-x}Al_xAs$ Isotype Heterojunctions in Doubleheterostructure Laser Material," J. Appl. Phys., *52* (October 1981), pp. 6054–58.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 2, February 1983
*Printed in U.S.A.*

# An Analysis of the Derivative Weight-Gain Signal From Measured Crystal Shape: Implications for Diameter Control of GaAs

By A. S. JORDAN, R. CARUSO, and A. R. VON NEIDA

(Manuscript received May 11, 1982)

*A commercially viable GaAs device technology for field-effect transistors, integrated circuits, and lasers is critically dependent on the availability of high-quality, single-crystal boules with controlled diameter. We have modeled a diameter control scheme based on the monitoring of crystal weight for liquid-encapsulated Czochralski (LEC) growth of GaAs. The presence of the $B_2O_3$ liquid encapsulant and significant capillary forces make the direct interpretation of the weight-gain signal and its time derivative (DWGS) more complicated in comparison with pulled materials such as oxide crystals. We have formulated a realistic model for the LEC growth of axisymmetric crystals and have derived the differential equation relating the time evolution of the DWGS to radius and length. We show that the magnitude of the DWGS at the crystal's "shoulder" is inversely related to the radius of curvature. Furthermore, the meniscus by itself gives rise to a precursor or early warning in the signal, which means that the maximum in DWGS precedes the maximum in shape by a few hundred seconds. The existence of a secondary maximum or aftershock in the signal that is the sole consequence of liquid encapsulation is also demonstrated. Excellent agreement has been obtained between DWGS and the signal predicted from the measured shape of a grown crystal. Thus, prospects for automatic diameter control are encouraging.*

## I. INTRODUCTION

GaAs is one of the key semiconductor materials that serves as a substrate for light-emitting diodes (LEDs), lasers, and field-effect transistors (FETs). Paralleling the development pattern of growth techniques for other single crystals, the issues of primary interest have

evolved from questions of quality (elimination of twinning and defects, doping uniformity) to that of economy of size. However, scaling up the dimensions of GaAs crystals grown by the liquid-encapsulated Czochralski (LEC) technique necessitates introducing sophisticated schemes for diameter control. By achieving satisfactory control, it will also be possible to run for extended periods of time with minimal supervision, and to attain higher crystal yields, as well as a reduction in defect generation brought on by shape change.

Since diameter control in Czochralski growth has been the subject of an excellent recent review by Hurle,[1] here a brief outline of previous efforts with respect to LEC growth will suffice. Unlike Si[2] and a wide variety of oxide crystals[3] (e.g., GGG, $LiTaO_3$) for which successful diameter measurement and control systems have been developed, the realization of a viable system for the LEC growth of III-V compounds has been much more difficult to achieve. This is largely due to effects associated with the growth chamber under high pressure, the presence of the encapsulating layer of $B_2O_3(\ell)$, and the phenomenon of anomalous density ($d_{liquid} > d_{solid}$) in some semiconductor materials.

A number of approaches to the control problem have been taken. Pruett and Lien,[4] and van Dijk et al.[5] have employed an X-ray beam passing through the high-pressure growth apparatus for GaP, designed to make the melt a high absorber of radiation relative to all other materials in the radiative path of the system. The subsequent use of an image intensifier gave an accurate television picture of the growth interface. Small changes in diameter could then be electronically extracted from the video signal and utilized for its control.[5]

Alternatively, considerable success has been achieved meeting the demand for large, closely controlled diameter GaP single crystals using a floating die technique. Cole et al.[6] employed a $Si_3N_4$ ring floating in the GaP melt beneath the $B_2O_3(\ell)$. This modified form of Stepanov ring[6,7] creates a long-term stable growth regime, permitting diameter tolerances of ±1 mm for a 50-mm diameter boule. The technique has also been applied to GaAs with different degrees of success, apparently depending on the crystallographic growth direction.[8]

In a third technique, advocated by Bardsley et al.,[9] a weight signal or a derivative weight-gain signal obtained by detecting the apparent weight change of either the crucible or crystal during Czochralski growth is used to control the power output and consequently the diameter. Bardsley et al. have considered both the theory[9] and its implementation by an analogue servo-system.[10] Although these authors have proposed reasonable initial postulates and expressions to calculate the weight gain, the detailed analysis is limited only to small perturbations in diameter and the formalism precludes growth by the LEC technique.

For some time now we have routinely measured the apparent weight of LEC-pulled GaAs crystals with a high sensitivity load cell placed in the high-pressure chamber in series with the crystal pull rod. The output of the cell is recorded as a dc level. Then, the derivative of the signal is taken electronically with a device developed especially for that purpose, and is also recorded. This activity has served as a useful qualitative guide in the manual control of diameter. The major objective of this paper is to develop the fundamental theory that governs the relationship between the shape of an axisymmetric crystal with arbitrary variations in its diameter (including the shoulder) grown by the LEC technique and the instantaneously detected derivative weight-gain signal. Besides gaining new physical insight, we expect that by means of these investigations we can focus on the important physical factors and analytical techniques essential to the eventual control of crystal diameter.

As a first step, a tractable model is formulated for LEC growth with a meniscus. The treatment leads to an explicit expression for the derivative weight-gain signal exclusively in terms of the crystal cross section and its first and second derivatives, in addition to geometrical and material parameters. Next, the numerical techniques required to perform computer simulations are outlined. Then, the key features of the signal are investigated by using the probability density function of the lognormal distribution for the crystal contour. Among the effects considered in detail are the shape of the shoulder and the presence or absence of the $B_2O_3(\ell)$ encapsulant and meniscus. Furthermore, the derivative weight-gain signal is predicted for four contour lines—generated by consecutive 90° axial rotations—of a specially prepared GaAs boule and compared with the signal measured during growth. Finally, in light of these modeling calculations, the prospective techniques for diameter control are discussed.

## II. THEORY

In this section, by a consideration of the relevant features of the LEC growth process, a realistic model is developed for the derivative weight-gain signal in a form suitable for computer analysis. To make the calculations tractable, but without a serious loss of generality, the following set of assumptions and constraints has been introduced:

(*i*) The crystal and crucible are axisymmetric with circular cross sections. Hence, the crystal's radius, $r$, can always be prescribed as a function of the axial location, $L$, which is monotonically related to the time elapsed since seeding.

(*ii*) The crucible containing the melt and the encapsulant is a right circular cylinder of radius, $R$. Less restrictively, departures from

a right cylinder are permitted below the liquid level prevailing at the completion of growth.

(*iii*) The effects of crucible and crystal rotation and the concomitant viscous drag are neglected.

(*iv*) As the initial melt level falls by distance $\ell$ (Fig. 1), the entire amount of the liquid is transferred to the solid.

(*v*) Seepage from the $B_2O_3(\ell)$ encapsulating column of mass, $m_B$, and volume, $V_B$, to the melt/crucible interface (walls) is minute.

(*vi*) The solid-liquid interface is planar. This allows the separation of the associated heat transfer problem from the calculations.

(*vii*) Capillary forces between the melt, $B_2O_3(\ell)$, and solid result in a meniscus of height, $h$, with a subtended angle between the vertical

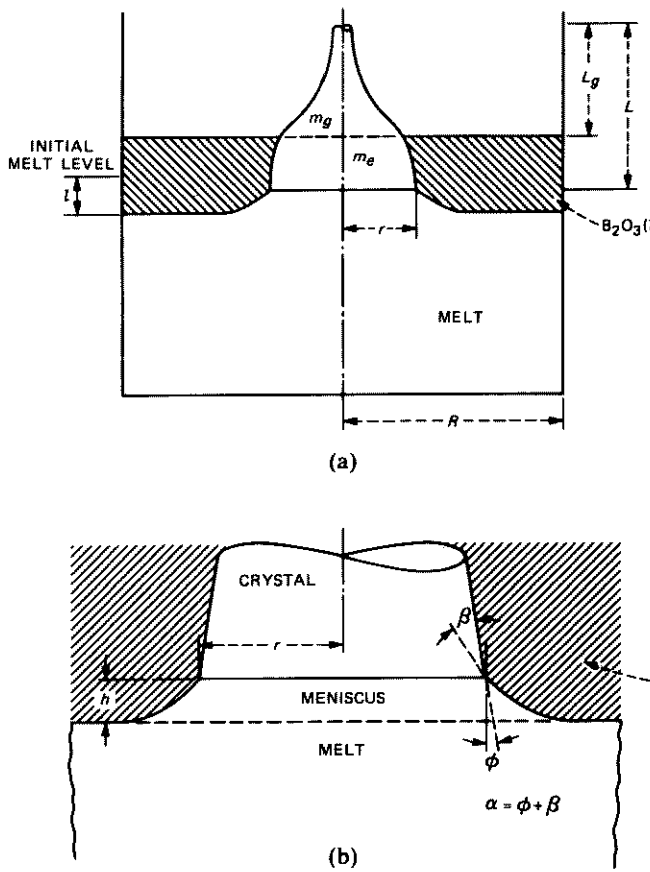



Fig. 1—(a) Schematic geometry of LEC pulling. (b) Enlarged view of the solid-liquid interface region.

and the tangent to the "skirt" of the meniscus (joining angle), $\alpha$ (Fig. 1).

(*viii*) As the crystal is pulled at constant rate $p$, contact with the melt is uninterrupted.

(*ix*) Physical constants required in the calculations are average values representative near the melting point of the crystal.

The preceding conditions in combination with the geometrical description of LEC pulling in Fig. 1 and Archimedes' principle permit writing the weight signal, $W_s$, corresponding to detected crystal mass, $m_s$, as the sum

$$W_s = m_s g = \underset{\substack{W_1 \\ \text{true weight}}}{} \quad \underset{\substack{-W_2 \\ \text{buoyancy correction to } W_1}}{} \quad \underset{\substack{+W_3 \\ \text{vertical projection of capillary force}}}{} \quad \underset{\substack{+W_4 \\ \text{weight of the cylindrical meniscus}}}{} \quad \underset{\substack{-W_5 \\ \text{buoyancy correction to } W_3 + W_4}}{} \tag{1}$$

where

$$W_1 = (m_e + m_g)g = mg \tag{2a}$$

$$W_2 = V_e \rho_B g \tag{2b}$$

$$W_3 = 2\pi r\sigma \cos \alpha \tag{2c}$$

$$W_4 = r^2 \pi h \rho_\ell g \tag{2d}$$

$$W_5 = V_{\text{men}} \rho_B g \tag{2e}$$

The symbols $m$, $V$, and $\rho$ without subscripts designate the mass, volume, and density of the growing crystal, respectively. The same letters with subscripts $B$, $\ell$, $e$, and $g$ refer to the $B_2O_3$ encapsulant, melt, and that portion of the crystal that resides either in the encapsulant or the gas phase, respectively. The symbols $\sigma$ and $V_{\text{men}}$ represent the surface tension and meniscus volume, respectively.

Since $V_e = (m - m_g)/\rho$, with the abbreviation $k_1 = \rho_B/\rho$, $W_1 - W_2$ in eqs. (2a) and (2b) can be transformed into

$$W_1 - W_2 = m(1 - k_1)g + k_1 m_g g. \tag{3}$$

Then, differentiating eqs. (3), (2c), (2d), and 2e with respect to time, $t$, and using eq. (1), we obtain the derivative weight-gain signal (DWGS) in the form

$$\frac{dW_s/g}{dt} = (1 - k_1)\frac{dm}{dt} + k_1 \frac{dm_g}{dt} + \frac{2\pi}{g}\sigma \cos \alpha \frac{dr}{dt} - \frac{2\pi\sigma}{g} r \sin \alpha \frac{d\alpha}{dt} + \rho_\ell \left(2r\pi h \frac{dr}{dt} + r^2 \pi \frac{dh}{dt}\right) - \rho_B \frac{dV_{\text{men}}}{dt}. \tag{4}$$

It can be readily shown that $dm/dt$ is a function of the other

variables. Clearly, if the crystal is axisymmetric, one has

$$\frac{dm}{dt} = \rho\pi r^2 \frac{dL}{dt}. \tag{5}$$

Moreover, the continuity of growth provides the relation

$$L = \ell + pt - h, \tag{6}$$

or in derivative form

$$\frac{dL}{dt} = \frac{d\ell}{dt} + p - \frac{dh}{dt}. \tag{7}$$

In addition, the conservation of matter during crystallization as the liquid level falls leads to

$$m = \rho_\ell(\pi R^2 \ell - V_{\text{men}}) \tag{8}$$

or

$$\frac{dm}{dt} = \rho_\ell \left( \pi R^2 \frac{d\ell}{dt} - \frac{dV_{\text{men}}}{dt} \right). \tag{9}$$

A combination of eqs. (5), (7), and (9) permits the elimination of $d\ell/dt$; hence eq. (5) can be rewritten as

$$\frac{dm}{dt} = \rho\pi r^2 \quad \frac{\left( \frac{k dV_{\text{men}}}{\pi \ dt} + pr^2 - r^2 \frac{dh}{dt} \right)}{\bar{R}^2 + p - \frac{dh}{dt}}, \tag{10}$$

where $k = \rho_\ell/\rho$ and $\bar{R}^2 = \mathrm{kR}^2 - \mathrm{r}^2$.

Substituting eq. (10) into eq. (4) for the DWGS yields after several algebraic operations

$$\frac{dW_s/g}{dt} = k_1 \frac{dm_g}{dt} + \frac{p(1-k_1)\pi\rho_\ell r^2 R^2}{\bar{R}^2} - \frac{2\pi\sigma}{g} r \sin\alpha \frac{d\alpha}{dt}$$
$$+ \frac{dV_{\text{men}}}{dt} \left[ (1-k_1)\rho_\ell \frac{r^2}{\bar{R}^2} - \rho_B \right] + r^2\pi \frac{dh}{dt} \rho_\ell \left[ 1 - (1-k_1)\frac{R^2}{\bar{R}^2} \right]$$
$$+ 2\pi \frac{dr}{dt} \left[ rh\rho_\ell + \frac{\sigma}{g} \cos\alpha \right]. \tag{11}$$

In essence, the general form of the DWGS is given by eq. (11). Since all the variables directly depend on $L$ or indirectly through $r$ on $L$, it is convenient to replace the derivatives $d/dt$ by $d/dL \cdot dL/dt$. Accordingly, we have

$$\frac{dW_s/g}{dt} = k_1 \frac{dm_g}{dL}\frac{dL}{dt} + \frac{p(1-k_1)\pi\rho_\ell r^2 R^2}{\bar{R}^2} - \frac{2\pi\sigma}{g} r \sin\alpha \frac{d\alpha}{dL}\frac{dL}{dt}$$

$$+ \frac{dV_{\text{men}}}{dL}\frac{dL}{dt}\left[(1-k_1)\rho_\ell \frac{r^2}{\bar{R}^2} - \rho_B\right] + r^2\pi \frac{dh}{dL}\frac{dL}{dt}\rho_\ell\left[1-(1-k_1)\frac{R^2}{\bar{R}^2}\right]$$

$$+ 2\pi \frac{dr}{dL}\frac{dL}{dt}\left(rh\rho_\ell + \frac{\sigma}{g}\cos\alpha\right). \quad (12)$$

Equation (12) completely describes the DWGS once the few remaining unknown functions are evaluated. These are the axial derivative of the crystal mass outside the $B_2O_3(\ell)(dm_g/dL)$, the macroscopic growth rate $(dL/dt)$, and the meniscus ($h$, $dh/dL$, $V_{\text{men}}$, and $dV_{\text{men}}/dL$) in terms of $r(L)$ and the required physical and geometrical constants. The crystal shape $r(L)$ we consider here as an input signal and the choice of different functional forms will be postponed to the next section.

### 2.1 Evaluation of $dm_g/dL$

From Fig. 1 we conclude that the total cylindrical volume occupied by the segment of the growing crystal in $V_e$, the encapsulant, and the meniscus are conserved. Thus we can write the integral relation

$$\int_0^L r^2\pi dL - \int_0^{L_g} r^2\pi dL + V_{\text{men}} + V_B = \pi R^2(L - L_g + h), \quad (13)$$

where the two integrals on the left-hand side are equal to

$$\int_{L_g}^{L} r^2(L)\pi dL.$$

Thus we can express $m_g/\rho$ as

$$m_g/\rho = \int_0^{L_g} r^2\pi dL = \int_0^L r^2\pi dL + V_{\text{men}} + V_B - \pi R^2(L - L_g + h).$$

Differentiating the above equation with respect to $L$ gives

$$\frac{dm_g/\rho}{dL} = r^2\pi + \frac{dV_{\text{men}}}{dL} - \pi R^2\left(1 + \frac{dh}{dL} - \frac{dL_g}{dL}\right). \quad (14)$$

To obtain $dL_g/dL$ we define a function $\phi(L, L_g) = 0$ by subtracting the right-hand side from both sides of eq. (13). Then, employing the well-known result for the differentiation of implicit functions we have

$$\frac{dL_g}{dL} = -\frac{\partial\phi/\partial L}{\partial\phi/\partial L_g}, \quad (15)$$

where

$$\frac{\partial \phi}{\partial L} = r^2(L)\pi + \frac{dV_{\text{men}}}{dL} - \pi R^2 \left(1 + \frac{dh}{dL}\right)$$

and

$$\frac{\partial \phi}{\partial L_g} = -r^2(L_g)\pi + \pi R^2.$$

The actual value of $L_g$ can only be determined by a numerical procedure. From eq. (13) we get

$$\frac{\int_0^L r^2 \pi dL + \frac{m_B}{\rho_B} + V_{\text{men}}}{\pi R^2} - (L + h) = \frac{\int_0^{L_g} r^2 dL}{R^2} - L_g, \tag{16}$$

where the left-hand side includes only known quantities.

While the entire crystal is submerged in $B_2O_3$, $m_g = 0$ and $dm_g/dL = 0$. In that case the counterpart of the volume balance in eq. (13) is of the form

$$\int_0^L r^2 \pi dL + V_B + V_{\text{men}} = \pi R^2 (L + h + \Delta), \tag{17}$$

where $\Delta$ is the distance of the tip of the crystal from the $B_2O_3(\ell)$ – gas interface. Rearranging eq. (17) yields

$$\frac{\int_0^L r^2 \pi dL + \frac{m_B}{\rho_B} + V_{\text{men}}}{\pi R^2} - (L + h) = \Delta. \tag{18}$$

Comparing eqs. (16) and (18) we observe that the left-hand sides are identical. As the crystal protrudes through the encapsulant, $\Delta$ changes from positive [eq. (18)] to negative [eq. (16)]. This provides an important clue in the computation of $L_g$. The numerical solution of eq. (16) for $L_g$ starts when $\Delta = 0$.

### *2.2 Determination of the macroscopic growth rate, dL/dt*

The macroscopic growth rate given in eq. (7) can be rewritten as

$$\frac{dL}{dt} = \frac{d\ell}{dL}\frac{dL}{dt} - \frac{dh}{dL}\frac{dL}{dt} + p. \tag{19}$$

Hence, by a rearrangement of eq. (19) $dL/dt$ becomes

$$\frac{dL}{dt} = \frac{p}{1 - d\ell/dL + dh/dL}. \tag{20}$$

To eliminate the melt coordinate $\ell$ we use the relation for the conservation of melt [eq. (8)], which gives

$$\ell = \frac{m + \rho_\ell V_{\text{men}}}{\rho_\ell \pi R^2} = \frac{\pi \int_0^L \rho r^2 dL + \rho_\ell V_{\text{men}}}{\rho_\ell \pi R^2}. \tag{21}$$

Differentiating eq. (21) with respect to $L$ yields

$$\frac{d\ell}{dL} = \frac{r^2}{kR^2} + \frac{dV_{\text{men}}/dL}{\pi R^2}. \tag{22}$$

Finally, introducing eq. (22) into eq. (20) gives the macroscopic growth rate in the form

$$\frac{dL}{dt} = \frac{p}{1 - \left(\frac{r^2}{kR^2} + \frac{dV_{\text{men}}}{dL} \Big/ \pi R^2\right) + \frac{dh}{dL}}. \tag{23}$$

### 2.3 Transformation of the crystal length to time coordinate

In general, $r$ is known as a function of $L$. On the other hand the DWGS is recorded as a function of time. Therefore, a transformation from $L$ to $t$ is necessary in the analysis. From the continuity condition [eq. (6)] $t$ can be expressed as

$$t = \frac{L + h(L) - \ell(L)}{p}. \tag{24}$$

Substituting eq. (21) into eq. (24) yields

$$t = \left(L + h - \frac{\int_0^L r^2 dL}{kR^2} - \frac{V_{\text{men}}}{\pi R^2}\right) \Big/ p. \tag{25}$$

### 2.4 Meniscus height and related properties

Mika and Uelhoff[11] have determined the meniscus shape and interface height, $h$, occurring during Czochralski growth by a numerical solution of the Euler-Laplace differential equation. Although, in principle, numerical results for the meniscus could be generated concurrently with the computation of the DWGS, here we prefer to rely on faster closed form solutions of the capillary equation. Fortunately, as shown by Mika and Uelhoff,[11] the analytical approximation of Tsivinskii[12] describes the exact values of $h$ with excellent precision over a wide range of joining angles, $\alpha$, and radii. We have verified that for $\alpha \lesssim -10°$ and $r \lesssim 0.5$ cm the error in using Tsivinskii's approximation is less than ~2 percent. Subsequent computer simulations have demonstrated that in practical situations, except when there is a very rapid

drop in radius, the stated limit on $\alpha$ is obeyed. The fact that for a short time after seeding $r$ is less than 0.5 cm is of minor importance owing to the initial insensitivity of the DWGS to $h$.

Therefore, Tsivinskii's equations[12] for the interface height and shape will provide the basis of our DWGS analysis. In our notation Tsivinskii's equation for $h$ is of the form

$$h = A[(1 - \sin\alpha + u^2)^{1/2} - u], \tag{26}$$

where

$$u = \frac{A\cos\alpha}{4r(L)}$$

and $A$ is a constant. According to Egorov, Tsivinskii and Zatulovskii's modification for LEC growth of Tsivinskii's original work $A$ is given by[13]

$$A = \sqrt{\frac{2\sigma}{(\rho_\ell - \rho_B)g}}. \tag{27}$$

The joining angle $\alpha$ is the sum of the growing angle, $\phi$, and contact (wetting) angle, $\beta$ (Fig. 1), i.e.,

$$\alpha = \phi + \beta. \tag{28}$$

In general, $\beta$ is a constant, while the growing angle is a function of the derivative $dr/dL$ according to[9]

$$\tan\phi = \frac{dr}{dL}, \qquad \phi = \tan^{-1}\frac{dr}{dL}. \tag{29}$$

Thus, $h$ [eq. (26)] can be expressed in terms of $r$ and $dr/dL$ at the solid-liquid interface.

Consequently, the required first derivative of $h$, $dh/dL$, includes the second derivative $d^2r/dL^2$. This can be readily shown by the substitution of the combined eqs. (28) and (29) into eq. (26). Then, using standard trigonometric relations for angle-sums and inverse functions and differentiating the expression thus obtained with respect to $L$ we find

$$\frac{dh}{dL} = \frac{A}{\left[1 + \left(\frac{dr}{dL}\right)^2\right]^{1/2}}\left[\frac{uv - \lambda}{(1 - \sin\alpha + u^2)^{1/2}} - v\right], \tag{30}$$

where

$$\lambda = \frac{1}{2}\frac{d^2r/dL^2}{1 + \left(\frac{dr}{dL}\right)^2}\left(\cos\beta - \frac{dr}{dL}\sin\beta\right)$$

$$v = -\frac{A}{4r^2}\left[\frac{rd^2r/dL^2}{1+\left(\frac{dr}{dL}\right)^2}\left(\frac{dr}{dL}\cos\beta + \sin\beta\right)\right.$$

$$\left. + \frac{dr}{dL}\left(\cos\beta - \frac{dr}{dL}\sin\beta\right)\right].$$

Another derivative of interest in eq. (12) is $d\alpha/dL$. From eqs. (28) and (29) we have

$$\frac{d\alpha}{dL} = \frac{d(\tan^{-1}dr/dL + \beta)}{dL} = \frac{1}{1+(dr/dL)^2}\frac{d^2r}{dL^2}. \tag{31}$$

A property closely related to the interface height is the volume of the skirt-shaped meniscus, $V_{\text{men}}$ (Fig. 1), and its axial derivative $dV_{\text{men}}/dL$. The quantity $V_{\text{men}}$ can only be evaluated by a numerical technique. Egorov et al.[13] provide the $x, y$ coordinates of the meniscus in the form of the integral

$$x = \int_h^y \frac{Q}{\sqrt{1-Q^2}}\,dy + r(L), \tag{32}$$

where

$$Q = \sin\alpha - \left(\frac{1}{A^2} + \frac{\cos\alpha}{2rh}\right)(y^2 - h^2).$$

Owing to axial symmetry summing up narrow segments of area $x^2\pi$ provides the volume of the meniscus as

$$V_{\text{men}} = \pi\delta \sum_{y_i=0}^{y_i=h-\delta} x^2(y_i), \tag{33}$$

where $\delta$ is the thickness of a segment.

Apart from performing numerical differentiation, no simple method exists for the evaluation of $dV_{\text{men}}/dL$. If the shape of the meniscus is that of a right cylinder, then $V_{\text{men}}$ and $dV_{\text{men}}/dL$ are given by

$$V_{\text{cyl}} = r^2\pi h$$

and

$$\frac{dV_{\text{cyl}}}{dL} = 2r\pi h\frac{dr}{dL} + r^2\pi\frac{dh}{dL}. \tag{34}$$

Simulations indicate that the numerical derivative $dV_{\text{men}}/dL$ is closely approximated by

$$\frac{dV_{\text{men}}}{dL} \simeq \frac{V_{\text{men}}}{V_{\text{cyl}}}\frac{dV_{\text{cyl}}}{dL}. \tag{35}$$

## III. RESULTS AND DISCUSSION

### *3.1 Computing methods*

We have succeeded in reducing the theoretical determination of the DWGS to a differential equation in $r$, $dr/dL$ and $d^2r/dL^2$. Therefore, if the shape of the crystal is given, the DWGS is calculable; conversely, the DWGS completely defines the radius of the crystal at any moment of interest. Here, we address ourselves to the first part of the problem, though the eventual objective is diameter control emphasizing the DWGS.

The program to evaluate the DWGS was written in HP Basic. A key feature in the efficient organization of the computations is the use of subroutines for $dm_g/dL$ [eq. (14)], $dL_g/dL$ [eq. (15)], $dL/dt$ [eq. (23)], $h$ [eq. (26)], $dh/dL$ [eq. (30)], $d\alpha/dL$ [eq. (31)], $V_{\text{men}}$ [eq. (33)], $dV_{\text{men}}/dL$ [eq. (35)], as well as for $r(L)$ and its first and second derivatives.

Numerical integration was necessary to obtain $V_{\text{men}}$ and the crystal volumes between $O$ and $L$ and $O$ and $L_g$. It was found advantageous to evaluate the crystal volumes in the main routine because during the simulated growth of a crystal incrementing $L$ in steps of 0.1 cm, the accumulating volume elements could be retained in memory.

As mentioned earlier, the quantity $L_g$ is only required after the seed emerges from the $B_2O_3(\ell)$ encapsulant. Mathematically speaking, this occurs when the left-hand side of eq. (16) or (18) becomes negative. To determine $L_g$, at each increment of $L$ the left-hand side of eq. (16) is evaluated. Then, a linear search is performed, substituting trial values of $L_g$ into the right-hand side of eq. (16) in the range $L_g' - 0.2$ to $L_g' + 0.4$ cm in steps of 0.01 cm, where $L_g'$ is the previous solution to eq. (16) at $L - 0.1$ cm. In case the 0.01-cm mesh around $L_g'$ was too coarse to yield a solution, an option with 0.001-cm increments was available.

There are several mathematical tools available to describe the shape of the grown crystal, $r(L)$. From the actual cross section a large number of $r$, $L$ coordinates can be generated in tabular form. Then, one obvious choice is to employ an interpretation scheme based on point-by-point tabular values of $r(L)$ in addition to numerical first and second derivatives. A more sophisticated alternative is to determine the Fourier coefficients of the crystal shape. However, we have learned by experience that the most efficient and satisfactory scheme to convert the tabular data into functional form is by using cubic spline regression.

A cubic spline function is a piecewise polynomial of degree three that joins adjacent polynomials in "knots." At the knots the functional values as well as the first and second derivatives are continuous. Ordinarily, a spline passes through all the points within the interval bounded by two knots. In contrast, in spline regression a cubic least-square fit for the same points is obtained. We have adapted, for the

present application, Wold's spline regression technique in terms of *B*-splines.[14]

Fortunately, the key features of the DWGS can be illuminated without invoking advanced data-fitting techniques. A judiciously chosen elementary transcendental function representing $r(L)$, which has $n$ continuous derivatives, will be found suitable to illustrate the effect of various factors governing the course of the DWGS.

### 3.2 Major influences on the derivative weight-gain signal

#### 3.2.1 Crystal shape

Typical LEC crystals, including GaAs, exhibit a pronounced "shoulder" as the radius rapidly expands from the seed. Boule-to-boule variation in the rate of approach to and radius of curvature of the shoulder is not uncommon. A mathematical description of the shoulder, in general, and a prototype crystal, in particular, is possible by means of the probability density function of the lognormal distribution.[15] Accordingly, we can write for a seed diameter of 0.6 cm

$$r(L) = C\left[\frac{1}{sL\sqrt{2\pi}}\exp - \frac{(\ln^2 L/L_m)}{2s^2}\right] + 0.3. \tag{36}$$

The median length, $L_m$, and the constant $s$ are chosen such that the maximum $r$ is 2.8 cm at $L = 2.2$ cm. From these conditions we have

$$L_m = 2.2\exp(s^2)$$

and a lengthy expression for $C$ that is also only dependent on the standard deviation, $s$. Hence, the function $r(L)$ can be constructed if $s$ is given.

In Figs. 2, 3, and 4 we present the crystal shape $r(L)$ for $s = 0.4$, 0.8, and 1.2, respectively. It can be seen that with decreasing $s$ the radius of curvature at the maximum decreases (i.e., $|d^2r/dL^2|$ increases). Furthermore, it can be shown that the rate of approach to the shoulder $(dr/dL)$ rapidly rises as $s$ diminishes.

Substituting eq. (36) and its first and second derivatives into the DWGS [eq. (12)] and the subsidiary equations yields the signal and the crystal cross section as a function of time, both of which are shown in Figs. 2, 3, and 4. The required parametric values are listed in Table I. The results demonstrate that the DWGS is extremely sensitive to $dr/dL$ and $|d^2r/dL^2|$. In fact, the steeper the rise in the shape, the sharper the peak in the DWGS and the larger its absolute magnitude.

Two other characteristic features of LEC pulling that are of crucial importance can also be observed in Figs. 2, 3, and 4. First, the maximum in DWGS precedes in time the maximum in shape by a few hundred seconds. This time lag we shall designate as the "precursor." The second property is the additional maximum in DWGS at a time when

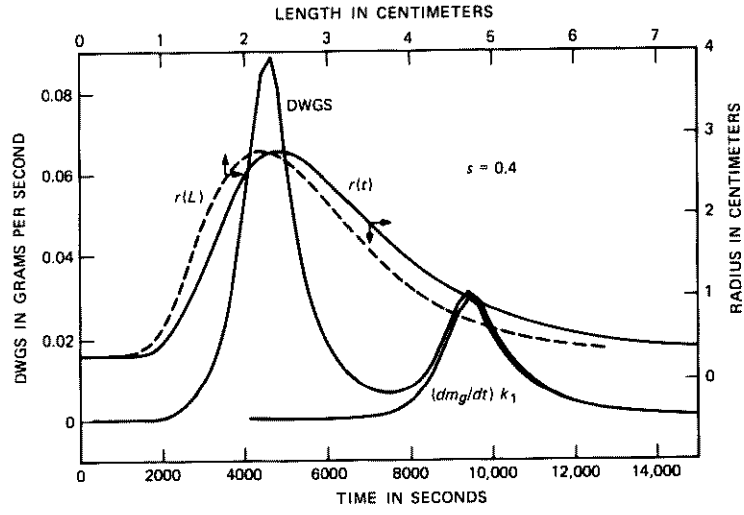



Fig. 2—The DWGS and crystal cross section as a function of time for LEC pulling. The shape versus length curve (- - -) is based on eq. (36), setting $s = 0.4$. The time derivative of the unencapsulated weight [$dm_g/dt$, eqs. (14) and (19)] is also shown.

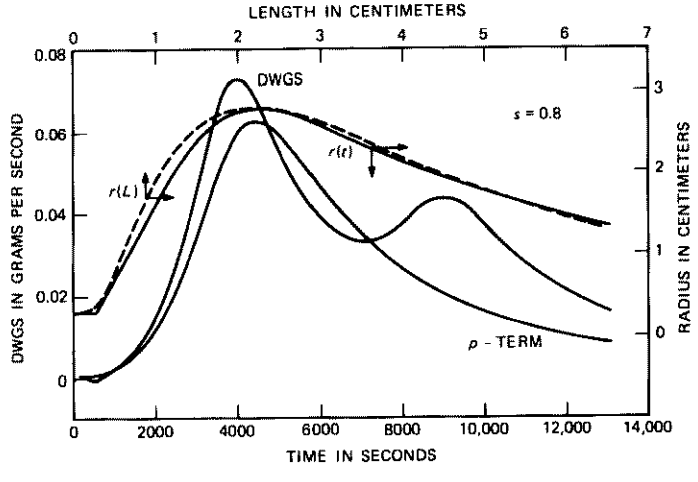



Fig. 3—The DWGS and crystal cross section as a function of time for LEC pulling. The shape versus length curve (- - -) is based on eq. (36), setting $s = 0.8$. The pull rate term in eq. (12)—the factor usually employed in conventional diameter control—is also shown.

the crystal's radius has already declined. We shall refer to this phenomenon as the "aftershock."

We have investigated the source of the precursor and aftershock. Simulations show that neither $r$ nor its derivatives bears any responsibility. Examining the variation of the interface height with time (Fig.

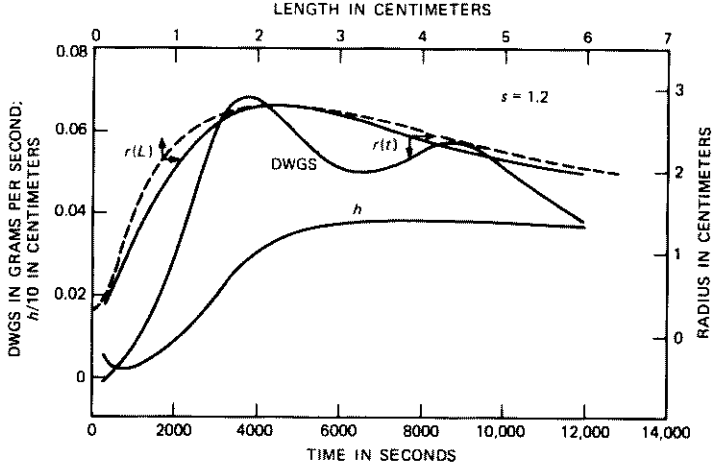



Fig. 4—The DWGS and crystal cross section as a function of time for LEC pulling. The shape versus length (- - -) curve is based on eq. (36), setting $s = 1.2$. The meniscus height [eq. (26)] is also shown.

Table I—Parameters for the calculation of the DWGS

| | |
|---|---|
| Density of crystalline GaAs, $d$[g/cm$^3$] | 5.17 |
| Density of molten GaAs, $d_l$[g/cm$^3$] | 5.71 |
| Density of $B_2O_3(\ell)$, $d_B$[g/cm$^3$] | 1.55 |
| Mass of $B_2O_3(\ell)$, $m_B$[g] | 129.5 |
| Pull rate, $\rho$[cm/s] | $3.6 \times 10^{-4}$ |
| Crucible radius, $R$[cm] | 3.9 |
| Wetting angle, $\beta$[deg] | 15 |
| Capillary constant, $A$[cm] | 0.4 |
| Surface tension, $\sigma$[dyne/cm] | 333 |

4) we note that starting with a small but finite value at the seed, $h$ saturates beyond the shoulder ($h \simeq 0.38$ cm).* Hence, no clue can be extracted from the form of $h$. Concentrating on the term including the pull rate in eq. (12), one concludes that this conventional description of the DWGS[3] holds quite well in the early phase of growth but departs from reality near the shoulder and beyond (Fig. 3). The maximum in the $p$ term and the shoulder perfectly coincide and no secondary maximum appears.

If, however, one examines the derivative $dm_g/dt$ [eqs. (14) and (23),

* The fact that at $t = 0$, $r \neq 0$, thus $h > 0$ is finite contradicts eq. (24). Therefore, to correct eq. (25) the residual time

$$t_0 = \left[ h(L=0) - \frac{V_{men}(L=0)}{\pi R^2} \right] \Big/ p$$

is always subtracted from $p$.

Fig. (2)], it is found that this quantity rises from zero to a maximum value precisely at the time of the aftershock and that the magnitude of the DWGS in this regime is essentially $dm_g/dt$. In view of the fact that $dm_g/dt$ is specifically associated with $B_2O_3(\ell)$ encapsulation, a reasonable hypothesis is that the aftershock is a consequence of the LEC technique.

### 3.2.2 Meniscus shape and liquid encapsulation

To isolate the factors leading to the precursor and aftershock we have evaluated the DWGS corresponding to the lognormal $r(L)$ [eq. (36), $s = 0.8$] in the absence of liquid encapsulation. In this case the major equations still hold provided $\rho_B$, $V_B$, $m_B$, $m_g$, and $dm_g/dt$ are taken as zero. In Fig. 5 we present the DWGS with and without interposing a meniscus between the melt and the growing crystal. It is immediately obvious that in both instances the aftershock disappears from the DWGS. Moreover, if the meniscus is also removed, the precursor is missing, i.e., the times to reach the peak in DWGS and cross section exactly coincide. In fact by simultaneously eliminating $B_2O_3(\ell)$ and the meniscus from the growth system, only the contribution of the conventional pull rate term[3] is retained in eq. (12).

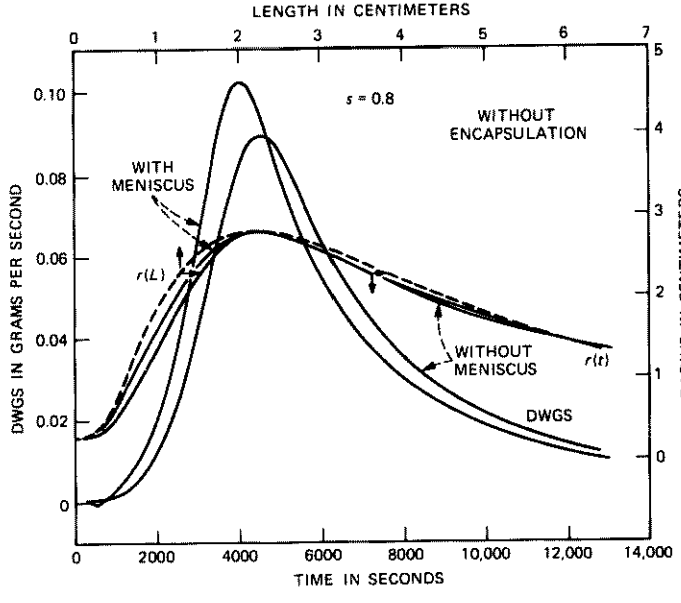



Fig. 5—The DWGS and crystal cross section as a function of time in normal Czochralski pulling without encapsulation. Curves with and without a meniscus present are given. The shape versus length curve (– – –) is based on eq. (36), setting $s = 0.8$. When the meniscus is absent, the pull rate term in eq. (12) exactly coincides with the curve shown.

Further confirmation of these effects is offered in Fig. 6. Here the calculation of the DWGS is repeated for LEC pulling without a meniscus. As we expected, though the precursor is missing the aftershock reappears. Apparently, up to and over the shoulder the DWGS is governed by the $p$ term. Near the secondary maximum the derivative $dm_g/dt$ becomes the dominant factor.

A more physical interpretation of the aftershock is possible by plotting the time-dependent radius, $r(L_g)$, of the crystal as it just protrudes through the $B_2O_3(\ell)$. In Fig. 6 we show both $r(L)$ and $r(L_g)$ as a function of time. Besides an expected displacement of the radius along the time axis one notes that the maximum in $r(L_g)$ occurs exactly at the time of the secondary maximum in DWGS. From these considerations a surprisingly simple explanation emerges. As more of the shoulder region becomes uncovered from the encapsulant, the DWGS registers the rapidly increasing true weight gain as opposed to the previously measured apparent (buoyancy-reduced) weight gain of the crystal. Hence, at the time of the maximum aftershock both the freshly grown layers at $L$, submerged in $B_2O_3(\ell)$, and the sizable exposed portions at $L_g$ are detected.

The discovery of the precursor and the aftershock has a significant bearing on the diameter control of GaAs. The precursor is an early warning signal that predicts a maximum in radius a few hundred seconds before the actual event. In other words the DWGs is already

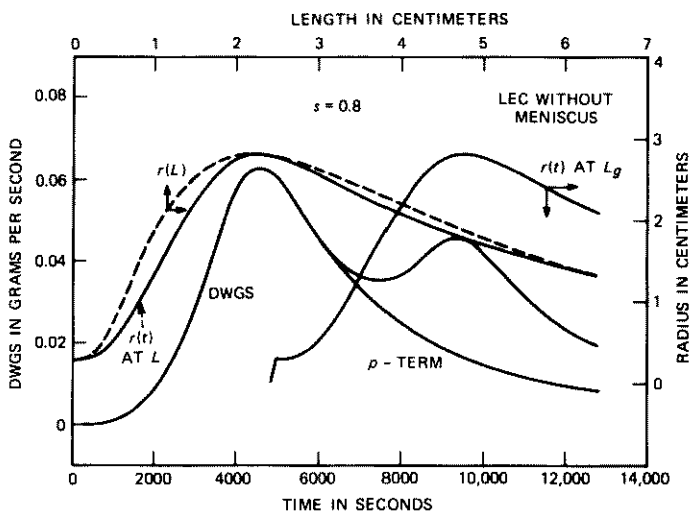



Fig. 6—The DWGS and crystal cross section as a function of time in LEC pulling, free of a meniscus. The radius is shown as a function of time at the solid-liquid interface, $L$, and at the top of the $B_2O_3$ layer, $L_g$. The pull rate term is also shown. The shape versus length curve (– – –) is based on eq. (36), setting $s = 0.8$.

dropping while the diameter is still increasing. Thus the controller has sufficient time to react to an unwanted change.

When the aftershock is observed there is a good chance that the controller misinterprets it as an unexpected gain in diameter and, if undesirable, will respond accordingly. Then, of course, a visible reduction in diameter will occur. By the time this is noticed, corrective action taken cannot restore the loss in diameter control. At best, beyond the shoulder a "sinusoidal" cross section with small amplitude results.

In view of these observations one cannot base the diameter control algorithm for GaAs on the customary $p$ term[3] in eq. (12). To be sure, from the seed up to near the shoulder perhaps it suffices. However, for the bulk of the boule, control by means of such a traditional method is clearly inadequate.

### *3.3 Comparison with experiment*

We have extracted the salient features of the DWGS arising during LEC pulling by the use of the lognormal probability function for the crystal's cross section. To compare the experimentally determined DWGS with the theoretical one the idealized crystal shape must be replaced by that of a GaAs boule grown by the LEC method under closely controlled conditions. Some of the growth parameters of interest for this specially prepared crystal are listed in Table I. The weight and length of the crystal are 523.3 grams and 6.6 cm, respectively.

The crystal coordinates ($r$ versus $L$) can be derived from a two-dimensional projection employing a digitizer. Owing to the eccentricity of the actual ⟨100⟩ boule, a single view is insufficient. We have obtained planar projections with sharp contours by a photographic technique. The crystal was placed on the top of a light box, backlit, and photographed using a high-contrast film. Then, the procedure was repeated following a 90-degree rotation around the axis. In this manner, the two photos provided a total of four cross sections. Digitizing the shape in 0.5-mm intervals 133 pairs of $(r, L)$ coordinates were collected for each of the four contours. At the shoulder the maximum error between the crystal and its projection is less than 0.5 percent. The computed mean weight is 528.6 g with a standard deviation of ±9.3 g, which should be compared with the measured weight of 523.3 g. Clearly, an excellent tabular description of the crystal's cross section has been accomplished. The data points for one of the four views are shown in Fig. 7.

The alternatives to represent the tabular data have been outlined earlier. In principle, because of close spacing, there would be no obstacle to a simple interpolation of the $r(L)$ values. However, as we illustrated in Fig. 7, using the numerical first and second derivatives may be problematic on account of noise. Indeed, when the DWGS was

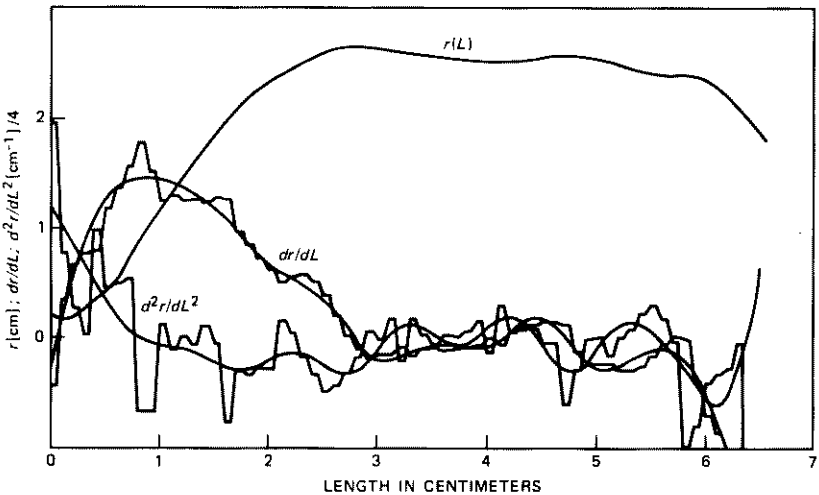



Fig. 7—One of the four experimental contours of a GaAs crystal and its cubic spline-regression representation (—). The cubic spline regression of the numerical first derivative $dr/dL$ is also shown. In addition, the numerical second derivative $d^2r/dL^2$ is given, together with the quadratic expression derived from the cubic spline of $dr/dL$.

computed from the numerical derivatives severe discontinuities appeared in the results. This suggests that the steep "jumps" in $d^2r/dL^2$ seriously affect at some locations the shape and magnitude of the DWGS.

A different approach is the finite Fourier transform of the $r(L)$ points. However, owing to slow convergence a large number of terms is required. Eight terms were found to be sufficient to describe the seed and shoulder regions. Beyond the shoulder 18 terms were necessary to fit the data, albeit with concomitant high-frequency noise. Since the number of required terms was unpredictable and the computation inefficient, Fourier representation of the $r(L)$ data was abandoned. Nonetheless, the DWGS based on Fourier analysis provided a reasonable description of the measurements.

Cubic spline regression[14] has proven to be the most promising method to cast the tabular data on crystal cross sections into functional form. At the middle of the axis, the crystal was separated into two slightly overlapping domains and within each 12 equally spaced knots were positioned. The outstanding spline fit to one of the contours is given in Fig. 7. There are two avenues open to obtain the first derivative $dr/dL$. One is a simple differentiation of the shape's spline, which results in a continuous quadratic representation. Unfortunately, in this case the second derivative becomes a continuous network of connected straight lines (the third derivative is discontinuous) leading

to a jagged DWGS. A more fruitful approach is to fit the numerical first derivative itself by a cubic spline. Then, the noisy second derivative is described by a quadratic expression. In Fig. 7 we show the cubic spline fit to the numerical first derivative data as well as to the quadratic function passing through the numerical $d^2r/dL^2$. We note that a reasonably good description of the derivatives has been obtained by the spline regression technique. Therefore, all four contours were treated in a like manner.

Having thus established the shape and shape derivatives for an actual GaAs crystal, one can readily evaluate the corresponding DWGS by means of the analytical tools described earlier. Among the input parameters listed in Table I, all the densities were obtained from a recent critical evaluation of Jordan.[16] The selected contact angle (15 degrees) is consistent with theoretical and experimental findings on Ge and Si.[17,18,19] The capillary constant $A$ and the surface tension, $\sigma$, are connected via eq. (27). Though the recommended value for $A$ by Egorov et al. is 0.48 cm,[13] we have achieved a slightly better fit to the experimental DWGS using $A = 0.4$ cm, which is equivalent to $\sigma = 333$ dynes/cm. The surface tension of GaAs thus obtained is consistent with that for other III-V compounds (InP[20] and GaSb[21])Si and Ge.[22]

The gross appearance of the DWGS is not overly sensitive to a change in $A$. Nonetheless, a detailed examination reveals that the peak corresponding to the shoulder becomes higher and broader as $A$ increases from 0.28 to 0.48 cm. With respect to the rest of the DWGS one can conclude that the peak-to-valley dynamic range diminishes as $A$ drops. Among the effects of the other parameters, we have observed a rise in DWGS with an increase in $p$ and a reduction of the time scale with a decrease in crucible radius.

In Fig. 8 we have reproduced the experimental derivative weight-gain signal as a function of time, as obtained on a carefully calibrated strip-chart recorder during the LEC growth of GaAs. The predicted DWGSs for the four contour lines as well as the time-dependent cross sections are also shown. Clearly the theoretical curves envelop the measured values, providing an excellent overall description.

The computed DWGSs exhibit both the early warning precursor and aftershock. In the inflection before the first peak we can discern the time when the seed first protrudes through the $B_2O_3(\ell)$. Since an actual crystal may possess additional cross-sectional bulges beside the shoulder, the secondary maximum in DWGS reflects both the memory of that earlier maximum in radius and the current expansion. To a small extent the DWGS is influenced by the input shape derivative function. For example, in the trough above the shoulder, a spline of the first derivative with many more knots would lead to a smoother transition to the next peak. Likewise, near the last peak, an improved

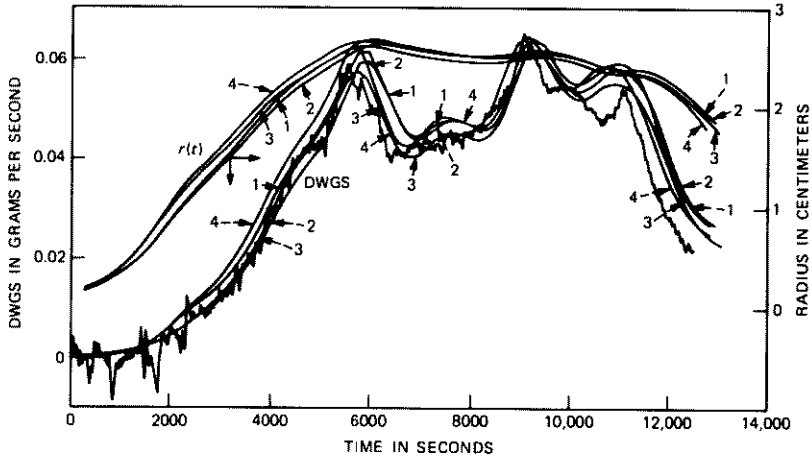



Fig. 8—DWGS (—) and crystal cross section as a function of time for four experimental contour lines of LEC-pulled GaAs. The measured DWGS was traced from the original strip-chart recording.

agreement with the experimental data is possible at the cost of a finer derivative spline. Near the bottom of the crystal the accord with the measured DWGS is more limited, perhaps on account of the increasingly negative turn in the joining angle $\alpha$ (between $-20$ and $-40$ degrees). Under those circumstances Tsivinskii's approximation for the meniscus properties becomes inaccurate.[11]

### *3.4 Prospects for diameter control*

The excellent agreement between predicted and experimental DWGS strongly suggests that diameter control of GaAs by the derivative measurement is feasible. There are two approaches to diameter control implied by the modeling calculations. One is a real-time instantaneous determination of the radius during growth from the DWGS by solving the differential equation (12). Then, depending on whether $r$ is increasing or decreasing from a preset limit, the $RF$ input power is changed by a suitable amount.

The other method is the construction of an "ideal" crystal with appropriate tolerances on a drawing board. Taking into account the tolerances for this crystal, a band of DWGS can be evaluated as has been done for Fig. 8. Subsequently, the control of diameter is reduced to the problem of adjusting the power input in such a fashion that a template established by the band of DWGSs is followed.

## IV. ACKNOWLEDGMENTS

We appreciate the continuing interest of J. W. Nielsen in this work and his critical reading of the manuscript. We thank L. J. Oster for the

expertly grown GaAs crystal and for his careful determination of the derivative weight-gain signal.

## REFERENCES

1. D. T. J. Hurle, "Control of Diameter in Czochralski and Related Crystal Growth Techniques," J. Crystal Growth, *42,* No. 1 (December 1977), pp. 473–82.
2. R. E. Lorenzini, F. S. Nuff, and D. J. Blair, "An Overview of Si Crystal Growing Processes," Solid State Techn., *17,* No. 2 (February 1974), pp. 33–6.
3. A. J. Valentino and C. D. Brandle, "Diameter Control of Czochralski Grown Crystals," J. Crystal Growth, *26,* No. 1 (November 1974), pp. 1–5.
4. H. D. Pruett and S. Y. Lien, "X-ray Imaging Technique for Observing LEC Crystal Growth," J. Electrochem. Soc., *121,* No. 6 (June 1974), pp. 822–6.
5. H. J. A. van Dijk, C. M. G. Jochem, G. J. Scholl, and P. van der Werf, "Diameter Control of LEC Grown GaP Crystals," J. Crystal Growth, *21,* No. 3 (February 1974), pp. 310–12.
6. M. Cole, R. M. Ware, and M. A. Whitaker, paper presented at ECCG1, Zurich (1976).
7. P. Schmaker, A. Kramer, and W. Staehlin, paper presented at ECCG1, Zurich (1976).
8. R. M. Ware, private communication.
9. W. Bardsley, D. T. J. Hurle, and G. C. Joyce, "The Weighing Method of Automatic Czochralski Crystal Growth, Part I," J. Crystal Growth, *40,* No. 1 (September 1977), pp. 13–20.
10. W. Bardsley, D. T. J. Hurle, G. C. Joyce, and G. C. Wilson, "The Weighing Method of Automatic Czochralski Crystal Growth, Part II," J. Crystal Growth, *40,* No. 1 (September 1977), pp. 21–8.
11. K. Mika and W. Uelhoff, "Shape and Stability of Menisci in Czochralski Growth and Comparison with Analytical Approximations," J. Crystal Growth, *30,* No. 1 (August 1975), pp. 9–20.
12. S. V. Tsivinskii, Inzh. Fiz Zh., *5* (1962), p. 59.
13. L. P. Egorov, S. N. Tsivinskii, and L. M. Zatulovskii, "The Shape of the Melt Column in Growing a Crystal from Under a Flux Layer," Bull. Acad. Sci. USSR, Phys. Ser., *40,* No. 7 (1976), pp. 172–4.
14. S. Wold, "Spline Functions in Data Analysis," Technometrics, *16,* No. 1 (February 1974), pp. 1–11.
15. Y. L. Maksoudian, *Probability and Statistics with Applications,* Scranton, PA: International Text Book Co., 1969, pp. 123–4.
16. A. S. Jordan, "An Evaluation of the Thermal and Elastic Properties Affecting GaAs Crystal Growth," J. Crystal Growth, *49,* No. 4 (August 1980), pp. 631–42.
17. P. I. Antonov, "Capillary Phenomena in Crystal Growth" in *Growth of Crystals,* Vol. *6A,* ed., N. N. Sheftal, New York: Consultants Bureau, 1968.
18. T. Surek and B. Chalmers, "The Direction of Growth of a Crystal in Contact with Its Melt," J. Crystal Growth, *29,* No. 1 (May 1975), pp. 1–11.
19. T. Surek, "The Meniscus Angle in Ge Crystal Growth from the Melt," Scripta Met., *10,* No. 5 (May 1976), pp. 425–31.
20. A. S. Popov and L. Demberel, "Surface Tension of Molten Stoichiometric InP," Kristall Und Technik, *12,* No. 11 (1977), pp. 1167–70.
21. M. Ya. Dashevskii, G. V. Kukuladze, V. B. Lazarev, and M. S. Mirgalovskii, "Surface Effects in GaSb Melts," Inorg. Materials, *3,* No. 9 (September 1967), pp. 1360–66.
22. J. C. Phillips and J. A. VanVechten, "Macroscopic Model of Formation of Vacancies in Semiconductors," Phys. Rev. Lett., *30,* No. 6 (February 1973), pp. 220–3.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 2, February 1983
*Printed in U.S.A.*

# Growth, Complexity, and Performance of Telephone Connecting Networks

By V. E. BENEŠ

(Manuscript received December 10, 1981)

*In an effort to free telephone traffic theory from some of its dependence on independence assumptions, and to reap some benefit from its traditional state equations, a systematic search is made to find relationships between load, loss, size, structure, and other network parameters that are simple, universal, and informative. Three principal topics are covered:*

*(i) A load-loss-size formula, linking some half-dozen network parameters by a rational function, and used repeatedly to give*

*(ii) Lower bounds on the number X of crosspoints in networks*

*(iii) Asymptotic results about blocking, growth, and complexity of selected network structures in passing from finite to "infinite" sources at constant load.*

*The major results in (ii) imply that for all practical networks on N terminals, the crosspoint count X must grow like N log N, i.e., incurring loss by restricting access or concentrating cannot avoid the N log N growth rate known to be exacted by nonblocking networks. The chief result under (iii) is that as a constant load is spread over N terminals, then the number X of crosspoints needed to keep loss less than $\epsilon > 0$ need grow only linearly with N, at a rate dependent on $\epsilon$, while the usage (erlangs carried per terminal) goes to zero.*

## I. INTRODUCTION

The relationships between traffic carried and traffic lost, between load and loss, have always been at the center of interest in telephone traffic theory. Since the time of Erlang,[1] over fifty years ago, the principal problems of traffic theory have been analytical: to predict mathematically, from the structure and mode of use of a switching or connecting network, and from the assumed stochastic behavior of the customers, how much traffic the network will *carry* on the average,

and how much it will *lose* as a result of blocking, overload, suboptimal routing, or incomplete searches for paths. As telephone networks have become larger, two more design parameters of interest have emerged and now command attention: the *size* of the network as measured by the number of crosspoints, and its *complexity* as measured, for example, by the number of stages of switching it has.

The probabilistic principle that it is very unlikely for more than a moderate number of customers to want to talk simultaneously has been the theoretical basis of traffic theory since its start. We can view it as an unrefined analog of the principle in information theory that separates a relatively small class of events that exhaust most of the probability from a remaining large class of very unlikely events. This principle has led quite naturally to the use of concentrators, and of networks in which blocking, mismatch, and overflow all can and do occur as it were by design. It is a function of traffic theory to articulate this principle in mathematical models for operating telephone networks, and to use such models to examine its implications for the growth and complexity of networks, as well as their loads and losses.

Even for the simplest stochastic models, progress with these tasks and problems has been very slow because of the combinatorial complexity of the network, the very large number of network states, and the lack of approximate methods. Thus, it is particularly important to find relationships between load, loss, size, and other network parameters that are simple, universal, and useful, even for very large networks. They should be simple in, for example, *not* requiring solution of very high-order systems of equations, universal in being relatively independent of network structure, and useful in providing inequalities, estimates of performance, and information about the growth of cost and complexity with network size.

In this paper we try systematically to sketch out some of these relationships and associated ideas. The results are of necessity spotty, and no claim is made of completeness or originality, only of rigor. Three principal topics are taken up here: (*i*) a load-loss formula, linking some half-dozen network and performance parameters by a rational function; (*ii*) lower bounds on the number of crosspoints in a network; (*iii*) asymptotic results about blocking, growth, and complexity of selected network structures in the limit of passage from finite sources to Poisson arrivals, with total offered traffic held constant.

## II. SUMMARY

The organization of the sequel is as follows: By way of some background, we start with discussions of blocking, loss, concentration, etc., and of their relation to the basic principles of telephone traffic theory and engineering. After various preliminary sections on model-

ling, we call attention to a (known) generalized Erlang formula that connects some of the important parameters of an operating network. We note its technical consequences, and use them repeatedly in the rest of the paper.

Next we take up the problem of the growth of the number of crosspoints with the number $N$ of terminals. The question we try to answer is this: When can the $N \log N$ order of growth necessary for nonblocking networks be reduced by allowing a fixed, small probability of blocking, using, for example, concentrators, or other forms of incomplete access? The answer is that it cannot, unless we consider a familiar, special kind of low traffic limit in which line usage vanishes. It is shown, quite generally, that networks arranged in stages *must* grow like $N \log N$ if certain (very reasonable and mild) traffic, access, and symmetry conditions are met. This result, similar to known results for nonblocking networks, implies that neither judicious concentration nor a nonzero loss can lower the order of growth from the $N \log N$ exacted by the nonblocking case.

The final sections describe various large networks with a simple structure vis-à-vis blocking; their loss probability can be calculated exactly in spite of the astronomical number of states. These networks are based on such structures as trunk groups, frames, and remote concentrators, all familiar to the traffic engineers. We are interested in studying these exact solutions as we let the network grow while keeping the total traffic constant; this kind of growth amounts to adding more and more customers, each of whom contributes less and less traffic, and results in a passage from finite to infinite sources (a Poisson process of arrivals) at constant offered load. The blocking formulas can be studied in this limit, and they lead to close connections with the classical Erlang function, $E(c, a)$. As an application, we can give methods of synthesizing very large networks with prescribed blocking probabilities. In particular, as a constant offered load is spread over more and more customers, the number of crosspoints sufficient to achieve less than $\epsilon$ in blocking need grow only linearly with the number of customers.

There is a bibliography of background reading and related work following the references.

## III. STATEMENT OF RESULTS

### *3.1 Generalized Erlang formula*

Using some standard "dynamic" assumptions[2] to describe random traffic, we show that half-a-dozen parameters, all characteristic of network size and performance, are related by a simple, rational function. These parameters are:

$N$ = number of terminals on a side of a two-sided network
= number of inlets = number of outlets
$m$ = (mean) carried load = equilibrium average number of calls in progress
$\lambda$ = calling rate per pair of idle customers
$bl$ = probability of blocking, from the "wire-chief's" point of view
$\sigma^2$ = variance of number of calls in progress,

and the formula states that,[2] very simply,

$$1 - bl = \frac{1}{\lambda} \frac{m}{(N-m)^2 + \sigma^2}. \tag{1}$$

For some purposes the parameters

$p = m/N$ = line usage = erlangs carried per inlet (outlet)
$a = \lambda N^2$ = total offered load (when everyone is idle)

are more significant or convenient, and using them the formula is recast as

$$a(1 - bl) = \frac{m}{(1-p)^2 + \sigma^2 N^{-2}}.$$

Indeed, there are many ways of twisting and inverting the basic formula, each one illuminating some special aspect; several such will appear later. It can be shown that if $N \to \infty$ while $a = \lambda N^2$ remains constant, then $p$ and $\sigma^2 N^{-2}$ go to zero, and we have Erlang's original result, the "tautology"

$$a(1 - bl) = m$$

or

$$\text{blocking} = \frac{\text{load lost}}{\text{load offered}}.$$

The formula (1), really a generalization of Erlang's loss formula, is useful for the following applications:

(*i*) Order of magnitude estimates

(*ii*) Asymptotic analyses for large networks, with or without a passage from finite to infinite sources

(*iii*) Bounds on the number of switches in a network

(*iv*) Growth and complexity bounds for networks with given load and loss constraints

(*v*) Synthesis of networks having prescribed parameters.

All these applications are illustrated in the text that follows.

### *3.2 Concentration*

The principle that high occupancy states are very unlikely has

suggested the use of concentration, and it is pertinent to assess the effect and value of concentration in large networks. Some results of this kind are in the next three subsections, and they warrant these conclusions:

(*i*) Practical networks must grow like constant $N \log N$, whether there is concentration or not. Concentration affects primarily the value of the constant, and of course the blocking and the carried load.

(*ii*) The extent of possible concentration is limited by the loss and the carried load. As might be expected, higher line usage and lower blocking imply less concentration.

### 3.3 Growth without concentration

In the prototypical networks without concentration (e.g., those made of stages of square switches), the number of crosspoints for $N$ customers must grow like $N \log N$ no matter what load is carried. Several arguments are given for similar lower bounds, some purely combinatorial, others involving traffic concepts and parameters. In particular, if blocking is to be kept less than $\epsilon$, and total traffic $a = \lambda N^2$ is kept constant while $N$ increases, the requisite networks must grow like $N \log N$ if they are made of stages of square switches: especially, for $s$ stages

$$X = \text{number of crosspoints} \geq N \log N + sN + \log(1 - \epsilon) - a.$$

### 3.4 Growth with full access, allowing concentration

Some simple and mild combinatorial properties, possessed by all practical networks, mandate an $N \log N$ order of growth in the number $X$ of crosspoints, even when concentration is permitted. A network provides "full access" if every inlet can reach every outlet by some path. A network is said to be "arranged in stages" (or "made" of stages) if its terminals are partitioned into sets $T_1, T_2, \cdots, T_{s+1}$, such that $T_1$ consists of the inlets, $T_2$ to $T_s$ are sets of internal nodes or junctors, and $T_{s+1}$ are the outlets, with crosspoints placed only between $T_i$ and $T_{i+1}$, $i = 1, \cdots, s$. (See Fig. 1). Here $s$ is the number of stages, and every call traverses each $T_i$ exactly once, in the specified order or its reverse. Finally, a network arranged in stages is called "symmetric" if it looks the same from each terminal in any given $T_i$; we content ourselves with this informal definition here; a precise one can be given in terms of group theory.[3]

We prove this fundamental telephonic fact: A symmetric network that provides full access and is arranged in stages must have at least

$$en \log N \quad e = 2.71828 \ldots$$

crosspoints, where $N$ is the number of inlets (outlets, too), and $n$ is the "neck size," defined as the size of the smallest $T_i$:

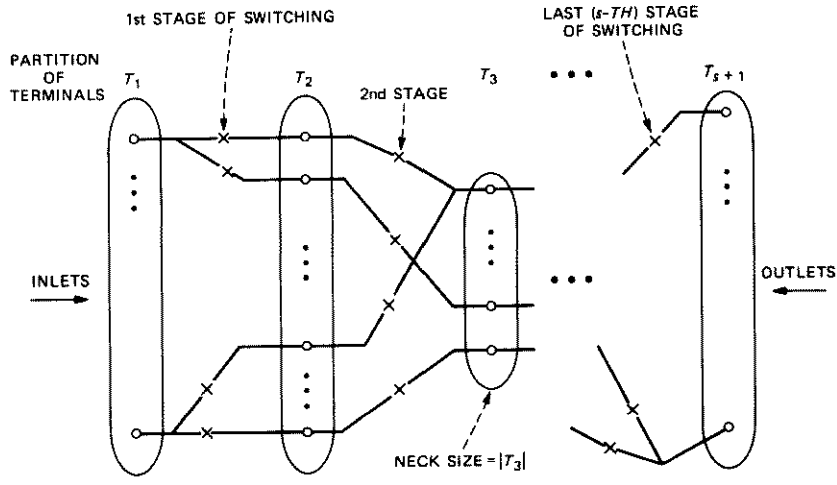



Fig. 1—Network arranged in stages.

$$n = \min_{1 \le i \le s+1} |T_i|, \quad |A| = \text{cardinality of } A.$$

The ratio $n/N$ is a global measure of concentration or expansion. If, in particular, all the $T_i$ are equinumerous, as happens if the network is made of stages of square switches, then the neck size is $N$, and there must be at least $eN \log N$ crosspoints, regardless of the traffic characteristics.

### *3.5 Growth without full access, allowing concentration*

The condition of full access is so reasonable that few engineers would consider a network that lacks it. Nevertheless, probabilistic arguments yield $N \log N$ lower bounds for the crosspoint count $X$ even when this condition is dropped. The point is that if the network is to carry a reasonable load, the "neck size" $n$ cannot be too small; especially, it follows from the Erlang formula (1) that $n$ must exceed $(1 - \sqrt{1-p})\,N$ for line usage $p = m/N$. Similar lower bounds involving also the required loss can be derived. These lower bounds put a limit on how much one can concentrate (measuring global concentration by neck size), and they lead to $N \log N$ lower bounds for $X$ even when there is not full access. For example, any network arranged in stages has

$$X \ge \frac{1}{2} e(1 - \sqrt{1-p})\, N \log N + 0(N)$$

if each customer's line carries $p$ erlangs. Thus any sequence of networks that grow in the strong sense that $p$ is bounded away from zero must grow like $N \log N$, if they are arranged in stages.

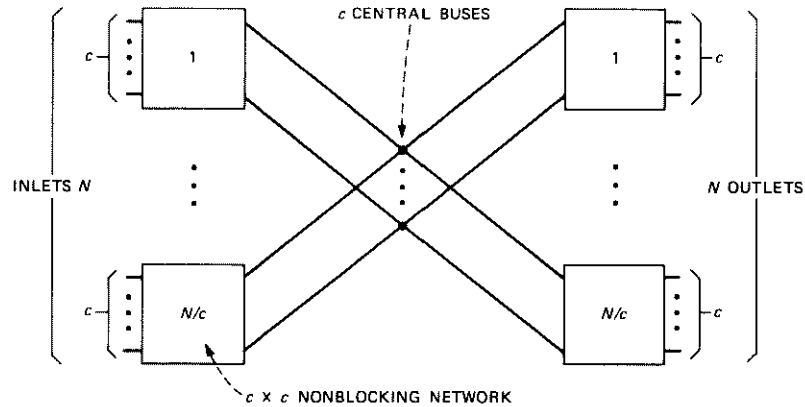



Fig. 2—Central bus network.

### *3.6 Asymptotic results*

Three network structures lead, in the model to be used, to statistical equilibrium equations that have the "product form" solution, and so all their interesting parameters can be calculated exactly from a partition function, and their behavior in the limit $N \to \infty$, $a = \lambda N^2 =$ constant, studied. They are (see Figs 2 through 4.)

(*i*) The central bus concept—two large $N$-to-$c$ nonblocking concentrators back to back, with $c$ central buses

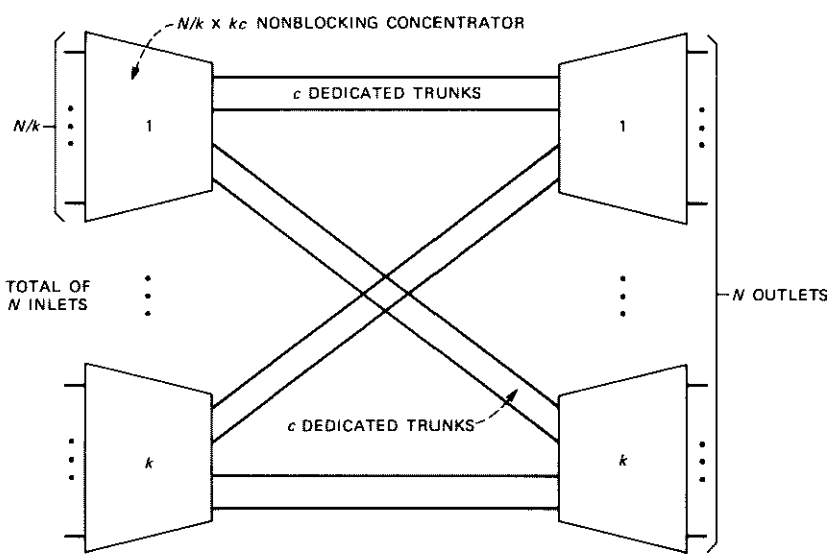



Fig. 3—Frame network.

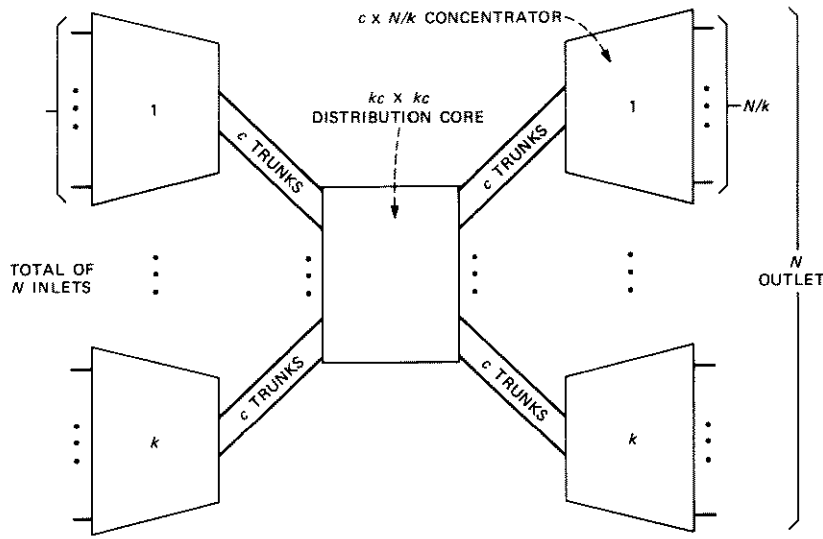



Fig. 4—Remote concentrator network.

(*ii*) The frame concept—large concentrators connected pairwise by dedicated groups of $c$ trunks

(*iii*) The remote concentrator concept—large concentrators, each connected to a central distribution core by its own group of $c$ trunks.

In the "weak" limit as $N \to \infty$, $a = \lambda N^2 =$ constant, the first two structures are (not surprisingly) closely connected to Erlang's formula for loss: their probabilities of loss are bounded above by, and approach, the value $E(c, a)$. The third is more subtle and elusive, since loss can occur in either one of two relevant trunk groups, but also more interesting. We can find $E(\cdot, \cdot)$-type bounds on the loss, but the question is whether asymptotically

$$\text{loss} \leq 1 - (1 - b)^2 + o(1),$$

where $b$ is the chance that all $c$ trunks on one (any) concentrator are busy, in the limit, remains open. This quadratic upper bound is what the loss would be in the limit if the two relevant trunk groups were independent, each with "blocking" $b$. Very few of the many "blocking polynomial" approximate formulas used in practice have been vindicated by so much as an inequality proved in a dynamical model.

It follows from our analyses that for each $\epsilon > 0$, and each of the three kinds of network structure considered, there exist arbitrarily large networks of that kind, with loss less than $\epsilon$, and total offered traffic $a = \lambda N^2 =$ constant, whose crosspoint count $X$ grows at most *linearly* with $N$. We view this as a limit result in the "weak" direction, since a constant amount of traffic $a = \lambda N^2$ is being divided up among

$N$ lines, $N \to \infty$. The carried loads converge, and so the usage $p = m/N$, the erlangs carried per line, goes to zero. This linear growth is not inconsistent with the $N$ log $N$ orders mentioned earlier; the latter follow from combinatorial properties, or are the result of a different "limiting direction" in which $p$ is bounded below away from zero. The above weak limit result is inappropriate for cases in which, as the network grows, each new customer is to supply a fixed number $p_0$ of added erlangs carried; this latter situation enjoins $N$ log $N$ growth. All the results about linear growth are given rigorous proofs, for the Markov process models adopted, by a passage from finite to infinite sources. In particular, no independence or other ad hoc assumptions are made to simplify the blocking estimate.

## IV. COMPROMISES AND TRADE-OFFS

No matter what technology is used to build it, whether Strowger switches, crossbars, solid-state crosspoints, or time-division, the design of a telephone connecting network is inevitably a compromise between the competing criteria of cost and performance. Restricting attention to the traffic and operational aspects, it is nearly a truism that an overengineered network rich in switches will given unblemished service at an unacceptable cost, while one meagerly endowed with switches can only provide poor service at bargain cost. The trick the switching engineer must perform is to come up with designs that avoid these naive extremes.

The engineer's task is also affected by more subtle considerations, such as the following: the same pool of switching gear can be organized in an efficient way that is combinatorially optimal for connecting many pairs of customers in different patterns, i.e., realizing many assignments readily. Unfortunately, these efficient ways involve many stages and so are usually very complex and difficult to control, because putting up a call or removing one requires a lot of information and many decisions. Or the same gear can be hooked up in an inefficient, simple network of a few stages, which is easy to operate, but since it lacks combinatorial power it will have noticeably higher loss. Examples can be found in systems in which equipment is dedicated to handle certain geographically defined kinds of traffic (see Fig. 3).

Obviously, then, there are many trade-offs available to the designer between complexity, cost, control, and the various performance parameters, such as load and loss. Part of what traffic theory must provide, as a proper theoretical underpinning to network design, is an account of the options that are available (or unavailable) in the way of equipment, load, control and structural complexity, growth, and incurred loss. Such an account would describe the achievable regions in parameter space, some of the outer limits of possible designs, and the

achievable rates of growth of various indices of performance and complexity.

In the past such information for the large networks, which are of chief interest, has been sparse and difficult to obtain by means other than simulation and admittedly fuzzy theory. Thus, it is pertinent to have an account that retains its accuracy and force as the size of the network increases without bound. To give a specific example of this kind of information, we can put the following questions: For $x$, $p$, and $\epsilon$ specified positive numbers, are there arbitrarily large networks that have blocking probability at most $\epsilon$, line usage at least $p$, and a number of crosspoints per terminal at most $x$? (To define blocking, let us assume for definiteness that call arrivals are random from finite sources by pairs, and holding times exponential—a "usual" model, that no one can quarrel with.) If there are such networks, how complex are they? That is, how fast does their complexity grow as measured by number of stages, amount of data needed to select a route, etc.?

## V. THE VALUE OF BLOCKING

It is widely believed among telephone switching engineers that a positive probability of blocking is worth a great deal of switching and control equipment. Probably for this good reason alone, the famous nonblocking networks first invented by Charles Clos[4] have never been utilized on any but a small scale. Put another way, the canard says that to eliminate the last little bit of blocking will take an inordinate increase in equipment. But precisely what does the word "inordinate" mean here? Can we replace it by a mathematical function $E(b)$, which increases as the blocking $b$ decreases, and whose interpretation is somehow that to achieve blocking $b$ you need at least $E(b)$ in equipment? And how is $E(b)$ related to the structure of the network, to its size measured by number $N$ of terminals on a side?

The remark that started the preceding paragraph is really only the tip of the iceberg: it is not exaggerating to claim that the mathematical basis of traffic engineering is the observation that very likely only a moderate number of telephone customers will want to talk to each other at the same time. The engineer provides enough lines, junctors, switches, trunks, etc., to take care of this overwhelmingly probable case, plus maybe a little extra for hedging his bets. The traffic theorist provides probabilistic models that make precise the meanings of "average," "likely," and "probable" in this setting. Of course the loads and losses the engineer seeks or achieves are subject to correction from customers' complaints, public commissions, and supervisors. But no matter what the numbers may be, the principle of not having to provide for the very unlikely events is universally accepted by all of those concerned. This principle amounts to public acceptance of pos-

itive blocking probability for public telephony; it stops designs from passing the inevitable knees in the cost curves, beyond which costs grow very fast for each increment in desired performance.

Naturally, then, we are interested in quantitative and mathematical expressions of the above principle. One can sensibly ask how much switching equipment can actually be saved by allowing a blocking probability of $\epsilon > 0$. Where in the system can you (and should you) save it? In particular, how is allowing a blocking of $\epsilon$ related to the rate of growth of the switching gear incurring that blocking as the number $N$ of terminals gets large? It is known[5] that if $\epsilon = 0$ the needed gear in crosspoints is bounded above and below by

$$\text{const. } N \log N.$$

These results are purely combinatorial, and involve neither probability models nor traffic parameters. But is $N \log N$ still the right order of growth when $\epsilon > 0$, and the problem is posed in the context of our "usual" model, with a traffic parameter entering, such as the calling rate $\lambda$ per idle inlet-outlet pair? The answer depends on just how $\lambda$ varies relative to $N$ as $N \to \infty$. If it is constant or decreases so slowly that the usage $p$ is bounded away from zero, then $N \log N$ growth is necessary; if $\lambda N^2$ remains constant, then for any $\epsilon > 0$, $X$ need grow only linearly with $N$.

## VI. THE KINDS AND ORIGINS OF LOSS

When we think about the probability of blocking in a network, it is useful to ask where and how it originates. At high levels of offered load, most of the loss incurred might be due primarily to frequent outright overloads of critical "bottlenecks," even without the occurrence of any combinatorial niceties such as mismatch. At lower levels of offered load, the fraction of loss owing to overloads is very small because the high occupancy states have very small probability, and most of the loss is due to mismatch in states of moderate occupancy and high total probability. It is not always possible to draw this distinction exactly in practice, nor is it necessary. It is important for theoretical purposes, though, because it suggests exactly analyzable large network structures (and models for them), in which the blocking is either all overload, or overload with certain simple kinds of mismatch. Such models can be used to study the trade-offs among the traffic and growth parameters; several are described in the latter half of this work.

## VII. THE VIRTUE OF CONCENTRATION

Let it be agreed that for many terminals $N$ and small calling rate $\lambda$ per idle terminal-pair, the chance that many customers will want to

talk simultaneously is very small; if the principle of not providing for the very unlikely event is to be taken seriously, how should switching equipment be arranged? A standard and traditional method is to *concentrate* traffic.

The most efficient networks known today concentrate traffic from very many lightly loaded terminals into a heavily loaded central distribution core, in which link occupancy may run as high as 0.8 to 0.9, and then expand it back out in an inverse manner. The number of terminals entering the core is typically much smaller than the number of inlets, a feature that leads to a natural bottleneck or what we later call a "neck size." The blocking incurred can be thought of as being of two kinds, or arising from two sources: concentrator blocking, the inability of free inlets or outlets to get a line to the distribution core, and internal blocking in the core itself. Each of these sources of loss may in turn be due mostly to overload or to mismatch.

Thus, to reap the economic and operational values consequent on allowing blocking, the possible or maximum numbers of calls in progress at various places in the network are intentionally limited so as to save switching gear, the argument being as before that while having more calls in progress at these places is possible, it is so (or sufficiently) unlikely that there is no point in providing for it. We ask, how effectively can such limits curb the growth without impairing service?

Now the known nonblocking networks achieve their perfect operation by systematic *expansion*, the provision of more paths than can actually be used at one time; this is the antithesis of the concentrator-cum-distribution core idea usually used in practice. These nonblocking networks exhibit $N \log N$ growth, as they must. Since concentration is the opposite of expansion, it should lead to a saving in crosspoints. How big a saving is it? Especially, when can concentration reduce the order of growth to something slower than $N \log N$?

## VIII. ASYMPTOTICS FOR LARGE NETWORKS

In seeking to answer some of the preceding questions, we shall examine the behavior of network parameters as the number $N$ of terminals on a side becomes arbitrarily large. To fix ideas we begin with some thought-experiments that will lead to more specific questions.

Accepting for a moment the conventional wisdom that all efficient large networks use concentrating switches, we imagine a sequence of such networks with more and more terminals, and ask: What can happen to the efficiency of these networks as they grow? Is it possible to keep the loss below a specified amount without having the crosspoint count or the number of stages grow very fast? Are there large networks which, though they may not be in the running as red-hot field designs

for a particular technology, nevertheless are of interest because their load, loss, and cost can be easily and accurately calculated or estimated? If there are such networks, what combinatorial features account for the ease of calculation? Where are "most" of the crosspoints, in the concentration stages, or in the distribution core?

## IX. LIMIT DIRECTIONS

Needless to say, some care must be taken in carrying out the analyses needed to answer these questions. For most purposes, it is enough to make more exact the way in which the traffic and performance parameters are to vary as $N$ grows; usually some group or function of them is constrained to stay in a given set. Such constraints define "directions" in which limits or other asymptotics are being sought, and they provide useful ways of looking at the performance of very large networks. Two such directions, leading to very different growth rates, will be of interest here.

### *9.1 The "weak" limit*

For example, we can let the offered load per idle pair $\lambda$ get small and $N$ get big, so that $a = \lambda N^2$ is a fixed constant. This amounts to letting the process of attempted calls become Poisson with rate $a$; the corresponding limit process is sometimes called a "passage from finite to infinite sources," and in traffic theory is often associated with familiar notions such as the Poisson approximation to the binomial distribution. We shall show by examples that some interesting limits of this kind exist and can be evaluated, leading to functions and concepts well-known in traffic theory, such as the Erlang loss $E(c, a)$. Such calculations lead to information about the growth rate of cost and complexity for large networks that have specified load and loss.

### *9.2 The strong constraint*

A very different condition, to be used later, is that the usage $p = m/N$ of our sequence of growing networks be bounded away from zero. It says roughly that each new customer, as $N$ grows, adds a fixed amount of carried load to the network, at least. This condition, incompatible with the "weak" limit, leads to $N \log N$ growth in crosspoints, and is not, as far as we know, associated with any limits in distribution, the way the weak limit is. That is why we call it a condition and not a limit. It is physically natural for networks to grow in this manner, and so this is an important condition to consider.

## X. NETWORK STATES

We shall use a model for the structural and combinatorial aspects of

a connecting network. This model arises by considering the network structure to be given by a graph $G$ whose vertices are the terminals of the network, and whose edges represent crosspoints between terminals by pairs, with some of the terminals designated as inlets or outlets. Calls in the network are described by paths on $G$ from an inlet to an outlet. Thus, a connecting network $\nu$ is a quadruple $\nu = (G, I, \Omega, S)$, where $G$ is a graph depicting networking structure, $I$ is the set of vertices of $G$ which are inlets, $\Omega$ is the set of outlets, and $S$ is the set of permitted or physically meaningful states. It is possible that $I = \Omega$ (one-sided network), that $I \cap \Omega = \phi$ (two-sided network), or that some intermediate condition obtain, depending on the "community of interest" aspects of the network $v$. Variables $w$, $x$, $y$, and $z$ at the end of the alphabet denote states, while $u$ and $v$ denote a typical inlet and a typical outlet, respectively.

A possible state $x$ can be thought of as a set of disjoint chains on $G$, each joining $I$ to $\Omega$. Not every such set of chains need represent a state in $S$: wastefully circuitous chains may be excluded from $S$. The set $S$ is partially ordered by inclusion $\leq$, where $x \leq y$ means that state $x$ can be obtained from state $y$ by removing zero or more calls. It is reasonable that if $y$ is a state and $x$ results from $y$ by removal of some chains, then $x$ should be a state too, i.e., $S$ should be closed under "hangups." It can be seen from this requirement that the set $S$ of permitted states has the structure of a semilattice, that is, a partially ordered system whose order relation is definable in terms of a binary operation $\cap$ that is idempotent, commutative, and associative, by the formula $x \leq y$ iff $x = x \cap y$. Here for $x \cap y$ we can simply use literal set intersections: $x \cap y$ is exactly the state consisting of those calls and their respective routes that are common to $x$ and $y$.

An *assignment* is a specification of what inlets are to be connected to what outlets. The set $A$ of assignments can be represented as the set of all fixed-point-free correspondences from subsets of $I$ to $\Omega$. The assignments form a semilattice in the same way that the states do, and $A$ is related to $S$ as follows: call two states $x$, $y$ in $S$ equivalent as to assignment, written $x \sim y$, iff all and only those inlets $u \in I$ are connected in $x$ to outlets $v \in \Omega$, which are connected to the same $v$ in $y$, though possibly by different routes. The realizable assignments can then be identified with the equivalence classes of states under $\sim$, and there is a natural map $\gamma: S \to A$, the projection that carries each state $x$ into the assignment $\gamma(x)$ it realizes, i.e., the equivalence class it belongs to under $\sim$.

With $x$ and $y$ states such that $x \geq y$, it is convenient to use $x - y$ to mean the state resulting from $x$ by removing from $x$ all the calls in $y$. Similarly, with $a$ and $b$ assignments such that $a \geq b$, we use $a - b$ to mean the assignment resulting from $a$ by dropping all the connections

intended in $b$. Note that here $x - y$, $a - b$ have their usual set-theoretic meaning.

It can now be seen that the map $\gamma$ is a semilattice homomorphism of $S$ into $A$, with the properties:

$$x \geq y \Rightarrow \gamma(x) \geq \gamma(y)$$

$$x \geq y \Rightarrow \gamma(x - y) = \gamma(x) - \gamma(y)$$

$$\gamma(x \cap y) \leq \gamma(x) \cap \gamma(y)$$

$$\gamma(x) = \phi \Rightarrow x = 0 = \text{zero state, with no calls up.}$$

Not every assignment need be realizable by some state of $S$. Indeed, it is common for practical networks to realize only a vanishing fraction of the possible assignments, and the networks that do realize every assignment, the so-called *rearrangeable* networks, have been the objects of substantial theoretical study. Thus, the image set $\gamma(S)$ of realizable assignments is typically much smaller than the set $A$ in which it is embedded. A *unit* assignment is, naturally, one that assigns exactly one outlet to some inlet, and it corresponds to having just one call in progress. It is convenient to identify calls $c$ and unit assignments, and to write $\gamma(x) \cup c$ for the larger assignment consisting of $\gamma(x)$ and the call $c$ together, with the understanding, of course, that $c$ is "new in $x$" in the sense that neither of its terminals is busy in $x$.

We denote by $A_x$ the set of states that are immediately above $x$ in the partial ordering $\leq$ of $S$, and by $B_x$ the set of those that are immediately below. Thus,

$$A_x = \{\text{states reachable from } x \text{ by adding a call}\}$$
$$B_x = \{\text{states reachable from } x \text{ by hangup}\}.$$

For $c$ new in $x$, let $A_{cx} = A_x \cap \gamma^{-1}[\gamma(x) \cup c]$; $A_{cx}$ is the subset of states of $A_x$ that could result from $x$ by putting up the call $c$, because $\gamma^{-1}\gamma(y)$ is precisely the equivalence class of $y$ under $\sim$. If $A_{cx}$ is empty then we say $c$ is blocked in $x$: there is no $y \in A_x$ that realizes the larger assignment $\gamma(x) \cup c$. It can be seen that with $F_x$ the set of new calls of $x$ that are not blocked, the family $\{A_{cx}, c \in F_x\}$ forms the partition of $A_x$ induced by equivalence $\sim$.

## XI. ROUTING OF CALLS

We shall use a routing matrix $R = (r_{xy})$ as a convenient formal description of how routes are chosen for calls. The class of routing matrices, $R$, can be described thus: for each $x \in S$ let $\Pi_x$ be the partition of $A_x$ induced by the relation $\sim$ of "having the same calls up", or satisfying the same assignment of inlets to outlets; it can be seen that $\Pi_x$ consists of exactly the sets $A_{cx}$ for $c$ free and not blocked

in $x$; for $Y \in \Pi_x$, $r_{xy}$ for $y \in Y$ is to be a probability distribution over $Y$, that is $r_{xx} \geq 0$ and $\sum_{y \in Y} r_{xy} = 1$; $r_{xy}$ is to be 0 in all other cases.

The interpretation of the routing matrix as a method of choice is to be this: any $Y \in \Pi_x$ represents all the ways in which a particular call $c$ (free and not blocked in $x$) *could* be completed when the network is in state $x$; for $y \in Y$, $r_{xy}$ is the chance (or fraction of times) that if call $c$ arises in state $x$ it will be completed by being routed in the network so as to take the system to state $y$. The distribution $\{r_{xy}, y \in Y\}$ indicates how the calling rate owing to $c$ is to be spread over the possible ways of putting up this call. Evidently, such a description of routing could be made time-dependent, and extended to cover refusal of unblocked calls as an option; we do not consider these possibilities here. The problem of choosing an optimal routing matrix $R$ has been worked on at some length.

## XII. STOCHASTIC MODEL

We now recall[2] a stochastic model for the traffic offered to a network. A Markov stochastic process $x_t$ taking values on $S$ can be based on these simple probabilistic and operational assumptions:

(*i*) Holding times of calls are mutually independent variates, each with the negative exponential distribution of unit mean.

(*ii*) If $u$ is an inlet idle in state $x \in S$, and $v \neq u$ is any outlet, there is a conditional probability $\lambda h + o(h)$, $\lambda > 0$, as $h \to 0$, that $u$ attempts a call to $v$ in $(t, t + h)$ if $x_t = x$.

(*iii*) A routing matrix $R = (r_{xy})$ is used to choose routes, as follows: If $c = \{(u, v)\}$ is a call free and not blocked in $x$, then the fraction of times that the system passes from $x$ to $y \in A_{cx}$ if $c$ arises when $x_t = x$ is just $r_{xy}$.

(*iv*) Blocked calls and calls to busy terminals are declined, with no change of state.

It is convenient to collect these assumptions into a transition rate matrix $Q = (q_{xy})$, the generator of $x_t$; this matrix is given by

$$q_{xy} = \begin{cases} 1 & \text{if } y \in B_x \\ \lambda r_{xy} & \text{if } y \in A_x \\ -|x| - \lambda s(x) & \text{if } y = x, \text{ with } s(x) = |F_x| \\ 0 & \text{otherwise,} \end{cases}$$

and the associated statistical equilibrium (or state) equations take the simple form

$$[|x| + \lambda s(x)]p_x = \sum_{y \in A_x} p_y + \lambda \sum_{y \in B_x} p_y r_{yx} \; x \in S,$$

where $\{p_x, x \in S\}$ is the asymptotic distribution of $x_t$. Here $|x|$ denotes the number of calls in progress in $x$, and $s(x)$ is the number of

unblocked idle inlet-outlet pairs in $x$, the possible "successes" in $x$; note that $s(x) = |F_x|$.

## XIII. PARAMETERS OF INTEREST FOR DESIGN AND ENGINEERING

We shall frequently use about a dozen basic parameters, characteristic of the operating network, and important for these reasons: They describe load, cost, and performance, or they can be measured readily, or they arise naturally in the associated traffic theory and are convenient for calculations and asymptotic analyses. For two-sided networks $\nu$, these parameters are the following:

$\lambda$ = calling rate per idle inlet-outlet pair
$N$ = number of terminals (inlets, or outlets) on each side
$bl$ = probability of blocking
$m$ = carried load = expected number of calls in progress
$p$ = usage = $m/N$ = erlangs carried per terminal
$\sigma$ = standard deviation of number of calls in progress
$X$ = total number of crosspoints
$s$ = number of stages (if $\nu$ is arranged in stages)
$a = \lambda N^2$ = total offered load when everyone is idle
$w = \max_{x \in S} |x|$ = maximum possible number of calls in progress
$n$ = "neck size," defined for $\nu$ arranged in stages separating junctor groups $T_1, T_2, \cdots T_{s+1}$, as the cardinality of the smallest $T_i$.

*Remark:* The parameter $\alpha = \lambda N^2$ is a convenient abbreviation for total offered load, especially for certain weak "large network" asymptotics for which $\lambda N^2$ is held constant as $\lambda \to 0$ and $N \to \infty$.

*Remark:* The ratios $w/N$ and $n/N$ are rough *global* measures of concentration, global because there could be, for example, remote local concentrators with a concentration ratio different from each of these. Clearly, $w \leq n$, when both $w$ and $n$ are defined.

Notation: We write $X(\nu)$, $p(\nu)$, etc, whenever it is necessary to express the dependence of a parameter on the network $\nu$.

## XIV. THE PARAMETER SURFACE

In the early[1] applications of traffic theory to trunking problems, a central role was played by Erlang's loss formula, which depended on two parameters, the load $a$, and the number $c$ of trunks. For connecting network studies, though, to take into account at least the size of the network and the "finite source" effect, if not other network features, a modification of Erlang's formula is more suitable. (The finite source effect is a recognition that busy terminals generate no traffic.) Such a formula has been derived in earlier work.[2] We shall exhibit many

useful results that follow from it, or from it together with reasonable but special hypotheses, such as the property of a network that it is made of square switches, or is arranged in stages, or provides full access.

For a two-sided network with $N$ terminals on a side, the load, loss, load deviation, and rate parameter, $\lambda$, are related by the following formula:

Generalization of Erlang's formula to networks:

$$1 - bl = \frac{1}{\lambda} \frac{m}{(N-m)^2 + \sigma^2} \tag{1}$$

$$\alpha(1 - bl) = \frac{m}{(1-p)^2 + (\sigma/N)^2}.$$

*Proof:* $1 - bl$ is the fraction of attempted calls that are not blocked. By the law of large numbers, this fraction is the rate of successful attempts divided by the total rate of attempts, both in equilibrium. For a state $x$ let $s(x)$ be the number of inlet-outlet pairs $(u, v)$ that are not blocked in $x$; "$s(x)$" stands for "successes in $x$." The success rate is then $\lambda \sum_{x \in S} p_x s(x)$ and the total attempt rate $\lambda \sum_{x \in S} p_x (N - |x|)^2$. However, in equilibrium, the rate in equals the rate out, so

$$\lambda \sum_{x \in S} p_x s(x) = \sum_{x \in S} p_x |x| = m.$$

Evidently the total attempt rate can be written as $\lambda[(N-m)^2 + \sigma^2]$, and the general Erlang formula is proved.

*Remarks:* The gist of the Erlang formula is that six of the parameters of interest cannot assume arbitrary values but must lie on a surface described by a simple rational function. It is apparent that similar but more complex formulas can be proved for one-sided networks, or two-sided ones with different numbers of terminals on each side; we shall not consider these, because the basic ideas are the same. Note that blocking, a complicated quantity in the model, is determined solely by the first two moments of the number of calls in progress. Also, if $N$ and $\lambda$ vary so that $(1-p)^2 + (\sigma/N)^2$ approaches 1, the formula approaches the exact form $m = \alpha(1 - bl)$ it has in the "true" Erlang case.[1]

## XV. SOME TECHNICAL RESULTS

The generalized Erlang formula has many useful consequences. Some of them are summarized in the next few results, all of which are additional relationships among the engineering parameters.

*Lemma:* $\dfrac{\sigma^2}{N^2} < p - p^2.$ (2)

*Proof:* Clearly, $E|x_t|^2 = \sum_{x \in S} p_x |x|^2 \geq Nm$. Hence,

$$\sigma^2 = E|x_t|^2 - (E|x_t|)^2 \leq Nm - m^2.$$

*Lemma:* $\dfrac{m}{a(1 - bl)} \leq 1 - p.$ (3)

*Proof:* Lemma (2) implies that

$$p^2 - 2p + 1 + \frac{\sigma^2}{N^2} < 1 - p.$$

By (1) the left-hand side is $m/a\ (1 - bl)$.
Representation of $N$:

$$N = \frac{m + \sqrt{m^2 - \left[1 - \dfrac{m}{a(1 - bl)}\right](m^2 + \sigma^2)}}{1 - \dfrac{m}{a(1 - bl)}}$$

$$= \frac{m(1 + \sqrt{1 - p})}{1 - \dfrac{m}{a(1 - bl)}}, \tag{4}$$

where

$$p = \left[1 - \frac{m}{a(1 - bl)}\right]\left[1 + \frac{\sigma^2}{m^2}\right]$$

and

$$0 < p < 1.$$

*Proof:* $N$ is one of the quadratic roots

$$\frac{m(1 \pm \sqrt{1 - p})}{1 - \dfrac{m}{a(1 - bl)}}. \tag{5}$$

To show that the plus branch is the right one we note that the quadratic (in $y$) function

$$y^2\left[1 - \frac{m}{a(1 - bl)}\right] - 2my + m^2 + \sigma^2 \tag{6}$$

equals $m^2 + \sigma^2$ at $y = 0$, and has a negative minimum at

$$y^* = \frac{m}{1 - \dfrac{m}{a(1-bl)}}.$$

This value is a minimum because the second derivative is

$$2\left[1 - \frac{m}{a(1-bl)}\right] > 2p$$

by Lemma (3). This value is negative because it is

$$m^2 + \sigma^2 - \frac{m^2}{1 - \dfrac{m}{a(1-bl)}} \tag{7}$$

and the Erlang formula (1) implies that

$$\sigma^2 + m^2 = 2mN - N^2\left[1 - \frac{m}{a(1-bl)}\right]. \tag{8}$$

Substituting this into (7) we see that the value is

$$-N^2\left[1 - \frac{m}{a(1-bl)}\right] + 2mN - \frac{m^2}{1 - \dfrac{m}{a(1-bl)}},$$

clearly negative. Thus $y^*$ separates the two real roots of (6). However, Lemma (3) implies that $N > y^*$, so $N$ must be given by the plus sign in (5). It is obvious that $p$ is positive; to show $p < 1$ we divide (8) by $m^2$ and multiply by $1 - \dfrac{m}{a(1-bl)}$ to get

$$p = 1 - \left\{1 - \frac{N}{m}\left[1 - \frac{m}{a(1-bl)}\right]\right\}^2 < 1,$$

which incidentally strengthens Lemma (3) to

$$p < 1 - \frac{m}{a(1-bl)} < 2p.$$

*Lemma: If $\nu$ is such that, at most, $w$ calls can be in progress, then for $w < N$*

$$(i)\ \ bl \geq 1 - \frac{m}{\lambda(N-w)^2}$$

$$(ii)\ \ w \geq N(1 - \sqrt{1-p})$$

$$(iii)\ \ p \leq \frac{2w}{N} - \left\{\frac{w}{N}\right\}^2. \tag{9}$$

*Proof:* In this situation the average calling rate is greater than or equal

to $(N - w)^2\lambda$, so $(i)$ follows from the generalized Erlang formula (1). Thus also

$$\frac{m}{\lambda N^2[(1-p)^2 + \sigma^2 N^{-2}]} \leq \frac{m}{\lambda(N-w)^2}(N-w)^2$$

$$\leq N^2[(1-p)^2 + \sigma^2 N^{-2}] = N^2 \frac{m}{a(1-bl)}.$$

But by Lemma (3),

$$\frac{m}{a(1-bl)} \leq 1 - p$$

and thus $(ii)$ and $(iii)$ via

$$N - w \leq N\sqrt{1-p}.$$

The same argument proves

$$w \geq N[1 - \sqrt{(1-p)^2 + \sigma^2 N^{-2}}].$$

## XVI. NONCONCENTRATING NETWORKS GROW LIKE *N* LOG *N* EVEN IN THE WEAK LIMIT

One way to display the value of concentrating traffic is to show how bad things are when you do not do it. We shall look at a large class of practical networks that have no concentration, and show that to all intents and purposes, these networks grow like $N \log N$ for $N$ terminals. In particular, without concentration, these networks have no way of trading off nonzero or even substantial blocking (up to some $\epsilon > 0$) for slow growth, i.e., slower than $N \log N$, or constant $\times N \log N$ with a constant that depends on usage $p$. Especially, these networks have $N \log N$ growth even in the weak limit $a = \lambda N^2 =$ constant; we show later that, in this special case, linear growth is achievable if concentration is used, even with blocking kept less than $\epsilon$.

Now the prototypical network without concentration is one that is made of *square* switches arranged in stages, and these constitute the class we consider. Since engineers are primarily interested in networks with a high degree of symmetry, which "look the same" from any terminal within a junctor group, or from any inlet, or outlet, we shall restrict attention to networks $\nu$ with these properties:

$(i)$ $\nu$ has $s \geq 1$ stages

$(ii)$ The switches in each stage are square, and identical.

Note that the number of stages is not fixed, and that different stages may have different numbers of switches; also, the interconnection patterns between the stages (in old terminology, the *cross-connect fields*) can represent arbitrary $N$-permutations.

*Theorem: Let $n_1, n_2, \cdots, n_s$ be $s$ divisors of $N$, possibly with repetitions, and consider a network $\nu$ with $N/n_i$ $n_i \times n_i$ switches in its $i$th stage, and blocking $bl < \epsilon$. Then its crosspoint count $X(\nu)$ satisfies*

$$X(\nu) \geqq N \sum_{i=1}^{s} n_i \geq N \log N + N(s + \log(1 - \epsilon)) - a. \tag{10}$$

*Remark:* The number of outlets reachable from an inlet is at most $\prod_{i=1}^{s} n_i$. If every inlet can reach every outlet, then this product exceeds $N - 1$, and in this "usual" case

$$\sum_{i=1}^{s} n_i \geq s + \log \prod_{i=1}^{s} n_i \geq s + \log N \tag{11}$$

and the conclusion of the theorem follows. In the opposite "unusual" case, $\prod_{i=1}^{s} n_i < N$, some calls are permanently blocked, and the network does not provide full access. Thus, one point of the theorem is that even in this combinatorially poor situation, blocking cannot save more than a linearly growing number of crosspoints when $a = \lambda N^2$ is constant or grows at most linearly, so that $X(\nu) = O(N \log N)$.

*Proof:* Only the "unusual" case $N > \prod_{i=1}^{s} n_i$ need be considered. Let $s(x)$ be the number of possible "successes" in $x$, i.e., of inlet-outlet pairs idle and not blocked in state $x$, so that

$$(N - |x|)^2 - s(x)$$

is the number of idle inlet-outlet pairs that cannot be connected in $x$. In the unusual case an idle inlet cannot reach at least $N - \prod_{i=1}^{s} n_i$ outlets. In state $x$, at most $|x|$ of these unreachable outlets are busy; thus there are at least

$$N - \prod_{i=1}^{s} n_i - |x|$$

idle outlets with no path to our test inlet. This being true for each idle inlet, and there being $N - |x|$ idle inlets, the number $\beta_x$ of blocked idle inlet-outlet pairs in state $x$ satisfies

$$\beta_x \geq (N - |x|) \left( N - \prod_{i=1}^{s} n_i - |x| \right),$$

or since $(N - |x|)^2 = s(x) + \beta_x$,

$$s(x) \leq (N - |x|) \prod_{i=1}^{s} n_i.$$

Averaging this inequality with respect to the equilibrium state probabilities $\{p_x, x \in S\}$, and noting that $m = \sum_{x \in S} |x| p_x = \lambda \sum_{x \in S} s(x) p_x$, we find

$$\frac{m}{\lambda} \leq (N-m) \prod_{i=1}^{s} n_i$$

$$\prod_{i=1}^{s} n_i \geq \frac{Nm}{a(1-p)}.$$

Therefore,

$$\begin{aligned} X(\nu) &= N \sum_{i=1}^{s} n_i \\ &\geq N\left(s + \log \prod_{i=1}^{s} n_i\right) \\ &\geq Ns + N \log N + N \log \frac{m}{a(1-p)}. \end{aligned}$$

To find a suitable lower bound on the last term we use the basic generalization (1) of Erlang's formula:

$$\begin{aligned} 1 - bl &= \frac{m}{\lambda(N-m)^2 + \lambda\sigma^2} = \frac{m}{a[(1-p)^2 + \sigma^2 N^{-2}]} \\ &\leq \frac{m}{a(1-p)^2}. \end{aligned}$$

Since the blocking $bl$ is less than $\epsilon$, one finds

$$\log(1-\epsilon) \leq \log(1-bl) \leq \log \frac{m}{a(1-p)} - \log(1-p)$$

$$X(\nu) \geq N \log N + sN + N \log(1-p) + N \log(1-\epsilon).$$

Next we argue that, as in Lemma (2),

$$\sigma^2 N^{-2} \leq p - p^2$$

$$(1-p)^2 + \sigma^2 N^{-2} \leq 1 - p,$$

and so by the Erlang formula (1) again,

$$p = \frac{1-bl}{N} a[(1-p)^2 + \sigma^2 N^{-2}] \leq \frac{1-p}{N} a \leq \frac{a}{N+a}$$

whence

$$\log(1-p) > \log N - \log(N+a) \geq -\frac{a}{N}$$

and the proof is complete.

## XVII. GROWTH OF NETWORKS ARRANGED IN STAGES

Almost all the connecting networks used in practice are *made of*

*stages*, or *arranged in stages*. This property means that the terminals of the network are partitioned into disjoint sets $T_1, T_2, \cdots, T_{s+1}$, which are simply ordered as indicated; the first set $T_1$ consists of the inlets, the next $s - 2$ sets are internal junctors, and the last set $T_{s+1}$ consists of the outlets; crosspoints are placed only from terminals in a given set $T_i$ to terminals in the next set $T_{i+1}$ in the ordering (see Fig. 1). Thus every call in progress is a path from an inlet to an outlet that passes through each set $T_i$ once in the order specified. The crosspoint pattern between successive sets is called a stage of switching, and is representable as a bipartite graph. The sets $T_i$ need not all have the same numbers of terminals; if they do not, there is expansion or concentration. The size of the smallest $T_i$ is called the "neck size," and is of course an upper bound on the number of calls in progress:

$$|x| \leq n = \min_{1 \leq i \leq s+1} |T_i|.$$

We shall restrict attention to *symmetric* networks, in which the network looks the same to every terminal in a given $T_i$, $i = 1, \cdots, s + 1$. A network provides *full access* if every inlet-outlet pair can be connected by a path through the network, with no doubling back allowed. Full access is a natural convenient condition that greatly simplifies arguments, but it is not necessary for proving $N \log N$ growth.

*Theorem: Let $\nu$ be a symmetric network with $N$ inlets (& outlets), with neck size $n$, providing full access through $s$ stages. Then the crosspoint count $X(\nu)$ satisfies*

$$X(\nu) \geq en \log N \quad e = 2.71828 \cdots . \tag{12}$$

*Proof: Let $n_i$, $i = 1, \cdots, s$,* be the number of crosspoints in stage $i$ connected to a junctor between stages $i$ and $i + 1$. By symmetry this number is the same for all such junctors or terminals. If stage $i$ is represented by a bipartite graph, $n_i$ is the degree of each "input" vertex. Thus, if the neck size is $n$, then stage $i$ has at least $nn_i$ crosspoints, and so

$$X(\nu) \geq n \sum_{i=1}^{s} n_i.$$

Since $\nu$ provides full access it must be true that

$$N \leq \prod_{i=1}^{s} n_i;$$

for an inlet can reach no more than $\Pi_{i=1}^{s} n_i$ outlets; so if $N$ exceeded this product there would be some it could not reach. Hence, by the inequality linking arithmetic and geometric means,

$$X(\nu) \geq n \sum_{i=1}^{s} n_i$$

$$\geq ns \left( \prod_{i=1}^{s} n_i \right)^{\frac{1}{s}}$$

$$\geq nsN^{\frac{1}{s}}.$$

But $sN^{\frac{1}{s}}$, viewed as a function of a real variable $s$, has a unique minimum at $s = \log N$, so that

$$X(\nu) \geq n \log N \, N^{\frac{1}{\log N}} = en \log N, \quad e = 2.71828 \cdots.$$

*Remark:* If the neck size is $N$, as it is for networks made of square switches, then for symmetric $\nu$ providing full access

$$X(\nu) \geq e \, N \log N.$$

*Lemma: If $\nu$ has neck size $n < N$, then*

$$n \geq N\left\{1 - \left[\frac{m}{a(1 - bl)}\right]^{1/2}\right\} \geq N\{1 - \sqrt{1 - p}\}. \tag{13}$$

*This result links the neck size n to the performance parameters m and bl and to the traffic parameter a. The second inequality is remarkable in involving only the line usage p; the higher p is to be, the closer the neck size must be to N.*

*Proof:* If $\nu$ has neck size $n$, then at most $n$ calls can be in progress at a time, so that $N - |x| \geq N - n$, and by the Erlang formula (1)

$$1 - bl = \frac{m}{\lambda \Sigma p_x (N - |x|)^2} \leq \frac{m}{\lambda (N - n)^2}$$

$$(N - n)^2 \leq N^2 \frac{m}{a(1 - bl)}.$$

The second inequality follows from Lemma (3); it leads at once to this basic result:

*Theorem: If $\nu$ is a symmetric network arranged in stages, providing full access and with each inlet carrying p erlangs, then*

$$X(\nu) \geq e(1 - \sqrt{1 - p})N \log N. \tag{14}$$

*Proof:* (12) and (13).

Discussion: This inequality says that any symmetric network providing full access has const. $N \log N$ crosspoints, where the constant depends only on the line usage, no matter what blocking is incurred. Using the

first inequality in (13) gives a larger constant, now dependent on loads (offered and carried) and blocking.

*Theorem: Let $\nu$ be a network arranged in stages, not necessarily providing full access. Then*

$$X(\nu) \geq e(1 - \sqrt{1-p})[\tfrac{1}{2}N \log N - \tfrac{1}{2}N \log \lambda + N \log(1 - bl) + N \log p]. \quad (15)$$

*Proof:* As in the proof of Theorem (12) we find

$$X(\nu) \geq ns\left(\prod_{i=1}^{s} n_i\right)^{\frac{1}{s}}. \quad (16)$$

Next, the averaging argument of Theorem (14) gives

$$\prod_{i=1}^{s} n_i \geq \frac{Nm}{a(1-p)} = \frac{p}{\lambda(1-p)}.$$

For $b > 0$, $sb^{\frac{1}{s}}$ assumes a unique minimum at $s = \log b$, and $(b)^{\frac{1}{\log b}} = e = 2.71828, \cdots$, so by (16),

$$X(\nu) \geq ne \log \frac{p}{\lambda(1-p)}$$

$$\geq e(1 - \sqrt{1-p})N[\log p - \log \lambda - \log(1-p)]. \quad (17)$$

The generalized Erlang formula (1) can be put in the forms

$$\lambda N(1 - bl)[(1-p)^2 + \sigma^2 N^{-2}] = p \quad (18)$$

$$\lambda N(1 - bl)(1-p)^2 + (1-p) - 1 + \lambda N(1 - bl)\sigma^2 N^{-2} = 0. \quad (19)$$

The second form is a quadratic equation for $1 - p$ whose solution, picking the plus branch, is

$$1 - p = \frac{\sqrt{1 + 4\lambda N(1-bl)[1 - \lambda(1-bl)\sigma^2 N^{-1}]} - 1}{2\lambda N(1-bl)}.$$

By (18) above, the quantity (factor) $1 - \lambda(1 - bl)\sigma^2 N^{-1}$ under the square root is equal to

$$1 - p\frac{\sigma^2}{(N-m)^2 + \sigma^2}$$

and lies strictly between 0 and 1. Therefore,

$$1 - p < \frac{\sqrt{1 + 4\lambda N(1-bl)} - 1}{2\lambda N(1-bl)}.$$

Let $y = 2\lambda N(1 - bl)$ for short, so that

$$1 - p < \frac{\sqrt{1 + 2y} - 1}{y}.$$

Hence,

$$\log(1 - p) < \log \frac{\sqrt{1 + 2y} - 1}{y} = \log \frac{2}{\sqrt{1 + 2y} + 1}$$

$$-\log(1 - p) > \log(\sqrt{1 + 2y} + 1) - \log 2$$

$$> \frac{1}{2} \log(1 + 2y) - \log 2$$

$$> \frac{1}{2} (\log 4 + \log \lambda + \log N + \log(1 - bl)) - \log 2.$$

Returning now to formula (17) we find

$$X(\nu) \geq e(1 - \sqrt{1 - p})N[\log p - \tfrac{1}{2} \log \lambda + \tfrac{1}{2} \log N + \tfrac{1}{2} \log(1 - bl)].$$

*Remark:* Theorem (15) shows that full access is not necessary for $N \log N$ growth; it just makes the constant bigger and the argument simpler.

*Theorem: Let $\nu_N$ be a sequence of networks on $N$ terminals arranged in stages, and such that*

$$\begin{aligned} &(i)\ p(\nu_N) \geq p_0 > 0 \\ &(ii)\ bl(\nu_N) \leq \epsilon \\ &(iii)\ \lambda(\nu_N) \text{ is bounded.} \end{aligned} \tag{20}$$

Then as $N \to \infty$

$$X(\nu_N) \geq \frac{e}{2} (1 - \sqrt{1 - p_0})N \log N + O(N).$$

*Proof:* Immediate from (15).

*Remarks:* Theorems (15) and (20) of course apply also to the networks made of stages of square switches considered in Theorems (10) and (11). However, it should be noted that the possible traffic asymptotics in the two theorems are different, although they might overlap. In (11) $a = \lambda N^2$ grows at most linearly, while in (20) it grows at least linearly; in (11) $a = \lambda N^2$ could be identically a constant (the weak limit case), so that $\lambda$ and $p$ both go to zero, and $X(\nu)$ grows like $N \log N$ instead of linearly as it might [see Theorem (24)]; in (20), on the other hand, $p$ is bounded away from zero, hence $\lambda N$ is also, so $a = \lambda N^2$ increases at least linearly, and $X(\nu)$ grows like constant $\times$ $N \log N$, with constant depending on the lower bound for $p$. The point is that the absence of concentration exemplified in the square switches compels $N \log N$ growth even in the weak limit ($a = \lambda N^2$ constant, $p$ vanishing), while

in general if concentration is allowed it takes the "strong condition" $p \geq p_0 > 0$ to force $N \log N$ growth.

## XVIII. ANALYZABLE LARGE NETWORKS

We turn now to the study of three simple patterns or structures for networks with concentration. Interest in them arises from the fact that their loads, losses, and complexity can be calculated or rigorously bounded for arbitrarily large values of $N$. Most of them embody features, such as frames and concentrators, which are familiar in telephone network design, and some provide tenuous links to previous approximate blocking formulas based on independence assumptions. These formulas suggest inequalities stating that certain natural "blocking polynomials" are in fact upper bounds on the probability of blocking; their proof or disproof has eluded us so far, but they are worth mentioning nevertheless.

## XIX. CENTRAL BUSES CONCEPT

A useful extreme case is a network that, like the trunk group, has no blocking states until a certain number $c$ of calls in progress is reached, at which point *all* calls are blocked. One way to build such a network is to concentrate both the $N$ inlets and the $N$ outlets down to $c$ terminals in a nonblocking way, and then to put a $c$-by-$c$ nonblocking network in between, as shown in Fig. 5. A better way results when we note that the central network is superfluous; all you need are *c central buses* with "expanding" networks on each side such that any idle terminal can reach an idle bus. An arrangement of this kind is shown in Fig. 2, in which each bus has an appearance on every (inlet or outlet) subnetwork; when these are nonblocking, a theoretically useful solvable case results. We call such networks *central bus networks.*

*Remark:* Clearly, the central bus idea for networks springs right out of the idea that in a large network with lightly loaded lines, only a

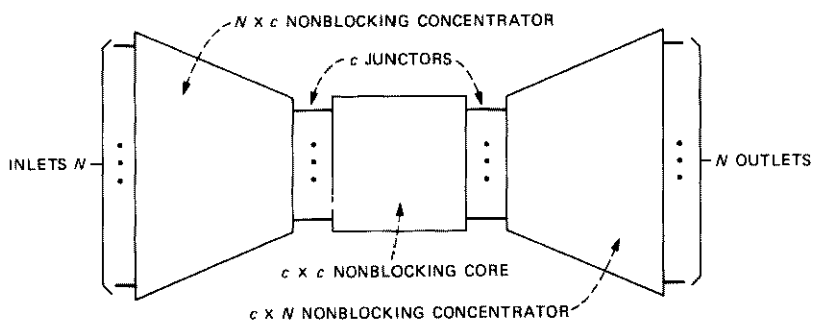



Fig. 5—Network nonblocking up to $c$ calls.

moderate number of customers will be talking. The thought is this: if the low and moderate occupancy states have all the probability, let's use all our switching gear to make *them* as nonblocking as possible, and just ban the unlikely high occupancy states altogether. Accordingly, the designer guesses or calculates what that moderate occupancy might be, on the average, and provides some larger number $c$ of central buses, with nonblocking access for everyone.

*Remark:* The central bus network is also a good candidate for the best disposition of a fixed number $X$ of crosspoints at very low traffic $\lambda$. For it is known[2] that

$$\text{blocking} = \text{const.}\ \lambda^m + o(\lambda^m) \quad \text{as} \quad \lambda \downarrow 0,$$

where

$$m = \min_{x \in S} \{|x|\text{: some call is blocked in } x\}.$$

Thus, of all networks made out of $X$ crosspoints, the ones for which $m$ is largest will have asymptotically least blocking at low traffic, as $\lambda \to 0$.

For traffic purposes, the number of calls in progress is an adequate notion of state for central bus networks. Under our assumptions, the equilibrium probabilities $p_n$ of $n$ calls up satisfy

$$p_n = p_0 \frac{\lambda^n}{n!} (N-n+1)^2(N-n+2)^2 \cdots N^2$$

and so the blocking is

$$bl = \frac{\dfrac{\lambda^c}{c!}(N-c)^2(N-c+1)^2 \cdots N^2}{N^2 + \displaystyle\sum_{j=1}^{c} \frac{\lambda^j}{j!}(N-j)^2(N-j+1)^2 \cdots N^2}.$$

We introduce the parameter $a = \lambda N^2$ by writing this as

$$bl = \frac{\dfrac{a^c}{c!}\left(1-\dfrac{c}{N}\right)^2\left(1-\dfrac{c-1}{N}\right)^2 \cdots \left(1-\dfrac{1}{N}\right)^2}{1 + \displaystyle\sum_{j=1}^{c} \frac{a^j}{j!}\left(1-\frac{j}{N}\right)^2\left(1-\frac{j-1}{N}\right)^2 \cdots \left(1-\frac{1}{N}\right)^2}. \tag{21}$$

If we extend each product in the denominator all the way up to $c$, and replace the 1 by the product in the numerator, we increase the formula to exactly the Erlang loss function $E(c, a)$. Thus,

$$bl < E(c, a). \tag{22}$$

A similar argument shows that the traffic carried, $m$, satisfies

$$m < a[1 - E(c, a)]. \tag{23}$$

Thus, for central bus networks the load and loss are bounded above by the corresponding Erlang load and loss for $c$ trunks and incoming traffic $a$.

## XX. LINEAR GROWTH IS POSSIBLE IN THE WEAK LIMIT

The central bus concept also leads to an asymptotic estimate of what is possible in the way of network growth. We prove

*Theorem: For every $\epsilon > 0$, there is an integer $c$, and a sequence $\nu_N$ of networks on $N$ terminals with $c$ central buses such that as $N \to \infty$ with $a \equiv$ constant*

$$\begin{aligned} &(i)\ bl(\nu_N) < E(c, a) < \epsilon,\ bl(\nu_N) \to E(c, a) \\ &(ii)\ m(\nu_N) < a[1 - E(c, a)],\ m(\nu_N) \to a[1 - E(c, a)] \\ &(iii)\ X(\nu_N) \le 136N \log_2 c + O(c) \\ &(iv)\ s(\nu_N) \le 4(1 + \log_2 c). \end{aligned} \tag{24}$$

*Remarks:* This result says that there exist arbitrarily large networks with specified blocking whose growth in crosspoints is linear (with slope dependent on blocking), whose complexity in number of stages is logarithmic, and whose load approaches a constant. Here the growth in "size" $N$ is accompanied by a diminution in the offered load $\lambda$ per idle pair, according to $\lambda N^2 \equiv a$, a constant, in a natural passage from finite to infinite sources. Because of the way these networks will be defined, they will be at the "combinatorially efficient, hard to control" end of the trade-off spectrum.

*Proof of Theorem (24):* Given $\epsilon > 0$, choose $c$ to be the smallest integer such that $E(c, a) < \epsilon$, and construct a sequence of networks $\nu_N$ with $N$ terminals on a side and $c$ central buses, with nonblocking access to an idle bus from each side. Property $(i)$ follows from (21), (22); $(ii)$ is a result of (23) and the general Erlang formula (1); we have

$$N = \frac{m(1 + \sqrt{1 - \rho})}{1 - \dfrac{m}{a(1 - bl)}}, \tag{4}$$

where $m$ is the load, $a = \lambda N^2$, $bl$ = blocking, and $\rho \in (0, 1)$. Since $m \le c$ we must have, by (4)

$$\frac{n}{a(1 - bl)} \to 1 \quad \text{as} \quad N \uparrow \infty.$$

But by $(i)$, $bl \to E(c, a)$, whence the limit in $(ii)$.

To complete the proof we use the basic bounds on the complexity of nonblocking networks given[5] by Bassalygo and Pinsker, according to whom each of the $c \times c$ nonblocking networks needed in Fig. 2 can be

made using at most $68c \log_2 c + O(c)$ crosspoints and $2(1 + \log_2 c)$ stages.

## XXI. A DIRECT ARGUMENT

It is a property of the "direction" picked for the asymptotics in Theorem (24) that the carried loads $m$ approach a limit, so that the line usage $p = m/N$ goes to zero. In practical terms this means that a fixed amount of traffic is being spread over more and more customers, while the load contribution from any one vanishes. Ultimately, then, this limit "direction" is suitable only for very many lightly loaded lines, and it would be more interesting to have similar or analogous results in which the carried load would increase as the networks grew, and the usage $p$ would be bounded away from zero, with the blocking always less than some prescribed number $\epsilon > 0$.

We know from Theorems (15) and (20) that even without the constraint "$bl < \epsilon$," such a positivity constraint on $p$ necessitates $N \log N$ growth for practical networks. It is instructive, however, to give a separate direct argument for this behavior in the case of the networks constructed in Theorem (24).

*Lemma: For* $1 > \epsilon > 0$, *let* $c$ *be the earliest integer such that* $E(c, a) \leq \epsilon$. *Then*

$$c \geq a(1 - \epsilon). \tag{25}$$

*Proof:* By hypothesis, $E(c, a) \leq \epsilon < E(c - 1, a)$. As is well-known, the Erlang function satisfies the recurrence

$$E(c, a) = \frac{1}{1 + \dfrac{c}{aE(c - 1, a)}}.$$

Thus,

$$\frac{1}{1 + \dfrac{c}{aE(c - 1, a)}} \leq \epsilon < E(c - 1, a).$$

Replacing $E(c - 1, a)$ on the left by the smaller $\epsilon$ will decrease the left, giving

$$\frac{1}{1 + \dfrac{c}{a\epsilon}} \leq \epsilon.$$

Proposition: Let $\nu$ be a central bus network constructed to have $bl \leq \epsilon$, as in Theorem (24), by the method[5] by Bassalygo and Pinsker. Then

$$X(\nu) \geq 8pN \log_2 N + 4Np[\log_2(1 - \epsilon) + \log_2 \lambda]. \tag{26}$$

*Proof:* $\nu$ achieves usage $p = m/N$, so there must be a state $x \in S$ with $|x| \geq Np$. The method of construction implies that $\nu$ is a network having $4 + 4 \log_2 c$ stages, where $c$ is the number of central buses, chosen to be the earliest integer $c$ such that $E(c, a) \leq \epsilon$. Thus every call passes through $4 + 4 \log_2 c$ crosspoints. Since two calls do not pass through the same crosspoints, the state $x$ has at least $4Np \log_2 c$ *busy* crosspoints, and so

$$X(\nu) \geq 4Np \log_2 c.$$

But by Lemma (25) and the choice of $c$, we have

$$\begin{aligned} c &\geq a(1 - \epsilon) = \lambda N^2(1 - \epsilon) \\ \log_2 c &\geq 2 \log_2 N + \log_2(1 - \epsilon) + \log_2 \lambda \qquad (27) \\ X(\nu) &\geq 8pN \log_2 N + 4Np[\log_2(1 - \epsilon) + \log_2 \lambda]. \end{aligned}$$

It follows that any sequence of such networks, growing so that $p$ is bounded away from zero, will grow like $N$ log $N$. For then $\lambda N$ is bounded below, and (27) implies

$$X(\nu) \geq 2pN \log_2 N + O(N), \quad \text{as} \quad N \to \infty.$$

## XXII. FRAME CONCEPT

The second kind of network structure we shall study is called a *frame*. It is familiar to engineers from the No. 5 and earlier crossbar systems. The idea of the frame is to mount all $N$ terminals on a side in $k$ groups of $N/k$ on subnetworks that are connected pairwise by dedicated junctor groups. In the two-sided case shown in Fig. 3, these connections are described by a complete bipartite graph. We shall suppose that the inlet subnetworks are $N/k$ by $kc$ and identical, and are mirror images of the outlet subnetworks, with $c$ trunks or junctors connecting each pair of inlet-outlet subnetworks. A solvable limiting case results when we make the subnetwork nonblocking, so that loss is due always to overload of one of the groups of $c$ junctors between a pair of subnetworks.

*Remark*: It is tempting to conjecture here that in the natural "weak" limit $N \to \infty, \lambda \to 0, \lambda N^2 \equiv a$, a constant, the loss for the frame network with nonblocking concentrators will approach the Erlang loss $E(c, ak^{-2})$ for $c$ trunks and Poisson traffic $ak^{-2}$.

It is not hard to see that if the subnetworks in Fig. 3 are nonblocking, then to define the transition rates that devolve from the stochastic model it is enough to know how many trunks are busy in each of the $k^2$ dedicated groups of size $c$. Therefore we can use, instead of our usual microscopic semilattice, a very much reduced[3] notion of state, as follows: we describe the system by a $k \times k$ matrix $x$ of integers $(x_{ij})$,

$0 \le i, j \le k$, with the interpretation that

$x_{ij}$ = number of calls in progress from subnetwork $i$ on the left to subnetwork $j$ on the right.

The $x_{ij}$ are restricted to be integers between 0 and $c$, the capacity of each $(i, j)$ trunk group. It is useful to have the notations

$$\begin{aligned} x_i &= \sum_{j=1}^{k} x_{ij} = \text{number of calls in progress on subnetwork } i \text{ on left} \\ x^j &= \sum_{i=1}^{k} x_{ij} = \text{number of calls in progress on subnetwork } j \text{ on right} \\ x^+(i,j) &= \text{the state resulting from } x \text{ when a new call is added to the } (i,j) \text{ trunk group} \\ x^-(i,j) &= \text{the state resulting from } x \text{ when a hangup occurs on the } (i,j) \text{ trunk group.} \end{aligned} \tag{28}$$

With $p_x$ the stationary probability of $x$, the statistical equilibrium equations are

$$p_x\left[\sum_{i,j=1}^{k} x_{ij} + \lambda \sum_{i,j=1}^{k} 1_{x_{ij}\neq c}(N - x_i)(N - x^j)\right]$$

$$= \sum_{i,j=1}^{k} [p_{x^+(i,j)}(x_{ij+1})1_{x_{ij}\neq c} + \lambda 1_{x_{ij}\neq 0} p_{x^-(i,j)}(N - x_i + 1)(N - x^j + 1)].$$

Here the indicator functions for $x_{ij} \neq 0$ and $x_{ij} \neq c$ give the right equations on the "boundary" of the state space. The solution of these equations has the convenient product form

$$\begin{aligned} p_x &= p_0 \frac{\lambda^{|x|}}{\prod_{i,j} x_{ij}!} \prod_{i,j} x_i! x^j! \binom{N}{x_i}\binom{N}{x^j} \\ &= p_0 \lambda^{|x|} \prod_{i,j} \binom{x_i}{x_{i1}, \cdots, x_{ik}}\binom{x^j}{x_{1j}, \cdots, x_{kj}}\binom{N}{x_i}\binom{N}{x^j}, \end{aligned}$$

where $p_0$ is the chance that no calls are up, given as the reciprocal of the normalizing sum as

$$p_0^{-1} = \sum_x \lambda^{|x|} \prod_{i,j} \binom{x_i}{x_{i1}, \cdots, x_{ik}}\binom{x^j}{x_{1j}, \cdots, x_{kj}}\binom{N}{x_i}\binom{N}{x^j}.$$

The sum over the states $x$ is over all $k \times k$ matrices whose entries are integers in $[0, c]$, including the identically 0 state, which contributes a term 1.

Under our symmetry assumptions the probability of blocking between two outer subnetworks $i$ and $j$ is the same as the overall loss,

and is given by

$$bl = \frac{\sum_x p_x 1_{x_{ij}=c}(N - x_i)(N - x^j)}{\sum_x p_x (N - x_i(N - x^j)}.$$

This formula does not depend on the normalizer $p_0^{-1}$. Although it is exact, it will be of interest to us primarily for the insight it gives into the asymptotic behavior of $bl$ as $N \to \infty$, $\lambda \to 0$, with $\lambda N^2 = a$, constant. To this end it is enough to look at $p_x/p_0$. Writing it in the form, for $x \neq 0$, and using $\sum_i x_i = \sum_j x^j = |x|$,

$$p_x/p_0 = \frac{\lambda^{|x|}}{\prod_{i,j} x_{ij}} \prod_{i,j} \prod_{m=0}^{x_i - 1} \left(\frac{N}{k} - m\right) \prod_{\ell=0}^{x^j - 1} \left(\frac{N}{k} - \ell\right)$$

$$= \left(\frac{\lambda N^2}{k^2}\right)^{|x|} \cdot \prod_{i,j} \cdot \frac{\left(1 - \frac{x_i - 1}{N/k}\right) \cdots \left(1 - \frac{1}{N/k}\right)\left(1 - \frac{x^j - 1}{N/k}\right) \cdots \left(1 - \frac{1}{n/K}\right)}{x_{ij}!}$$

(the products interpreted as 1 if $x_i$ or $x^j$ is 0) one can see that

$$\lim_{\substack{N \to \infty \\ \lambda \to 0 \\ \lambda N^2 = a}} p_x/p_0 = \frac{(ak^{-2})^{|x|}}{\prod_{i,j} x_{ij}!};$$

write $\alpha = ak^{-2}$ for simplicity; the probability of loss goes to

$$\frac{\frac{\alpha^c}{c!} \sum_{x_{ij}=c} \frac{\alpha^{|x|-c}}{\prod_{k,\ell \neq i,j} x_{k\ell}!}}{\sum_{n=0}^{c} \frac{\alpha^n}{n!} \sum_{x_{ij}=n} \alpha^{|x|-n} \Big/ \prod_{k,\ell \neq i,j} x_{\ell k}!}.$$

It can be verified that for the various $n$ up to $c$, the $\sum_{x_{ij}=n}$ summands in the denominator are all equal, and equal to the $\sum_{x_{ij}=c}$ sum in the numerator. Thus, as conjectured,

$$\lim_{\substack{n \to \infty \\ \lambda \to 0 \\ \frac{\lambda N^2}{k^2} = a}} bl = \frac{\frac{\alpha^c}{c!}}{\sum_{n=0}^{c} \frac{\alpha^n}{n!}} = E(c, \alpha) = E(c, ak^{-2}). \tag{29}$$

Table I—Load loss relationships

| | Loss | Load |
|---|---|---|
| Central bus | $E(c, a)$ | $a[1 - E(c, a)]$ |
| Frame | $E(c, ak^{-2})$ | $a[1 - E(c, ak^{-2})]$ |

As for central bus networks, one can get linear growth in the weak limit, described as follows:

*Theorem: For every $\epsilon > 0$, there is an integer $c$ and a growing sequence $\nu_N$ of two-sided frame networks on $N$ terminals, with $N/k$ subnetworks on a side, such that as $N \to \infty$, $\lambda \to 0$, and $a = \lambda N^2 \equiv$ constant,*

$$\begin{aligned} &(i)\ bl(\nu_N) \to E(c, ak^{-2}) \leq \epsilon \\ &(ii)\ m(\nu_N) \to a[1 - E(c, ak^{-2})] \\ &(iii)\ X(\nu_N) \leq 2k[68N \log_2 c + O(c)] \\ &(iv)\ s(\nu_N) = 4(1 + \log_2 kc). \end{aligned} \tag{30}$$

*Remark:* The reader can check that it is quite proper to regard the frame network with $k$ concentrating subnetworks on a side as a system of $k^2$ central bus networks: the loss is asymptotically Erlang $E(c, \cdot)$ in both cases, and the carried load for the frame is $k^2$ times the carried load on the corresponding central bus, as shown in Table I.

*Proof of Theorem (30):* This proof is very much like that of (24). We chose $c$ to be the least integer such that $E(c, ak^{-2}) \leq \epsilon$, and construct nonblocking $N/k \times kc$ concentrators for the subnetworks, each using no more than $68(N/k)\log_2(kc) + O(kc)$ crosspoints and $2(1 + \log_2 kc)$ stages. Convergence of the loss has been proved as (29), and that of the load follows as usual from the Erlang formula (1), as in (24).

## XXIII. REMOTE CONCENTRATOR CONCEPT

A third network structure worth looking at consists of concentrating subnetworks each connected only to one and the same central "core" network by a group of $c$ high-usage trunks, as in Fig. 4. We call this structure, well-known to traffic engineers, the "remote concentrator concept;" it represents an extreme form of the advice to separate concentration and distribution in the network. We shall suppose that the subnetworks are divided into two groups, one carrying the inlets, the other the outlets, so that the whole network is still two-sided. When the outer subnetworks (concentrators) and the inner core are all nonblocking, a useful solvable case results, and we can again guess that as the right limit is taken there will be some connection with Erlang's $E$ function. Since the success of a call attempt depends

entirely on finding first a free trunk into the core, and then one going out from it to the destination concentrator, it is tempting to guess that asymptotically the loss is given by the "blocking polynomial" $1 - (1 - b)^2$, where $b$ is the chance that all $c$ trunks in a group are busy. This simple guess is probably not true, because of the correlation between loads on trunk groups; it may nevertheless be a bound, although we have not been able to prove this: the question whether the blocking owing to simultaneously full groups is larger or smaller than $b^2$ is open. We can, however, give Erlang $E$ bounds on both kinds of blocking, as well as exact loss formulas for finite $N$ in terms of logarithmic derivatives of a partition function.

Let $k$ be a divisor of $N$, fixed henceforth, to be interpreted as the number of concentrators on the inlet side of the network, each with $N/k$ inlets, $c$ trunks to the core, and nonblocking. The outlet side is similarly constituted. As a notion of state we can take the matrices $x = (x_{ij}, 0 \le i, j \le k)$ with the meaning

$$x_{ij} = \text{number of calls in progress from inlet concentrator } i \text{ to outlet concentrator } j.$$

These matrices are subject to the condition that both rows and columns must not sum to more than $c$, the trunk group size. This is the same notion of state used for frame networks, except that there the *entries* had to be at the most $c$, while here the row and column sums are thus bounded.

Using the same notations (28) as for the frame networks, we can put down the following equilibrium equations:

$$p_x \left[ \sum_{i,j=1}^{k} x_{ij} + \lambda \sum_{i,j=1}^{k} 1_{x_i < c, x^j < c} \left( \frac{N}{k} - x^j \right) \right]$$
$$= \sum_{i,j=1}^{k} \left[ p_{x^+(i,j)}(x_{ij} + 1) 1_{x_{ij} < c} + \lambda 1_{x_{ij} > 0} p_{x^-(i,j)} \left( \frac{N}{k} x_{i+1} \right) \left( \frac{N}{k} - x^j + 1 \right) \right].$$

Again, the indicator factors give the right equations at the boundary of the state space, and the solution has the same product form as for frame networks:

$$p_x = p_0 \lambda^{|x|} \prod_{i,j=1}^{k} \binom{x}{x_{i1}, \cdots, x_{ik}} \binom{x^j}{x_{ij}, \cdots, x_{kj}} \binom{N/k}{x_i} \binom{N/k}{x^j},$$

where $p_0$ is the chance of no calls in progress, the reciprocal of the normalizer

$$\sum_{x \in S} \lambda^{|x|} \prod_{i,j=1}^{k} \binom{x_i}{x_{i1}, \cdots, x_{ik}} \binom{x^j}{x_{ij}, \cdots, x_{kj}} \binom{N/k}{x_i} \binom{N/k}{x^j}.$$

The sum over the states $x \in S$ is over all $k \times k$ matrices with nonnegative integer entries, and row *and* column sums at most $c$.

By symmetry, the probability of blocking between two remote concentrators $i$ and $j$ is the same as the overall loss, and is given by

$$bl = \frac{\sum_{x \in S} p_x[1_{x_i=c} \wedge 1_{x^j=c}] \left(\frac{N}{k} - x_i\right)\left(\frac{N}{k} - x^j\right)}{\sum_{x \in S} p_x \left(\frac{N}{k} - x_i\right)\left(\frac{N}{k} - x^j\right)}.$$

This formula depends only on the known ratios $p_x/p_0$. We examine its asymptotic comportment as $\lambda \to 0$, $N \to \infty$, with $\lambda N^2 \equiv a$, constant. The argument for (29) gives

$$\lim_{\substack{N\to\infty \\ \lambda\to 0 \\ \lambda N^2 = a}} p_x/p_0 = \frac{\left(\frac{a}{k^2}\right)^{|x|}}{\prod_{i,j=1}^{k} x_{ij}!}.$$

## XXIV. THE PARTITION FUNCTION

We have, for the remote concentrator concept,

$$p_x = p_0 \lambda^{|x|} \prod_{i,j=1}^{|x|} \binom{x_i}{x_{i1}, \cdots, x_{ik}} \binom{x^j}{x_{1j}, \cdots, x_{kj}} \binom{N/k}{x_i} \binom{N/k}{x^j}$$
$$= p_0 \lambda^{|x|} c(x).$$

Hence, introducing the generating function

$$\phi(y) = 1 + \sum_{j=1}^{kc} y^j \sum_{\substack{|x|=j \\ x \in S}} c(x),$$

the moments of the number of calls in progress can be expressed as logarithmic derivatives; especially,

$$m = \lambda \frac{d}{d\lambda} \log \phi(\lambda)$$

$$\sigma^2 = \lambda^2 \frac{d^2}{d\lambda^2} \log \phi(\lambda) + \lambda^2 \frac{d}{d\lambda} \log \phi(\lambda),$$

and by the generalized Erlang formula (1), the blocking is determined via

$$1 - bl = \frac{1}{\lambda} \frac{m}{(N-m)^2 + \sigma^2}$$
$$= \frac{\frac{d}{d\lambda} \log \phi(\lambda)}{\left[N - \lambda \frac{d}{d\lambda} \log \phi(\lambda)\right]^2 + \lambda^2 \frac{d^2}{d\lambda^2} \log \phi(\lambda) + \lambda^2 \frac{d}{d\lambda} \log \phi(\lambda)}.$$

The interested reader can verify that analogous results hold in the Erlang case of $c$ trunks offered traffic $a$: the generating function is

$$\phi(y) = 1 + y + \frac{y^2}{2} + \cdots + \frac{y^c}{c!}$$

so that

$$1 - bl = 1 - E(c, a) = \frac{\phi'(a)}{\phi(a)}$$

$$m = a(1 - E(c, a)) = a \frac{d}{da} \log \phi(a).$$

The essential form of these relationships persists in the weak limit $\lambda \to 0$, $N \to \infty$, $\lambda N^2 = a \equiv$ constant. We write

$$p_x = p_0 a^{|x|} \prod_{i,j=1}^{k} \binom{x_i}{x_{i1}, \cdots, x_{ik}} \binom{x^j}{x_{1j}, \cdots, x_{kj}} \frac{(N/k)!(N/k)!N^{-2|x|}}{\left(\frac{N}{k} - x_i\right)!x_i!\left(\frac{N}{k} - x^j\right)!x^j!}$$

and take the limit as above. Stirling's formula implies that the partition function becomes

$$\phi(y) = 1 + \sum_{\ell=1}^{kc} y^\ell \sum_{\substack{|x|=\ell \\ \text{row sums} \le c \\ \text{column sums} \le c}} k^{-2\ell} \cdot \prod_{i,j=1}^{k} \frac{\binom{x_i}{x_{i1}, \cdots, x_{kj}} \binom{x^j}{x_{1j}, \cdots, x_{kj}}}{x_i!x^j!}. \quad (31)$$

Then since $(N - m)^2 + \sigma^2 \sim N^2$ in the weak limit, one finds

$$m \to a \frac{d}{da} \log \phi(a) \quad (32)$$

$$1 - bl \to \frac{d}{da} \log \phi(a). \quad (33)$$

*Theorem: For every integer $k$ and every $\epsilon > 0$, there is an integer $c$, and a sequence $\nu_N$ of "remote concentrator" networks, with $c$ trunks from each of $k$ $N/k \times c$ concentrators on a side to a central core $kc \times kc$, such that as $\lambda \to 0$, $N \to \infty$, $\lambda N^2 = a \equiv$ constant*

$$\begin{aligned} &(i)\ bl(\nu_N) \to 1 - \frac{d}{da} \log \phi(a) < \epsilon \\ &(ii)\ m(\nu_N) > a \frac{d}{da} \log \phi(a) \\ &(iii)\ X(\nu_N) \le \frac{2N}{k} [68c \log_2 c + O(c)] + 68kc \log_2 kc + O(kc) \\ &(iv)\ s(\nu_N) = 6(1 + \log_2 c) + 2 \log_2 k. \end{aligned} \quad (34)$$

*Proof:* Again, this proof is like that of (24). Using (32), the blocking can be written as

$$bl = 1 - \frac{m}{\lambda N^2 \sum_{x \in S} p_x \left(1 - \frac{x_i}{N}\right)\left(1 - \frac{x^j}{N}\right)} = 1 - \frac{m}{a + o(1)}$$

$$= 1 - \frac{d}{da} \log \phi(a) + o(1).$$

So with $\lambda N^2 = a$, fixed, pick the integer $c$ in the partition function $\phi(y)$ defined by (31) so that

$$1 - \frac{d}{da} \log \phi(a) < \epsilon.$$

This is possible because the limit blocking must decrease to 0 with increasing $c$. Thus, (32) proves (*i*) and (*ii*); (*iii*) and (*iv*) follow by the same kind of concentrator construction as before, using the method of Bassalygo and Pinsker.[5]

## XXV. BLOCKING INEQUALITIES FOR REMOTE CONCENTRATOR CONCEPT

When the concentrating subnetworks and the core are nonblocking, it is possible to derive some interesting Erlang $E$ bounds for the blocking in a remote concentrator structure. We first note that the contributions to blocking are of two kinds: a call is blocked between subnetworks $i$ and $j$ because $x_i = c$, or $x^j = c$, or both, i.e., $x_i + x^j = 2c$. Thus,

$$bl = \frac{\sum_{x \in S} p_x \left(\frac{N}{k} - x_i\right)\left(\frac{N}{k} - x^j\right)(1_{x_i = c} + 1_{x^j = c} - 1_{x_i + x^j = 2c})}{\sum_{x \in S} p_x \left(\frac{N}{k} - x_i\right)\left(\frac{N}{k} - x^j\right)}$$

$$= \sum_{x \in S} p_x (1_{x_i = c} + 1_{x^j = c} - 1_{x_i + x^j = 2c}) + o(1).$$

By symmetry the $1_{x_i = c}$ and $1_{x^j = c}$ terms contribute the same amount, so the problem of estimating the blocking reduces to estimating:

(*i*) the probability $Pr\{x_i = c\} = \sum_{x_i = c} p_x$ of having "all trunks busy" on concentrator $i$, and

(*ii*) the "double trouble" term $Pr\{x_i + x^j = 2c\}$.

It is convenient to define, for states $x \in S$, and $1 \le i, j \le k$,

$$s^{ij}(x) = (1 - 1_{x_i = c})(1 - 1_{x^j = c})\left(\frac{N}{K} - x_i\right)\left(\frac{N}{k} - x^j\right).$$

This is the number of unblocked idle inlet-outlet pairs $(u, v)$ with $u$ on concentrator $i$ and $v$ on concentrator $j$.

*Lemma: For integers $0 \le t \le w = max_{x \in S}|x|$, the chance of t calls up on concentrator i can be represented as*

$$Pr\{x_i = t\} = Pr\{x_i = 0\} \frac{a^t}{t!} \prod_{\ell=0}^{t-1} \eta_\ell, \tag{35}$$

*where*

$$\eta_\ell = N^{-2\ell} \sum_{x_i = \ell} \frac{p_x}{Pr\{x_i = \ell\}} \sum_{j=1}^{k} s^{ij}(x)$$

$$= N^{-2\ell} E\{\text{number of unblocked calls on } i | \ell i\text{-trunks are busy}\}$$

$$\le k^{-1}.$$

*Proof:* This follows from the form of the statistical equilibrium equations, which says that the rate into a set is the rate out of it. We use the sets $\{x \in S: x_i = t\}$, which partition $S$ and communicate by pairs in a simply ordered array except at the endpoints $\ell = 0$ and $\ell = c$. The result follows by iteration. In a similar way it is found that

*Lemma:*

$$Pr\{x_i + x^j = t\} = Pr\{x_i + x^j = 0\} \frac{a^t}{t!} \prod_{\ell=0}^{t-1} \xi_\ell, \tag{36}$$

*where*

$$\xi_\ell = N^{-2\ell} \sum_{x_i + x^j = \ell} \frac{p_x}{Pr\{x_i + x^j = \ell\}} \sum_{m=1}^{k} [s^{im}(x) + (1 - \delta_{im}) s^{mj}(x)]$$

$$N^{-2\ell} E\{\text{number of unblocked calls to } i \text{ or } i | x_i + x^j = \ell\}$$

$$\le \frac{2k - 1}{k^2}.$$

*Theorem:*

$$Pr\{x_i = c\} \le E(c, ak^{-2})$$

$$Pr\{x_i + x^j = 2c\} = E(2c, 2ak^{-1} - ak^{-2}). \tag{37}$$

*Proof:* Introduce the function on the positive orthant

$$f(y_1, \cdots, y_c) = \frac{y_1 y_2 \cdots y_c}{1 + y_1 + y_1 y_2 + \cdots + y_1 y_2 \cdots y_c} = \frac{N}{D};$$

this is increasing in each $y_i$ there, since

$$\frac{\partial f}{\partial y_i} = \frac{N(1 + y_2 + y_2 y_2 + \cdots + y_1 y_2 \cdots y_{i-1})}{y_i D^2} > 0.$$

By normalization, $\sum_{i=0}^{c} Pr\{x_i = t\} = 1$, so Lemma (35) says that

$$Pr\{x_i = c\} = f\left(a\eta_0, \frac{a}{2}\eta_1, \cdots, \frac{a}{c}\eta_{c-1}\right)$$

$$\leq f\left(a/k, \frac{a/k}{2}, \cdots, \frac{a/k}{c}\right) = E(c, a/k).$$

The proof for the "double trouble" term is analogous, from Lemma (36).

## XXVI. CONCLUSIONS AND PROSPECTS

For the narrow class of probability models for telephone networks described by "finite sources, exponential holding times," we have shown that the $N \log N$ rate of growth (of the number $X$ of crosspoints) characteristic of nonblocking networks extends also to those with blocking. This narrow class provided easy methodological devices for carrying out the proofs. Extensions to more general statistics have been mentioned in an interesting series of papers by N. Pippenger, listed in the bibliography. However, his results are either combinatorial or restricted to Markovian models similar to ours. Since some of his principal demonstrations depend on what amounts to the old "lost calls held" convention applied to finite sources, his results are strictly not comparable to those given here. Extensions to the distribution-free context remain to be made. As Pippenger suggests, the most useful tools are likely to be the entropy concept and ergodic theory.

## REFERENCES


1. E. Brockmeyer, H. L. Halström, and A. Jensen, *The life and work of A. K. Erlang*, Acta Polytechnica Scandinavica, Mathematics and Computing Machinery Series, No. 6, Copenhagen, 1960.
2. V. E. Beneš, "Markov Processes Representing Traffic in Connecting Networks," B.S.T.J., *42*, No. 6 (November 1963) pp. 2795–838.
3. V. E. Beneš, "Reduction of Network States Under Symmetries," B.S.T.J., *57*, No. 1 (January 1978), pp. 111–49.
4. C. Clos, "A Study of Nonblocking Switching Networks, B.S.T.J., *32*, No. 2 (March 1953), pp. 406–24.
5. L. A. Bassalygo and M. S. Pinsker, "Complexity of an Optimal Nonblocking Switching Network Without Reconnections," Problemy Peredachi Informatsii, *9* (1973), pp. 84–7; translated into English in Problems of Information Transmission, *9* (1974), pp. 64–6.


## BIBLIOGRAPHY OF BACKGROUND READING AND RELATED WORK


Beneš, V. E., *Mathematical Theory of Connecting Networks and Telephone Traffic*, New York: Academic Press, 1965.

Ikeno, N., "A Limit on Crosspoint Number," 1959 International Symposium on Circuit and Information Theory, Los Angeles, CA, June 16–18, 1959.

Pippenger, N., "The Complexity Theory of Switching Networks," Sci. D. Thesis, Mass. Inst. Technology, 1973.

Pippenger, N., "Complexity of Seldom Blocking Networks," Proc. IEEE Communications Conference, 1976, paper 7–8.

Pippenger, N., "On the Complexity of Strictly Nonblocking Networks," IEEE Trans. Communications, *COM-22* (1974), pp. 1890–2.

Shannon, C. E., "Memory Requirements in a Telephone Exchange, B.S.T.J., *29*, No. 3 (July 1950), pp. 343–9.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 2, February 1983
*Printed in U.S.A.*

# Upper and Lower Bounds on Mean Throughput Rate and Mean Delay in Memory-Constrained Queueing Networks

By E. ARTHURS and B. W. STUCK

(Manuscript received April 27, 1982)

*Operators use terminals to enter transactions into a system and then wait for the system to respond. The system contains serially reusable resources, and can hold a maximum number of jobs. Each job requires a total mean amount of service at each stage. We calculate upper and lower bounds on the mean throughput rate and mean delay as a function of model parameters, and present examples that show these bounds are* sharp, *in the sense that they are achievable given only mean values. We also present partial results for closed queueing networks where the long-term, time-averaged distribution of number of jobs at each node in the network obey so-called* product form *separation of variable type of probability distributions. Examples and data from actual systems illustrate the utility of the work.*

## I. INTRODUCTION

At present there is great interest in modeling the traffic-handling characteristics of computer and communication systems using queueing networks.[1-4] The change in cost of electronic solid-state circuitry[5,6] and rising personnel costs[7,8] offers strong incentive to design cost-effective digital systems.

Computer communications systems can often be modeled quite naturally by a network of queues, where a job receives service at one stage or queue and then migrates to another stage, until it is completely serviced. Examples of actual systems and associated models are presented in later sections of this report. This class of models captures several fundamental phenomena of such systems, including asynchronous and concurrent execution of different jobs and different amounts of service required at each stage of execution. To answer whether this

type of model is valid, controlled experimentation and measurement must be carried out, and goals or criteria must be set for judging goodness of fit. Finally, one would like to use these models to predict or extrapolate behavior into unknown regions of operation to guide decision making.

Here we focus on one technique for bounding the mean throughput rate and mean delay of an abstraction of a computer communications system. This is only one factor among many others, such as cost, flexibility, and reliability, that must be considered in choosing one design over another for a given application. We drop these other factors from consideration after this point in the interest of brevity.

Broadly speaking, there are two reasons for wanting to quantify the traffic-handling capabilities in a computer communication system:

(*i*) Cost reduction of an existing service or product:

(a) In an existing system, it is often possible to modify existing scheduling policies to improve performance at an acceptable cost. An example would be to change from one memory partition per application program to a memory pool shared among all application programs.

(b) In a system handling a fixed set of job types, different equipment configurations can accomplish these jobs at different costs. Which should be chosen? An example would be to compare using two slow disks versus one fast disk.

(*ii*) Comparisons are often desired between current operations and wholly new modes of operations. An example would be using an existing batch computer system for time sharing, using the existing time sharing system for electronic mail, or using existing word processors for voice-annotated text services.

To quantify these issues, typically two stages are involved: the first is *synthesis*, where goals are stated along with different alternatives for reaching those goals, while the second is *analysis*, where the performance (here the mean throughput rate of finishing jobs and the mean delay for each stage of job execution) is quantified. Goals may be either oriented toward the total system, such as total number of jobs of a given type that are handled during an hour, or toward an individual, such as the mean delay to handle one or more stages of a job; along with goals such as these there should be some measure of the *sensitivity* of the goals to different operating points, and so forth.

Analysis often begins by postulating a set of parameters that carry or capture specific operational aspects, drawing inferences based on these parameters (either by mathematical analysis or by discrete event simulation),[9-11] measuring actual or simulated operation, and then repeating this process until it is felt that additional work is no longer warranted.

Our goal here is to demonstrate how to carry out part of this process by a straightforward technique for obtaining bounds on mean throughput rate and mean delay, given only mean value information for the service required at each stage of job execution. In our opinion, there are three principal contributions:

(*i*) A new technique for obtaining a *lower* bound on mean throughput rate and an associated *upper* bound on mean delay. Earlier workers (e.g., Ref. 1, pp. 212–25) obtained an *upper* bound on mean throughput rate and an associated *lower* bound on mean delay. Furthermore, we present an example that shows that given *only* mean values for the amount of service required at each visit to each stage in the queueing network model, *either* bound can be approached arbitrarily close, depending upon on the amount of fluctuation present about the mean service times. This shows that these bounds are *sharp*, much as was done earlier for loss systems.[12,13] The interested reader is referred to related works.[14,15]

(*ii*) A new technique for calculating both *upper* and *lower* bounds on the mean throughput rate and mean delay for a class of closed queueing networks whose long-term, time-averaged distribution for the number of jobs in system obeys a so-called *product form* or separation of variables decomposition[2,3] (for an application case study, see Ref. 16). The upper bound on mean throughput rate is the reciprocal of the total mean time to execute the transaction plus the *average* time spent in execution per node, while the lower bound on mean throughput rate is the reciprocal of the total mean time to execute the transaction plus the *maximum* mean time spent in execution per node. The tightness of these bounds will thus depend on how close these factors are to one another (Zahorjan et al., developed these results independently;[17] our derivation is felt to be more straightforward).

(*iii*) Data from controlled experimentation on actual computer and computer communication systems is presented to validate the approach presented here.

The examples presented are deliberately elementary, chosen for tractability. Everything of interest can be represented by formulas. Furthermore, this approach is a natural starting point for virtually any study of traffic-handling performance, can be refined in a variety of ways, used to check and bound much more complex analyses or simulations, and can be immediately related to measurements in an actual system. Often data are simply not available to describe the arrival statistics and service required for each step of each job, such as would be needed in simulation studies; this suggests using a mean value (distribution free) analysis, rather than more stringent distributional assumptions, and then assessing performance sensitivity by

varying the mean value, rather than investing effort in simulation studies. We advocate *synthesis* via *analysis* of the performance of a given configuration. The approach adopted here is not exhaustive, but it is fundamental. The examples show that only two avenues are available for improving computer communication system performance, reducing the time to handle a given task (i.e., speedup) and handling two or more tasks simultaneously [i.e., concurrency, either real (multiplexing multiple resources) or apparent (via scheduling a single resource)].

## II. A MODEL OF A PROCESSOR AND DISK SYSTEM

In this section we deal with a mathematical abstraction of an on-line transaction processing system.

### *2.1 Model description*

Clerks at terminals spend a mean amount of time reading, thinking, and entering the transaction, and then wait for the system to respond before repeating this cycle. Each transaction requires a mean amount of processor time and disk secondary storage access time to be completely executed. The system is configured with a finite amount of memory, and hence can hold a maximum number of jobs at any one time. Figure 1 shows a hardware block diagram of the system. The cycle that a job or transaction follows can be described by a path through a network of queues. The first stage or queue is associated with operators at terminals entering each transaction. Next, each job enters a *staging* queue, where it waits if there are already the maximum number of allowable jobs in the system, and otherwise it immediately enters the system. Once inside the system, a job will receive some processing, then require accessing some data from secondary disk storage, then some processing, and so forth until it is completely executed. Finally, control will return to the operator at the terminal and the process begins anew. Figure 2 shows a queueing network block diagram for this system, consisting of four queues: one for operator jobs, one for staging, one for processors, and one for disks.

The ingredients we need are

(*i*) The number of clerks, $C$, actively submitting transactions to the system

(*ii*) The number of processors, $P$, and disks, $D$, connected by a common switch. (The switch is assumed to be much faster than any step of job execution involving either a processor or a disk, and is ignored from this point on.)

(*iii*) The maximum number of jobs allowed inside the system at any one time, $M$

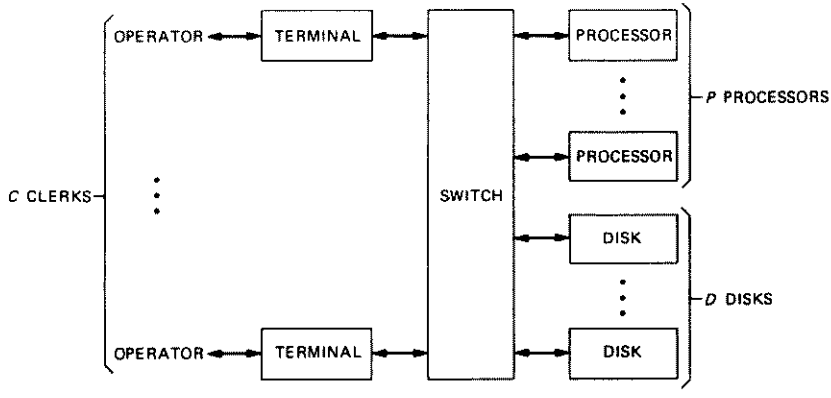



Fig. 1—Block diagram of the system hardware.

(*iv*) The mean time each operator spends reading, thinking, and entering each transaction, denoted $T_{\text{think}}$

(*v*) The mean processor time, $T_{\text{processor}}$, and mean disk access time, $T_{\text{disk}}$, per transaction.

The outputs of the analysis are the mean throughput rate of executing transactions and the mean response time (as seen by operators, including both execution time plus waiting time) as a function of model parameters. No job is assumed to be capable of executing in parallel with itself. The operating system multiplexes available jobs among available processors and disks to achieve some degree of concurrent use of resources.

From the vantage point of an operator at a terminal, we see that each transaction undergoes two stages of processing:

(*i*) A stage spent preparing the transaction for execution, with mean time interval, $T_{\text{think}}$

(*ii*) A stage spent waiting for the transaction to execute, with mean time interval, $R$.

For one operator at one terminal, the mean cycle time per transaction is simply the sum of the mean preparation time and mean delay. Hence, when $C$ operators are active, the mean throughput rate equals simply $C$ times the mean throughput rate for one operator:

$$\text{mean throughput rate} = \lambda = \frac{C}{T_{\text{think}} + R}.$$

If we rewrite this equation to find the mean response time, we see

$$R = \frac{C}{\text{mean throughput rate}} - T_{\text{think}}.$$

These two relationships will be fundamental in determining feasible operating regions for mean throughput rate and mean delay.

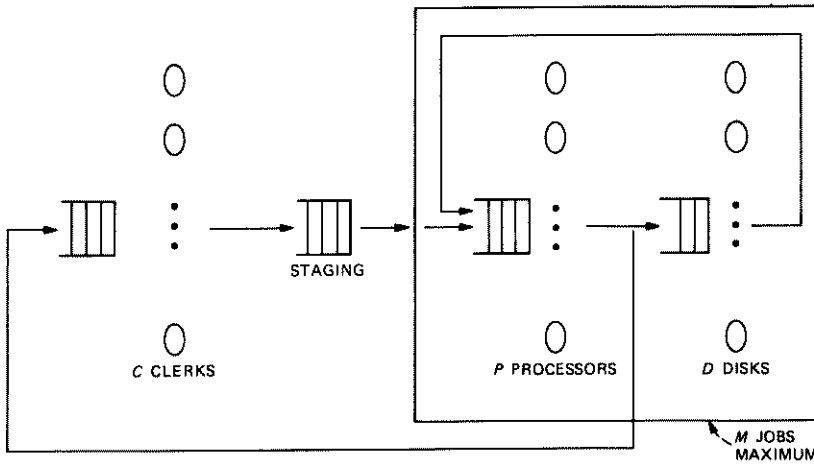



Fig. 2—Block diagram of the system queueing network.

### 2.2 *Mathematical problem statement*

The system state space is denoted by $\Omega$, where

$$\Omega = \{(J_{\text{operator}}, J_{\text{stage}}, J_{\text{processor}}, J_{\text{disk}}) \mid J_{\text{operator}} + J_{\text{stage}} + J_{\text{processor}} + J_{\text{disk}} = C;\ J_{\text{processor}} + J_{\text{disk}} = \min[M, C - J_{\text{operator}}]\}.$$

At any instant of time, the system is in a state given by ($J_{\text{operator}}$, $J_{\text{stage}}$, $J_{\text{processor}}$, $J_{\text{disk}}$). Each component is integer valued, and refers to the number of jobs or transactions either in execution or waiting to be executed at that stage. The admissible state space is constrained because

(*i*) There are at most $C$ jobs being worked on at any one time

(*ii*) The system can hold at most $M$ jobs at any one time.

In a later section, we will show that the mean number of tasks in execution with operators, processors, and disks, averaged over a suitably long time interval, equals the mean throughput rate, $\lambda$, multiplied by the total mean execution time for that stage. This is summarized in the following equations, where $E(\cdot)$ denotes a time average of the argument:

$$E[\min(J_{\text{operator}}, C)] = \lambda T_{\text{think}}$$

$$E[\min(J_{\text{processor}}, P)] = \lambda T_{\text{processor}}$$

$$E[\min(J_{\text{disk}}, D)] = \lambda T_{\text{disk}}.$$

In a later section, we show that $\lambda$ can be upper and lower bounded,

given only this information, in terms of model parameters, as follows:

$$\lambda_{\text{lower}} \le \lambda \le \lambda_{\text{upper}}$$

$$\lambda_{\text{lower}} = \frac{C}{T_{\text{think}} + \dfrac{C}{\min(C, M, P)} T_{\text{processor}} + \dfrac{C}{\min(C, M, D)} T_{\text{disk}}}$$

$$\lambda_{\text{upper}} = \min\left[\frac{\min(C, M, P)}{T_{\text{processor}}}, \frac{\min(C, M, D)}{T_{\text{disk}}}, \frac{\min(C, M)}{T_{\text{processor}} + T_{\text{disk}}}, \frac{C}{T_{\text{think}} + T_{\text{processor}} + T_{\text{disk}}}\right].$$

Each of the upper bounds on mean throughput rate has a physical interpretation, as follows:

(*i*) The processors are limiting the maximum mean throughput rate

$$\lambda_{\text{upper}} = \frac{\min(C, M, P)}{T_{\text{processor}}}$$

(*ii*) The disks are limiting the maximum mean throughput rate

$$\lambda_{\text{upper}} = \frac{\min(C, M, D)}{T_{\text{disk}}}$$

(*iii*) The clerks are limiting the maximum mean throughput rate

$$\lambda_{\text{upper}} = \frac{C}{T_{\text{think}} + T_{\text{processor}} + T_{\text{disk}}}$$

(*iv*) Memory is limiting the maximum mean throughput rate

$$\lambda_{\text{upper}} = \frac{\min(C, M)}{T_{\text{processor}} + T_{\text{disk}}}.$$

The lower bound on mean throughput rate has the physical interpretation of executing jobs one at a time on each processor/disk pair. The upper bound on mean throughput rate is associated with the best possible concurrency, while the lower bound on mean throughput rate is associated with the worst possible parallelism. The upper bound on mean throughput rate yields a lower bound on mean delay; the lower bound on mean throughput rate yields an upper bound on mean delay:

$$\frac{C}{\lambda_{\text{upper}}} - T_{\text{think}} \le R \le \frac{C}{\lambda_{\text{lower}}} - T_{\text{think}}.$$

These bounds define an admissible or feasible region of operation and are plotted in Figs. 3 and 4 for the case of one processor and one disk, versus $C = M$.

Two regimes are evident: a *lightly loaded* regime, where the number

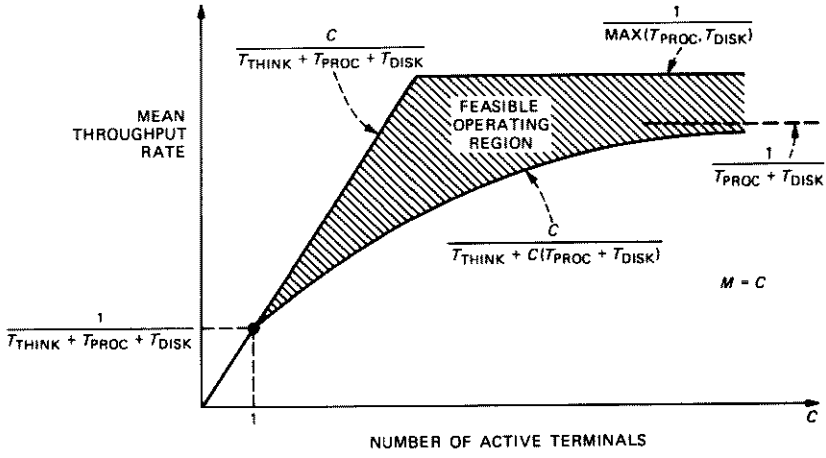



Fig. 3—Mean throughput rate versus number of active terminals.

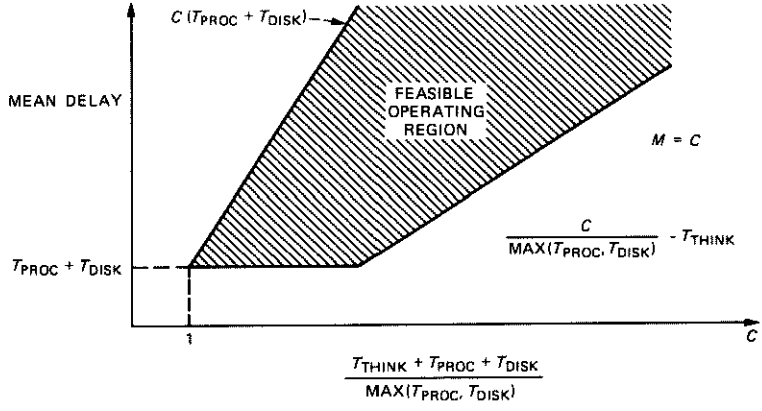



Fig. 4—Mean delay versus number of active terminals.

of *clerks* is directly proportional to the mean throughput rate and the mean delay is independent of the number of clerks, so the clerks are a bottleneck; and a *heavily loaded* regime, where the on-line computer communication system is the bottleneck, with the mean throughput rate independent of the number of clerks and the mean delay directly proportional to the number of clerks.

If we vary the number of clerks, there is a natural breakpoint between these two regimes:

$$\text{breakpoint number of operators} = \lambda_{\text{upper}}[T_{\text{processor}} + T_{\text{disk}} + T_{\text{think}}].$$

As long as the number of clerks is well below this breakpoint, the

clerks and not the system will be limiting the mean throughput rate, and the mean delay per transaction will be approximately the mean delay to execute a transaction with no contention. With the number of clerks well above this breakpoint, the system will be limiting the mean throughput rate, and the mean delay per transaction will be well in excess of that time to execute a transaction with no contention. Analysis suggests measurements to determine where these two regimes lie; synthesis would involve choosing which regime we wish to operate in (remember, we will always have *some* bottleneck!) and designing the system accordingly.

### *2.3 Impact of memory constraint for one processor and one disk*

Here are two possible scheduling policies for a system with one processor and one disk:

(*i*) Only one job is allowed into the system to be executed at any one time. This is called ***single-thread*** scheduling, and corresponds to the maximum number of jobs in the system equal to one, $M = 1$.

(*ii*) More than one job is allowed in the system to be executed at any one time. This is called ***multiple-thread*** scheduling, and corresponds to $M > 1$. For $M = 1$ we see

$$\frac{C}{T_{\text{think}} + C(T_{\text{processor}} + T_{\text{disk}})} \leq \lambda_{\text{single thread}}$$

$$\leq \min\left(\frac{C}{T_{\text{think}} + T_{\text{processor}} + T_{\text{disk}}}, \frac{1}{T_{\text{processor}} + T_{\text{disk}}}\right) M = 1$$

If we allow multiplexing of the processors and disks amongst transactions, then $M > 1$ is allowed, but now one or the other of the two serially reusable resources will become completely utilized for $M$ sufficiently large:

$$\frac{C}{T_{\text{think}} + C(T_{\text{processor}} + T_{\text{disk}})} \leq \lambda_{\text{multiple thread}}$$

$$\leq \min\left(\frac{C}{T_{\text{think}} + T_{\text{processor}} + T_{\text{disk}}}, \frac{1}{T_{\text{processor}}}, \frac{1}{T_{\text{disk}}}\right) M > 1$$

In either case, the lower bound on mean throughput rate is identical, but the upper bound can be different, owing to different bottlenecks:

(*i*) The number of clerks is a bottleneck

$$\lambda_{\text{single thread}}, \lambda_{\text{multiple thread}} \leq \frac{C}{T_{\text{think}} + T_{\text{processor}} + T_{\text{disk}}}$$

(*ii*) The processor is a bottleneck

$$\lambda_{\text{multiple thread}} \leq \frac{1}{T_{\text{processor}}}$$

(*iii*) The disk is a bottleneck

$$\lambda_{\text{multiple thread}} \leq \frac{1}{T_{\text{disk}}}$$

(*iv*) Memory is limiting the maximum mean throughput rate

$$\lambda_{\text{single thread}} \leq \frac{1}{T_{\text{processor}} + T_{\text{disk}}}.$$

Provided that the clerks or operators are not limiting the maximum mean throughput rate, the ratio of the two different upper bounds is an indication of the gain owing to scheduling or allowing more than one job inside the system at any one time:

$$\frac{\lambda_{\text{multiple thread}}}{\lambda_{\text{single thread}}} = \frac{T_{\text{processor}} + T_{\text{disk}}}{\max(T_{\text{processor}}, T_{\text{disk}})}.$$

For one processor and one disk, this gain owing to scheduling can be at most two, no matter what $T_{\text{processor}}$ or $T_{\text{disk}}$ are! Moreover, this will only be achieved when $T_{\text{processor}}$ equals $T_{\text{disk}}$, but in general these two mean times will *not* be equal and hence the gain will *not* be as great as a factor of two; for example, if $T_{\text{disk}}$ were ten times as great as $T_{\text{processor}}$, then the gain would be at most ten percent, and other factors ignored in this analysis may swamp this.

### 2.4 Asymptotics

We close with an investigation of asymptotic behavior of this system. One type of asymptotic analysis is to let all parameters be fixed except one, and the final one becomes progressively larger and larger. Here a natural candidate for such a parameter is the number of operators or jobs circulating in the system $C$. As the number of operators or clerks becomes large, $C \to \infty$, we see

$$\frac{1}{\dfrac{T_{\text{processor}}}{\min(P, M)} + \dfrac{T_{\text{disk}}}{\min(D, M)}} \leq \lambda$$

$$\leq \min\left(\frac{P}{T_{\text{processor}}}, \frac{D}{T_{\text{disk}}}, \frac{M}{T_{\text{processor}} + T_{\text{disk}}}\right), C \to \infty.$$

This in turn will yield upper and lower bounds on mean response time:

$$\left(\frac{C}{\lambda_{\text{lower}}} - T_{\text{think}}\right) \to \infty \leq R \leq \left(\frac{C}{\lambda_{\text{upper}}} - T_{\text{think}}\right) \to \infty \; C \to \infty.$$

In other words, the mean throughput rate lies between two *finite* bounds, while the mean response time is *infinite* (will exceed any finite threshold as we add more and more clerks).

A second type of asymptotic analysis is to fix the ratio of two

parameters, and allow them both to become progressively larger, holding all other parameters constant. Here, a natural candidate is the ratio of the number of jobs over the mean think time per operator, which we denote by $\alpha$, which is a measure of the *total* offered rate of submitting jobs:

$$\alpha \equiv \frac{C}{T_{\text{think}}},\ C \to \infty,\ T_{\text{think}} \to \infty.$$

We allow the number of jobs or terminals to become large, as well as the mean intersubmission time of jobs from each terminal, thus weakening the contribution to the total offered rate of each terminal. In the literature, an analogous procedure is called passing from the so-called *finite source* to *infinite source* arrival process (e.g., see Ref. 18, pp. 102–3), granted certain additional distribution assumptions that we do *not* make here (e.g., see Ref. 18, pp. 80–2). If we fix $\alpha$ while allowing $C$, $T_{\text{think}} \to \infty$, we see

$$\frac{\alpha}{1 + \alpha\left[\dfrac{T_{\text{processor}}}{\min(P, M)} + \dfrac{T_{\text{disk}}}{\min(D, M)}\right]} \le \lambda \le \min\left(\frac{P}{T_{\text{processor}}}, \frac{D}{T_{\text{disk}}}, \frac{M}{T_{\text{processor}} + T_{\text{disk}}}\right).$$

This in turn yields the following lower bound on mean delay:

$$R \ge \begin{cases} \infty & \alpha > \dfrac{1}{\max\left[\dfrac{T_{\text{processor}}}{\min(P, M)}, \dfrac{T_{\text{disk}}}{\min(D, M)}, \dfrac{M}{T_{\text{processor}} + T_{\text{disk}}}\right]} \\ T_{\text{processor}} + T_{\text{disk}} & \alpha < \dfrac{1}{\max\left[\dfrac{T_{\text{processor}}}{\min(P, M)}, \dfrac{T_{\text{disk}}}{\min(D, M)}, \dfrac{T_{\text{processor}} + T_{\text{disk}}}{M}\right]}. \end{cases}$$

The remaining case, an upper bound on mean delay or mean response time, is trivial:

$$R \le \infty,\ \alpha \textit{ fixed},\ C \to \infty,\ T_{\text{think}} \to \infty.$$

Additional (distributional) information must be available to allow us to handle the case where

$$\alpha = \frac{1}{\max\left[\dfrac{T_{\text{processor}}}{\min(P, M)}, \dfrac{T_{\text{disk}}}{\min(D, M)}, \dfrac{T_{\text{processor}} + T_{\text{disk}}}{M}\right]}.$$

Inituitively we see that if the total mean arrival rate is less than the

upper bound on the mean throughput rate, then the system is capable of having a *finite* lower bound for mean response time; when the total mean arrival rate is greater than the upper bound on the mean throughput rate, then the mean response time lower bound is *infinite*.

Note that the mean throughput rate lies between two *finite* limits, while the mean response time can lie between a *finite* and *infinite* value, given only mean value information, i.e., the mean response time is *not* well bounded given only this amount of information. This is well known in other types of queueing systems, such as the M/G/1 system (e.g., see Ref. 18, pp. 189–92), where the *mean* delay depends not only on the *first* moment of the service time distribution but also the *second* moment of the service time distribution: mean value information does *not* specify the mean delay in such systems by itself, but rather we need the actual distribution of service time to deal with this issue.

## III. PROTOTYPE DIRECTORY ASSISTANCE SYSTEM CASE STUDY

Here is a case study in using these techniques. A prototype of an on-line transaction processing system was built to handle telephone number directory assistance queries. In a typical cycle of operation, a person at a terminal would

(*i*) Receive a query from a customer via voice telephone

(*ii*) Enter the given information into a computer terminal while talking to the customer

(*iii*) Wait for the system to respond with the answer to the query

(*iv*) Tell the customer over the voice telephone the reply

(*v*) Close out customer interaction

(*vi*) Wait to receive the next customer query.

The hardware configuration for the system consisted of $C$ terminals, a single processor, a single disk controller, and a single disk spindle. An operating system coordinated scheduling and management of these devices, while a set of prototype application programs handled transaction processing.

Measurements on the prototype system in operation showed that

(*i*) The mean time spent by a person talking, reading, and thinking, denoted by $T_{\text{think}}$, was twenty seconds

(*ii*) The mean processor time per transaction was broken down into three sets of application programs

(a) The operator interface front-end programs consumed 180 milliseconds of processor time per query on the average

(b) The index manipulation application programs consumed 420 milliseconds of processor time per query on the average

(c) The data retrieval application programs consumed 330 milliseconds of processor time per query on the average

(d) Miscellaneous application programs that were invoked for

accounting, system administration, and other purposes consumed one hundred and forty milliseconds (140 ms) per query

Hence, the total mean processor time per query, $T_{\text{processor}}$, was 1.07 seconds

(*iii*) The mean number of disk accesses per query was twenty six (26), with the disk capable of making one access every twenty five milliseconds (25 ms), which results in a mean time the disk is busy per query, denoted $T_{\text{disk}}$, of six hundred fifty milliseconds (650 ms).

The above measurements on total mean processor time and disk access counts were based on examining the mean resources required for one hundred different transactions to the system; the measurement error on the processor time was felt to be under ten milliseconds, while the measurement error on the number of disk accesses was felt to be under one access. For this level of analysis, the upper and lower mean value bounds on mean response time are given by

$$\max\left[T_{\text{processor}} + T_{\text{disk}}, \frac{C}{\max(T_{\text{processor}}, T_{\text{disk}})} - T_{\text{think}}\right] \leq R \leq C(T_{\text{processor}} + T_{\text{disk}}),$$

while the associated upper and lower mean value bounds on mean throughput rate are given by

$$\frac{C}{T_{\text{think}} + C(T_{\text{processor}} + T_{\text{disk}})} \leq \lambda \leq \min\left[\frac{C}{T_{\text{think}} + T_{\text{processor}} + T_{\text{disk}}}, \frac{1}{\max(T_{\text{processor}}, T_{\text{disk}})}\right].$$

These bounds are plotted in Figs. 5 and 6, along with observed data gathered over an eight-hour time interval with twelve $C = 12$ operators and calculations based upon a closed queueing network model obeying product-form-type solution. The goodness of fit of the closed queueing network model to actual data was felt to be acceptable for the purposes at hand; the mean value lower bound on mean delay and upper bound on mean throughput rate were also felt to give an indication of performance limitations at an early stage of development, which the data gathering and refinement via a closed queueing network model only strengthened further. Note that the system is achieving a great deal of concurrency, because the actual mean throughput rate is much closer to the upper bound, not the single-thread lower bound. Similar observations hold for mean delay.

## IV. PROTOTYPE DATA BASE ADMINISTRATION SYSTEM CASE STUDY

A transaction processing system administers the data base for a

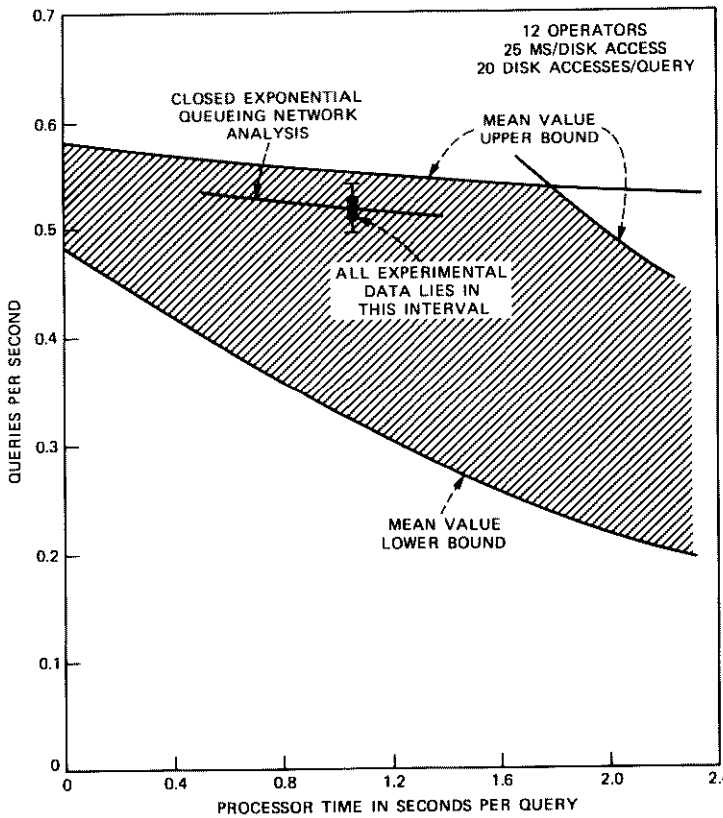



Fig. 5—Mean throughput rate (bounds and data) versus $T_{\text{processor}}$.

second system that switches telephone calls; hence this system is called a *front-end* system to the *back-end* telephone call switching system. Transactions involve additions, deletions, and changes to existing telephone numbers in the switching system files. A prototype system had a hardware configuration consisting of a single processor, a single disk controller, and a single disk spindle, with a fixed number of asynchronous terminals. This same prototype had an operating system to coordinate and schedule these resources, while application programs handled the transaction processing. The application programs were structured into a front end for handling operator terminal interactions, a data base management system, and a back-end communications system for interacting with various switching systems.

The same formulas for upper and lower mean value bounds on mean response time and mean throughput rate hold as in the previous example, except for a change in the numbers.

Two sets of measurements were gathered, one at the start of per-

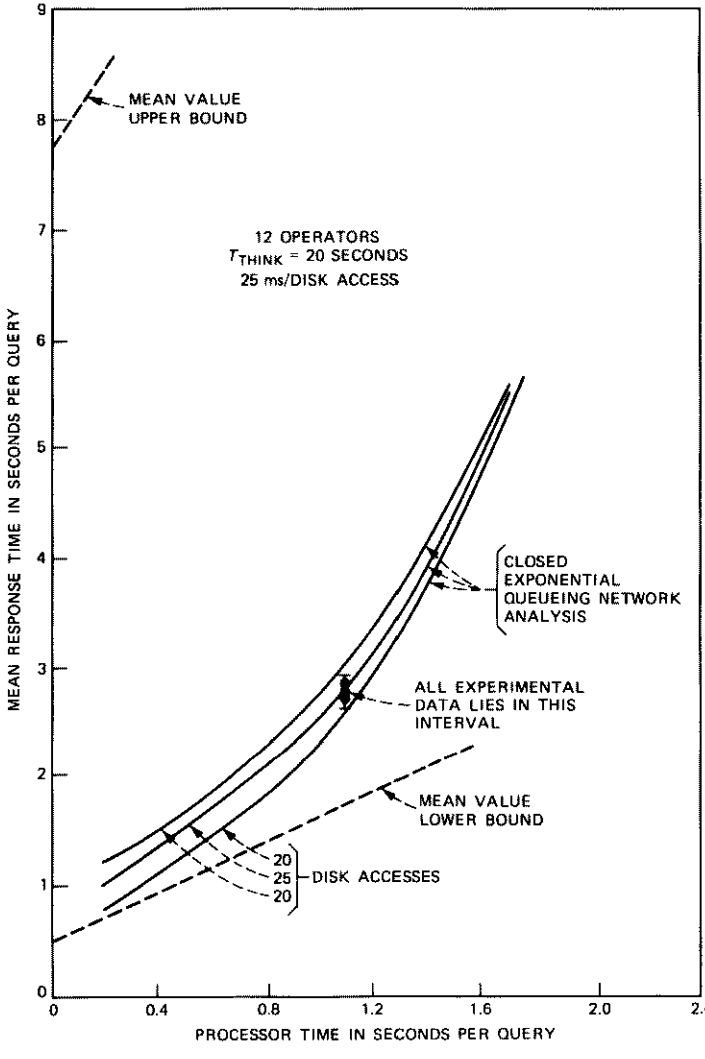



Fig. 6—Mean delay (bounds and data) versus $T_{processor}$.

formance analysis, labeled *initial* in Table I, and one after completing two months of performance analysis, which involved recoding application programs to take better advantage of operating system features, with the same hardware configuration, labeled *final* in Table I. Measurements were carried out in a controlled environment where the actual hardware, operating system, and application programs were used, but the operator behavior was simulated by a second computer. The behavior of each operator was modeled by a script, involving a time for reading, thinking, and typing, followed by a time waiting for

the system to respond. After an initial startup transient the measurements of response time were quite predictable for all operations, with the measurement error being one second at most. Each operator submitted tens of jobs, and the results were averaged over all operators and all jobs, so the final statistics were felt to be statistically reproducible, to within a fraction of a second.

Figures 7 and 8 plot the mean value upper and lower bounds as well as data from these measurements for the mean response time and mean throughput rate as a function of number of operators. The goodness of fit to mean value bounds was felt to be acceptable for the purpose. Unlike the first case study, the data here clearly shows that a great deal of fluctuation was encountered in system operation under load: for the initial system, the fluctuations were so great that the system apparently was always executing only one transaction at a time, while for the final system, as load built up, the system effectively moved from a regime of two tasks making use of both serially reusable resources to a regime where only one task at a time was in execution. This is in contrast to the other set of data, where the system is always achieving a great deal of concurrency under load. A closed exponential

Table I—Prototype system measurements

| Quantity | Initial (seconds) | Final (seconds) |
|---|---|---|
| $T_{think}$ | 15.0 | 15.0 |
| $T_{processor}$ | 8.2 | 3.5 |
| $T_{disk}$ | 5.0 | 0.5 |

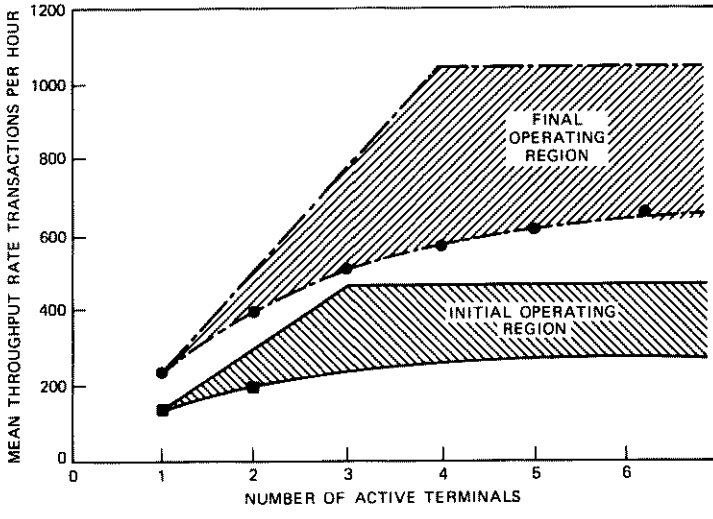



Fig. 7—Mean throughput rate (bounds and data) versus number of active clerks.

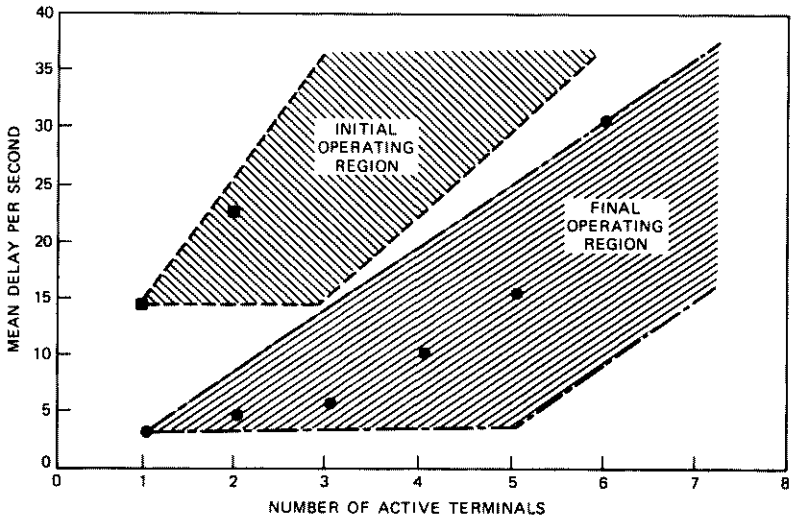



Fig. 8—Mean delay (bounds and data) versus number of active clerks.

queueing network model of this system would predict behavior that closely tracked the upper bound on mean throughput rate and lower bound on mean delay, and would simply not allow for sufficient fluctuation to drive operation into a mode of operation of executing one task at a time. In fact, this suggested a problem with *memory management* that was forcing the system into this mode of operation; an obvious test that was not carried out owing to lack of time was to add more main memory to see if more concurrency might be achieved.

## V. A MODEL OF FLOW CONTROL OVER A SINGLE LINK

An on-line communications system consists of operators at terminals who send messages to one another. The system consists of a transmitter and a receiver, with communication channels connecting the transmitter and receiver. The receiver is capable of buffering only a maximum number of messages at any one time, which is a memory constraint.

### *5.1 Model description*

A communications system is composed of a transmitter processor, a receiver processor, a set of buffers each capable of holding one message at the receiver, and a noiseless communications link. Here are the steps involved in sending a message from the transmitter to the receiver:

(*i*) The transmitter processes a message. This step has a mean

duration $T_{\text{trans}}$ at the transmitter, and it requires both the transmitter *and* a buffer at the receiver.

(*ii*) The message is sent over the link from the transmitter to the receiver. This step has a mean duration $T_{\text{trans-rec}}$.

(*iii*) The receiver processes the message. This step has a mean duration $T_{\text{rec}}$.

(*iv*) An acknowledgment of correct receipt of the message is sent from the receiver to the transmitter. This step has a mean duration $T_{\text{rec-trans}}$. At the start of this step, the receiver marks the buffer free.

(*v*) The transmitter processes the acknowledgment. This step has a mean duration of $T_{\text{ack}}$.

At the end of this step, the transmitter marks the buffer free.

We assume from this point on that the time required by the transmitter to process the acknowledgment is zero. Figure 9 shows a hardware block diagram of the system. Figure 10 shows a queueing network block diagram of the system. The system state is denoted by $\Omega$ where

$$\Omega = \{(J_{\text{trans}}, J_{\text{trans-rec}}, J_{\text{rec}}, J_{\text{rec-trans}}) \mid J_{\text{trans}} + J_{\text{trans-rec}} + J_{\text{rec}} + J_{\text{rec-trans}} = B\}$$

At any instant of time, the system is in a state given by a four tuple, $(J_{\text{trans}}, J_{\text{trans-rec}}, J_{\text{rec}}, J_{\text{rec-trans}})$, where each component is nonnegative and integer valued, and the state space constraint is obeyed.

The mean throughput rate is denoted by $\lambda$. The mean number of jobs in execution in the transmitter and in the receiver equals the mean throughput rate multiplied by the total mean execution time, as shown in a later section. We denote by $E(\cdot)$ the time average of the argument, and write:

$$\lambda T_{\text{trans}} = E[\min(J_{\text{trans}}, P_{\text{trans}} = 1)]$$

$$\lambda T_{\text{rec}} = E[\min(J_{\text{rec}}, P_{\text{rec}} = 1)].$$

Our goal is to find upper and lower bounds on mean throughput rate, subject to meeting state space constraints.

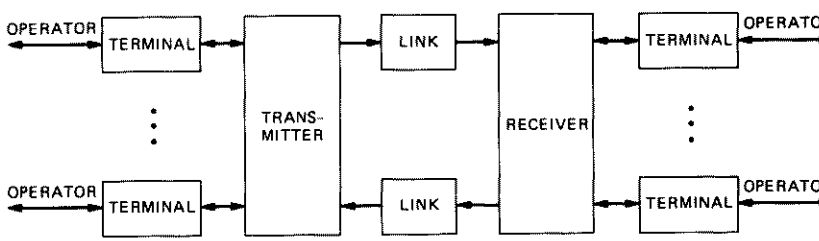



Fig. 9—Hardware block diagram.

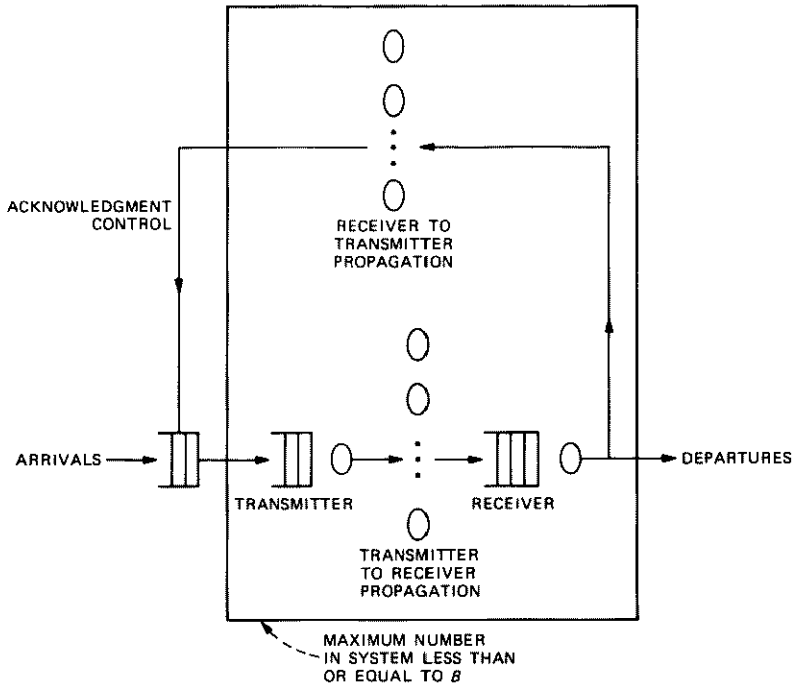



Fig. 10—Queueing network model.

In a later section, we show

$$\frac{B}{BT_{\text{trans}} + T_{\text{trans-rec}} + BT_{\text{rec}} + T_{\text{rec-trans}}} = \lambda_{\text{lower}} \leq \lambda$$

$$\lambda \leq \lambda_{\text{upper}} = \min\left(\frac{1}{T_{\text{trans}}}, \frac{1}{T_{\text{rec}}}, \frac{B}{T_{\text{trans}} + T_{\text{trans-rec}} + T_{\text{rec}} + T_{\text{rec-trans}}}\right).$$

The physical interpretation of the upper bound on mean throughput rate is as follows

(*i*) The transmitter is the bottleneck

$$\lambda_{\text{upper}} = \frac{1}{T_{\text{trans}}}$$

(*ii*) The receiver is the bottleneck

$$\lambda_{\text{upper}} = \frac{1}{T_{\text{rec}}}$$

(*iii*) The number of buffers is the bottleneck

$$\lambda_{\text{upper}} = \frac{B}{T_{\text{trans}} + T_{\text{trans-rec}} + T_{\text{rec}} + T_{\text{rec-trans}}}.$$

The physical interpretation of the lower bound is that at most one message at a time is being handled by the system.

Figures 11 through 13 plot these upper and lower bounds, as well as the results of an exponential queueing network analysis,[1-3] for the special case where

$$T_{\text{trans}} = T_{\text{rec}},\ T_{\text{trans-rec}} = T_{\text{rec-trans}}$$

for three different cases, where the propagation delay is much smaller, equal, and much larger than the mean processing time at either end of the link. The fraction of time the queueing network model predicts the system to be in state $\boldsymbol{J}$ is denoted by $\pi(\boldsymbol{J})$, where

$$\pi(\boldsymbol{J}) = \frac{1}{G} {T_{\text{trans}}}^{J_{\text{trans}}} \frac{{T_{\text{trans-rec}}}^{J_{\text{trans-rec}}}}{J_{\text{trans-rec}}!} {T_{\text{rec}}}^{J_{\text{rec}}} \frac{{T_{\text{rec-trans}}}^{J_{\text{rec-trans}}}}{J_{\text{rec-trans}}!}.$$

The system partition function denoted $G$ is chosen to normalize the probability distribution:

$$\sum_{\boldsymbol{J} \in \Omega_B} \pi(\boldsymbol{J}) = 1.$$

### 5.2 Negligible link propagation delay

We now restrict attention to the special case where the propagation delay is negligible compared to the processing at either end of the link, from this point on. For one buffer, the mean throughput rate is upper bounded by

$$\lambda \leq \lambda_{\text{upper}} = \frac{1}{T_{\text{trans}} + T_{\text{trans-rec}} + T_{\text{rec}} + T_{\text{rec-trans}} + T_{\text{ack}}} = \frac{1}{T_{\text{trans}} + T_{\text{rec}}}.$$

There is no concurrency or parallel execution of messages, and the

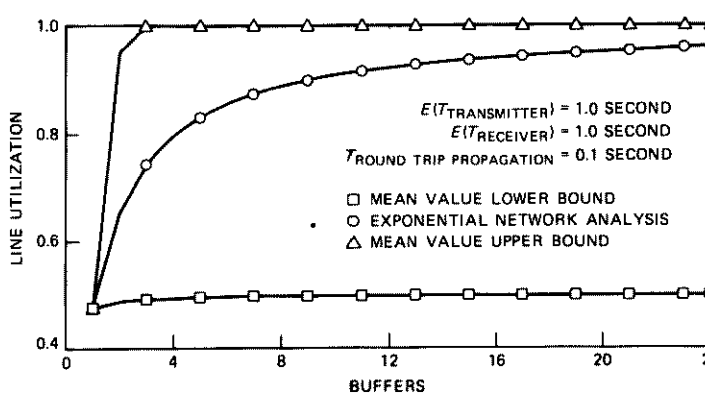



Fig. 11—Line utilization vs. number of buffers ($T_{\text{trans-rec}} = T_{\text{trans}}/10$).

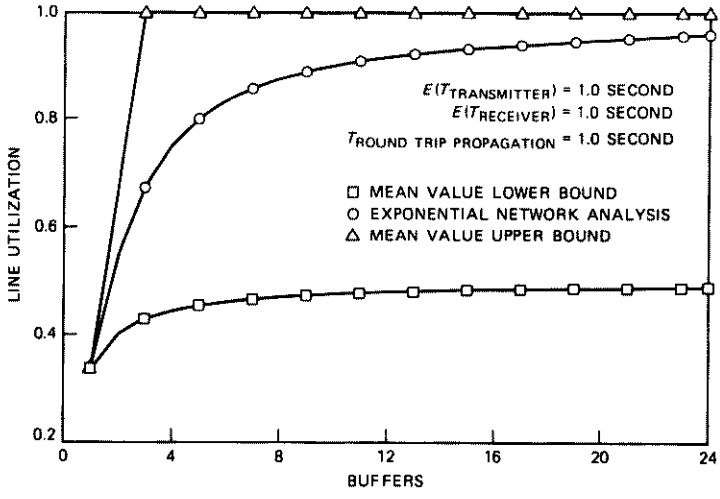



Fig. 12—Line utilization versus number of buffers ($T_{trans\text{-}rec} = T_{trans}$).

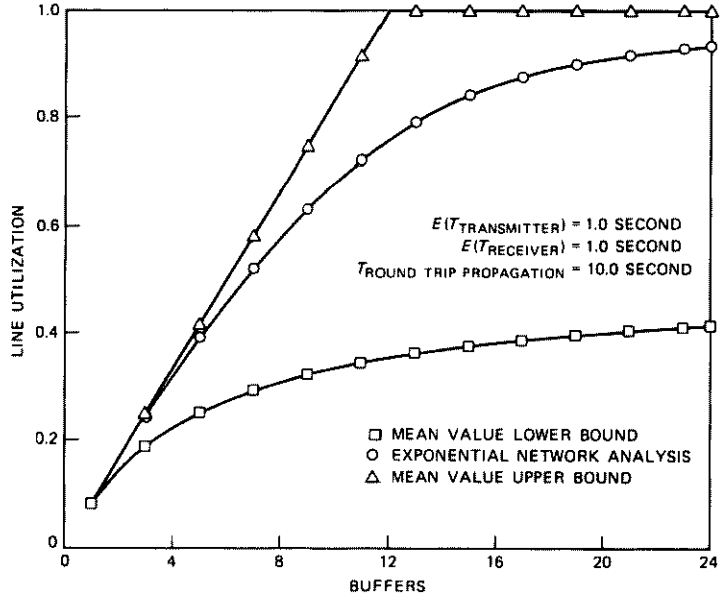



Fig. 13—Line utilization versus number of buffers ($T_{trans\text{-}rec} = 10\ T_{trans}$).

total time required for message handling is the sum of the individual steps.

For more than one buffer, this will yield an upper bound on the mean throughput rate of simply $B$ times the mean throughput rate for

one buffer:

$$\lambda \leq \lambda_{\text{upper}} = \frac{B}{T_{\text{trans}} + T_{\text{trans-rec}} + T_{\text{rec}} + T_{\text{rec-trans}} + T_{\text{ack}}} = \frac{B}{T_{\text{trans}} + T_{\text{rec}}}.$$

On the other hand, as the number of messages increases, then either the transmitter or the receiver (or both) will become completely busy, yielding different upper bounds on mean throughput rate:

(*i*) The transmitter is a bottleneck

$$\lambda \leq \lambda_{\text{upper}} = \frac{1}{T_{\text{trans}} + T_{\text{ack}}} = \frac{1}{T_{\text{trans}}}$$

(*ii*) The receiver is a bottleneck

$$\lambda \leq \lambda_{\text{upper}} = \frac{1}{T_{\text{rec}}}.$$

Combining all this, we see

$$\lambda \leq \lambda_{\text{upper}} = \min\left(\frac{1}{T_{\text{trans}} + T_{\text{ack}}}, \frac{1}{T_{\text{receiver}}}, \frac{B}{T_{\text{trans}} + T_{\text{trans-rec}} + T_{\text{rec}} + T_{\text{rec-trans}} + T_{\text{ack}}}\right)$$

$$\lambda \leq \lambda_{\text{upper}} = \min\left(\frac{1}{T_{\text{trans}}}, \frac{1}{T_{\text{rec}}}, \frac{B}{T_{\text{trans}} + T_{\text{rec}}}\right).$$

Increasing the number of buffers from one to two, $B = 1$ to $B = 2$ always increases the maximum mean throughput rate, and now we see

$$\lambda \leq \lambda_{\text{upper}} = \min\left(\frac{1}{T_{\text{trans}}}, \frac{1}{T_{\text{rec}}}\right) \qquad B > 1.$$

Furthermore, this increase is maximized for $T_{\text{trans}} = T_{\text{rec}}$, and then the upper bound *doubles* in going from one buffer to more than one buffer. Why is this so? By having more than one buffer, both the transmitter and receiver can simultaneously be filling and emptying a buffer, allowing greater concurrency or parallelism compared with the single-buffer case. We also note that allowing more than two buffers, e.g., *infinite* buffers, will not increase the upper bound on the maximum mean throughput rate any further. This is because there are only two serially reusable resources, a transmitter and a receiver, so once they are concurrently busy, no further gains can be achieved.

For the lower bound on mean throughput rate, we see that

$$\lambda \geq \lambda_{\text{lower}} = \frac{B}{BT_{\text{trans}} + BT_{\text{rec}}} = \frac{1}{T_{\text{trans}} + T_{\text{rec}}},$$

which is identical to the upper bound for $B = 1$. Why is this so? There may be significant fluctuation about the mean values shown above,

and in the limit of one big swing about the mean value all of the messages will pile up at one stage in the network and nothing will be transmitted until buffers become available.

### 5.3 Impact of fluctuations

We now examine one special case of this problem in detail, where $T_{\text{trans-rec}} = T_{\text{rec-trans}} = T_{\text{ack}} = 0$, and we wish to study the impact of fluctuations about mean values on system performance. We assume the transmitter processing times are sequences of independent identically distributed exponential random variables with mean $T_{\text{trans}}$. We assume the receiver processing times are sequences of independent identically random variables with common hyperexponential distribution $G_{\text{receiver}}(X)$:

$$G_{\text{receiver}}(X) = (1 - \alpha) + \alpha(1 - e^{-X\mu_{\text{rec}}}).$$

In words, a fraction $1 - \alpha$ will require zero processing time at the receiver, while a fraction $\alpha$ will require an exponentially distributed amount of processing time with mean $1/\mu_{\text{rec}}$. The parameter $\alpha$ gives us an additional degree of freedom to model fluctuations in the receiver processing times. For this case, we choose to fix the *squared coefficient of variation* denoted by $C^2$, which for the random variable $X$ is defined as the ratio of the variance to square of the mean (the standard deviation, measured in units of mean value, squared):

$$\text{squared coefficient of variation} = \frac{\text{variance}(X)}{E^2(X)} \equiv C^2.$$

When this is zero, the variance is zero, and there is zero fluctuation about the mean. When this is one, we have an exponential distribution, where the standard deviation equals the mean. When this is greater than one, the standard deviation is greater than the mean. For this particular case, we see $0 < \alpha \leq 1$ and hence

$$C^2 = \frac{2}{\alpha} - 1 \geq 1.$$

If the mean is fixed but $\alpha$ is varied from one (the exponential distribution case, where the fluctuations are the order of the mean) to zero (increasing fluctuations about the mean), with most jobs taking zero time but a few taking a very long time, we can gain insight into the impact on performance. Since we have fixed the squared coefficient of variation, the mean is also fixed, since

$$T_{\text{rec}} = \frac{\alpha}{\mu_{\text{rec}}}.$$

The distribution of the number in the receiver subsystem at the

completion of processing at the receiver of a message is denoted by $F(K)$, $K = 0, \cdots, B$. If none are left behind, then the mean time to the next completion epoch is $T_{\text{trans}} + T_{\text{rec}}$. If more than zero are left behind at the receiver, then the mean time to the next completion epoch is $T_{\text{rec}}$. The mean throughput rate is given by

$$\lambda = \frac{1}{F(0)(T_{\text{trans}} + T_{\text{rec}}) + [1 - F(0)](T_{\text{rec}})} = \frac{1}{F(0)T_{\text{trans}} + T_{\text{rec}}}.$$

Once we find the distribution of the number of messages in the system at completion epochs, we are done. However, this is a well-known result (see Ref. 18, pp. 235–40), and we merely summarize the known formulas here for the sake of completeness:

$$F(0) = \frac{Q(0)}{\sum_{K=0}^{B-1} Q(K)}.$$

The terms $Q(K)$, $K = 0, \cdots, B - 1$ are given implicitly via the following moment-generating function $\zeta(X)$:

$$\zeta(X) = \sum_{K=0}^{\infty} Q(K)X^K = \frac{(1 - X)E[e^{-\lambda(1-X)T_{\text{rec}}}]}{E[e^{-\lambda(1-X)T_{\text{rec}}}] - X}.$$

Illustrative numerical results are plotted in Figs. 14 through 16 assuming the transmitter and receiver service times are independent, identically distributed, exponential random variables. We note that for the special case where $T_{\text{rec}} = T_{\text{trans}} = 1$, the mean throughput rate is given by

$$\lambda = \frac{1 + \dfrac{2(B + 1)}{C^2 + 1}}{2 + \dfrac{2(B - 1)}{C^2 + 1}} \qquad C^2 \geq 1.$$

Here we see that as $C^2 \to \infty$ that the mean throughput rate approaches the lower bound of 1/2 arbitrarily close, i.e., there is no concurrency or gain in going to more than one buffer if the fluctuations are too great. On the other hand, as $B \to \infty$ for $C^2$ fixed, the mean throughput rate approaches one, which is the best possible. The numerical plots show in which regime which phenomenon (the fluctuations or the buffering and concurrency) dominates the actual mean throughput rate. The impact of speed mismatch (i.e., as the transmitter and receiver mean message execution times start to differ) tends to swamp the impact of fluctuations: the greater the speed mismatch, the greater concurrency is achieved, because the exact mean throughput rate approaches the upper bound closer and closer as the speed mismatch increases be-

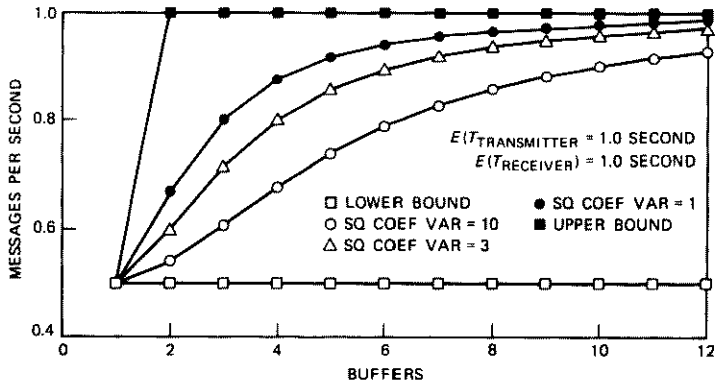



Fig. 14—Mean throughput rate vs. number of buffers for a closed queueing network model ($T_{trans}$ = 1.0 and $T_{rec}$ = 1.0).

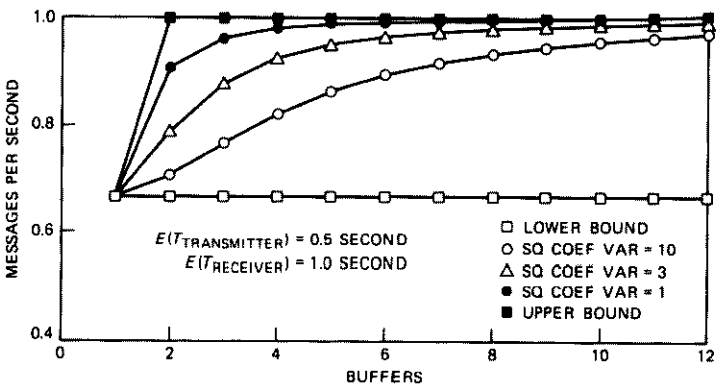



Fig. 15—Mean throughput rate vs. number of buffers for a closed queueing network model ($T_{trans}$ = 0.5 and $T_{rec}$ = 1.0).

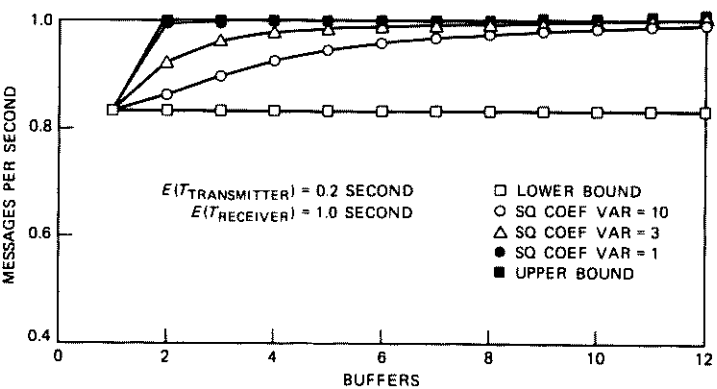



Fig. 16—Mean throughput rate vs. number of buffers for a closed queueing network model ($T_{trans}$ = 0.2 and $T_{rec}$ = 1.0).

tween transmitter and receiver. Note that the upper bound on mean throughput rate corresponds to a squared coefficient of variation of zero, while the lower bound corresponds to a squared coefficient of variation that becomes infinite.

We now discuss this phenomenon in more detail, because the formulas give only one way of understanding this model. Figure 17 shows a sample path generated from a simulation of the model, for a total number of five jobs in the system. In the initial part of the simulation, the first stage fluctuates between four and five jobs, while the second stage fluctuates between zero and one job; in the final part of the simulation, the situation is reversed; after sufficiently long time, we would return to the first case. When most of the jobs are at one stage, the mean throughput rate is roughly the reciprocal of the time to execute one job from start to finish, and there is no concurrency. The other cases, where there are multiple jobs at each stage, are transient and the system spends relatively little time in these states.

The analysis developed above can make these intuitive notions more precise. For example, the mean fraction of time that there are zero jobs at the receiver is

$$\text{fraction of time zero jobs at receiver} = \frac{F(0)}{F(0) + \dfrac{T_{\text{rec}}}{T_{\text{trans}}}},$$

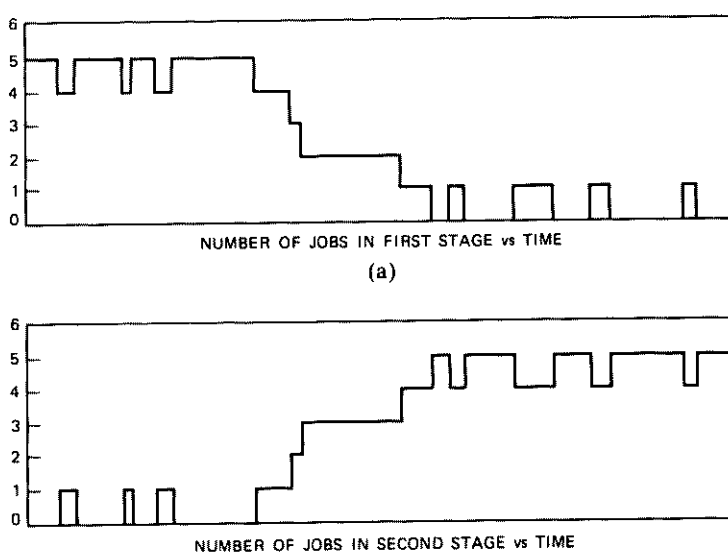



Fig. 17—Sample path generated from a simulation of the model for (a) first stage vs. time and (b) second stage vs. time.

while the fraction of time all the jobs are at the receiver is

$$\text{fraction of time all jobs at receiver} = 1 - \frac{1}{F(0) + \dfrac{T_{\text{rec}}}{T_{\text{trans}}}}.$$

As we allow $\alpha \to 0$, i.e., as the fluctuations and squared coefficient of variation become larger, while the mean time spent at the transmitter and receiver stay fixed, we see that the sum of these two fractions can be made *arbitrarily* close to one, which is what the simulation result in Fig. 17 shows. At the same time, we see that the mean sojourn time in the state where the receiver is empty is given by

$$\text{mean sojourn time in idle receiver state} = \sum_{K=1}^{\infty} (1-\alpha)^{K-1} \alpha K T_{\text{trans}} = \frac{T_{\text{trans}}}{\alpha} \to \infty \quad \alpha \to 0.$$

Put differently, if one were to measure the operation of this system, the system might be in the receiver idle state for the entire duration of the observation process, and the other state of the receiver having all jobs (which will also become successively longer and longer as $\alpha \to 0$) will never be observed, or vice versa! In Fig. 17, this would correspond to gathering data in the first part of the simulation, never in the second part, or vice versa.

### 5.4 Queueing network analysis for negligible propagation delay

In a later section, we show that the mean throughput rate is upper and lower bounded by

$$\lambda_{\text{lower}} \le \lambda \le \lambda_{\text{upper}}$$

$$\lambda_{\text{lower}} = \frac{1}{T_{\text{trans}} + T_{\text{rec}} + \max(T_{\text{trans}}, T_{\text{rec}})}$$

$$\lambda_{\text{upper}} = \frac{1}{T_{\text{trans}} + T_{\text{rec}} + 1/2(T_{\text{trans}} + T_{\text{rec}})}.$$

The mean value bounds, the queueing network upper and lower bounds, and exact queueing network analysis mean throughput calculations are all plotted in Figs. 18 through 20 for $T_{\text{rec}} = 1.0$ and $T_{\text{trans}} = 1.0, 0.5, 0.2$. The queueing network bounds are identical to the exact analysis when the transmitter and receiver execute messages at the same rate. When the transmitter becomes faster than the receiver, the bounds and exact analysis tend to track the upper bound on mean throughput rate; in other words, the speed mismatch is of greater importance than the impact of fluctuations.

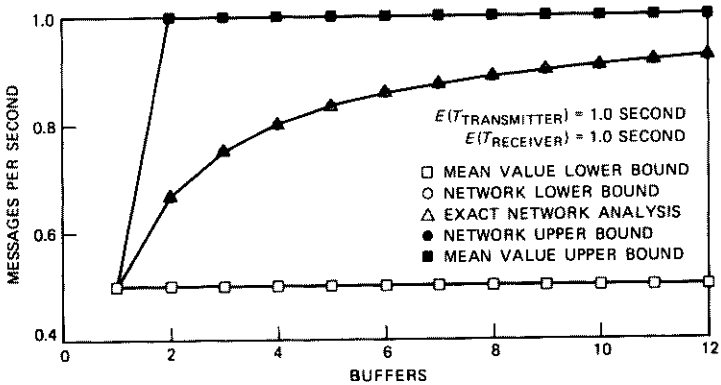



Fig. 18—Mean throughput rate vs. number of buffers for zero propagation time ($T_{trans}$ = 1.0 and $T_{rec}$ = 1.0).

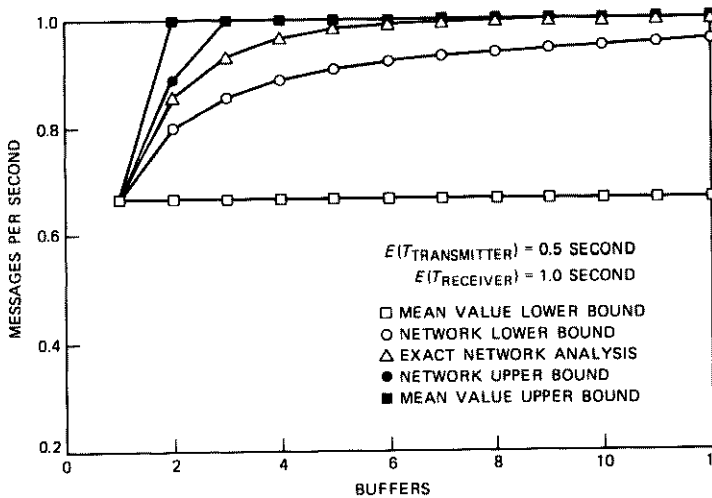



Fig. 19—Mean throughput rate vs. number of buffers for zero propagation time ($T_{trans}$ = 0.5 and $T_{rec}$ = 1.0).

### *5.5 Experimental data*

To test predictions of this analysis against actual operations, a series of experiments were carried out to determine the mean maximum throughput rate of a data communications link constructed with two computers, one transmitting and one receiving, over a data link where the link propagation time was negligible compared to the data communications processing at either end of the link or the data transmission time of a packet over this link. The test described here involved sending 51,200 bytes of data over a 9600-b/s data link; similar results were found for a 1200-b/s data link. The source data were encoded

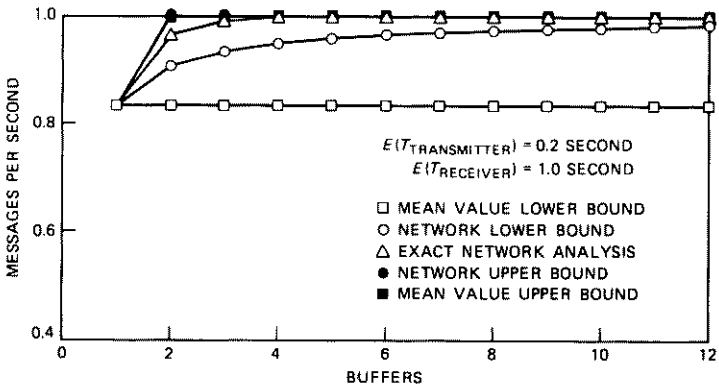



Fig. 20—Mean throughput rate vs. number of buffers for zero propagation time ($T_{trans}$ = 0.2 and $T_{rec}$ = 1.0).

into packets containing either 32, 64, 128, or 256 bytes (one byte equals eight bits) of data. The system and numerous other details of the experiment will be described elsewhere in a different report.

We wish to test the gain in going from start-stop or single buffering to double buffering and to greater than double buffering; our previous analysis assumes that a mean value of data communications processing time at the transmitter and receiver adequately characterizes the system performance.

The experiment involved simply measuring the time required to transmit 51,200 bytes of data over each link for each size packet. No processing was done on the data at either the transmitter or receiver other than to do the data communications processing required for correct operation. The transmitter and receiver processes resided in the same PDP 11/45 computer with a *UNIX**-like operating system environment.

Table II summarizes the results of that experiment. The time required to send each of 51,200 bytes of data plus two additional bits (for parity and control) over a serial 9600-b/s data link is 53.3 seconds; thus, the data link transmission speed and not the transmitter or receiver is limiting data flow here. This can also be seen directly by noting that the link utilization is approaching one hundred per cent in Table II. This table shows that double buffering at the receiver offers substantial improvement in mean message throughput over single buffering, and there is no apparent advantage in terms of throughput in choosing a receiver buffer larger than two (e.g., seven was tried). Finally, this suggests that for this purpose this level of analysis is

* Trademark of Bell Laboratories.

Table II—PDP 11/45 loop-around experiment—maximum mean throughput rate for transmission of 51,200 bytes over 9600-b/s data link

| Number of Buffers | Packet Size (bytes) | Time (seconds) | Maximum Throughput (b/s) | Link Utilization (percent) |
|---|---|---|---|---|
| 1 | 32 | 160.0 | 3200 | 33 |
| 1 | 64 | 125.0 | 4096 | 43 |
| 1 | 128 | 90.0 | 5688 | 59 |
| 1 | 256 | 80.0 | 6400 | 67 |
| 2 | 32 | 80.0 | 6400 | 67 |
| 2 | 64 | 58.5 | 8752 | 91 |
| 2 | 128 | 55.5 | 9225 | 96 |
| 2 | 256 | 55.0 | 9309 | 97 |
| 7 | 32 | 64.0 | 8000 | 83 |
| 7 | 64 | 58.0 | 8827 | 92 |
| 7 | 128 | 55.0 | 9309 | 97 |
| 7 | 256 | 54.5 | 9412 | 98 |

appropriate, i.e., that other phenomena that are present are in fact negligible for these purposes, as shown by the data.

## VI. CONCLUSIONS

A performance study of an computer communication system may be carried out in at least one of three ways:

(*i*) Mean value analysis, as described here[14,15]

(*ii*) Jackson queueing network analysis[3]

(*iii*) Discrete event simulation model.[9–11]

In this paper we have demonstrated the ability of the mean value analysis to present a clear picture of the dependence of computer communication system performance on the values of the model parameters. The mean value analysis is a simple, flexible, inexpensive approach to performance analysis and should always be used, even if it is required to supplement the analysis with one or both of the other techniques. The other approaches quantify the impact of fluctuations about mean values on performance, refining the mean value analysis.

The utility or validity of any of these approaches cannot be judged in the abstract: whichever approach or combination of methods is most appropriate must be judged in terms of the data gathering and measurements, and how the data is used to draw inferences concerning cause and effect phenomena, coupled with the spectrum of practical feasible alternatives. The mean value approach presented here is simply one tool for carrying out this complex decision-making process.

## VII. ACKNOWLEDGMENT

The authors are indebted to numerous colleagues throughout Bell Laboratories for stimulating this work, people involved with the de-

velopment of over one hundred diverse systems who have shared with us their insight into design and analysis considerations over the past five years. The authors are also indebted to Bell Laboratories for providing such a stimulating work environment.

## REFERENCES


1. L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*, Chapters Five and Six, New York: Wiley Interscience, 1976.
2. F. P. Kelly, "*Networks of Queues*," Advances in Applied Probability, *8* No. 2 (June 1976), pp. 416–32.
3. F. P. Kelly, *Reversibility and Stochastic Networks*, Chichester: Wiley, 1979.
4. C. Sauer and K. Chandy, *Computer Systems Performance Modeling*, Englewood Cliffs, NJ: Prentice Hall, 1981.
5. M. Phister, Jr. *Data Processing: Technology and Economics*, Second Edition, Bedford, MA: Digital Press, 1979.
6. A. F. Shackil, "*Design Case History: Wang's Word Processor*," IEEE Spectrum, *18*, No. 8 (August 1981), pp. 29–33.
7. G. H. Engel, J. Groppuso, R. A. Lowenstein, and W. G. Traub, "*An Office Communications System*," IBM Systems Journal, *18*, No. 4 (1979), pp. 402–31.
8. R. P. Uhlig, D. J. Farber, and J. H. Bair, *The Office of the Future: Communication and Computers*, Amsterdam: North-Holland, 1979.
9. G. S. Fishman, *Concepts and Methods in Discrete Event Digital Simulation*, New York: Wiley, 1973.
10. G. S. Fishman, *Principles of Discrete Event Simulation*, New York: Wiley, 1978.
11. H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Reading, MA: Addison Wesley, 1978.
12. J. M. Holtzman, "*The Accuracy of the Equivalent Random Method with Renewal Inputs*," B.S.T.J., *52*, No. 9 (September 1973), pp. 1673–9.
13. A. E. Eckberg, "*Sharp Bounds on Laplace-Stieltjes Transforms, with Applications to Various Queueing Problems*," Mathematics of Operations Research, *2*, No. 2 (1977), pp. 135–42.
14. K. Omahen, "*Capacity Bounds for Multiresource Queues*," J.A.C.M., *24* (1977), pp. 646–63.
15. P. J. Denning and J. P. Buzen, "*The Operational Analysis of Queueing Network Models*," Computing Surveys, *10*, No. 3 (1978), pp. 225–61.
16. S. S. Lam, "*Queuing Networks with Population Size Constraints*," IBM J. Research and Development, *21*, No. 7 (July 1977), pp. 370–8.
17. J. Zahorjan, K. C. Sevick, D. L. Eager, and B. Galler, "*Balanced Job Bound Analysis of Queueing Networks*," Communications of the ACM, *25*, No. 2 (February 1982), pp. 134–41.
18. R. B. Cooper, *Introduction to Queueing Theory*, Second Edition, New York: North Holland, 1981.
19. J. D. C. Little, "*A Proof of the Queueing Formula* $L = \lambda W$," Operations Research, *9* (1961), pp. 383–7.
20. W. S. Jewell, "*A Simple Proof of*: $L = \lambda W$," Operations Research, *15* (1967), pp. 1109–16.
21. R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Reading, MA: Addison-Wesley, 1967; Little's formula, pp. 18–19.
22. W. L. Smith, "*Renewal Theory and Its Ramifications*," J. Royal Statistical Society (Series B), *20*, No. 2 (February 1958), pp. 243–302.
23. G. P. Klimov, *An Erogodic Theorem for Regenerating Processes*, Theory of Probability and Its Applications, *21*, No. 2 (June 1976), pp. 392–5.


## APPENDIX A

### *Little's Law*

Jobs enter a system, spend time within the system, and depart. The system attributes of interest here are:

(*i*) $L(t)$ denotes the number of jobs in the system at time $t$

(*ii*) $C(t)$ denotes the number of completions in the time interval $(0, t]$

(*iii*) Every job that enters the system leaves the system.

Our goal is to relate the mean throughput rate of jobs, the mean time a job spends in the system, and the mean number of jobs in the system.

The total mean time spent by all the jobs in the system is simply the area underneath the function $L(t)$:

$$\text{total mean time in system by all jobs} = \int_0^t L(\tau)d\tau.$$

The total mean time spent in the system by any one job is given by

$$\text{mean time in system per job in } (0, t] \equiv \frac{\int_0^t L(\tau)d\tau}{C(t)}.$$

We multiply and divide by $t$ as follows:

$$\text{mean time in system per job} = \frac{\int_0^t L(\tau)d\tau}{t} \times \frac{1}{C(t)/t}.$$

The first term is simply the mean number of jobs in the system, averaged over a time interval of duration $t$:

$$\text{mean number of jobs in system in } (0, t] \equiv \frac{\int_0^t L(\tau)d\tau}{t}.$$

The second term is simply the mean throughput rate:

$$\text{mean throughput rate in } (0, t] \equiv \frac{C(t)}{t}.$$

Hence, we have shown that the mean number of jobs inside the system equals the mean throughput rate multiplied by the mean time in system per job, all over an interval of duration $(0, t]$:

mean number of jobs in system in $(0, t]$

= mean throughput rate in $(0, t] \times$ mean time in system/job in $(0, t]$.

If the observation interval becomes infinite, $t \to \infty$, and the mean values defined here in fact stabilize and do not fluctuate, then we have what is called Little's Law.[19-21] These other derivations rely on averaging over an ensemble of equally likely experiments, and draw on deep results from the theory of stochastic processes;[22,23] the difficulty is in showing that the limits in fact exist in a meaningful mathematical

sense. Here we focus exclusively on time averages of quantities of interest, since these can be readily measured.

We close with an application of this result that we will use in the following section. Jobs arrive for processing by a system. Each job requires a total mean amount of service $T$. The system consists of a single queue feeding $P$ identical processors. At any given instant of time, there are $J$ jobs in the system, either running or waiting to run. The mean throughput rate of jobs is denoted by $\lambda$.

We now restrict attention to a subsystem of the total system, the subsystem of jobs in execution. Since we have $P$ processors, the number of jobs in execution at any instant of time is $\min[J, P]$. Hence, we see that the mean number of jobs in execution, averaged over a time interval, equals the mean throughput rate multiplied by the mean time a job spends in execution:

$$\text{mean number of jobs in execution} \equiv E[\min(J, P)] = \lambda T.$$

The actual service pattern of the job is not of interest here: a job may actually consist of a series of steps with different processing at each step, and at the conclusion of each step of execution the job returns to the end of the queue (or to some point in the queue based upon the step) until it is completely executed.

## APPENDIX B

### *Mean Value Analysis*

We now present the mathematical analysis to justify assertions in earlier sections. The model we deal with is a system handling only one type of job or transaction. Each transaction consists of one or more steps; at each step, a given amount of a serially reusable resource is required for a given time interval. A resource is any entity that is required for subsequent execution of a transaction; examples of physical hardware resources are processors, memory, disk spindles, disk controllers, communication links, local backplane buses and so forth; example of logical software resources are files, tables, messages, semaphores, and so forth. Here the first step of each transaction involves entering the transaction into the system via an operator at a terminal; the second step of each transaction involves placing the transaction in a staging queue where it will wait if there are more than a given maximum number of jobs already in the system and otherwise will enter the system immediately; and it will go through one or more additional steps inside the system, where the job holds a single serially reusable resource for each step of execution and then moves on, until the job is completely executed and control returns to the operator at the terminal. For each step of each transaction, we are given the amount of each resource and the *mean* time required to hold that set

of resources. We denote by $T_K$ the total mean time spent by a transaction holding resource type $K$, which we stress is the sum total execution time of all visits to that stage by a transaction.

The mathematical model consists of

(*i*) $N + 2$ stages of stations: station 0 is associated with operators at terminals, station 1 is the staging station, and stations $2, \cdots, N + 1$ ($N$ total) are associated with a single, serially reusable resource

(*ii*) Stage $K = 0, 2, \cdots, N + 1$ has $P_K$ identical parallel servers or processors

(*iii*) A maximum of $M$ jobs can be held at all stages $K = 2, \cdots, N + 1$

(*iv*) Each job moves from station to station, and requires $T_K$ total mean amount of service time at stage $K = 0, 2, \cdots, N + 2$.

Figure 21 is a queueing network block diagram of this system.

We denote by $\lambda$ the total mean throughput rate of completing jobs; $R$ denotes the total mean response time (queueing or waiting time plus execution time) per job. The system state space is denoted by $\Omega$. Elements in the state space are denoted by $\boldsymbol{J} = (J_0, \cdots, J_{N+1})$. $J_K$, $K = 0, 2, \cdots, N + 1$ denotes the number of jobs either waiting or in execution at stage $K$.

Feasible elements in the state space obey the following constraints:

(*i*) The total number of tasks in the system is fixed at $P_0$

$$P_0 = |\boldsymbol{J}| = \sum_{K=0}^{N+1} J_K.$$

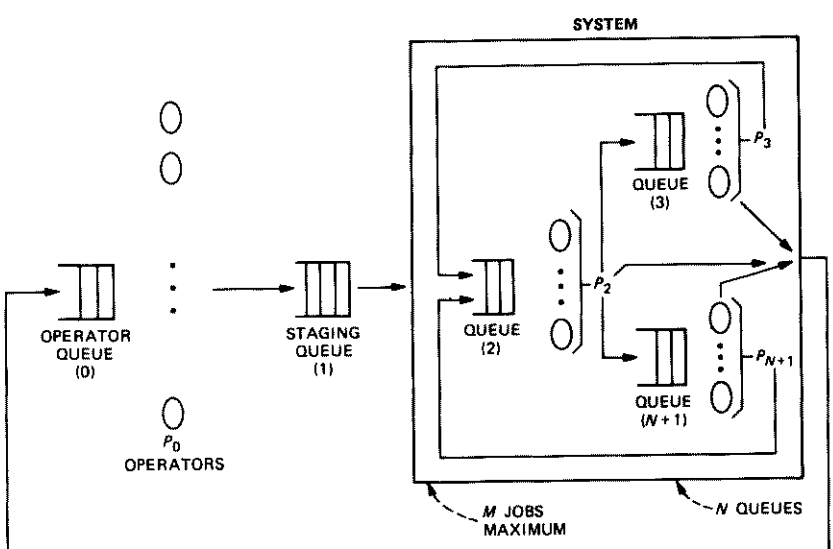



Fig. 21—Block diagram of the queueing network model.

(*ii*) There can be at most a maximum of $M$ jobs inside the system:

$$\sum_{K=2}^{N+1} J_K = \min[M, P_0 - J_0].$$

Combining all these, we see that elements $J$ in $\Omega$ are nonnegative integer-valued tuples such that

$$J \in \Omega = \{V \mid V = (V_0, \cdots, V_{N+1}); \quad V_N \geq 0 \; K = 0, \cdots, N+1; \quad \sum_{K=0}^{N+1} V_K = P_0; \quad \sum_{K=2}^{N+1} V_K = \min(M, P_0 - V_0)\}.$$

The number of jobs *in execution* at stage $K = 0, 2, \cdots, N+1$ is given by $\min(J_K, P_K)$ at any given instant of time. From the previous section, Little's Law allows us to write:

mean number in execution at stage $K$

$$\equiv E[\min(J_K, P_K)] = \lambda T_K \quad K = 0, 2, \cdots, N+1,$$

where $E(\cdot)$ denotes the time average of the argument. Our goal is to find upper and lower bounds on $\lambda$ subject to the state space constraints on $J_K$, $K = 0, \cdots, N+1$.

Since mean throughput rate and mean response time or delay are related via

$$\lambda = \frac{P_0}{T_0 + R}$$

we will also obtain associated lower and upper bounds on mean delay.

### *B.1 Lower bound on mean throughput rate*

We first divide both sides of the following equation

$$E[\min(J_0, P_0)] = \lambda T_0$$

by $P_0$. In like manner, we divide both sides of the following equations

$$\lambda T_K = E[\min(J_K, P_K)] \quad K = 2, \cdots, N+1$$

by $\min(M, P_0, P_K)$. Now we add up these $N+1$ equations:

$$\frac{E[\min(J_0, P_0)]}{P_0} + \sum_{K=2}^{N+1} \frac{E[\min(J_K, P_K)]}{\min(M, P_0, P_K)} = \lambda \left[ \frac{T_0}{P_0} + \sum_{K=2}^{N+1} \frac{T_K}{\min(M, P_0, P_K)} \right].$$

Now we interchange the mean value with the summation on the left-hand side:

$$E\left[\frac{\min(J_0, P_0)}{P_0} + \sum_{K=2}^{N+1} \frac{\min(J_K, P_K)}{\min(M, P_0, P_K)}\right]$$
$$= \lambda\left[\frac{T_0}{P_0} + \sum_{K=2}^{N+1} \frac{T_K}{\min(M, P_0, P_K)}\right].$$

Our goal is to lower bound the left-hand side by one, which will yield a lower bound on $\lambda$.

Two cases can arise. First, there can exist one $I = 2, \cdots, N+1$ such that $P_I \leq J_I$. Since all the terms on the left-hand side are non-negative, we can lower bound the left-hand side by ignoring all of these terms except term $I$:

$$\frac{\min(J_0, P_0)}{P_0} + \sum_{K=2}^{N+1} \frac{\min(J_K, P_K)}{\min(M, P_0, P_K)} \geq \frac{\min(J_I, P_I)}{\min(M, P_0, P_I)}$$
$$\geq \frac{P_I}{\min(M, P_0, P_I)} \geq 1 \quad I = 2, \cdots, N+1.$$

Second, for all $K = 0, 2, \cdots, N+1$, $P_K > J_K$ and hence

$$\min(J_K, P_K) = J_K \quad K = 2, \cdots, N+1.$$

Two subcases arise: if $P_0 - J_0 \leq M$ then there is no waiting by any job in the staging queue, and

$$\frac{J_0}{P_0} + \sum_{K=2}^{N+1} \frac{J_K}{\min(M, P_0, P_K)} \geq \frac{J_0}{P_0} + \frac{P_0 - J_0}{P_0} = 1.$$

The other subcase is if $P_0 - J_0 > M$ and then there is waiting in the staging queue, so

$$\frac{\min(J_0, P_0)}{P_0} + \sum_{K=2}^{N+1} \frac{J_K}{\min(M, P_0, P_K)} \geq \sum_{K=2}^{N+1} \frac{J_K}{\min(M, P_0, P_K)} = \frac{M}{M} = 1.$$

Hence, we see that

$$\lambda\left[\frac{T_0}{P_0} + \sum_{K=2}^{N+1} \frac{T_K}{\min(M, P_0, P_K)}\right] \geq 1$$

and we obtain the desired lower bound:

$$\lambda_{\text{lower}} = \frac{P_0}{T_0 + \sum_{K=2}^{N+1} \frac{P_0}{\min(M, P_0, P_K)} T_K}.$$

The total mean time to execute a job at each stage in the system has been stretched from $T_K$, $K = 2, \cdots, N+1$ to $\tilde{T}_K$, $K = 2, \cdots, N+1$, where

$$\tilde{T}_K = \frac{P_0}{\min(M, P_0, P_K)} T_K \geq T_k \quad K = 2, \cdots, N+1$$

$$\lambda_{\text{lower}} = \frac{P_0}{T_0 + \sum_{K=2}^{N+1} \tilde{T}_K},$$

which is one way of quantifying the slowdown at each node owing to congestion.

### *B.2 Upper bound on mean throughput rate*

From the definition of $\lambda$ we see

$$\lambda = \frac{E[\min(J_K, P_K)]}{T_K} \leq \frac{\min(P_K, P_0, M)}{T_K} \quad K = 0, 2, \cdots, N+1.$$

From this same identity, we obtain a second upper bound:

$$\lambda T_K \leq E(J_K) \quad K = 0, 2, \cdots, N+1$$

$$\rightarrow \lambda\left(T_0 + \sum_{K=2}^{N+1} T_K\right) \leq E\left(J_0 + \sum_{K=2}^{N+1} J_K\right) = P_0.$$

The constraint on the maximum number of jobs inside the system can be written as

$$\sum_{K=2}^{N+1} J_K \leq \min(P_0, M).$$

If we use Little's Law, we see

$$\lambda \sum_{K=2}^{N+1} T_K = \sum_{K=2}^{N+1} E(J_K) \leq \min(M, P_0).$$

In summary, we have shown

$$\lambda \leq \min \left\{ \min_{K=0,2,\cdots,N+1} \left[ \frac{\min(P_0, P_K, M)}{T_K} \right], \frac{P_0}{T_0 + \sum_{K=2}^{N+1} T_K}, \frac{\min(M, P_0)}{\sum_{K=2}^{N+1} T_K} \right\}.$$

### *B.3 Interpretation*

One intuitive explanation for these bounds is the following. To achieve the upper bound on mean throughput rate, each step of job execution has little fluctuation relative to its mean value, and jobs interleave with one another. The mean throughput rate can be upper bounded via the following mechanisms:

(*i*) The total number of jobs circulating in the system is limiting the mean throughput rate; in this regime, as we increase the number

of jobs, the mean throughput rate increases in roughly the same proportion

(*ii*) One stage is executing jobs at its maximum rate, limiting the mean throughput rate; in this regime, as we increase either the speed of each processor at the stage, or the number of processors with the same speed, the mean throughput rate increases in roughly the same proportion

(*iii*) The constraint on the maximum number of jobs in the system is limiting the mean throughput rate; in this regime, as we increase the allowable maximum number of jobs in the system, the mean throughput rate increases accordingly.

To achieve the lower bound on mean throughput rate, each step of job execution has large fluctuations relative to its mean value, so that all jobs in the system are congested at one node. A different way of gaining insight into this lower bound is to replace the service or processing time distribution at each node with a bimodal distribution with the same mean as the old distribution, where $(1 - \epsilon_K)$ denotes the fraction of jobs at stage $K$ that are executed in "zero" time and $\epsilon_K$ is the fraction of jobs at stage $K$ that are executed in time $1/\mu_K$ such that $T_K = \epsilon_K/\mu_K$. Here in normal operation two things can occur: the mean time for a job to cycle through the network will be roughly zero, since most stages will take zero time, and hence the number of jobs in circulation will limit the mean throughput rate, or one stage of execution will take a time that is much longer relative to all the other times, and hence all but one or two jobs will be congested at one node, thus limiting the mean throughput rate.

## APPENDIX C

### *Product Form Distribution Results*

The mathematical model considered in this section is a special case of that considered in the previous section:

(*i*) One type of job that migrates amongst $S$ stations or stages

(*ii*) A *single* processor available to execute a job at stage $K = 1, \cdots, S$

(*iii*) $N$ tasks or jobs circulate among the nodes

(*iv*) $T_K$ denotes the total mean amount of service required by a job summed over all its visits to stage $K = 1, \cdots, S$.

The system state is denoted by $\Omega$:

$$\Omega = \left\{ (J_1, \cdots, J_S) \mid \sum_{K=1}^{S} J_K = N \right\}.$$

At any given instant of time, the system is in state $\boldsymbol{J} = (J_1, \cdots, J_S)$,

where $J_K$, $K = 1, \cdots, S$ denotes the number of jobs at node $K$ (both waiting and in execution). The long-term time-averaged distribution of number of jobs at each node at an arbitrary instant of time is assumed from this point on to obey a so-called *product form* or separation of variables formula

$$\text{PROB}(J_1 = K_1, \cdots, J_S = K_S) = \frac{1}{G_N} \prod_{I=1}^{S} T_I^{K_I} \qquad (K_1, \cdots, K_S) \in \Omega$$

$$G_N = G_N(T_1, \cdots, T_S) = \sum_{J \in \Omega} \prod_{I=1}^{S} T_I^{J_I}.$$

The interested reader is referred to the literature[3] for probabilistic assumptions that lead to this type of probability measure on $\Omega$, the admissible state space. $G_N$ is the *system partition function* chosen to normalize the product form to a probability measure.[3] Granted these assumptions, we observe that the mean throughput rate of jobs making a complete cycle of the system is given by

$$\lambda = \frac{\text{PROB}(J_K > 0)}{T_K} = \frac{G_{N-1}(T_1, \cdots, T_S)}{G_N(T_1, \cdots, T_S)}.$$

Our goal is to obtain *tighter* upper and lower bounds on mean throughput rate and hence mean delay than we obtained in the previous section, using this additional information. We begin by observing that

$$\lambda \sum_{K=1}^{S} T_K = \frac{G_{N-1}(T_1, \cdots, T_s) \sum_{K=1}^{S} T_K}{G_N(T_1, \cdots, T_S)}$$

is a symmetric function of the $S$ variables $T_K$, i.e., we do not change the value of the function when we interchange any two variables. This property allows us to show that this function has its maximum when all the variables are equal to one another. This follows from calculating first the gradient of the function at that point and showing that it is zero, and second showing that the Hessian, the determinant of all partial second derivatives, is negative definite at that point. An alternate way of seeing this holds is to realize that the gradient is zero (from the symmetry of the function) at the point where all coordinates are unity, so this point must either be a minimum or a maximum; we then evaluate the function at a neighboring point, say the point where all coordinates except one equal zero, and see that this is *less* than at the point where all coordinates are one, so this must be a maximum. Hence, we see

$$\lambda \sum_{K=1}^{S} T_K \leq \frac{SG_{N-1}(T_1 = 1, \cdots, T_S = 1)}{G_N(T_1 = 1, \cdots, T_S = 1)}$$

$$\leq \frac{S\binom{S+N-2}{N-1}}{\binom{S+N-1}{N}} = \frac{SN}{S+N-1}.$$

We now rearrange this upper bound to see

$$\lambda \leq \frac{N}{\sum_{K=1}^{S} T_K + (N-1)T_{\text{average}}}.$$

The first term in the denominator is the mean time for a job to make a complete cycle through the network:

$$T_{\text{cycle}} = \sum_{K=1}^{S} T_K.$$

The second term in the denominator is the mean amount of time per node spent by a job in one cycle of the network:

$$T_{\text{average}} = \frac{1}{S} \sum_{K=1}^{S} T_K.$$

The same method is now used to obtain a lower bound on the mean throughput rate, by observing that

$$\frac{1}{\lambda} = \frac{G_N(T_1, \cdots, T_S)}{G_{N-1}(T_1, \cdots, T_S)}.$$

Without loss of generality, we number the nodes such that node $S$ is the node that will do the greatest amount of processing on a job on the average during one cycle:

$$T_S = \max_{K=1,\cdots,S} T_K.$$

This can be used to rewrite the above expression:

$$\frac{1}{\lambda} = T_S + \frac{G_N(T_1, \cdots, T_{S-1})}{G_{N-1}(T_1, \cdots, T_S)}.$$

Since the second term is positive, this immediately gives us an *upper* bound on the mean throughput rate:

$$\lambda \leq \frac{1}{T_S}.$$

In words, node $S$ is the *bottleneck* node, in this sense.

We now rewrite our expression for the reciprocal of the mean throughput rate:

$$\frac{1}{\lambda} = T_S + \sum_{K=1}^{S-1} T_K F(T_1, \cdots, T_S)$$

$$F(T_1, \cdots, T_S) = \frac{G_N(T_1, \cdots, T_{S-1})}{G_{N-1}(T_1, \cdots, T_S) \sum_{K=1}^{S-1} T_K}.$$

We now manipulate this expression as we did above:

$$F(T^1, \cdots, T^S) \leq \frac{G^N(T^1 = 1, \cdots, T^{S-1} = 1)}{(S-1)G^{N-1}(T^1 = 1, \cdots, T^S = 1)}$$

$$= \frac{\binom{S+N-2}{N}}{(S-1)\binom{S+N-2}{N-1}} = \frac{\frac{(S+N-2)!}{N!(S-2)!}}{\frac{(S-1)(S+N-2)!}{(S-1)!(N-1)!}} = \frac{1}{N}.$$

Combining all this, we see

$$\frac{1}{\lambda} \leq T_S + \frac{1}{N} \sum_{K=1}^{S-1} T_K.$$

Rearranging, we see

$$\frac{N}{\sum_{K=1}^{S} T_K + (N-1)T_{\max}} \leq \lambda.$$

The first term in the denominator is the mean time for one job to make one complete cycle of the network:

$$T_{\text{cycle}} = \sum_{K=1}^{S} T_K.$$

The second term is the maximum mean time a job spends at any one node in the network:

$$T_{\max} = \max_{K=1,\cdots,S} T_K.$$

In summary, we see

$$\frac{N}{\sum_{K=1}^{S} T_K + (N-1)T_{\max}} \leq \lambda \leq \min\left[\frac{1}{T_{\max}}, \frac{N}{\sum_{K=1}^{S} T_K + (N-1)T_{\text{average}}}\right]$$

$$\frac{N}{T_{\text{cycle}} + (N-1)T_{\max}} \leq \lambda \leq \min\left[\frac{1}{T_{\max}}, \frac{N}{T_{\text{cycle}} + (N-1)T_{\text{average}}}\right].$$

If the *average* time per node spent in execution by one job during a cycle and the *maximum* time per node per job are roughly comparable to one another, these bounds will be quite close to one another.

## CONTRIBUTORS TO THIS ISSUE

**E. Arthurs,** Ph.D., 1957 (Electrical Engineering), Massachusetts Institute of Technology; M.I.T., 1957–1962; Bell Laboratories, 1962—. From 1957 to 1962 Mr. Arthurs was on the faculty of the M.I.T. Electrical Engineering Department. Since joining Bell Laboratories in 1962, he has worked on a variety of computer and communication network problems.

**Václav E. Beneš,** A.B., 1950, Harvard College; M.A. and Ph.D., 1953, Princeton University; Bell Laboratories, 1953—. Mr. Beneš has pursued mathematical research in traffic theory, stochastic processes, frequency modulation, combinatorics, servomechanisms, stochastic control, and filtering. In 1959–60 he was visiting lecturer in mathematics at Dartmouth College. In 1971 he taught stochastic processes at SUNY Buffalo, and in 1971–72 he was Visiting MacKay Lecturer in electrical engineering at the University of California in Berkeley. He is the author of *General Stochastic Processes in the Theory of Queues* (Addison-Wesley, 1963), and of *Mathematical Theory of Connecting Networks and Telephone Traffic* (Academic Press, 1965). Member, American Mathematical Society, Association for Symbolic Logic, Institute of Mathematical Statistics, SIAM, Mathematical Association of America, IEEE, Phi Beta Kappa.

**Ronald Caruso,** B.S., 1956, Rutgers University; M.S., 1964, Stevens Institute of Technology; Bell Laboratories, 1968—. Mr. Caruso is engaged in materials characterization of semiconductor crystals and epitaxial iron garnet films. Member, American Chemical Society, Phi Beta Kappa, Pi Mu Epsilon, Phi Lambda Upsilon.

**Gerard J. Foschini,** B.S.E.E., 1961, Newark College of Engineering, Newark, NJ; M.E.E., 1963, New York University, New York; Ph.D., 1967 (Mathematics), Steven Institute of Technology, Hoboken, NJ; Bell Laboratories, 1961—. Mr. Foschini has been with Bell Laboratories, Holmdel, NJ, since 1961. He initially worked on real-time program design. For many years he worked in the area of communication theory. In the spring of 1979 he taught at Princeton University. Mr. Foschini has supervised planning the architecture of data communications networks. Currently, he is involved with digital radio research. Member, Sigma Xi, Mathematical Association of America, IEEE, New York Academy of Sciences.

**Craig A. Gaw,** B.S. (with Distinction), 1970, M.S., 1974, Ph.D., 1979 (Electrical Engineering), Northwestern University; Argonne National Laboratory, 1967–1970; Bell Laboratories, 1978—. At Bell Laboratories, Mr. Gaw has been engaged in the characterization of GaAs double heterostructure injection laser material and devices. He is particularly interested in identifying parameters that affect device quality and reliability. Member, Tau Beta Pi, Sigma Nu, Sigma Xi, IEEE, EMSA.

**Basil W. Hakki,** B.S.E.E., 1957, M.S., 1958, Ph.D., 1960 (Electrical Engineering), University of Illinois; Bell Laboratories, 1963—. After joining Bell Laboratories, Mr. Hakki was involved in the design and analysis of GaAs two-valley microwave oscillators. He then worked with III-V light-emitting diodes and lasers. His work on GaAs double heterostructure injection lasers covered many aspects of the laser, including reliability, device design for cw and high-power operation, and device physics. He is currently involved in the study of quaternary lasers.

**Walter R. Holbrook,** B.S.M.E., 1969, Lafayette College; Bell Laboratories, 1959—. At Bell Laboratories Mr. Holbrook's work included the design of light-emitting diodes. Currently, he is a member of the Laser Development Department.

**Andrew S. Jordan,** B.S. (Metallurgy), 1959, Pennsylvania State University; Ph.D. (Metallurgy), 1965, University of Pennsylvania; Bell Laboratories, 1965—. Mr. Jordan has worked mainly in the area of compound semiconductors. He had been involved in the growth, phase equilibria, and impurity incorporation of ZnTe, CdTe, GaP, and GaAs. More recently, he has studied the degradation and reliability of GaP LEDs. Currently, he is engaged in modeling GaAs crystal growth. Member, Electrochemical Society.

**Chinlon Lin,** B.S.E.E., 1967, National Taiwan University; M.S., 1970, University of Illinois; Ph.D., 1973, University of California at Berkeley; Bell Laboratories, 1974—. Mr. Lin has worked on tunable dye lasers, short-pulse generation, nonlinear optics in fibers for frequency conversion, single-mode fiber dispersion and bandwidth studies, picosecond-injection-laser pulse generation and high-speed optoelectronics. At Berkeley, he received an IBM Fellowship and a Lankersheim Scholarship. He is currently in the Physical Optics and Electronics Research Department. Mr. Lin received an IEE (London) Electronics Letters Premium Paper Award in 1980 for a paper on zero-

dispersion-wavelength tailoring in single-mode fibers. Senior Member, IEEE, Topical Advisor for Fiber and Integrated Optics for the Optical Society of America.

**Pao-Lo Liu,** B.S., 1973 (Physics), National Taiwan University; M.S., 1976, Ph.D., 1979 (Applied Physics), Harvard University; Bell Laboratories, 1979—. At Bell Laboratories, Mr. Liu has worked on picosecond pulse generation, high-speed integrated optical modulators. He is currently a member of the Coherent Optics Research Department.

**Dan L. Philen,** B.S., 1968 (Chemistry), Auburn University; Ph.D., 1975 (Physical Chemistry), Texas A&M University; Georgia Institute of Technology, 1976–1979; Bell Laboratories, 1979—. Since joining Bell Laboratories Mr. Philen has been engaged in exploratory measurements on optical fiber properties. Member, American Chemical Society, Optical Society of America, Sigma Xi, Sigma Pi Sigma, Phi Lambda Upsilon.

**Jack Salz,** B.S.E.E., 1955, M.S.E., 1956, and Ph.D., 1961, University of Florida; Bell Laboratories, 1961—. Mr. Salz first worked on remote line concentrators for the electronic switching system. Since 1968 he has supervised a group engaged in theoretical studies in data communications and is currently a member of the Communications Methods Research Department. During the academic year 1967–68, he was on leave as Professor of Electrical Engineering at the University of Florida. In Spring 1981, he was a visiting lecturer at Stanford University. Member, Sigma Xi.

**B. W. Stuck,** S.B., S.M., Sc.D. (Electrical Engineering), Massachusetts Institute of Technology in 1969, 1969, and 1972, respectively; Bell Laboratories, 1972—. Since joining Bell Laboratories in 1972, Mr. Stuck has worked on a variety of digital communication and computer systems. Member, MAA, SIAM, IMS, ORSA, IEEE.

**Akira Tomita,** B.S., 1974, M.S., 1978 (Applied Physics), Hokkaido University, Sapporo, Japan; M.S., 1979, Ph.D., 1980 (Optics), University of Arizona; Bell Laboratories, 1980—. Mr. Tomita has worked in the field of nonlinear optical phase conjugation. He is currently working on lightwave telecommunication systems.

**A. R. Tynes,** B.S., 1950 (Engineering Physics), Montana State University; M.S., 1953 (Physics), Ph.D., 1963 (Physics), Bell Labora-

tories, 1961—. Presently Mr. Tynes is a member of the Undersea Systems Laboratory.

**Allyn R. Von Neida,** B.S. (E.E.), B.S. (Metallurgy), 1954, Lehigh University; Ph.D., 1960, Yale University; Bell Laboratories, 1961—. Mr. Von Neida has worked on materials for magnetic memories and is now engaged in crystal growth and characterization of semiconductors. Member, AIME, American Physical Society, Sigma Xi.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Brainard R. C., Scattaglia J. V., **Programmable Test-Bed for Composite Television Coding.** Smpte J 91(10):906–911, 1982.

Cleveland W. S., Devlin S. J., **Calendar Effects in Monthly Time-Series—Modelings and Adjustment.** J Am Stat A 77(379):520–528, 1982.

Cleveland W. S., Harris C. S., McGill R., **Judgments of Circle Sizes on Statistical Maps.** J Am Stat A 77(379):541–547, 1982.

Daubechies I., Klauder J. R., **Constructing Measures for Path-Integrals.** J Math Phys 23(10):1805–1822, 1982.

Nikolakopoulou G. A., Edelson D., Schryer N. L., **Modeling Chemically Reacting Flow Systems. 2. An Adaptive Spatial Mesh Technique for Problems With Discontinuities and Steep Fronts.** Comput Chem 6(3):93–99, 1982.

Willie J. S., **Covariation of a Time-Series and a Point Process.** J Appl Prob 19(3):609–618, 1982.

Willie J. S., **Measuring the Association of a Time-Series and a Point Process.** J Appl Prob 19(3):597–608, 1982.

Worrall B. M., Hall M. A., **The Analysis of an Inventory Control Model Using Polynomial Geometric-Programming.** Int J Prod 20(5):657–667, 1982.

## ENGINEERING

Alferness R. C., **Wave-Guide Electrooptic Modulators.** IEEE Micr T 30(8):1121–1137, 1982.

Antler M., **Field Studies of Contact Materials—Contact Resistance Behavior of Some Base and Noble-Metals.** IEEE Compon 5(3):301–307, 1982.

Chen C. Y., Bethea C. G., Cho A. Y., Garbinski P. A., **Temporal Resolution of an $Al_xGa_{1-x}$ As/GaAs Bias-Free Photodetector.** Electr Lett 18(20):890–891, 1982.

Feuer A., Barmish B. R., **Instability of Optimal Aim Control—Reply (Letter).** IEEE Auto C 27(5):1140, 1982.

Howard R. E., Jackel L. D., Swartz R. G., Grabbe P., Archer V. D., Epworth R. W., Hu E. L., Tennant D. M., Voshchenkov A. M., **Buried Channel MOSFETs With Gate Lengths From 2.5-μm to 700-A.** Elec Dev L 3(10):322–324, 1982.

Hwang J. C. M., Flahive P. G., Wemple S. H., **Performance of Power FETs Fabricated on MBE-Grown GaAs-Layers.** Elec Dev L 3(10):320–321, 1982.

Lee T. P., Burrus C. A., Liu P. L., Dentai A. G., **High-Efficiency Short-Cavity InGaAsP Laser With One High-Reflectivity Mirror.** Electr Lett 18(19):805–806, 1982.

Lemons R. A., Bosch M. A., **Electrically Amplified Optical-Recording.** Elec Dev L 3(9):254–255, 1982.

Lin C., Tomita A., Tynes A. R., Glodis P. F., Philen D. L., **Picosecond Dispersionless Transmission of InGaAsP Injection-Laser Pulses at the Minimum Chromatic Dispersion Wavelength in a 27-km-Long Single-Mode Fiber.** Electr Lett 18(20):882–884, 1982.

Logan R. A., Vanderziel J. P., Temkin H., Henry C. H., **InGaAsP/InP (1.3-μm) Buried-Crescent Lasers With Separate Optical Confinement.** Electr Lett 18(20):895–896, 1982.

Luryi S., Kazarinov R. F., **On the Theory of the Thermionic Emission Transistor. 2. TET as an Element of Logic-Circuits.** Sol St Elec 25(9):933–942, 1982.

Luryi S., Kazarinov R. F., **Optimum Baritt Structure.** Sol St Elec 25(9):943–945, 1982.

Martin R. D., Thomson D. J., **Robust-Resistant Spectrum Estimation.** P IEEE 70(9):1097–1115, 1982.

Minford W. J., **Accelerated Life Testing and Reliability of High k-Multilayer Ceramic Capacitors.** IEEE Compon 5(3):297–300, 1982.

Schiavone J. A., **Microwave Radio Meterology—Diurnal Fading Distributions.** Radio Sci 17(5):1301–1312, 1982.

Thomson D. J., **Spectrum Estimation and Harmonic-Analysis (Review or Bibliog.).** P IEEE 70(9):1055–1096, 1982.

Troxel D. E., Schreiber W. F., Burzinski N. J., Matson M. D., **Interactive Enhancement of Tone Scale.** Opt Eng 21(5):841–846, 1982.

Tsang W. T., Logan R. A., Ditzenberger J. A., **Ultralow Threshold, Graded-Index Wave-Guide, Separate Confinement, CW Buried-Heterostructure Lasers.** Electr Lett 18(19):845–847, 1982.

Wong D. Y., Juang B. H., Gray A. H., **An 800 bit/s Vector Quantization LPC Vocoder.** IEEE Acoust 30(5):770–780, 1982.

Luss H., **Operations-Research and Capacity Expansion Problems—A Survey (Review or Bibliog.).** Operat Res 30(5):907–947, 1982.

## MANAGEMENT/ECONOMICS

Guasch J. L., Weiss A., **An Equilibrium-Analysis of Wage Productivity Gaps.** Rev Econ S 49(4):485–497, 1982.

Hakansson N. H., **Changes in the Financial Market—Welfare and Price Effects and the Basic Theorems of Value Conservation.** J Finance 37(4):977–1004, 1982.

## PHYSICAL SCIENCES

Ackerman J. R., Kohler B. E., Huppert D., Rentzepis P. M., **Radiationless Decay of 1,3,5,7-Octatetraene.** J Chem Phys 77(8):3967–3973, 1982.

Alferness R. C., Buhl L. L., **High-Speed Wave-Guide Electrooptic Polarization Modulator.** Optics Lett 7(10):500–502, 1982.

Allara D. L., **Analysis of Surfaces and Thin-Films By IR, Raman, and Optical Spectroscopy (Review or Bibliog.).** Acs Symp S 1982(199):33–47, 1982.

Auston D. H., Smith P. R., **Picosecond Optical Electronic Sampling—Characterization of High-Speed Photodetectors.** Appl Phys L 41(7):599–601, 1982.

Bean J. C., Rozgonyi G. A., **Patterned Silicon Molecular-Beam Epitaxy With Sub-Micron Lateral Resolution.** Appl Phys L 41(8):752–755, 1982.

Bertz S. H., Dabbach G., **Factors Governing the Thermal-Stability of Organocopper Reagents—Two New Classes of Heterocuprates With Greatly Improved Thermal-Stability.** J Chem S CH1982(18):1030–1032, 1982.

Bhat R., Oconnor P., Temkin H., Dingle R., Keramidas V. G., **Acceptor Incorporation in High-Purity OMCVD Grown GaAs Using Trimethyl and Triethyl Gallium Sources.** Inst Phys C1982(63):101–106, 1982.

Bhat R., Keramidas V. G., **Comparative-Study of GaAs Grown by Organo-Metallic Chemical Vapor-Deposition (OMCVD) Using Trimethyl and Triethyl Gallium Sources.** P Soc Photo 323:104–109, 1982.

Chabal Y. J., Rowe J. E., Poate J. M., Franciose A., Weaver J. H., **Stoichiometry and Structural Disorder Effects on the Electronic-Structure of Ni and Pd Silicides.** Phys Rev B 26(6):2748–2758, 1982.

Chin A. K., Zipfel G. L., Mahajan S., Ermanis F., DiGiuseppe M. A., **Cathodoluminescence Evaluation of Dark Spot Defects in InP/InGaAsP Light-Emitting-Diodes.** Appl Phys L 41(6):555–557, 1982.

Cone R. L., Ender D. A., Otteson M. S., Fisher P. L., Friedman J. M., Guggenheim H. J., **Multiresonant 2-Photon-Absorption-Induced 4-Wave Mixing in Crystalline Rare-Earth Insulators.** AIP Conf PR1982(90):471–477, 1982.

Daniels J. M., Cladis P. E., Finn P. L., Powers L. S., Smith J. C., Filas R. W., Goodby J. W., Leslie T. M., **Polarization Absorption-Spectroscopy—Determination of the Direction and Degree of Orientation of Absorption Transitions.** J Appl Phys 53(9):6127–6136, 1982.

Dentai A. G., Burrus C. A., Lee T. P., Campbell J. C., Copeland J. A., Oliver J. D., **LPE InGaAs/InP With $N_D$-$N_A$ Greater-Than $5 \times 10^{14}$ $cm^{-3}$ for Photosensitive Devices.** Inst Phys C1982(63):467–471, 1982.

Donnelly V. M., Flamm D. L., Collins G., **Laser Diagnostics of Plasma-Etching—Measurement of $Cl_2^+$ in a Chlorine Discharge.** J Vac Sci T 21(3):817–823, 1982.

Donnelly V. M., Karlicek R. F., **Development of Laser Diagnostic Probes for Chemical Vapor-Deposition of InP/InGaAsP Epitaxial Layers.** J Appl Phys 53(9):6399–6407, 1982.

Dudderar T. D., Gilbert J. A., **Fiberoptic Measurement of the Deformation Field on a Remote Surface Using Numerically Processed White-Light Speckle.** Appl Optics 21(19):3520–3527, 1982.

Dupuis R. D., Lynch R. T., Thurmond C. D., Bonner W. A., **Growth of InP by Metalorganic Chemical Vapor-Deposition (MOCVD).** P Soc Photo 323:131–136, 1982.

Eisenstein G., Vitello D., **Chemically Etched Conical Microlenses for Coupling Single-Mode Lasers into Single-Mode Fibers.** Appl Optics 21(19):3470–3474, 1982.

Fleury P. A., **(RS) Phase-Transitions, Critical Phenomena and Instabilities (Review or Bibliog.).** Usp Fiz Nau 138(1):129–145, 1982.

Fork R. L., **Physics of Optical Switching.** Phys Rev A 26(4):2049–2064, 1982.

Forrest S. R., Kaplan M. L., Schmidt P. H., Venkatesan T., Lovinger A. J., **Large Conductivity Changes in Ion-Beam Irradiated Organic Thin-Films.** Appl Phys L 41(8):708–710, 1982.

Gallagher P. K., Gyorgy E. M., Jones W. R., **An Evolved Gas-Analysis Study of the Reduction of Nickel-Oxide by Hydrogen.** J Therm Ana 23(1–2):185–192, 1982.

Gedanken A., Robin M. B., Kuebler N. A., **Non-Linear Photochemistry in Organic, Inorganic, and Organo-Metallic Systems.** J Phys Chem 86(21):4096–4107, 1982.

Gibbs H. M., Jewell J. L., Moloney J. V., Tarng S. S., Tai K., Watson E. A., Gossard A. C., McCall S. L., Passner A., Venkatesan T. N., **Switching of a GaAs Bistable Etalon—External Switching On and Off, Regenerative Pulsations, Transverse Effects, and Lasing.** P Soc Photo 321:67–74, 1982.

Gibson J. M., Bean J. C., Poate J. M., Tung R. T., **The Effects of Nucleation and Growth on Epitaxy In the $CoSi_2$/Si System.** Thin Sol Fi 93(1–2):99–108, 1982.

Ginsberg A. P., Lindsell W. E., Sprinkle C. R., West K. W., Cohen R. L., **Disulfur and Diselenium Complexes of Rhodium and Iridium.** Inorg Chem 21(10):3666–3681, 1982.

Harrison T. R., Johnson A. M., Tien P. K., Dayem A. H., **$NiSi_2$-Si Infrared Schottky Photodetectors Grown by Molecular-Beam Epitaxy.** Appl Phys L 41(8):734–736, 1982.

Heaven M. C., Clyne M. A. A., **Interpretation of the Spontaneous Predissociation of $Cl_2[B^3\pi(0\hat{u})]$.** J Chem S F2 78(P8):1339–1344, 1982.

Heaven M., Miller T. A., English J. H., Bondybey V. E., **Laser-Induced Fluorescence-Spectra of YAG-Laser Vaporized-$SE_2$.** Chem P Lett 91(4):251–257, 1982.

Heimann P. A., Murarka S. P., Sheng T. T., **Electrical-Conduction and Breakdown in Oxides of Polycrystalline Silicon and Their Correlation With Interface Texture.** J Appl Phys 53(9):6240–6245, 1982.

Huse D. A., **Tricriticality of Interacting Hard Squares—Some Exact Results.** Phys Rev L 49(16):1121–1124, 1982.

Jackel J. L., Rice C. E., Veselka J. J., **Proton-Exchange for High-Index Wave-Guides in $LiNbO_3$.** Appl Phys L 41(7):607–608, 1982.

Jackel J. L., Rice C. E., **Variation in Wave-Guides Fabricated by Immersion of $LiNbO_3$ in $AgNO_3$ and $TiNO_3$—The Role of Hydrogen.** Appl. Phys L 41(6):508–510, 1982.

Jayarama A., Batlogg B., Maines R. G., Bach H., **Effective Ionic Charge and Bulk Modulus Scaling in Rocksalt-Structured Rare-Earth Compounds.** Phys Rev B 26(6):3347–3351, 1982.

Jin B. J., Lin T. H., Chu C. W., Shen Y. S., **Magnetic Phase-Transitions in $CrBe_{13}$ Under High-Pressures.** Phys Rev B 26(7):3878–3881, 1982.

Karlicek R. F., Donnelly V. M., Johnston W. O., **Laser Spectroscopic Monitoring of a Hydride Transport Vapor-Phase Epitaxy (VPE) Reactor.** P Soc Photo 323:62–66, 1982.

Lax M. et al., **Non-Adiabatic Formulation for Radiationless Transitions Induced by Classical Lattice-Vibrations.** Phys Rev B 26(7):3547–3558, 1982.

Leung S. Y., Schumake N. E., **Slider Induced Convection in Horizontal Liquid-Phase Epitaxy (LPE) System.** P Soc Photo 323:156–163, 1982.

Levy N., **Analysis of an Epoxy Curing Reaction by Differential Scanning Calorimetry (Review or Bibliog.).** Acs Symp S 1982(197):313–327, 1982.

Liao P. F., Stern M. B., **Surface-Enhanced Raman-Scattering on Gold and Aluminum Particle Arrays.** Optics Lett 7(10):483–485, 1982.

Liminga R. et al., **Gamma-Lithium Iodate Structure at 515 K and the $\alpha$-$LiIO_3$ to $\gamma$-$LiIO_3$ to $\beta$-$LiIO_3$ Phase-Transitions.** J Chem Phys 77(8):4222–4226, 1982.

Liu P. L., **Bandwidth, Field Distribution, and Optimal Electrode Design for Wave-Guide Modulators.** J Appl Phys 53(10):6681–6686, 1982.

Lloyd J. R., Nakahara S., **Formation and Growth of Voids and or Gas-Bubbles in Thin-Films.** Thin Sol Fi 93(3–4):281–286, 1982.

Logan R. A., **Optical-Device Structures Grown by Liquid-Phase Epitaxy (LPE).** P Soc Photo 323:150–155, 1982.

Lourenco J. A., **The Effect of LPE Growth Atmosphere on Thermal-Decomposition of (100) S-InP Substrates.** J Cryst Gr 59(3):563–566, 1982.

Marra W. C. et al., **X-Ray-Diffraction Studies—Melting of Pb Monolayers on Cu(110) Surfaces.** Phys Rev L 49(16):1169–1172, 1982.

Mcilrath T. J., Freeman R. R., **Laser Techniques for Extreme Ultraviolet Spectroscopy (Editorial).** AIP Conf Pr 1982(90):1–8, 1982.

Miller D. A. B., Chemla D. S., Eilenberger D. J., Smith P. W., Gossard A. C., Tsang W. T., **Large Room-Temperature Optical Nonlinearity in $GaAs/Ga_{1-x}Al_xAs$ Multiple Quantum Well Structures.** Appl Phys L 41(8):679–681, 1982.

Mills A. P., **Surface-Analysis and Atomic Physics With Slow Positron Beams.** Science 218(4570):335–340, 1982.

Nakahara S., **Phenomenological Description of Thin-Film Inter-Diffusion.** P Soc Photo 346:39–46, 1982.

Orenstein J., Baker G. L., **Photogenerated GaP States in Polyacetylene.** Phys Rev L 49(14):1043–1046, 1982.

Oron M., Tamari N., Shirikman H., Burrus C. A., **Lasing Properties of InGaAsP Buried Heterojunction Lasers Grown on a Mesa Substrate.** Appl Phys L 41(7):609–611, 1982.

Paek U. C., Peterson G. E., Carnevale A., **Effects of Depressed Cladding on the Transmission Characteristics of Single-Mode Fibers With Graded-Index Profiles.** Appl Optics 21(19):3430–3436, 1982.

Pearsall T. P., Hermann C., **Direct Measurement of Alloy Disorder in Lattice-Matched $Ga_xIn_{1-x}AsP_{1-y}$ Alloys ($y \cong 2.2x$).** Inst Phys C1982(63):269–274, 1982.

Petroff P. M., Gossard A. C., Logan R. A., Wiegmann W., **Toward Quantum Well Wires—Fabrication and Optical-Properties.** Appl Phys L 41(7):635–638, 1982.

Pfeiffer L., Walstedt R. E., Bell R. F., Kovacs T., **Temperature-Dependence of the Orthorhombic Charge-Density-Wave Parameters in 2H-$TaSe_2$ by Se-77 Nuclear Magnetic-Resonance.** Phys Rev L 49(16):1162–1165, 1982.

Phillips T. G., Dolan G. J., **SiS Mixers.** Physica B&C 110(1–3):2010–2019, 1982.

Portal J. C., Nicholas R. J., Brummell M. A., Cho A. Y., Cheng K. Y., Pearsall T. P., **Quantum Transport in GaInAs-AlInAs Heterojunctions, and the Influence of Intersubband Scattering.** Sol St Comm 43(12):907–911, 1982.

Powers L., **X-Ray Absorption-Spectroscopy Application to Biological Molecules (Review or Bibliog.).** Bioc Biop A 683(1):1–38, 1982.

Presby H. M., **Geometric Measurement of Preform Rods and Starting Tubes.** Appl Optics 21(19):3528–3530, 1982.

Revesz A. G., Wemple S. H., **The Optical-Properties of Noncrystalline Silicon and $Si_{1-x}H_x$ Films.** Phys St S-A 72(2):721–729, 1982.

Reynolds A. H., Straub K. D., Rentzepis P. M., **Picosecond Spectroscopy of Cu(II) Cytochrome-C.** Biophys J 40(1):27–31, 1982.

Reynolds C. L., Tamargo M. C., Gaw C. A., **Influence of Cooling Rate on the Short-Time LPE Growth of the Active Layer in (Al,Ga)As DH Lasers.** J Cryst Gr 59(3):525–530, 1982.

Rogovin D., Nagel J., **Quantum-Theory of the DC Josephson Effect—Static Tunneling Characteristics of Ultra-Small Josephson-Junctions.** Phys Rev B 26(7):3698–3732, 1982.

Rosenbaum T. F., Rupp L. W., Thomas G. A., Chen H. S., Banavar J. R., Varma C. M., **Observation of Electron-Spin-Resonance Nonlinearities in a Spin-Glass (Letter).** J Phys C 15(27):L975–L979, 1982.

Schneemeyer L. F., Miller B., **InP and CdS Photo-Anodes in Concentrated Aqueous Iodide Electrolytes.** J Elchem So 129(9):1977–1981, 1982.

Schwartz G. P., Sunder W. A., Griffiths J. E., Gualtieri G. J., **Condensed Phase-Diagram for the In-As-O System.** Thin Sol Fi 94(3):205–212, 1982.

Schwartz G. P., Griffiths J. E., Gualtieri G. J., **Thermal-Oxidation and Native Oxide Substrate Reactions on InAs and InXGa$_1$-XAs.** Thin Sol FI 94(3):213–222, 1982.

Shay J. L., Schiavone L. M., Epworth R. W., Taylor D. W., **Performance of a 7-Segment Iridium Oxide Electrochromic Display.** J Appl Phys 53(9):6004–6006, 1982.

Silfvast W. T., Wood O. R., **Recombination Lasers in the Vacuum Ultraviolet.** AIP Conf PR1982(90):128–136, 1982.

Skocpol W. J., Jackel L. D., Hu E. L., Howard R. E., Fetter L. A., **One-Dimensional Localization and Interaction Effects in Narrow (0.1 $\mu$m) Silicon Inversion-Layers.** Phys Rev L 49(13):951–955, 1982.

Skocpol W. J., Voshchenkov A. M., Howard R. E., Hu E. L., Jackel L. D., Epworth R. W., Fetter L. A., Grabbe P., Tennant D. M., **Preliminary Observation of 1-D Effects in Narrow Si MOSFET Structures.** Physica B&C 110(1–3):2105–2107, 1982.

Stolen R. H., Botineau J., Ashkin A., **Intensity Discrimination of Optical Pulses With Birefringent Fibers.** Optics Lett 7(10):512–514, 1982.

Temkin H., Joyce W. B., Chin A. K., DiGiuseppe M. A., Ermanis F., **Effect of p-n-Junction Position on the Performance of InGaAsP Light-Emitting Diodes.** Appl Phys L 41(8):745–747, 1982.

Tenan M. A., Hackwood S., Beni G., **Friction in Capillary Systems.** J Appl Phys 53(10):6687–6692, 1982.

Unguris J., Seiler A., Celotta R. J., Pierce D. T., Johnson P. D., Smith N. V., **Spin-Polarized Inverse Photo-Electron Spectroscopy of Solid-Surfaces—Ni(110).** Phys Rev L 49(14):1047–1050, 1982.

Vanderzi J. P., Tsang W. T., **Integrated Multilayer GaAs-Lasers Separated by Tunnel-Junctions.** Appl Phys L 41(6):499–501, 1982.

Vandover R. B., Bacon D. D., Sinclair W. R., **Superconductive Tunneling Into NbN Deposited Near Room-Temperature.** Appl Phys L 41(8):764–766, 1982.

Wagner W. R., Cho A. Y., **$Al_{0.3}Ga_{0.7}P_{0.01}As_{0.99}$ GaAs-Laser Heterostructures Grown by Molecular-Beam Epitaxy.** J Appl Phys 53(9):6032–6036, 1982.

Wang T. T., West J. E., **Polarization of Poly(Vinylidene Fluoride) by Application of Breakdown Fields.** J Appl Phys 53(10):6552–6556, 1982.

Weber T. A., Stillinger F. H., **Dynamical Study of the $H_3O_2^+ + H_3O_2^-$ Neutralization Reaction Using the Polarization Model.** J Chem Phys 77(8):4150–4155, 1982.

White J. C., Henderson D., **Anti-Stokes Raman Lasers.** AIP Conf PR1982(90):117–127, 1982.

Woodruff D. P., Smith N. V., Johnson P. D., Royer W. A., **K-Resolved Inverse Photo-Electron Spectroscopy and its Application to Cu(001), Ni(001), and Ni(110).** Phys Rev B 26(6):2943–2955, 1982.

## SOCIAL AND LIFE SCIENCES

Chance B., Moore J., Powers L., Ching Y., **A Redox Equilibrator for the Preparation of Cytochrome-Oxidase of Mixed-Valence States and Intermediate Compounds for X-Ray Synchrotron Studies.** Analyt Bioc 124(2):239–247, 1982.

Chance B. et al., **Synchrotron X-Ray Studies of Biological Preparations at Low-Temperatures With Optical Monitoring of Sample Integrity.** Analyt Bioc 124(2):248–257, 1982.

Day M. C., Stone C. A., **Developmental and Individual-Differences in the Use of the Control-of-Variables Strategy.** J Educ Psyc 74(5):749–760, 1982.

Friedman J. M., Stepnoski R. A., Noble R. W., **Time-Resolved Resonance Raman Studies of Carp Hemoglobin.** FEBS Letter 146(2):278–282, 1982.

Krauskopf J., Williams D. R., Heeley D. W., **Cardinal Directions of Color Space.** Vision Res 22(9):1123–1131, 1982.

Weisstein N., Williams M. C., Harris G. S., **Depth, Connectedness, and Structural Relevance in the Object-Superiority Effect—Line Segments Are Harder to See in Flatter Patterns.** Perception 11(1):5–17, 1982.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 March 1983 Number 3, Part 1

Copyright © 1983 American Telephone and Telegraph Company, Printed in U.S.A.

# Measurements of Selective Near-In Sidelobe Reduction of a Pyramidal, Horn-Reflector Antenna

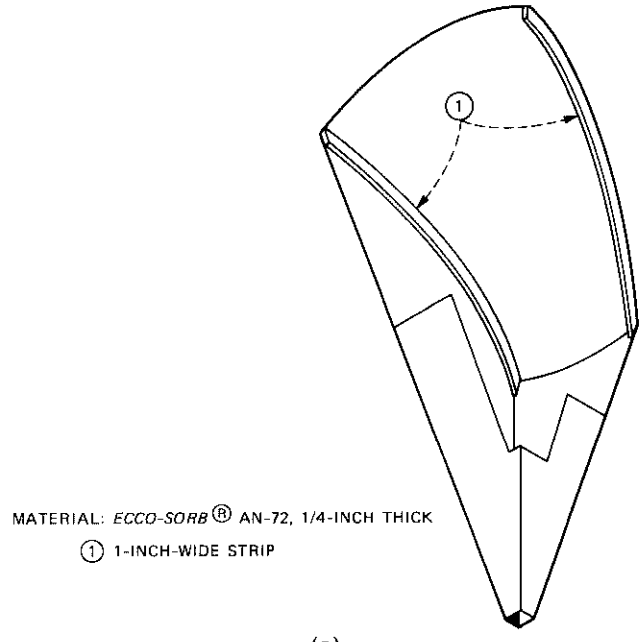
By R. A. SEMPLAK

(Manuscript received January 18, 1982)

*This paper describes measurements of a simple but effective means for reducing selective near-in sidelobe levels of a pyramidal horn-reflector antenna by using microwave absorber to modify the electric field across the reflector surface of the horn-reflector antenna. Examples of several modifications (made in the transverse plane for transverse polarization) are discussed and compared with data obtained before modification. Good reductions are achieved in the region of 2 to 6 degrees from the main beam. For example, an improvement was obtained on the order of 4 dB in the first sidelobe level with corresponding improvements of 8 and 10 dB in the angular region of the second and third sidelobe levels. One would expect some of the far-out sidelobe regions to increase; however, the reductions obtained in the levels of the near-in sidelobes could warrant a trade-off.*

## I. INTRODUCTION

It is well known that radio interference from adjacent paths limits the number of converging routes of a common carrier microwave radio system, and in recent years demands have been made to improve the sidelobe performance of the pyramidal, horn-reflector antenna. The use of blinders[1] (stepped extensions to the side walls of the antenna aperture) provide a degree of far sidelobe reduction, i.e., lobes beyond




(a)

Fig. 1a—A cutaway view of the pyramidal, horn-reflector antenna showing the placement of the 1-inch strips of microwave-absorbing material.

35 degrees from the axis of the main beam. Two methods now exist for dealing with the troublesome reflections from the flat weather cover of the horn-reflector antenna. One eliminates the reflections by using a focused weather cover.[2] The other uses a bottom-edge blinder to direct the reflections upward. An experimental investigation by C. A. Siller of Bell Laboratories showed that lining the sidewalls of the horn-reflector antenna with microwave absorber would reduce the sidelobe levels for angles greater than 40 degrees from the axis of the main beam, but this degraded the near-in sidelobes. A proposal by D. C. Hogg of Bell Laboratories for apodizing the horn-reflector antenna by applying a graded microwave absorber on the weather cover of the antenna might accomplish the desired sidelobe reduction. However, to achieve reduction in the sidelobe level by this technique one would have to suffer not only a reduction in antenna gain of at least 3 dB, but also an increase in the 3-dB beamwidth of the antenna. Such an increase in the beamwidth of the main beam could further degrade system performance. An analysis of available data on path interference, conducted by R. H. Turrin of Bell Laboratories, indicates that the most severe levels of interference occur at small angles (i.e., within ten degrees of the main beam), where the antenna discrimination is less than at large angles.

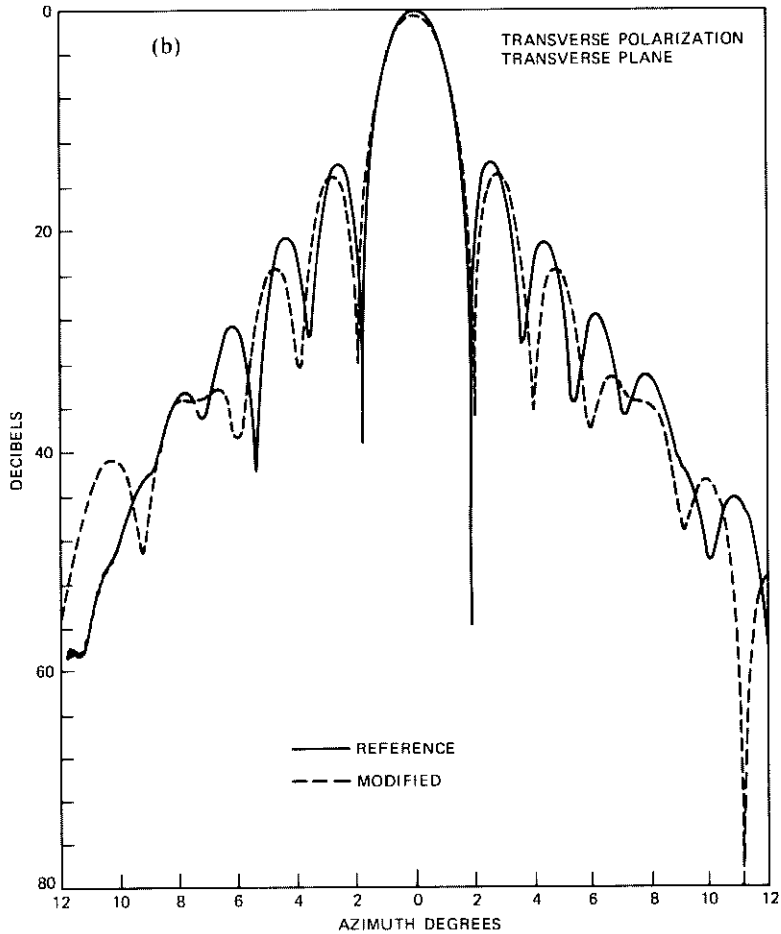



Fig. 1b—Comparison of the radiation patterns for transverse polarization of the modified (with absorber) antenna, the dashed curve, and the solid curve of the unmodified antenna. The gain loss owing to the absorber is on the order of 0.6 dB.

To continue the investigation for ways of reducing the near-in sidelobe levels of the horn-reflector antenna, a scale-model was built. The model has a numerically machined, precision reflector, and the scaling factor is 7.5, which means that measurements made at a frequency of 30 GHz will represent the performance of a full-size antenna measured at a frequency of 4 GHz. The discussion of the measurements made on the scaled model at a frequency of 30 GHz and the comparisons with data obtained by others on full-sized antennas are presented elsewhere.[3]

Inherent in the design of the pyramidal, horn-reflector antenna is a problem that results from illuminating the aperture with a dominant

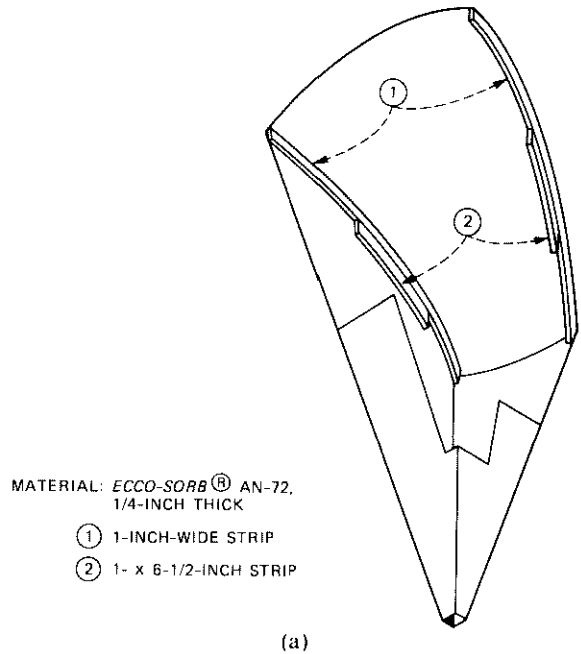



Fig. 2a—A cutaway view of the pyramidal, horn-reflector antenna showing an additional strip of absorbing material added to the sidewall of the horn.

waveguide mode.[4] The theoretically obtainable off-axis radiation levels in the transverse plane for transverse polarization are therefore considerably higher than one would like, i.e., they are essentially the equivalent of an aperture with constant illumination.

It should be remembered that longitudinal polarization and longitudinal plane of antenna rotation, for radiation measurements, are aligned with the pyramidal horn axis, i.e., vertically polarized in a terrestrial radio system. Transverse polarization and transverse plane indicate that the electric field and the plane of antenna rotation are perpendicular to the horn axis, i.e., horizontally polarized in a terrestrial radio system.

In view of the information obtained from Siller and Hogg, it appeared that a more fruitful approach to the reduction of the near-in sidelobes would be to take advantage of the reflection occurring at the surface of the paraboloidal section and attempt to modify the electric-field distribution across the surface of the reflector. This could be accomplished by introducing a microwave-absorbing material directly on the surface of the reflector.

In the discussion that follows, the microwave absorbing material* is

* *ECCOSORB* AN-72, a product of Emerson and Cuming, Inc.

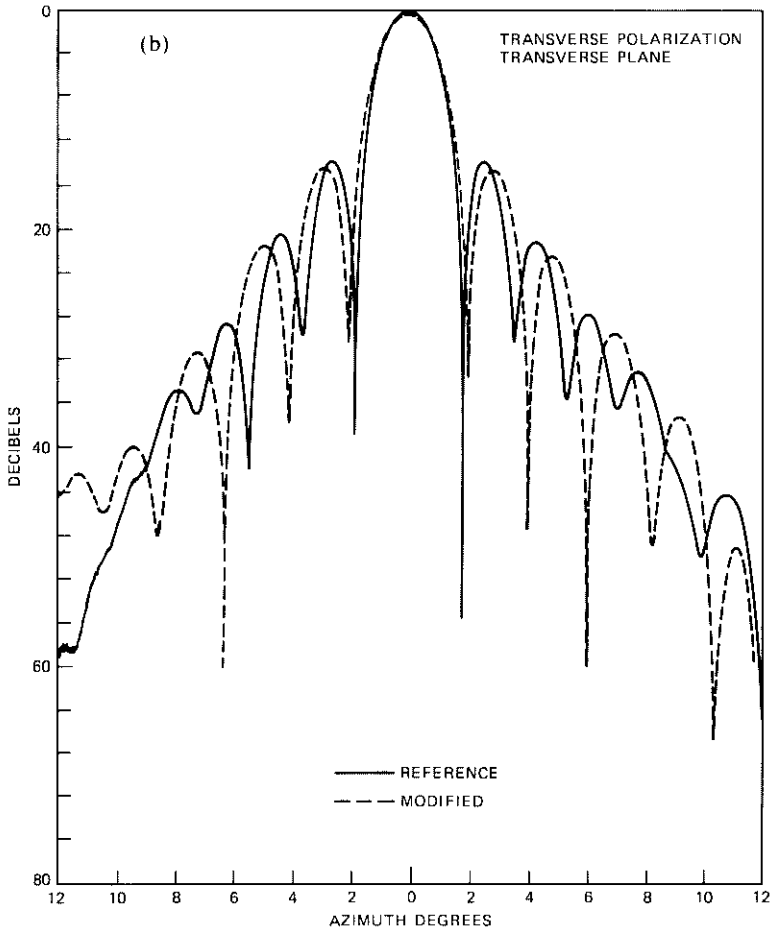


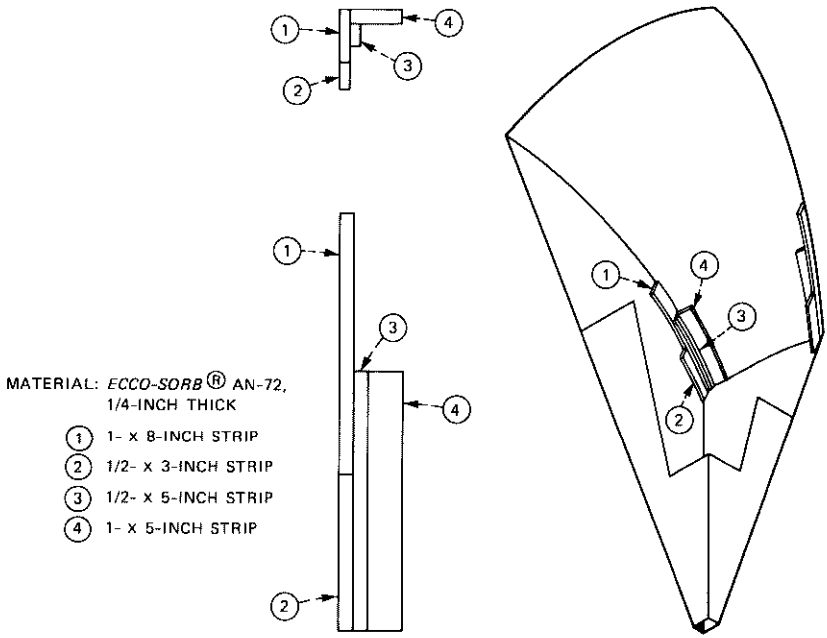

Fig. 2b—Comparison of the radiation patterns for transverse polarization of the modified (with absorber) antenna, the dashed curve, and the solid curve of the unmodified antenna.

attached to the surface of the reflector with a soluble floor-tile cement. The radiation patterns were made at a frequency of 30 GHz in the transverse plane for transverse polarization.

## II. DISCUSSION

From the many combinations (of absorbing-material configuration and placement on the surface of the reflector) that were measured, several were selected for discussion here.* For example in Fig. 1a, the

* The examples in the order cited serve to illustrate the evolution of these combinations.




(a)

Fig. 3a—A cutaway view of the pyramidal, horn-reflector antenna showing the placement of strips of microwave absorber in the rear corners on the reflector.

placement of the absorbing material on the reflector is shown in the cutaway view of the horn-reflector antenna. Here, the 1/4-inch-thick edge of the absorbing material is fastened to the reflector surface with the one-inch width of the material fastened to the sidewall of the horn. The absorbing material extends from the back wall to the edge of the aperture. The radiation pattern for this combination is shown by the dashed curve of Fig. 1b; the solid curve is the reference radiation pattern of the horn-reflector antenna, i.e., with nothing introduced into the antenna. In spite of the obvious lowering of the near-in sidelobes, which in this instance were 0.6, 3.0, and 6.0 dB for the first, second, and third sidelobes, respectively, there was no measurable increase in the 3-dB beamwidth. Although there is a gain reduction of 0.5 dB, this has been accounted for when stating the sidelobe reductions. One should note that the modified radiation pattern has very good symmetry and the angular displacements of the maxima for the sidelobes of the modified antenna have increased.

One might say that the observed results are typical of what would be expected when the aperture area is reduced. But consider Fig. 2a.

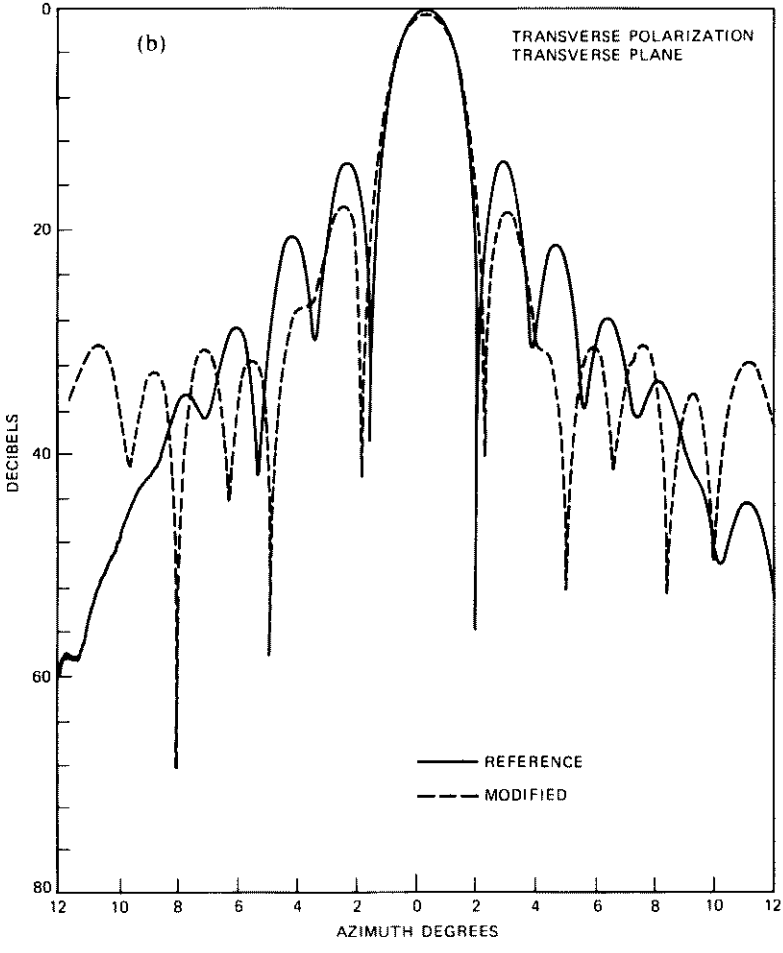



Fig. 3b—Comparison of the radiation patterns for transverse polarization of the modified (with absorber) antenna, the dashed curve, and the solid curve of the unmodified antenna. For this configuration the gain loss is 0.5 dB.

Here, an additional piece of absorber was placed on the sidewalls of the horn. No further reduction in the effective aperture area has taken place. However, the electric field distribution has again been modified. This is clearly seen by the dashed curve Fig. 2b, which shows a slight decrease in the near-in sidelobe levels of the modified antenna. The gain loss is essentially the same, but now there is a just discernible increase in the 3-dB beamwidth of the main beam. There is a further displacement on the sidelobe maxima for the modified antenna. Again the modified radiation pattern shows good symmetry. This configura-

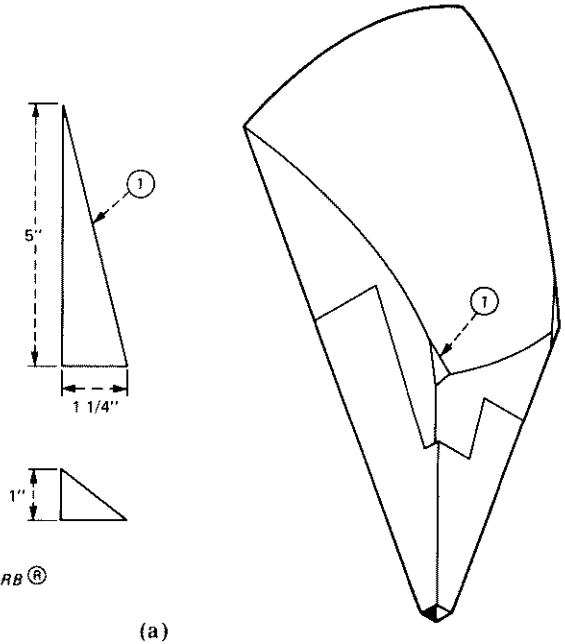



Fig. 4a—A cutaway view of the pyramidal, horn-reflector antenna showing the placement of a triangular wedge of absorbing material in the rear corners of the reflector.

tion could reduce specific interferences (i.e., at about 6 degrees), but the pattern envelope is slightly degraded.

It soon became apparent that one could achieve very good results by working in areas on the reflector that have the highest electric fields. The cutaway view of Fig. 3a shows the absorbing material confined to the rear corners of the reflector. The rather dramatic effects for this combination are shown by the dashed curve of Fig. 3b. Here, the first sidelobes are reduced by 4 dB, and the reduction out to about 5 degrees is remarkable. Again, the radiation pattern of the modified antenna shows good symmetry. The disconcerting feature of the radiation pattern for this configuration is the increase in the sidelobe levels beyond 8 degrees. In view of the number of edges that were introduced by the absorber, it seems that the increase in the sidelobe level beyond 8 degrees was due in part to scattering from these multiple edges.

An example, for another configuration, where an attempt was made to minimize the number of edges is shown in the cutaway view of Fig. 4a. Indeed, as one can see in Fig. 4b (when compared to Fig. 3b), there is an improvement in the region beyond 8 degrees. Here the loss in gain was of the order 0.2 dB and one can easily see the reduction in

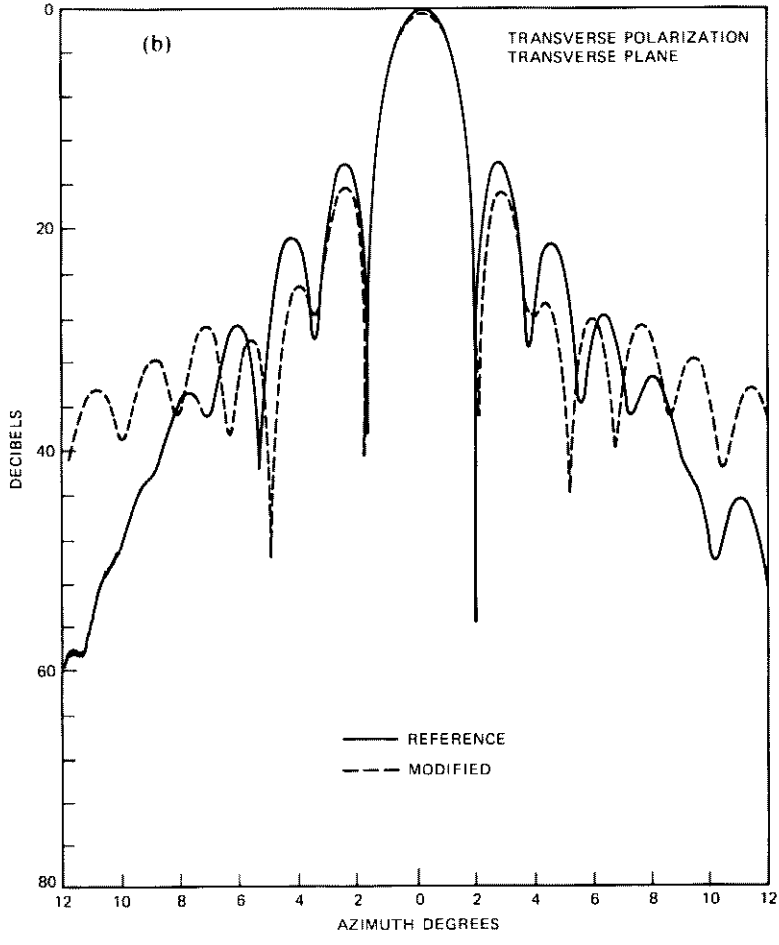
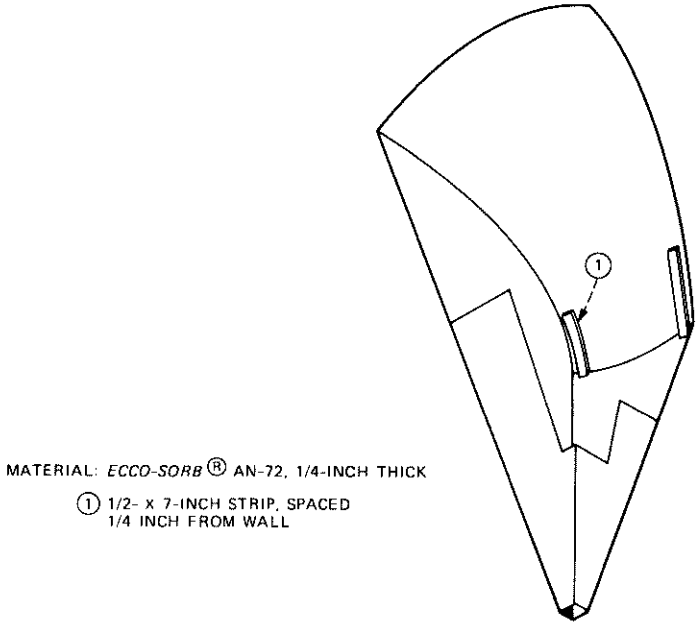



Fig. 4b—Comparison of the radiation patterns for transverse polarization of the modified (with absorber) antenna, the dashed curve, and the solid curve of the unmodified antenna. The gain loss is 0.3 dB.

the first and second sidelobe levels. Again, the symmetry of the pattern is preserved.

The last configuration to be discussed is shown by the cutaway view of Fig. 5a. In this configuration, a simple strip of absorbing material is placed on the reflector and spaced one-quarter inch from the sidewall. As shown by the dashed curve of Fig. 5b, the loss in gain is of the order 0.2 dB, and there is an appreciable reduction in the second and third sidelobe levels. Note the improvement obtained by this configuration in the region beyond 8 degrees when compared with the previously discussed modifications.




(a)

Fig. 5a—A cutaway view of the pyramidal, horn-reflector antenna showing the placement of a strip of absorbing material on the reflector surface.

Radiation patterns made for longitudinal polarization (in the transverse plane) indicate that the modifications introduced on the reflector were transparent for this polarization. Therefore, these patterns are not included here. Measurements in the longitudinal plane were not made, but one would expect to obtain similar performance. Further, the effects on far-out sidelobes were not evaluated.

## III. CONCLUSIONS

A simple but effective means for reducing selected near-in sidelobe levels has been developed and the measurements are presented and discussed. For example, if an interferer were located at the maxima of a third sidelobe, then by modifying the electric field distribution on the reflector one could reduce the amplitude at the angle of the third sidelobe at least 10 dB (as shown in Figs. 3b and 5b). Better results can be obtained for reduction in the angular region of the second sidelobe. Even the first sidelobe can be reduced by more than 4 dB. One would expect some of the far-out sidelobe regions to increase; however, the reductions obtained in the levels of the near-in sidelobes could warrant a trade-off.

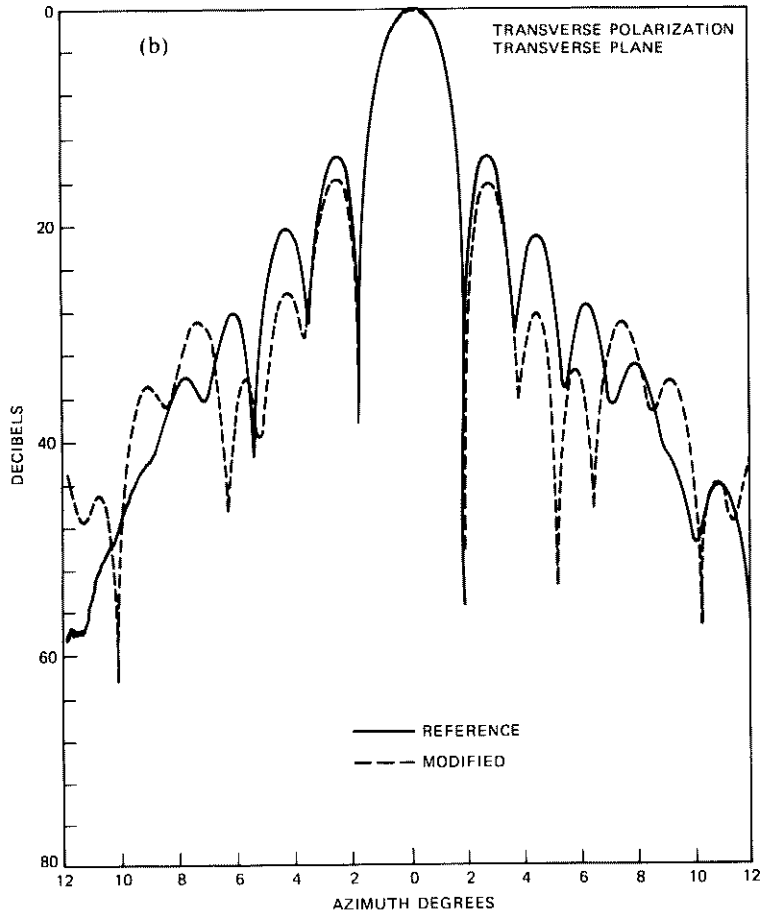



Fig. 5b—Comparison of the radiation patterns for transverse polarization of the modified (with absorber) antenna, the dashed curve, and the solid curve of the unmodified antenna. The gain loss is 0.3 dB.

## REFERENCES


1. D. T. Thomas, "Analysis and Design of Elementary Blinders for Large Horn Reflector Antennas," B.S.T.J., *50*, No. 8 (November 1971), pp. 2979–95.
2. R. A. Semplak, "Horn Reflector Antenna-Eliminating Weather Cover Reflections," B.S.T.J., *59*, No. 8 (October 1980), pp. 1333–42.
3. R. A. Semplak, "A 30-GHz Scale-Model, Pyramidal, Horn-Reflector Antenna," B.S.T.J., *58* (July–August 1979), pp. 1551–56.
4. A. B. Crawford, D. C. Hogg, and L. E. Hunt, "A Horn-Reflector Antenna for Space Communications," B.S.T.J., *40*, No. 4 (July 1961), pp. 1095–116.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# An Experimental Study of Atmospheric Optical Transmission

By B. G. KING, P. J. FITZGERALD, and H. A. STEIN

(Manuscript received August 5, 1982)

*This paper reports measurements made on a 23-mile, experimental, atmospheric, optical-transmission link for possible use as a standby substitute for microwave radio when the radio suffers severe multipath or obstruction fading. To allow comparison of transmission on a microwave and on an optical path, we used two parallel systems. One, a microwave system at 11 GHz, allowed frequency-selective fading to be measured, and the other, an optical system at 6328Å, allowed amplitude changes of the received optical signal to be obtained. The measured clear-air loss on the optical path is 27 dB. This measurement is made up of 17 dB of atmospheric scattering and 10 dB due to the receiving antennas intercepting only 10 percent of the beam at the receiver. The signal-to-noise ratio, calculated using measured background sky-noise and measured received power, is about 60 dB for a 100-MHz band. The beam diameter was measured to be 32 feet where the signal is down 20 dB. On the single occasion when frequency-selective microwave fading was observed, there was no fading of the optical signal. We find that it is necessary to control the transmitter elevation angle with a servo error signal from the receiver; the azimuth angle needs only occasional manual correction. The optical beam can be automatically reacquired after severe atmospheric attenuation, and that scintillation is usually several decibels, and occasionally as much as 10 dB.*

## I. INTRODUCTION

We report system parameters measured on a 23-mile, atmospheric, optical-transmission path. The object of the investigation is to determine whether a modulated optical signal transmitted through the atmosphere could be used as a stand-by substitute for a microwave radio link when transmission over the radio path is impaired by clear-air multipath or obstruction fading.

So far, we have learned that:

(*i*) The control of the transmitter elevation angle can be accomplished with an error signal from the receiver. This compensates for atmospheric refractive index gradient changes, which change the curvature of the optical beam.

(*ii*) There is very little need for horizontal beam correction.

(*iii*) The transmitter can be scanned up and down when severe path loss has attenuated the signal below the detection level so as to reestablish the link after the path loss is reduced.

(*iv*) The clear-air path loss is 27 dB. Of this, 17 dB is due to scattering and 10 dB is due to the receiver intercepting only 10 percent of the beam.

(*v*) The received beam is 20 feet in diameter where the received signal is 10 dB below its value at the beam center and 32 feet in diameter where the signal is down 20 dB.

(*vi*) The calculated clear-air peak-signal-to-average-noise ratio, including the background light, is between 57 and 63 dB, using a 20-mW laser.

It has been determined[1] that rain and fog can cause as much as 0.1-dB/ft optical-transmission loss, and, at such times, optical transmission on a 20- to 30-mile path is very difficult. However, it is believed that most microwave multipath fading occurs in relatively clear weather, and to date, our measurements have shown this to be true. Thus, when microwave clear-air fading occurs, the attenuation of the optical signal should be low.[2] Recently, Schiavone[3] has shown that 7 percent of the 30-dB clear-air microwave fade-time at Palmetto, Georgia, occurs jointly with visibilities of 4 miles or less. The remaining 93 percent occurs during times of better visibility.

Further, because of the very narrow beam that can be obtained with light,* there should be no multipath fading of the optical signal.* Again, our limited measurements to date have shown no multipath fading at optical frequencies while multipath fading was occurring at 11 GHz.

## II. LOCATION OF THE EXPERIMENT

Figure 1 shows the location of the experiment and also a vertical profile of the path. About three miles of the 23-mile path are over tidal water.

The transmitter is mounted in a cab at the top of a 100-foot steel tower at Murray Hill. The tower is shown in Fig. 2. The optical beam

* For a 1-foot antenna, at 6328Å, the ratio, $D/\lambda$, is $4.8 \times 10^5$, whereas for a 6-foot antenna at 4 GHz, $D/\lambda$ is $2.4 \times 10^1$. This results in a 3-dB width at the receiver of 0.25 feet for the optical beam and 4800 feet for the 4-GHz beam.

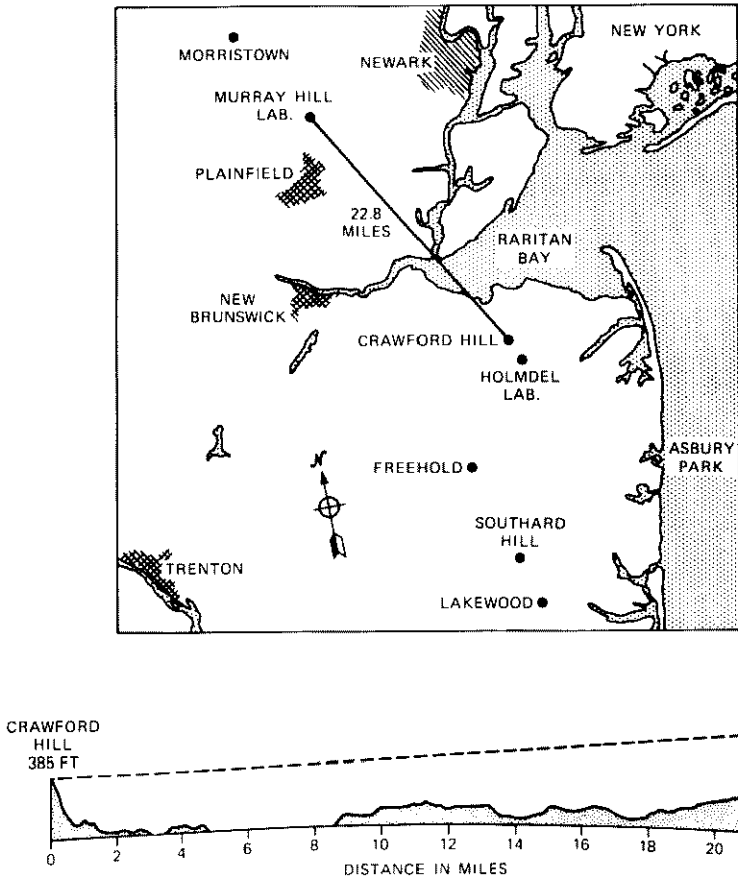



Fig. 1—Map and profile of the transmission path.

is transmitted through the cylinder protruding from the flat surface at the top of the tower. Two microwave antennas on the tower are used to measure multipath fading on the same path (11 and 30 GHz).

The optical receiver at Crawford Hill is shown in Fig. 3. The two microwave antennas on the left of the photograph are part of the microwave multipath-fading experiment.

### III. THE TRANSMITTER

Figures 4 and 5 show a diagram and a photograph of the transmitter, respectively. It is composed of a rigid but movable telescope framework, 10-feet long and 1-1/2-feet wide and high, mounted at one end on an orthogonal pair of gimbals and supported on a fulcrum at the other. The telescope gimbals are mounted on one end of a rigid *L*-shaped base, while the other end of the base provides a mount for a

Fig. 2—Transmitter tower at Murray Hill.

second fulcrum. The fulcrum of the telescope and that of the base are about three inches apart. A steel lever pivots on the stationary fulcrum of the base and supports the fulcrum of the telescope. The length of the lever is such that a one-mil (0.001-inch) movement at its free end ("A") causes the telescope to move 0.46 μrad, or a beam motion of about 6.4 feet at the receiver. The lever and fulcrum allow both vertical and horizontal adjustments of the telescope direction.

The position of the free end of the lever is controlled by two small motors. Two tones (900 and 1100 Hz), delivered over a pair of telephone lines from the receiver, control the vertical and horizontal position of the transmitter telescope. The response is relatively slow; a one-second burst of one of the tones moves the optical beam 13.5 inches at the receiver 23 miles away (9.3 μrad/s).

The laser power is 15 mW at 6328Å. The beam is modulated by a chopper wheel, which produces 450 pulses per second, with equal off and on time. To derive a synchronizing signal, light reflected from the chopper is detected and the 450-Hz tone is transmitted over a third

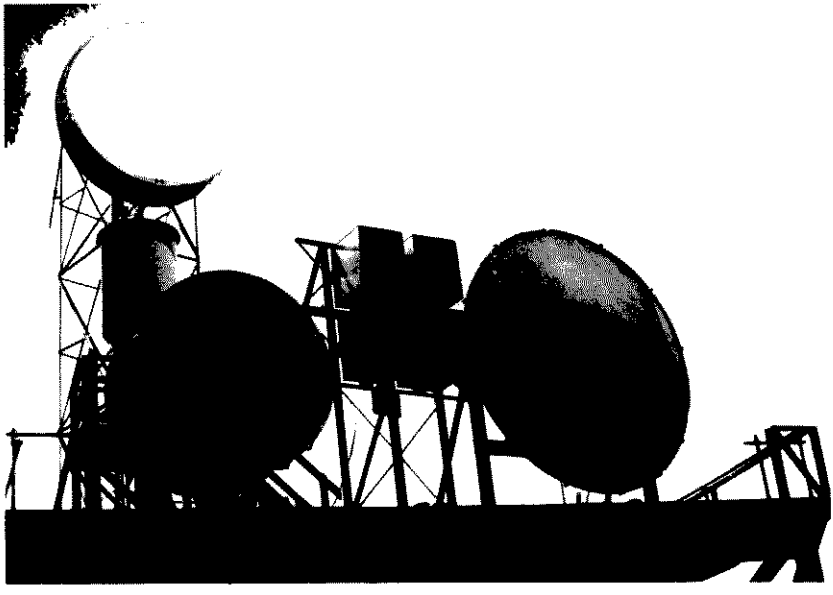


Fig. 3—Receiver assembly at Crawford Hill.

telephone line to the receiver. There, the tone is used to phase-lock the local oscillators of the homodyne detectors.

The rays of the optical signal at the transmitter are shown in Fig. 4. The focal point of the 12-inch mirror is made to fall, by the angle (45 degrees) and position of the small (2-inch diameter) plane mirror inside the telescope, just outside the top of the telescope. To allow sighting through this transmitting telescope for alignment on the receiver, the beam is collimated with a 10-diopter lens. A prism inserted just above this lens bends the beam outward, and allows precise alignment by eye. When used as a transmitter, the collimated beam is focused by a second 10-diopter lens at the focus of a 10-power microscope objective.

## IV. THE RECEIVER

The configuration of optical receiving antennas is shown in the photograph of Fig. 3 and the sketch of Fig. 6. Each element is composed of a 24- × 18-inch plastic Fresnel lens mounted in a weathertight, sheet-metal cone, with a photomultiplier at the focus. To reduce background noise, the optical signal passes through a 6328Å filter. This filter reduces the background current 22 dB and the signal 3 dB, for a net 19-dB improvement. In addition, the aperture (0.5 inch) of the photomultiplier is reduced to 0.1 inch with an iris. This reduces the background current 14 dB. Finally, to narrow the part of the sky seen by the photomultiplier, a rectangular bundle of thirty rectangular

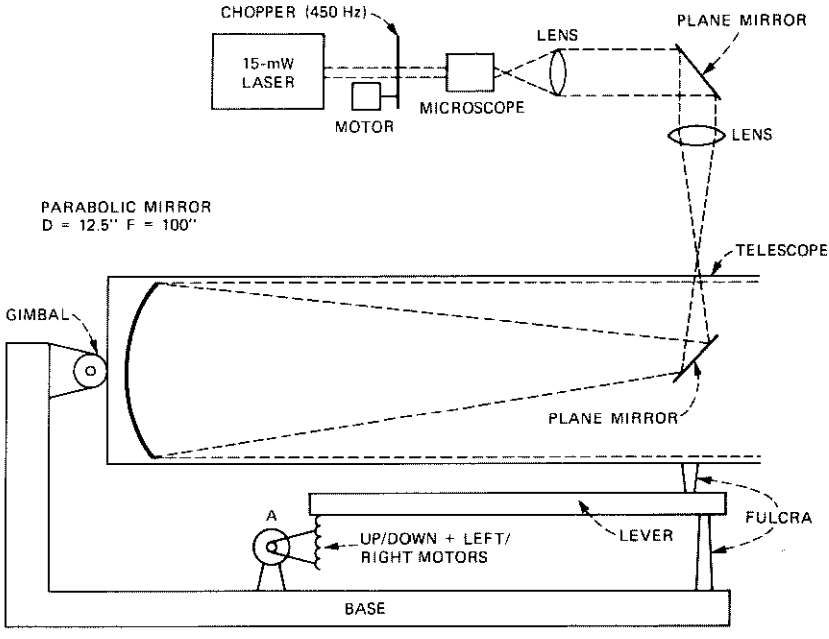



Fig. 4—A diagram of the transmitter.

(4 × 4 inches) tubes is mounted in front of the Fresnel lens. This reduces the background current 4 dB. There are four optical antennas in the receiver mount.

As we see in Fig. 7, the 450-Hz signals from the four photomultipliers are passed through an automatic gain control (AGC) step attenuator (0 to 40 dB in four steps of 10 dB). Each of the four signals is then amplified and band-limited in a homodyne detector. The output signals of the four homodyne detectors are indicated as *A*, *B*, *C*, and *D* in Fig. 7. These outputs are dc voltages linearly proportional to the power of the incoming 450-Hz optical signal. An input signal of 100 $\mu$V rms to each homodyne detector produces 10V dc at its output.

The four signals, *A*, *B*, *C*, and *D* in Fig. 7, are combined in a summing amplifier to give an indication of the optical signal strength and the integrated sum is recorded on a chart recorder. The sum is also used to control the AGC and to put the system into a search mode when the sum signal falls below a fixed threshold.

The four signals, *A*, *B*, *C*, and *D* in Fig. 7, are used to control the elevation angle of the transmitter. The signals *A* and *B* from the top two antennas are added, as are the signals *C* and *D* from the bottom. The sum *A* plus *B* is subtracted from the sum *C* plus *D* in a comparator. If the difference is positive, the lower half of the receiver is receiving a stronger signal, and if the difference exceeds a threshold, an 1100-Hz

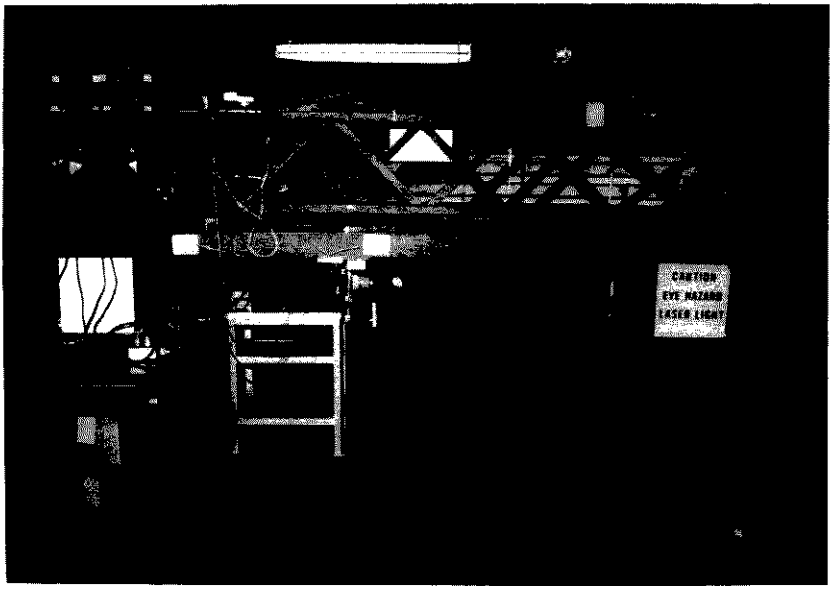



Fig. 5—Transmitter assembly.

tone is applied to the phone line. This tone causes the transmitter to elevate the beam-launch angle. A negative difference causes a 900-Hz tone, which lowers the beam-launch angle. In this way, the transmitter keeps the beam centered on the receiver.

The search control is designed to bring the optical beam back onto the receiver after a deep fade of the optical signal. The search control goes into action when:

(*i*) The sum signal falls below a threshold, and also

(*ii*) All of the attenuators are out of the AGC (zero AGC loss).

In the search mode, a sequence of 200 up commands (1100 Hz) (1 second on, 6 seconds off) is followed by a similar sequence of 200 down commands (900 Hz).* To assure that the search control is in sole command, when it is in operation the up/down servo is disabled. This vertical scanning continues until the sum signal exceeds a threshold, at which time control is handed off to the vertical servo controller. All but one of the thresholds are compared with the sum voltage ($A + B + C + D$). Figure 8 shows the thresholds and the action taken.

First, the summing amplifier saturates at 9.5 volts, which sets the upper limit. The attenuators are inserted, 10 dB at a time and with a

* Two hundred seconds of command takes the transmitter from top to bottom of its range of motion (1495 μrad at the transmitter or 182 feet vertical motion at the receiver).

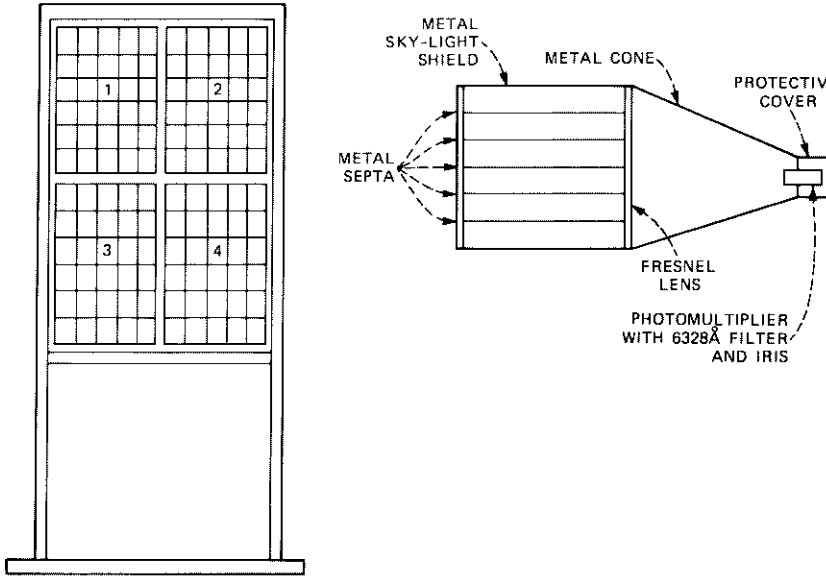



Fig. 6—A diagram of the receiver assembly.

10-second delay between insertions, when the sum voltage is 5 volts or more. The attenuators are taken out, 10 dB at a time and at 10-second intervals, when the sum voltage is 50 millivolts or less. The 100/1 threshold ratio is required to minimize attenuator switching owing to scintillation.

One of the thresholds in Fig. 8 is not compared with the sum voltage but rather with the difference $(A + B)-(C + D)$. This difference voltage is the up/down servo command and the minimum threshold is 20 mV. As the sum voltage increases, part of the sum is used to increase this threshold, which minimizes rapid hunting at high signal levels.

The last pair of thresholds turns the search control on and off. The search is turned on when the sum voltage is less than 200 mV and off when it exceeds 500 mV. These rather high thresholds are determined by the clear-air off-axis signal of 150 mV. It was found that, in very clear weather, when the transmitter was moved as far as it could be off the receiver, there was still a sum signal of 150 mV.

## V. THE BEAM

### *5.1 Received beam diameter*

The shape of the optical beam at the receiver was determined by incrementally scanning the transmitter and recording the received signal. The distance of motion of the lever in the transmitter ($A$ in Fig.

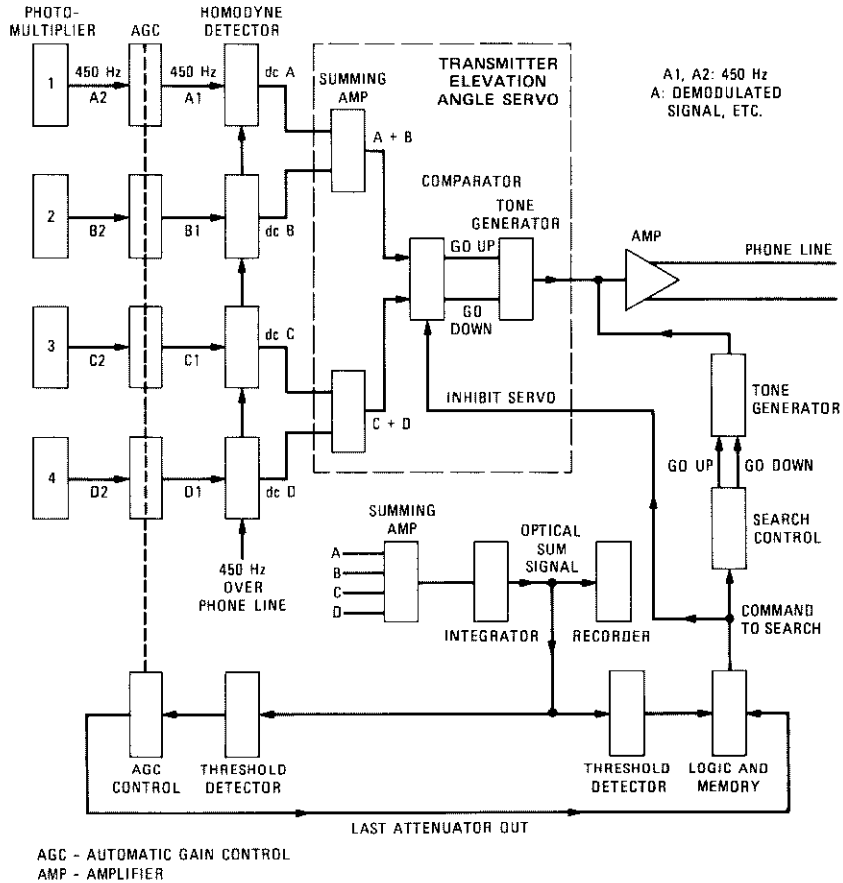



Fig. 7—An electrical diagram of the receiver.

4) was measured and converted to feet of optical-beam motion at the receiver (0.46 μradians of beam motion for each 0.001-inch motion at *A*). This is the abscissa in Fig. 9. The ordinate is simply the sum voltage converted to dB. At the 10-dB point, the beam is about 20 feet in diameter, and at 20 dB, the beam is about 30 feet in diameter.

### *5.2 Path loss*

The receiving antenna is about 3 × 4 feet, and the beam decreases only about 0.5 dB over this area. If all of the power at the receiver is assumed to lie within a 32-foot diameter circle (signal down 20 dB), and the power distribution over this area is calculated using the data of Fig. 9, the power into the 12-square-foot receiver is down about 10 dB below the total power. So, what might be considered the geometric efficiency, that is to say, the fraction of the total optical power in the beam at the receiver that is intercepted by the receiver, is 10 percent.

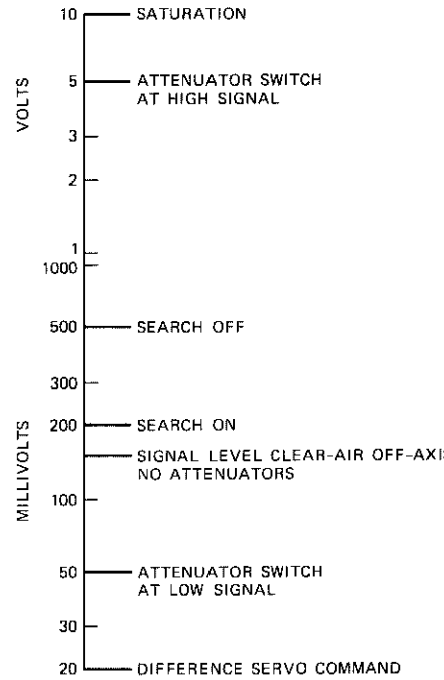



Fig. 8—Optical-system-control voltage levels.

However, there is another source of loss. Energy is scattered by the atmosphere. To determine the amount of this loss, the signal was measured at the end of the transmitter telescope, using a Fresnel lens and photomultiplier identical to those at the receiver. Twenty-seven dB of optical attenuation were inserted in the narrow collimated beam between the two 10-diopter lenses (at the transmitter) to bring the measured signal to the value observed at the receiver. Thus, we conclude that the scattering causes some 17 dB of path loss, and the total path loss, using four receiving antennas, is 27 dB. The *Handbook of Optics* indicates a 16-dB loss on a 23-mile path.[4]

### 5.3 Vertical beam motion

The transmitter is capable of 1495 μrad of vertical motion. This was consistent with the beam movement measured by Ochs and Lawrence,[5] who reported a maximum of 1023 μrad of vertical motion on a 28-mile path. The path from Murray Hill to Crawford Hill is 23 miles. This 1495 μrad of vertical motion has been found to be adequate to compensate for atmospheric changes on this path. Figure 10a shows a typical early-morning decrease of the transmitter elevation angle. The movement is 478 μrad at a 0.07-μrad/s rate. The largest early-morning

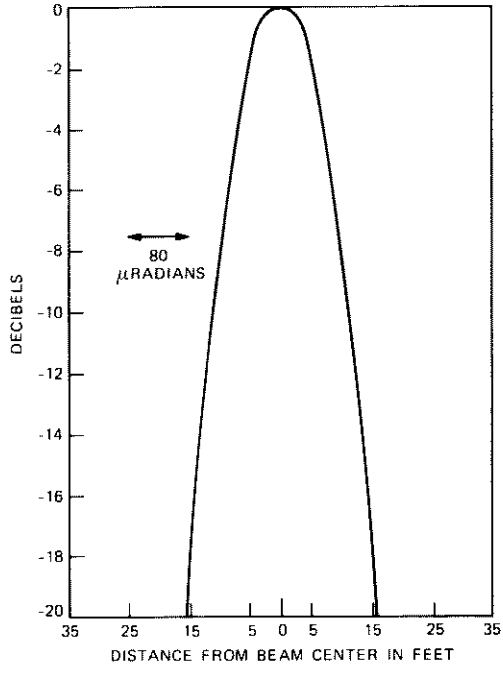



Fig. 9—Optical beam power vs distance from beam center.

movement recorded is shown in Fig. 10b. Here the transmitter moved 990 μrad at a rate of 0.18 μrad/s. The fastest transmitter movement observed was 0.89 μrad/s, well below the 9.3-μrad/s capability of the system. Ochs and Lawrence show a rate of 0.12 μrad/s on a 28-mile path.

The operation of the search control is shown in Fig. 10c. A heavy rainstorm brought the signal below the control threshold at about 16:20 and the transmitter began to search. It takes 39 minutes to complete an up/down scan. At 20:50 the signal was strong enough to turn off the search control, and the system returned to normal operation.

### *5.4 Attenuator operation*

Fog and rain can attenuate the optical signal severely. It was found that the up/down servo control was operable when the signal had faded 40 dB below its clear-air value. To keep the control voltages within a reasonable range, an AGC was used, and is shown in Fig. 7. When the sum-signal exceeds 5 volts a 10-dB attenuator is switched into the signal path, and when it falls below 50 mV a 10-dB attenuator is removed. Figure 11 shows a signal fade which, at 17:25, caused a 10-dB attenuator to be removed.

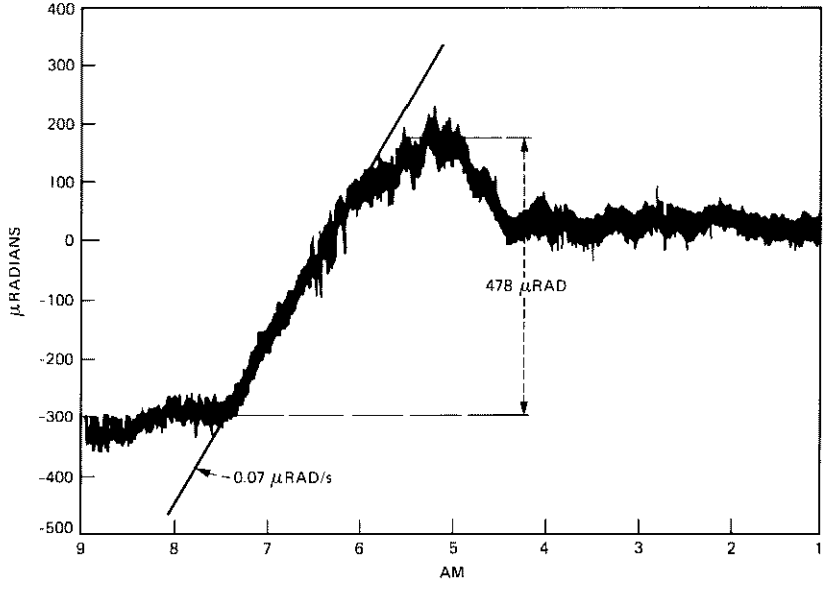



Fig. 10a—Transmitter elevation angle vs time (sunrise).

### 5.5 Beam microstructure

In addition to the effect of fog and rain, which causes the optical signal to be attenuated by about the same amount for all the receiving antennas, there is fine structure in the path, which causes small variations of the signal in each of the four antennas.[6] A measurement of this phenomenon is shown in Fig. 12a, which shows the output signal of the homodyne detectors with a 6-dB/octave, 0.1-second, time-constant output filter.

Small cells of heated and cooled air cause the velocity of the light to change slightly and differently in each ray. At the receiver, the signal is brighter where the preponderance of rays reaching the antenna are in phase, and dimmer where some of the rays to the antenna are out of phase with the remainder.

Ochs[6] reports that the delayed correlation between horizontal detectors is used to measure wind velocity. If this is done with the data of Fig. 12a, there is about a 0.1-second delay, left to right, in a distance of some 18 inches. This indicates a 10-mph east to west component of wind.

The photomultiplier in Antenna 1, which is seen in Figure 12a, was about 2 to 3 dB less sensitive than the other three. The signal from any one antenna shown in Fig. 12a varies as much as 5 dB in the 2.4-second record, and shows a maximum rate of change of about 18 dB/s. In Fig. 12b, the sums of the top $(A + B)$ and bottom $(C + D)$

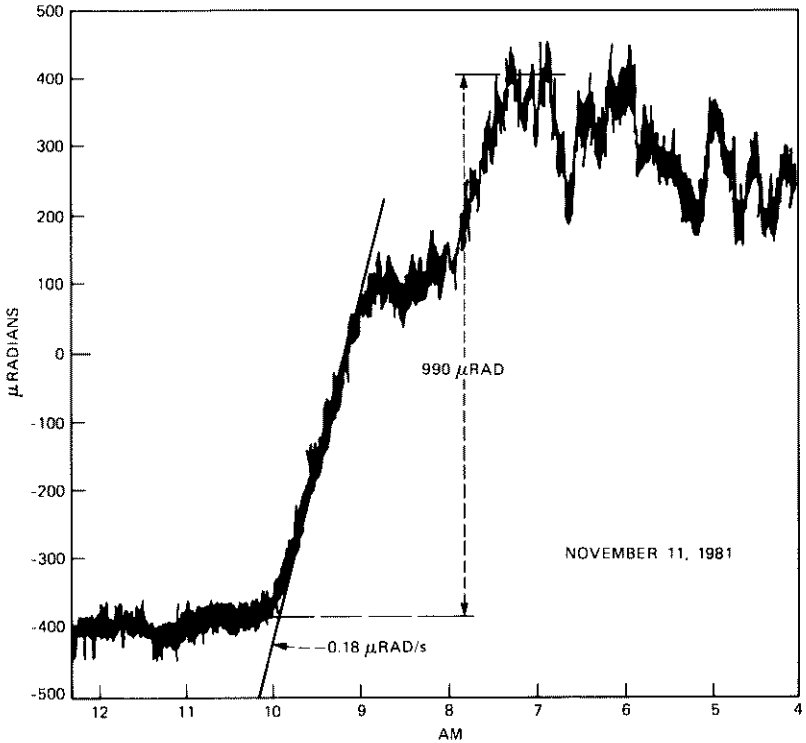



Fig. 10b—Transmitter elevation angle vs time (sunrise).

antennas are shown. At 0.5 second, the up/down servo would command the transmitter to go up, at 1.3 seconds to go down, and at 2 seconds to go up. It is to be observed that the two sums do not change as much (about 3 dB) nor as rapidly (about 11 dB/s) as the separate signals. Finally, the sum of all four signals is shown in Fig. 12c. The sum signal changes about 2 dB with a maximum rate of change of 7 dB/s. It is clear that increasing the size of the receiving antenna reduces the magnitude and rate of rapid signal fluctuations.

### *5.6 Scintillation*

The gross effect of beam microstructure on the sum signal is usually called scintillation. Scintillation is present in microwave radio signals, and is known to increase with frequency, all other things being equal. As was shown in Section 5.5, the optical scintillation decreased with increasing antenna area. Because of the large optical antennas the scintillation measured on the optical path is no worse than that measured on the parallel 30-GHz path, though larger than that on the parallel 11-GHz path. Hannan et al.[7] report optical scintillation as severe as 10 dB.

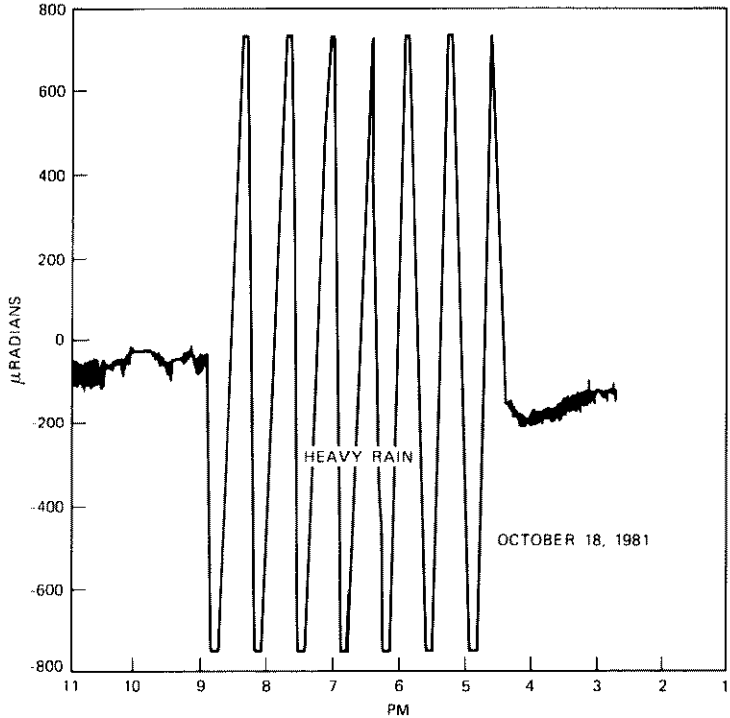



Fig. 10c—Transmitter elevation angle vs time (search mode).

Figure 13 shows the 450-Hz modulation signal, obtained directly from one of the photomultipliers on a clear, cool, sunny day. In Fig. 13a the maximum signal is about 500 mV (across 10 kilohms) and the minimum about 380 mV, a ratio of about a 1.2 dB. Of interest here is the structure seen at the maximum of the negative-going signal. It is clear that even with this large antenna the beam microstructure has components above one kilohertz. In Fig. 13b, taken at a ten times slower rate, the maximum signal is about 500 mV and the minimum about 180, a change of 4 dB. There are several instances where the rate of change is very rapid. For example, the signal drops from 450 mV to 300 mV in two pulses, or a change of 1.8 dB in 4.4 ms (or about 400 dB/s).

### *5.7 Optical signal during an 11-GHz multipath fade*

Figure 14a shows an 11-GHz, clear-air, multipath-fading and enhancement event. The fading is frequency selective with a maximum slope of about 2 dB over the 40-MHz band.

Figure 14b shows the same event along with the optical signal. The normal, clear-air signal level of the 11-GHz event was 2 volts. There is

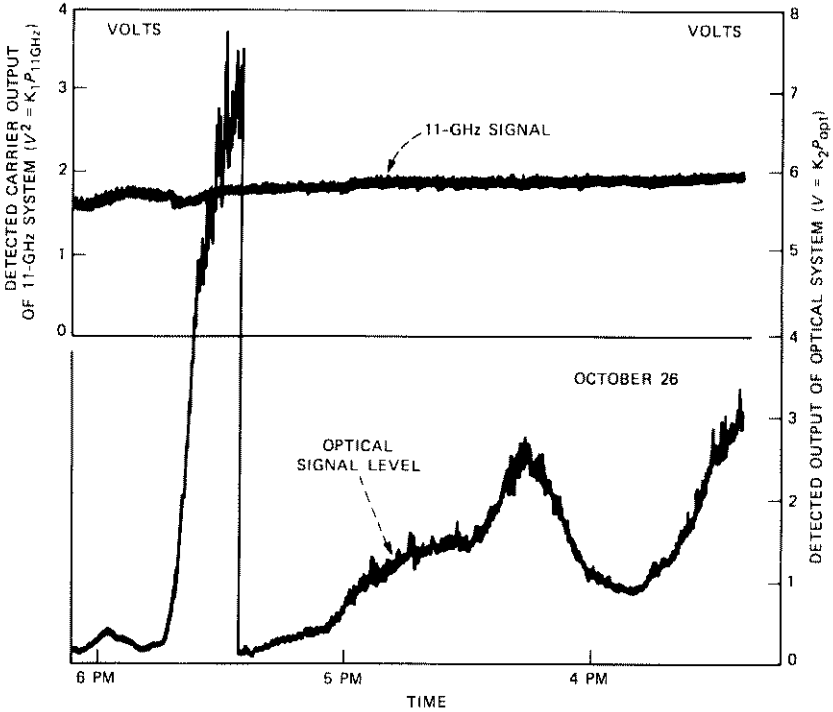



Fig. 11—Microwave and optical signal vs time (attenuator removed).

almost 6 dB of enhancement and about 12 dB of fading. The optical signal shows no fading and very little scintillation.

## VI. NOISE

In the preceding sections it has been shown that a beam of light can be transmitted 23 miles. By using feedback from the receiver to control the transmitter elevation angle, the rather narrow beam ($2.4 \times 10^{-4}$ radians between −20 dB levels) can be kept centered on the receiver. And finally, by scanning the transmitter vertically, transmitter elevation control can be reestablished after a deep fade.

There is still the question of the ability of the optical system to satisfactorily carry information at the rates necessary. As is shown in the appendix, there are modulators available that will impress the information of a microwave intermediate frequency (IF) signal onto the light beam. The remaining question is, can the signal be received with sufficient accuracy to provide a satisfactory replica of the impressed signal?

The photomultipliers are fast enough for either frequency modulation (FM) or pulse code modulation (PCM), having a 3-ns rise time. If

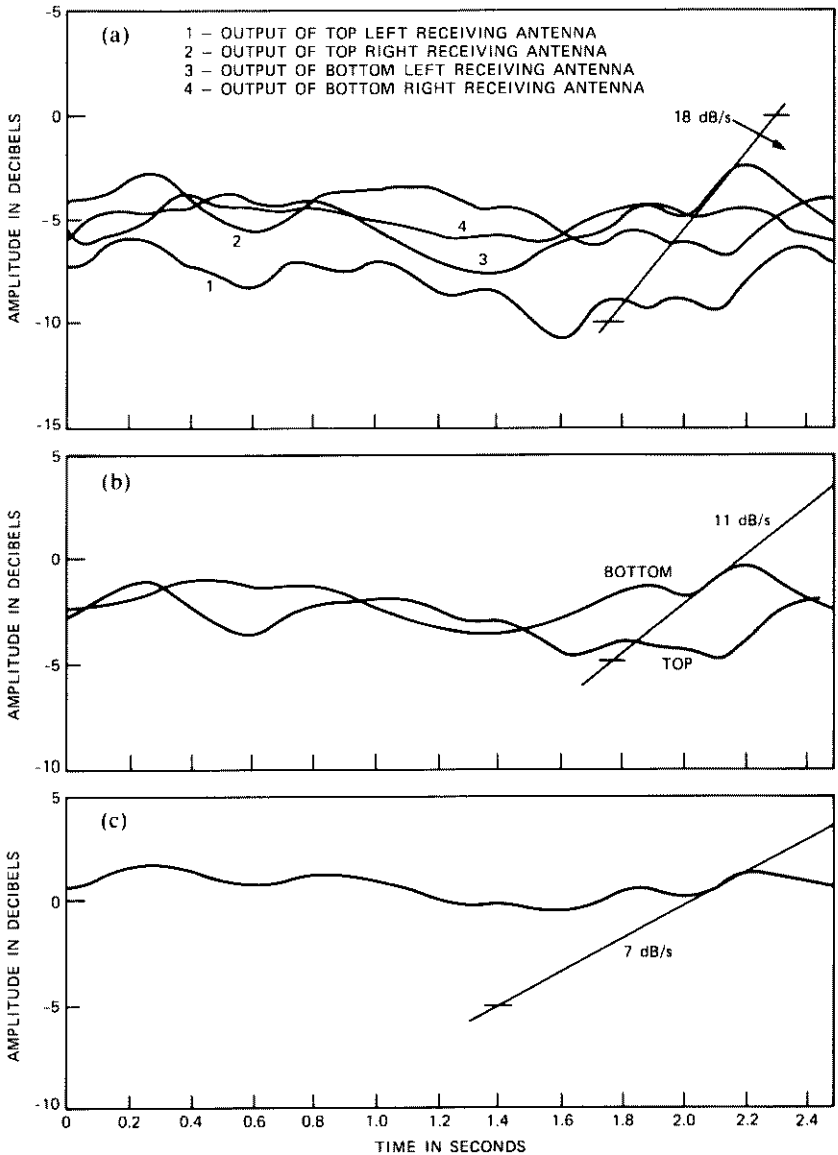



Fig. 12—(a) Demodulated output of four individual photomultipliers vs time. (b) Demodulated output of upper and lower photomultiplier pairs vs time. (c) Demodulated output of sum signal of four photomultipliers vs time.

the 4-MHz baseband signal is sampled at 8 MHz and encoded into 8-bit PCM,* [8] the rate is only 64 MHz. This is well within the 100-MHz capability of the modulator and photomultiplier. In a commercial

* Because noise is spread over the entire band, 8-bit PCM is adequate.

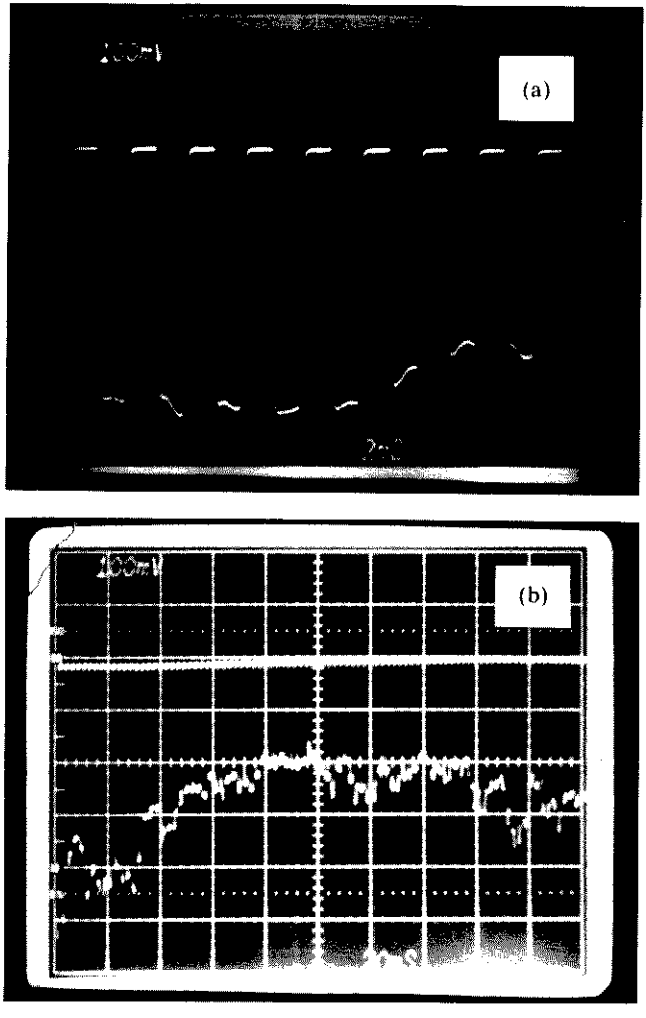



Fig. 13—(a) The ac of one photomultiplier vs time (2 ms/div). (b) The ac of one photomultiplier vs time (20 ms/div).

system, the photomultipliers would almost certainly be replaced with p-i-n diodes, and they are even faster.

So the only question is the noise generated in the receiver. There are four sources of noise in an atmospheric receiver:

(*i*) Thermal noise
(*ii*) Shot noise of the signal current
(*iii*) Shot noise of the current because of background illumination
(*iv*) Shot noise of the dark current.

The thermal noise has a mean-square value of

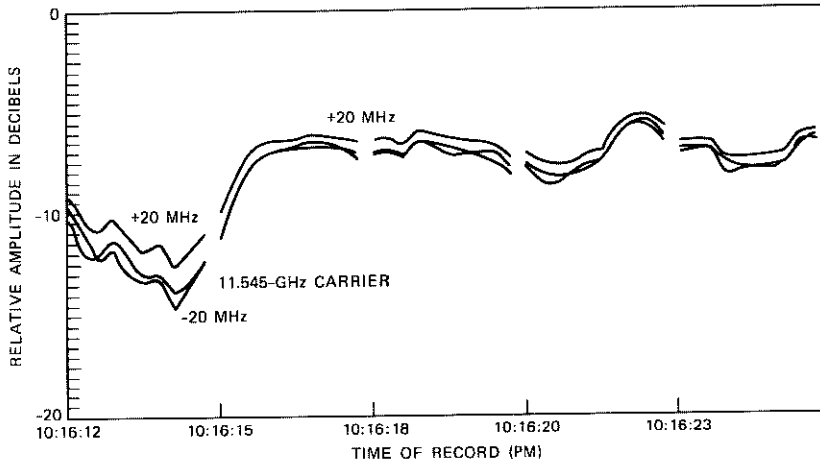



Fig. 14a—Detected amplitude of 11-GHz tones vs time.

$$\overline{i^2} = \frac{4\mathrm{K}TB}{R},$$

where K is Boltzman's constant ($1.38 \times 10^{-23}$), $T$ is temperature (300°K), $B$ is the bandwidth ($10^8$ Hz), and $R$ is the amplifier input resistance.

The thermal noise can be made very small by using a transimpedance amplifier. But we will calculate the noise using an input resistance ranging from 50 to 1000Ω, which with the interelectrode capacitance of 3 pF gives a time constant 0.15 to 3 ns.

The average current squared ($\overline{i^2_{\mathrm{sig}}}$) of the shot noise of the signal current ($I_{\mathrm{sig}}$) is simply

$$\overline{i^2_{\mathrm{sig}}} = 2\mathrm{e}\frac{I_{\mathrm{sig}}}{2}B,$$

where e is the electron charge ($1.6 \times 10^{-19}$) coulombs.

The noise owing to the background illumination is

$$\overline{I^2_{\mathrm{back}}} = 2\mathrm{e}I_{\mathrm{back}}B,$$

where $I_{\mathrm{back}}$ is the dc current due to background illumination.

By the use of the hood in front of the Fresnel lens and the iris and red filter at the photomultiplier, the background illumination, under the worst condition, causes three times as much current as the average signal current if the anode voltage is limited to 800 volts (maximum anode voltage is 1500). Therefore, $I_{\mathrm{back\ max}} = 3\ I_{\mathrm{sig}}/2$. Most of the time, the two currents are about equal (i.e., $I_{\mathrm{back}} = I_{\mathrm{sig}}/2$). Of course, at night, there is virtually no background current. The peak signal current is

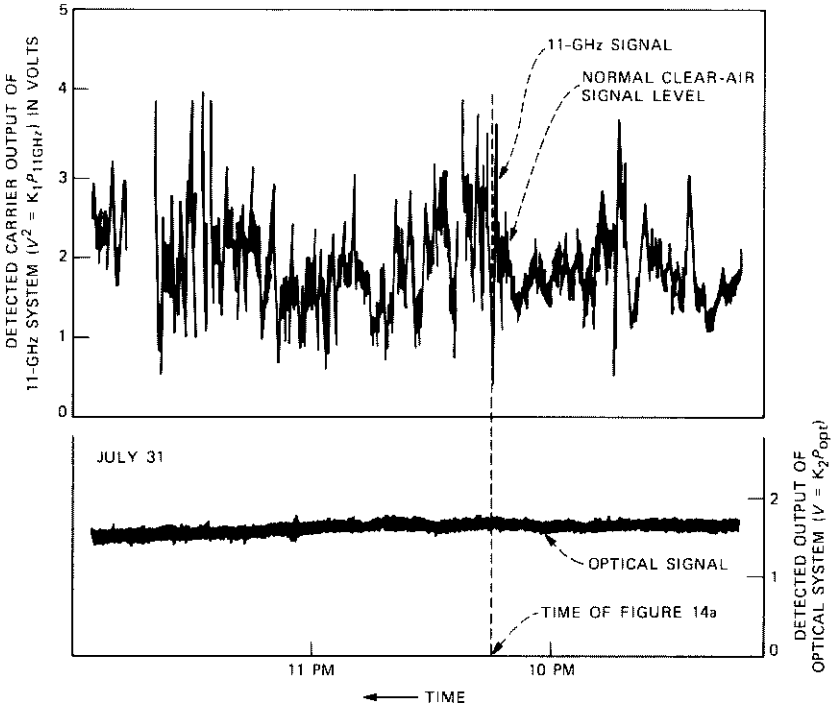



Fig. 14b—Detected amplitude of 11-GHz carrier and optical signal vs time.

about 50 $\mu$A and the highest background current was 75 $\mu$A. The dark current is three orders of magnitude smaller than the signal current, and can be ignored.

A useful measure of a receiver is the signal-to-noise ratio, and we can calculate it for this receiver with the information we have.

Figure 13b shows the output of one photomultiplier on a typical day. The voltage across a 10-kilohm resistor varies between 500 mV and 175 mV. The average, without scintillation, is 340 mV or a current of 34 $\mu$A, and from the four antennas the average sum is 136 $\mu$A. The background current was about equal to the average signal current at the time of the measurement. The peak-signal-to-average-noise ratio for an on/off PCM signal can be written as

$$\frac{S}{N} = \frac{(I_{sig})^2}{2e\left(\frac{1}{2} I_{sig}\right) B\left(1 + \frac{2I_{back}}{I_{sig}}\right) + \frac{4KTB}{R}}.$$

The calculated peak-signal-to-average-noise ratio S/N, using an average signal current of 136 $\mu$A, is shown in Table I.

Scintillation can cause a considerable variation of received signal. It

Table I—Calculated peak-signal-to-average-noise ratio

| $I_{back}/I_{sig\ avg}$ | $R$ (Ω) | Signal to noise (dB) |
|---|---|---|
| 0 | 50 | 58 |
| 0 | 100 | 60 |
| 0 | 1000 | 67 |
| 1 | 50 | 57 |
| 1 | 100 | 60 |
| 1 | 1000 | 65 |
| 3 | 50 | 57 |
| 3 | 100 | 59 |
| 3 | 1000 | 63 |

usually amounts to plus or minus several decibels. Under severe conditions it has been seen to cause ±10 dB variation of received signal.[7]

## VII. CONCLUSIONS

We have shown that:

(*i*) The elevation angle of an optical transmitter must be continuously adjusted to compensate for beam-curvature changes in the atmosphere, and that this can be done by a servo signal from the receiver.

(*ii*) The azimuth angle of an optical transmitter need not be corrected on a continuous basis. Occasional manual adjustment (every month or so) is adequate.

(*iii*) Heavy fog or rain will cause a total loss of optical signal, but the optical system can be reestablished by scanning the transmitter vertically.

(*iv*) The clear-air path loss is 27 dB on a 23-mile path.

(*v*) The received beam diameter is 32 feet, where the power is 20 dB below the level at the center of the beam.

(*vi*) The sky light can cause as much as three times as much current in the receiver as the average signal current, and this gives a system peak-signal-to-average-noise ratio of about 60 dB.

(*vii*) Modulators exist that should comfortably accommodate either the microwave IF signal (centered at 70 MHz) or a digitized version of the 4-MHz baseband.

(*viii*) The received signal scintillates several decibels most of the time, and can get as severe as 10 dB. The frequency rate of scintillation contains components up to several kilohertz.

(*ix*) During a clear-air microwave fade, the optical signal did not fade, and in fact, had much less scintillation than usual.

## REFERENCES


1. H. J. Schulte and B. G. King, unpublished work.
2. B. G. King and L. J. Greenstein, unpublished work.
3. A. Schiavone, unpublished work.
4. *Handbook of Optics*, New York: McGraw-Hill, 1978, pp. 14.16–17.
5. G. K. Ochs and R. S. Lawrence, "Measurements of Laser Beam Spread and Curvature Over Near Horizontal Atmospheric Paths," U.S. Department of Commerce/Environmental Sciences Service Administration, February 1969, ERL 106-WPL 6.
6. G. K. Ochs, private communication.
7. W. J. Hannan, J. Bordogna, and D. Karlsons, "Practical Aspects of Injection Laser Communication Systems," RCA Rev., *27*, No. 4 (December 1967), pp. 609–19.
8. J. C. Candy, private communication.


## APPENDIX

### *Modulation of Light*

There are two well-developed optical-modulation techniques. Both involve transparent solids that allow an electrical signal applied to the solid to modify the transmission of light through the solid. In one, the transparent body is mechanically driven by a piezoelectric driver and the pressure standing-wave is employed to form an optical diffraction grating in the body of the solid. This grating causes part of the light to be bent out of its unmodified path and this deflected part of the beam is used as the modulated optical signal. The frequency of the signal that causes this grating is relatively high, on the order of 100 MHz, and is resonant in the crystal. Digital or analogue modulation is impressed on the light beam by either keying the high-frequency signal off and on or linearly modulating its magnitude. Because of the time required for the resonance to build up and to decay, the modulating signal is usually much smaller than the resonant frequency, on the order of a few megabits or megacycles.

The other type of modulator uses a crystal in which the polarization angle of the light is rotated by the electrical signal. The crystal tends to be square in cross section (about 1/2 mm on a side) and about 1 cm long. Metal is plated onto two opposite long surfaces, and the signal voltage is applied to these. The light is passed through the long axis of the crystal in a path parallel to the plated surfaces. The light coming out of the crystal has some polarization, and is passed through a polarizer plate with its surface normal to the optical beam.

When voltage is applied, the polarization of the light out of the crystal rotates in proportion to the voltage applied, and after passing through the output polarizer, the output optical power is changed. The process is continuous, that is to say, the light can be brought back to the original polarization by applying enough voltage. The power out of the output polarizer varies as

$$P_{out} = K_1 P_{in} \sin^2 K_2 V_{in}, \tag{1}$$

where $K_1$ is the maximum through-transmission, $K_2 = 2\pi/V_{360}$, and $V_{360}$ is the voltage needed to cause 360° of phase rotation. If the voltage applied is a sinewave $V_{in} = V_0 \sin \omega t$, and the crystal is biased with a dc voltage, $V_B$, then the output power is

$$\begin{aligned} P_{out} &= K_1 P_{in} \sin^2(K_2 V_B + K_2 V_0 \sin \omega t) \\ &= \frac{K_1 P_{in}}{2}[1 - \cos 2(K_2 V_B + K_2 V_0 \sin \omega t)] \\ &= \frac{K_1 P_{in}}{2}[1 - \cos(2K_2 V_B)\cos(2K_2 V_0 \sin \omega t) \\ &\quad + \sin(2K_2 V_B)\sin(2K_2 V_0 \sin \omega t)]. \end{aligned} \tag{2}$$

It is well known that

$$\cos(a \sin\theta) = J_0(a) + 2J_2(a)\cos 2\theta + 2J_4(a)\cos 4\theta + \cdots$$

and

$$\sin(a \sin\theta) = 2J_1(a)\sin\theta + 2J_3(a)\sin 3\theta + 2J_5(a)\sin 5\theta + \cdots.$$

Let $2K_2V_B = b$ and $2K_2V_0 = a$. Then

$$\begin{aligned} P_{out} = \frac{K_1 P_{in}}{2}\{1 &- \cos b[J_0(a) + 2J_2(a)\cos 2\omega t + \cdots] \\ &+ \sin b[2J_1(a)\sin \omega t + 2J_3(a)\sin 3\omega t + \cdots]\}. \end{aligned} \tag{3}$$

To avoid all even harmonics, set $b = \pi/2$

$$\begin{aligned} b = 2K_2 V_B = \frac{\pi}{2} &= 2\frac{2\pi}{V_{360}} V_B \\ V_B &= \frac{V_{360}}{8}. \end{aligned} \tag{4}$$

With this bias, only the fundamental and its odd harmonics are present in the modulated signal.

The fraction of the optical power going into the harmonics is determined by the amplitude of the impressed signal, $V_0$. The argument of the Bessel functions is

$$a = \frac{\pi}{2}\frac{V_0}{V_B}. \tag{5}$$

Because of the nonlinearity of the modulator there is a relatively strong (−27 dB) third harmonic at only 20-percent modulation of the fundamental [$m = 2J_1(4\pi V_0/V_{360})$].

However, such nonlinearity has little effect on an FM signal so long as the amplitude of the FM signal is constant. The fundamental and its sidebands can be filtered from the higher harmonics, and with no

second harmonic present, this is easily done with a microwave intermediate frequency (IF) signal. If the IF signal is centered at 70 MHz, the deviation is ±4 MHz and the baseband signal bandwidth is 4 MHz. The highest sideband of the fundamental, using Carson's rule, is $f_c + \Delta f + f_B = 70 + 8 + 4 = 82$ MHz. The lowest sideband of the third harmonic is $3f_c - 3\Delta f - f_B = 210 - 24 - 4 = 182$ MHz. This should be easily taken care of by the filter in a microwave receiver.

All of the above assumes that the modulation is FM. This requires that the bias be set at $V_{360}/8$. However, binary PCM transmission calls for setting the bias so that no light is transmitted when the signal is a zero and maximum light is transmitted when the signal is a one. Thus, for PCM the bias $V_B$ is set at zero and the digital signal "one" is set at $V_B = V_{360}/4$. The harmonic generation during the transition is of little concern.

Acousto-optic and electro-optic modulators are commercially available. The acousto-optic modulators allow modulation up to about 3.5 MHz with rise and fall times of 120 ns, which is inadequate for most microwave IF signals. The modulation efficiency is about 85 percent, and the extinction ratio is about 1000:1. The electro-optic modulators allow modulation up to 100 MHz with rise and fall times of 3.5 ns. The depth of modulation at 6328Å is 85 percent and the extinction ratio is 500:1. The electro-optic modulators require a feedback bias control.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# Star Network With Collision-Avoidance Circuits

By A. ALBANESE

(Manuscript received July 15, 1982)

*This paper describes the implementation of a multiple-access star network that uses a new collision-avoidance circuit to avoid collision of packets and to take full advantage of wideband transmission systems. The analysis shows that the resulting network has a channel capacity equal to 1, has good stability under heavy traffic conditions, allows transmission of packets shorter than the network round trip time, and provides distributed switching. Collision-avoidance circuits have been built and operated up to 50 Mb/s.*

## I. INTRODUCTION

Carrier-Sense Multiple-Access local-area networks with collision detection (CSMA-CD)[1] have been implemented using lightwave technology,[2] but the use of lightguides is questioned because the networks do not benefit from the full bandwidth of the lightguides. In these networks, the lightguides are used to reduce ground-loop voltage, electromagnetic interference (EMI), size, and weight of the cable.

CSMA-CD networks have the disadvantage of requiring a large packet size when they operate at high bit rates. For example, a 1-km-long (2-km round trip) passive star network,[2] operating at 150 Mb/s, has an efficiency of 5 percent for 256 bits/packet, 44 percent for 4096 bits/packet, and 93 percent for 65,536 bits/packet. In addition to the low efficiency there is an associated instability problem that appears when the network reaches the channel capacity.[3] Without traffic restrictions or under heavy traffic conditions the network becomes unstable and crashes.

This paper describes a new method to eliminate the collisions caused by simultaneous transmission of packets in a star network. The method consists of placing collision-avoidance circuits in the node at the center of the star. This collision-avoidance circuit does the work of a "traffic cop," in that it lets pass the packets that arrive while the node is idle and blocks those that arrive while the node is busy. The packets that

make it through the "traffic cop" are broadcast to all users including the sender, and the blocked ones are retransmitted (by the originating users) until they succeed in getting through the node.

The insertion of collision-avoidance circuits in a star network results in a network that has the combined advantages of a zero-length CSMA-CD (high throughput and good stability) with those of ALOHA (transmission of packets shorter than the network round trip time). The resulting network is more efficient and stable than CSMA-CD networks and takes full advantage of high-bit-rate transmission. These characteristics make the network attractive for transmission at high bit rates and make lightwave networks suitable for local area networks.

The "traffic cop" requires active components at the center of a passive star, which poses a disadvantage in terms of reliability. This problem is minimized by using a small number of components in the circuit, conserving the distributed-switching feature of multiple-access networks, and providing centralized maintenance.

The sections that follow describe the architecture, protocol, implementation, and analysis of the network.

## II. NETWORK ARCHITECTURE

Figure 1 shows the case of a local-area network where all users are connected in a star configuration to a central node (N) by a user-interface circuit (UIC). This interface consists of a transmitter (T), a receiver (R), and a logic circuit (C) that enable the receiver that first receives a packet after the node is idle. The node consists of a short bus with a maximum length equal to the length that makes the time of a bus round trip equal to the period of one bit. Each user is connected to the UIC by a user's link.

## III. PROTOCOL

The protocol between the user and the network can be summarized as follows: the user-interface circuit blocks those packets arriving at the central node before they could cause a collision, and the users keep retransmitting unsuccessful packets until they succeed in getting through the node.

The description of the protocol is divided into the user and node protocols.

### *3.1 User protocol*

The user has input and output buffers to store the received and transmitted packets. When a user end has a packet in the output buffer to be transmitted:

(*i*) It transmits the packet at once.

(*ii*) It waits the user's round trip time, $T_{RT}$.

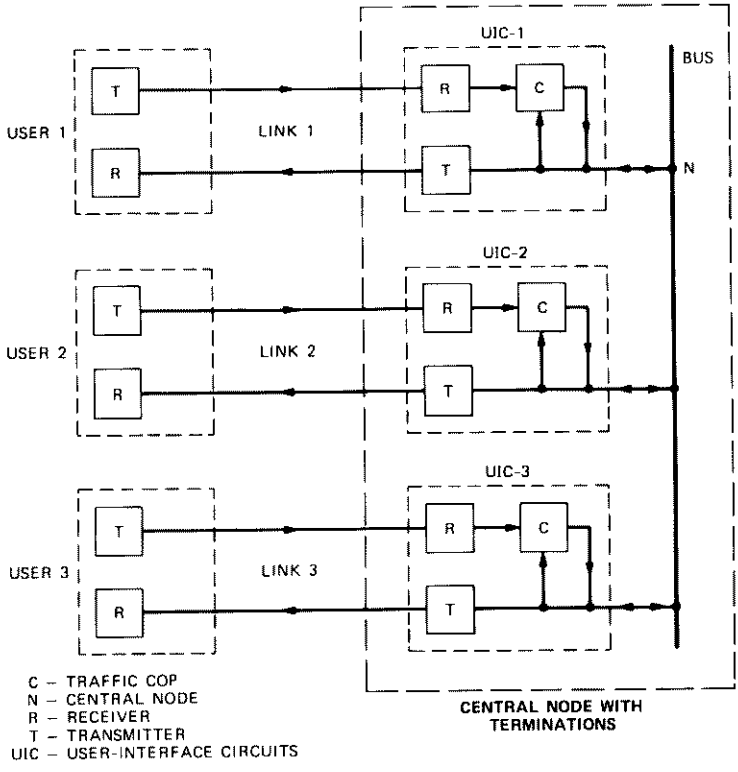



Fig. 1—A local-area network showing how all users are connected in a star configuration to a central node by independent user-interface circuits.

(*iii*) It checks the input buffer, and if a packet was received, it analyzes the origination and destination.

(*iv*) If a packet was not received, or if the received packet is not the same as the transmitted one, the packet was blocked at the node, and the whole process is repeated until the packet succeeds in getting through the node.

### *3.2 Node protocol*

The transmitters and receivers at the central node are controlled in the following way:

(*i*) All the transmitters are connected to the node N at all times. Therefore, all the users are continuously receiving the packets broadcast by the node.

(*ii*) All the receivers are disconnected from the node.

(*iii*) The receivers and the node are continuously monitored for "idle" or "busy" status.

(*iv*) When a packet arrives at the receiver, the receiver becomes

"busy." A transition of the receiver from "idle" to "busy" while the node is "idle" connects the receiver to the node, and changes the node status to "busy." A transition of the receiver from "idle" to "busy" while the node is "busy" is ignored and the receiver remains disconnected. In this way, only the receiver with the first arriving packet is connected to the node, and all the others are ignored or disconnected.

($v$) The node status returns to "idle" when the broadcast packet ends and the receiver is disconnected.

This protocol is performed by a logic circuit (C) whose functions are shown in Fig. 2. The receiver (R) and the transmitter (T) are part of the user's link and they vary according to the type of link. The logic circuit consists of an AND gate and four flip-flops. The four flip-flops are wired to perform the receiver-monitor, node-monitor, arbiter, and hold-on functions. The AND gate connects and disconnects the receiver to the node (N) following the status of the hold-on flip-flop. Figure 3 shows the timing diagram for the different parts of the circuit.

The receiver monitor and the node monitor are retriggerable single-shot circuits that detect the presence of a packet by sensing the carrier. They go "high" at the arrival of a packet and return "low" at the end of the packet.

The arbiter is a D-type flip-flop that produces an "I was first" pulse when the receiver monitor goes "busy" while the node monitor is "idle." The "I was first" pulse indicates that the receiver has the first arriving packet and sets the hold-on flip-flop "high." All "idle" to "busy" transitions of the receiver monitor that occur while the node monitor is "busy" are ignored and they do not set the hold-on flip-flop.

The hold-on flip-flop goes "high" following the "I was first" pulse

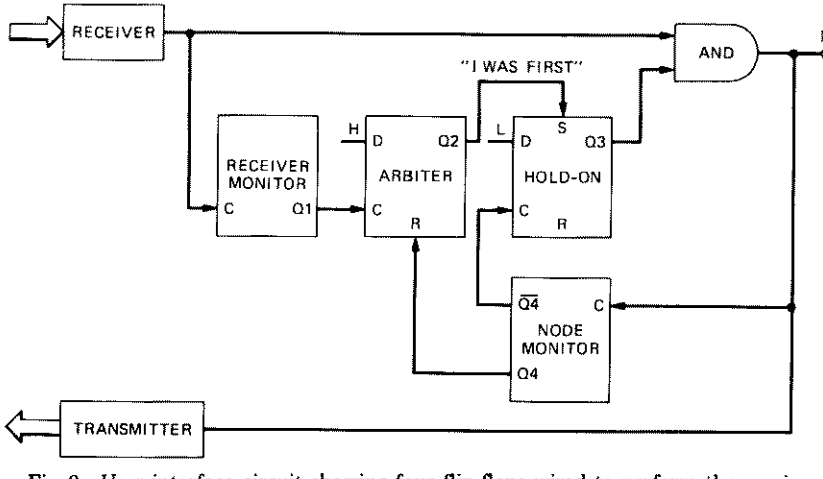



Fig. 2—User-interface circuit showing four flip-flops wired to perform the receiver-monitor, node-monitor, arbiter, and hold-on functions.

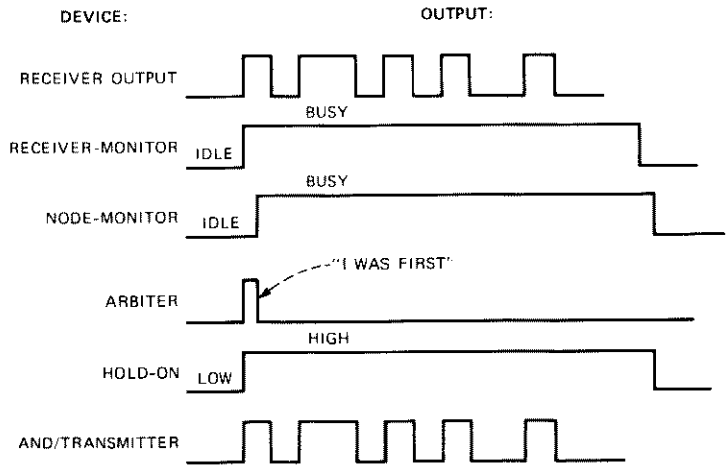



Fig. 3—A timing diagram of the user-interface circuit. The hold-on goes high only when the receiver monitor goes high before the node monitor does.

and returns "low" when the node monitor goes "idle." The output of the hold-on flip-flop activates the AND gate that connects and disconnects the receiver to the node (N).

The reader may wonder what happens when two or more packets arrive simultaneously. This is a rare event because it takes less than 20 ns to determine which was the first packet arriving. But suppose anyway that two packets arrive within a period of 20 ns; then a collision is possible because two receivers will be simultaneously connected to the node. An exclusive-OR circuit can be installed in the way shown by Fig. 4 to handle this rare event. This additional circuit resets the hold-on flip-flop ("low") any time that the receiver signal is different from that of the transmitter and while the node is "busy." The two receivers remain connected until the signal for one of the receivers differs from its transmitter signal. The receiver with the first low bit stays connected and the others are disconnected. Figure 5 shows the timing diagram of the different components when two users are randomly transmitting words.

## IV. ANALYSIS

To study the system let us consider a network of $K$ users each successfully transmitting $\lambda$ packets/second, each $\tau$ seconds long through the node. Then one defines $S = K\lambda\tau$, which is the average channel utilization, also known as the traffic intensity or traffic throughput.

$S$ is also the probability of a user encountering a busy node in any attempt at transmitting a packet. The probability of successfully

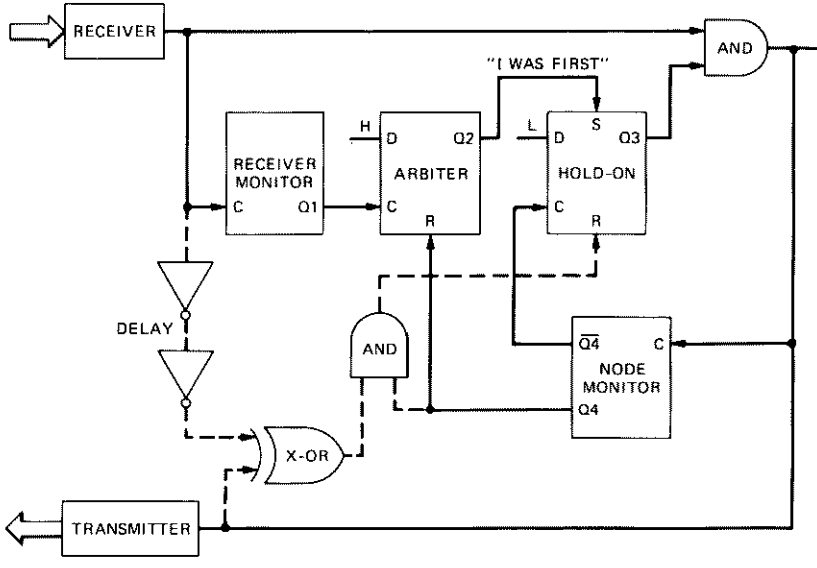



Fig. 4—Addition to the circuit to handle simultaneous events. The dashed lines indicate the circuit added to Fig. 3 to determine the priority of packets that arrive within the circuit response time.

transmitting a packet is $1 - S$. Therefore, the average number of transmissions required to get a packet through the node is

$$N = \frac{1}{1 - S} \tag{1}$$

and the average traffic offered by the users to the network is

$$G = SN = \frac{S}{1 - S}; \tag{2}$$

consequently, $S$ can be expressed in terms of $G$ as

$$S = \frac{G}{G + 1}. \tag{3}$$

Another parameter of importance in the network is the average user's transmission delay. This delay is computed adding all the possible delays weighted by their probabilities:

$$\begin{aligned} D &= T_1(1 - S) + (2T_1 + T_2)S(1 - S) \\ &\quad + (3T_1 + 2T_2)S^2(1 - S) + \cdots \\ &= T_1N + T_2SN, \end{aligned} \tag{4}$$

where $T_1 = T_{RT} + \tau$, $T_2$ is the retransmission delay in addition to $T_1$,

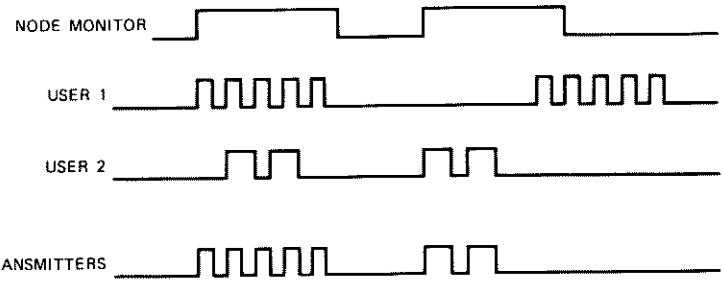



Fig. 5—Timing diagram showing two users randomly transmitting. The circuit broadcasts the packets of user 1 and user 2 that arrive only while the bus is idle.

and $T_{RT}$ is the user's round trip time. The delay is reduced by making $T_2$ small comparable to $T_1$, which leaves only the first term in eq. 4.

The above equations are analogous to those of a zero-length CSMA-CD network where the ratio $T_{RT}/\tau = 0$ makes the maximum value of $S$ equal to 1.

More efficient modes of operation are possible but they will increase the complexity of the stations. For example, a clock may be installed at the node N to broadcast a burst to indicate the beginning of a frame, and have reserved channel assignment. This arrangement decreases the average delay but it requires a reservation protocol.

## V. CONCLUSION

A new arrangement for a star network was proposed and a central-node configuration, as shown in Fig. 1 with UIC, was built and operated with packets up to 50 Mb/s. The node eliminates collisions by resolving the right-of-way when several packets arrive at the node. The node lets pass the packet that arrives first and blocks all other packets that would collide with the first one. Meanwhile, the users keep retransmitting packets until they get through the node.

The analysis shows that the average number of retransmissions depends on the traffic intensity at the node, and the retransmission of unsuccessful packets does not degrade the traffic throughput of the node. Also, the channel capacity has the maximum value 1, and it is not a function of the ratio between the packet size and the network-round-trip time.

The analysis shows that at moderate traffic intensities (50 percent) the average number of retransmissions is two. Traffic intensities approaching 100 percent may cause a large number of retransmissions. A 75-percent traffic intensity requires an average of four retransmissions, 80 percent requires five, and 90 percent requires ten.

High-bit-rate services can be provided simultaneously with low-bit-rate services without changing the traffic intensity simply by increasing

the transmission bit rate and holding the packet duration and the packet frequency constant.

## REFERENCES


1. R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Commun. ACM, *19,* No. 7 (July 1976), pp. 395–404.
2. E. G. Rawson, R. M. Metcalfe, R. Norton, A. B. Nafarrete, and D. Cronshaw, "Fibernet: A Fiber Optic Computer Network Experiment," IEEE Trans. Commun., *COM-26,* No. 7 (July 1978), pp. 983–90.
3. L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I—Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," IEEE Trans. Commun., *COM-23,* No. 12 (December 1975), pp. 1400–16.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# Pressure-Volume-Temperature Behavior in the System $H_2O$-NaOH-$SiO_2$ and Its Relationship to the Hydrothermal Growth of Quartz

By E. D. KOLB, P. L. KEY, R. A. LAUDISE, and E. E. SIMPSON*

(Manuscript received September 21, 1982)

*We have measured the pressure-volume-temperature relations in the high-pressure solutions used to grow electronic quartz and used this data to establish safe operating conditions for commercial production. High-temperature aqueous solution (hydrothermal) quartz growth, because of the importance of its product to electronics, must be ranked as one of the more important crystal-growth processes. We report here a convenient laboratory method for hydrothermal p-V-T measurements and give pressure data in 1.0-mol NaOH and in 1.0-mol NaOH saturated with quartz as a function of temperature up to 450°C and 30,000 psi. These results are compared with pressures measured on production-sized equipment. The results are used to establish the temperature at which the gas phase disappears under various conditions. The steels used for construction of high-pressure production autoclave equipment are brittle below a specific temperature, which increases slowly with service. Our p-V-T data can be used to assure that high pressures are avoided at temperatures where the autoclave is brittle. Finally, the depressions of pressure are used to glean information about the nature of the solute species present during growth, and can ultimately be of use in quartz rate and perfection studies.*

## I. INTRODUCTION

Commercial hydrothermal growth of single-crystal α-quartz has been practiced for more than twenty years. It is arguably second only to Si

* Western Electric, Merrimack Valley Works, No. Andover, Massachusetts 01845.

in its importance to electronics. The hydrothermal *mineralizer* or solvent used is either NaOH or $Na_2CO_3$ and, particularly in the case of the more generally used NaOH, both the physical chemistry and crystal-growth aspects have been quite extensively studied, including phase relations,[1] solubility,[2] growth kinetics,[3] distribution of impurities (especially OH),[4] perfection,[5] and electrical properties[6] of grown quartz. However, the pressure-volume-temperature (p-V-T) relations for neither $Na_2CO_3$ nor NaOH for mineralizer solutions saturated with quartz have been determined. Data on p-V-T for aqueous solutions of NaOH[7-9] and $Na_2CO_3$[7] are available, but their relationship to the silica-saturated solution used in growth is tenuous. For our purposes the system $H_2O$—NaOH—$SiO_2$ is of particular interest since it is most used and probably most studied. We have directed our studies to it and have adopted it for production.[10] Indeed, except for recent p-V-T measurements in the system $H_2O$—$H_3PO_4$—$AlPO_4$,[11] under conditions of $AlPO_4$ saturation like those used for $AlPO_4$ growth, no p-V-T measurements of mineralizers saturated with the solutes used in hydrothermal crystal growth have been made. In the paper of Kolb et al.,[11] we described a technique and equipment for rather rapid p-V-T measurements that can be easily applied to other saturated hydrothermal systems, and reviewed the literature summarizing p-V-T measurements in hydrothermal mineralizers. As a result of this work and our research and factory experience with quartz, it is our belief that p-V-T measurements in the system $H_2O$—NaOH—$SiO_2$ would be particularly useful for the following reasons:

(*i*) It has been shown[12] that the brittle-ductile transition temperature of low-carbon steels of the sort used in hydrothermal autoclaves gradually increases when such steels are aged in the temperature range of quartz growth. For safety reasons it is important to be sure no autoclave is exposed to substantial pressure while in the brittle regime. To assure this, accurate p-V-T data for the hydrothermal solution are essential. The need for such data becomes particularly important when autoclaves have been in service for periods of years.

(*ii*) Pressures are regularly measured, monitored, and used for control in commercial growth. In such growth a temperature differential, $\Delta T$, between dissolving-nutrient zone and seed-growth zone, is necessary to produce supersaturation. It would be especially useful to have p-V-T data for both isothermal *and* $\Delta T$ conditions for comparison with production data. Pressure changes during growth could possibly be used to monitor internal $\Delta T$ changes during growth.

(*iii*) Pressure depression in comparison with $H_2O$ and $H_2O$—NaOH may be useful in gleaning information concerning the species present during hydrothermal growth.

(*iv*) Pressure data may provide insights into kinetics, distribution

of impurities, and perfection, which could further improve the speed, economics, and perfection of commercial growth.

For the foregoing reasons we decided to study and here report p-V-T measurements in the system $H_2O$—NaOH—$SiO_2$ under isothermal conditions and with temperature differentials over the range of conditions relevant to quartz growth. For comparison purposes we also studied the system $H_2O$—NaOH.

## II. EXPERIMENTS

This section describes procedures used in laboratory measurements. Measurements in production vessels are described later. The apparatus and procedures used are similar to those reported in our study of $H_2O$—$H_3PO_4$—$AlPO_4$,[11] except that Pt-lined Morey vessels were not required when quartz was present. This is due to the fact that low-carbon steel autoclaves are relatively inert in the system $H_2O$—NaOH—$SiO_2$ because of the formation of insoluble sodium-iron silicates that protectively coat the vessel.[5] Thus a 1-1/4-in ID x 12-inch IL autoclave with a modified Bridgman closure made by Autoclave Engineers (Erie, PA) fabricated from Timken 17-22-AS steel was used. A series of runs at fills of 65, 75, 80, 83, 86, and 89 percent were made as were calibration runs using pure water at fills of 70, 80.1, and 80.9 percent. Pressure was measured with 0- through 5000-psi (350-bar), 0- through 20,000-psi (1380-bar), and 0- through 40,000-psi (2760-bar) Bourdon gauges (Heise Co., Newtown, Conn.), precalibrated on a dead-weight tester. Pressures above the coexistence curve were measured with gauges with a resolution of ±20 psi (~1 bar), while those along the coexistence curve (where liquid and vapor are in equilibrium) were obtained using the 0- through 5000-psi gauge whose resolution is ±5 psi (~0.3 bar). The pressure gauges were connected to the cover of the autoclave by 0.017-inch ID stainless capillary tubing brazed to a 1/4-inch diameter high-pressure cone seal.

Separate measurements in the system $H_2O$—NaOH were made using a Pt-lined Morey autoclave and pressure take-off of the sort described in our aluminum phosphate p-V-T work.[11] Autoclave volume for both autoclaves was measured by filling with $H_2O$ as previously described.[11] The volumes of the steel plunger and steel coupling were calculated from dimensions. The Bourdon-gauge tube volumes were determined by forcing $H_2O$ through the tube with a calibrated syringe that was fitted to the inlet and measuring the water collected at the exit (gauge-bleed port). The volume of $H_2O$ in the tube is arrived at as a difference that is taken only when bubble-free water is collected at the exit. The volumes of the steel and Pt-lined pressure take-offs and the stainless steel capillary were determined as described previously.[11] Table I lists typical volumes.

Table I—Typical volumes

| | | | |
|---|---|---|---|
| Bridgman Autoclave | 250.0 cc | Morey Pt-Lined Autoclave | 30.0 cc |
| Steel Take-off | 1.00 cc | Pt-lined Take-off | 0.15 cc |
| Steel Capillary | 0.30 cc | Steel Capillary | 0.30 cc |
| Gauge 40,000 psi | 18.0 cc | | |
| 20,000 psi | 5.0 cc | | |
| 5,000 psi | 13.0 cc | | |

The placement of heaters and thermocouples for the Pt-lined autoclave was as described previously,[11] while for the Bridgman autoclave heating was provided by a hot plate on the bottom and band heaters appropriately spaced so as to obtain minimum $\Delta T$ or, if desired, a particular $\Delta T$. An improved temperature-control system using an Electronix III controller with silicon-controlled rectifiers (Leeds and Northup, N. Wales, PA) was employed in the present work. Temperatures are estimated to be controlled within ±0.5°C.

To assure saturation and protect the steel autoclave walls from attack, quartz plates 1- x 2- x 0.04-inches were mounted in the Bridgman autoclave, along with 75 gm of small-particle quartz nutrient. Standard NaOH solutions of 1.000 ± 0.003 mol concentration were purchased from Fisher Scientific.

## III. RESULTS AND DISCUSSION

### *3.1 Calibration*

As a check of our procedures we elected to investigate how well our system reproduced the p-V-T data of Kennedy[13] for pure water. Three runs were made at fills of 70, 80.1, and 80.9 of the autoclave. The p-T behavior at 80.1- and 80.9-percent fill was measured in the unlined Bridgman vessel up to pressures of ~25,000 psi (~1725 bar) to simulate our measurements in $H_2O$—NaOH—$SiO_2$. The p-T behavior at 70-percent fill was measured in the Pt-lined Morey vessel with a Pt lead through to the stainless steel capillary coupling so as to simulate our measurements in $H_2O$—NaOH. Of necessity we were limited to the pressure capability of Morey autoclaves (~10,000 psi ~700 bar) for all work in lined vessels.

Table II gives representative pressures obtained in these runs and the equivalent percent fill for $H_2O$ from Kennedy's data.[13] Furthermore, in Table II the experimental percent fills are corrected for: (*i*) temperature expansion of the vessel, (*ii*) pressure dilatation of the vessel, and (*iii*) compression of the water in the lines and gauge. The temperature expansion of the vessel was obtained using eq. (1) of Kolb and Laudise.[11] The Morey and Bridgman vessels were made of Timken 17-22-AS steel so that the coefficient of expansion in Fig. 5 of Kolb and Laudise[11] was used. The pressure dilatation of the vessels was calculated using eq. (2) of Kolb and Laudise[11] and the values $Y$ and $\nu$

Table II—Representative pressures and percent fills for $H_2O$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Experimental Percent Fill | 80.9 | 80.9 | 80.9 | 80.9 | 80.1 | 80.1 | 80.1 | 70 | 70 |
| Measured Temperature (°C) | 250 | 300 | 350 | 400 | 300 | 350 | 400 | 350 | 400 |
| Measured Pressure (bar) | 46.9 | 676 | 1307 | 2060 | 572 | 1172 | 1765 | 455 | 903 |
| Experimental Percent Fill Corrected for Temperature Expansion and for Initial Fill on a Volume Basis (a) | 80.0 | 79.8 | 79.6 | 79.4 | 79.0 | 78.8 | 78.7 | 68.8 | 68.6 |
| Experimental Percent Fill Corrected for (a) + Pressure Dilatation (b) | 79.9 | 79.7 | 79.5 | 79.2 | 78.9 | 78.7 | 78.5 | 68.8 | 68.6 |
| Experimental Percent Fill Corrected for (a) + (b) + Compressibility of $H_2O$ in Lines, Leads and Pressure Gauge (c) | 79.9 | 79.5 | 79.1 | 78.7 | 79.7 | 78.4 | 78.0 | 68.5 | 68.0 |
| Kennedy[13] Percent Fill for Measured $p$-T | 79.7 | 79.8 | 79.3 | 79.8 | 78.5 | 78.2 | 77.7 | 68.4 | 67.6 |

for Timken 17-22-AS suggested.[11] The correction owing to the compressibility of $H_2O$ in the tubing connecting the high-pressure gauges and to water in the Bourdon tubes of the gauges was made using the specific volumes of water as a function of $p$ and $T$, as tabulated in Landolt-Börnstein.[14]

In addition, it should be pointed out that Kennedy's p-V-T data[13] are given in terms of specific volume, $\bar{V}$, $cm^3/g$. Our experimental percent fill (as reported in the first row of Table II) is based on the quantity: volume $H_2O$/free volume of autoclave. The corrected fills shown in the fourth, fifth, and sixth rows of Table II are multiplied by the density of water at room temperature so that the fill in these rows is reported as gm $H_2O$/free volume of autoclave and can be directly compared with Kennedy's data (last row of Table II). As can be seen, the agreement between our corrected data and Kennedy's data is excellent, suggesting that our procedures are reasonable. For the convenience of those wishing to apply corrections at other conditions, Table III summarizes some useful data.

### 3.2 p-V-T behavior of $H_2O$-NaOH-$SiO_2$

Figures 1 and 2 show the p-T behavior of 1.0-mol NaOH saturated with $SiO_2$ at fills of 65, 75, 80, 83, 86, and 89 percent. The concentration of 1.0-mol NaOH was chosen as comparable to conditions used in commercial growth. Figure 1 shows the low-pressure region, while Fig. 2 covers higher pressures. The line $A$-$B$ is the coexistence curve. The fills shown on these figures are based on the initial cold volumes of solution and autoclave and no correction for temperature or pressure dilatation of the vessel is made to them. This is consistent with the definition of fill that is conventional in crystal growth. Such corrections could be made if desired using the data of Table III and the procedures outlined in the previous section and in Kolb and Laudise.[11]

Equilibrium was assured by holding at temperature until pressure stabilized, usually after a few hours. Pressures were not recorded until they were invariant for 12 or more hours. Pressures were observed to reproduce regardless of whether the equilibration temperature was approached from above or below. As a further check to ensure the 12-hour equilibration time was sufficient, a 1.0-mol NaOH run at 89-percent fill was equilibrated with quartz for one week at 214°C and then the temperature was abruptly raised to 303°C. The dependence of pressure on time in this experiment is shown in Fig. 3. As we can see, pressure equilibrium is again established after 250 min. It should be noted that the pressure (23,350 psi, 1610 bar) for 89-percent fill at 303°C for 250 min in Fig. 3 is the same as the pressure at 303°C in Fig. 2. Thus, 12-hour equilibrium times are more than sufficient.

The points in Figs. 1 and 2 at 65-percent fill were obtained in a

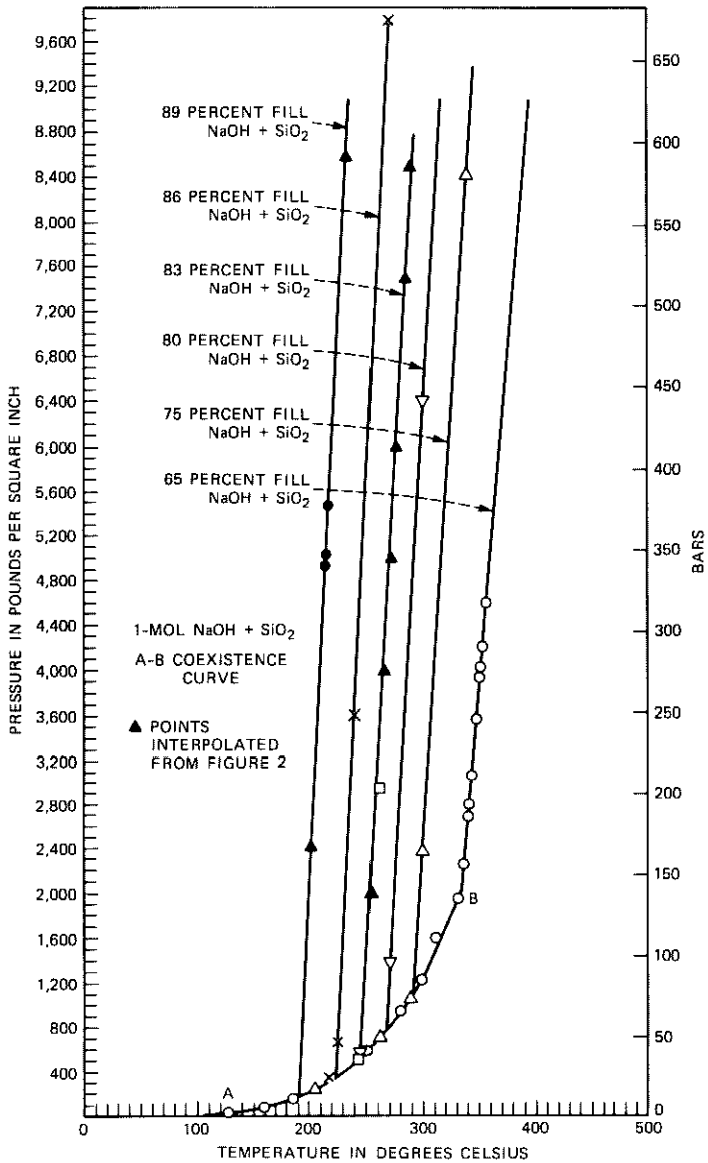



Fig. 1—p-T behavior of 1.0-mol NaOH saturated with quartz in a low-pressure region at several percent fills.

Morey autoclave using a 5000-psi gauge with a resolution of ±5 psi. These data are used to define the curve *A-B* where liquid and vapor saturated with quartz coexist. As we can see, at a particular fill once the autoclave has filled with liquid, the pressure departs from the coexistence curve and is linear with temperature; the slope of the *p-T*

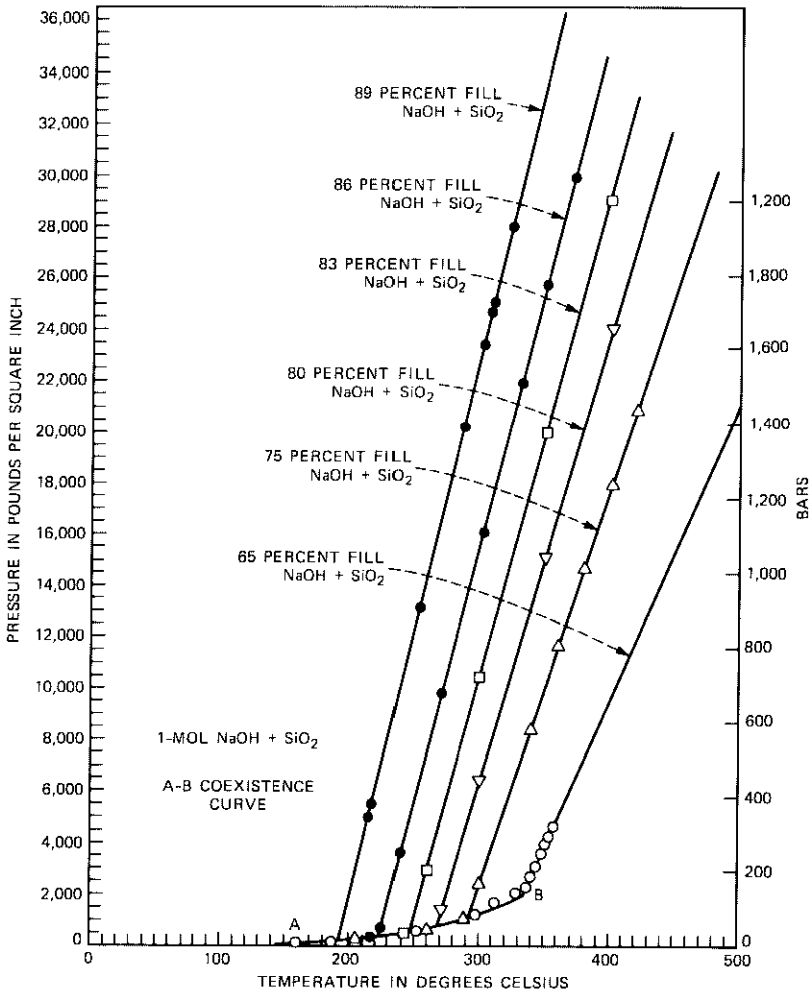



Fig. 2—p-T behavior of 1.0-mol NaOH saturated with quartz in a high-pressure region at several percent fills.

curves at constant-percent fill, $(\partial p/\partial T)_{\%f}$ is constant, and greater at higher-percent fills. Indeed, the data of Fig. 1 may be used to show the temperature at which the system departs from the coexistence curve as shown in Fig. 4.

The data of Figs. 1 and 2 may be displayed as in Fig. 5 to show the dependence of pressure on fill at constant temperatures. Similar to results on $H_2O$[13] and $H_2O$—$H_3PO_4$—$AlPO_4$,[11] the slope of these curves is not a constant.

### *3.3 p-V-T behavior of $H_2O$-NaOH*

The Pt-lined Morey vessel was used to determine pressures in NaOH solutions. Figure 6 shows typical results, which are compared to

Table III—Data for temperature and pressure expansion of autoclaves

| Temperature (°C) | Expansion Coefficient Timken 17-22-A (in/in°F) | $V/V_0^*$ | Young's Modules Timken 17-22-A* (psi) |
|---|---|---|---|
| 250 | $6.92 \times 10^{-6}$ | 1.0086 | $2.84 \times 10^7$ |
| 300 | $7.13 \times 10^{-6}$ | 1.0108 | $2.78 \times 10^7$ |
| 350 | $7.31 \times 10^{-6}$ | 1.0131 | $2.74 \times 10^7$ |
| 400 | $7.44 \times 10^{-6}$ | 1.0153 | $2.68 \times 10^7$ |
| 450 | $7.53 \times 10^{-6}$ | 1.0175 | |

* $V$ = Volume at temperature, $V_0$ = volume at 25°C, $\nu$ (Poisson's ratio) = 0.3, independent of temperature over this range.

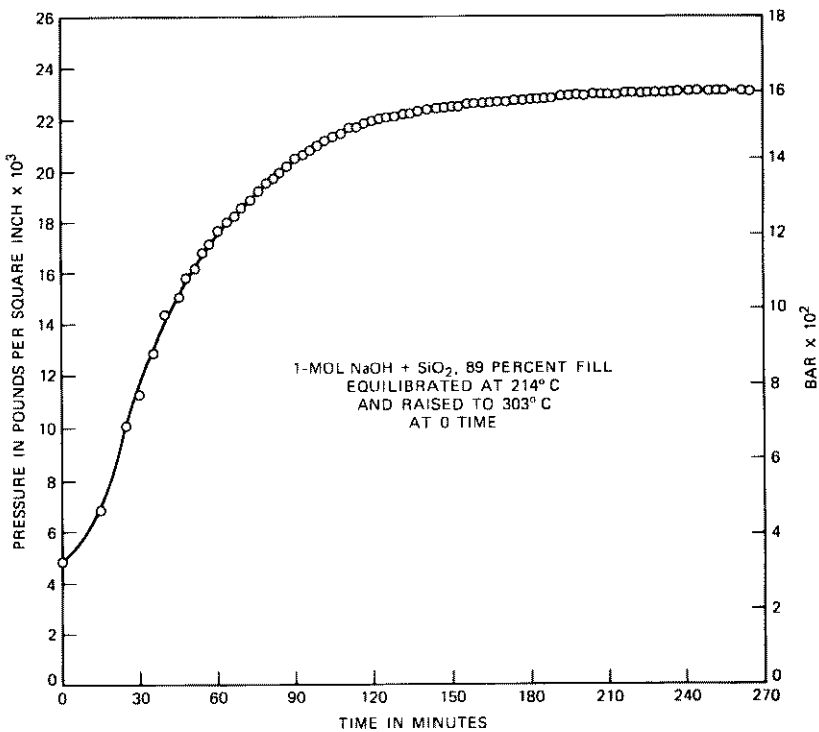



Fig. 3—Dependence of pressure on time when temperature of equilibrated vessel is increased from 214°C to 310°C (89-percent fill 1.0-mol NaOH initially saturated with quartz).

pressures of similar solutions saturated with quartz and with pure water. Some data from Samoilovich[15] and Kijama[16] are also plotted. However, the data of Kanahara, Yamasaki, and Matsuoka[9] for NaOH-$H_2O$ are not plotted. Their pressures for 1-mol NaOH are much lower than ours or those of Refs. 15 and 16. Liebertz[7] reports pressures for 4-mol NaOH, which are much higher than those of Kanahara et al.[9] but

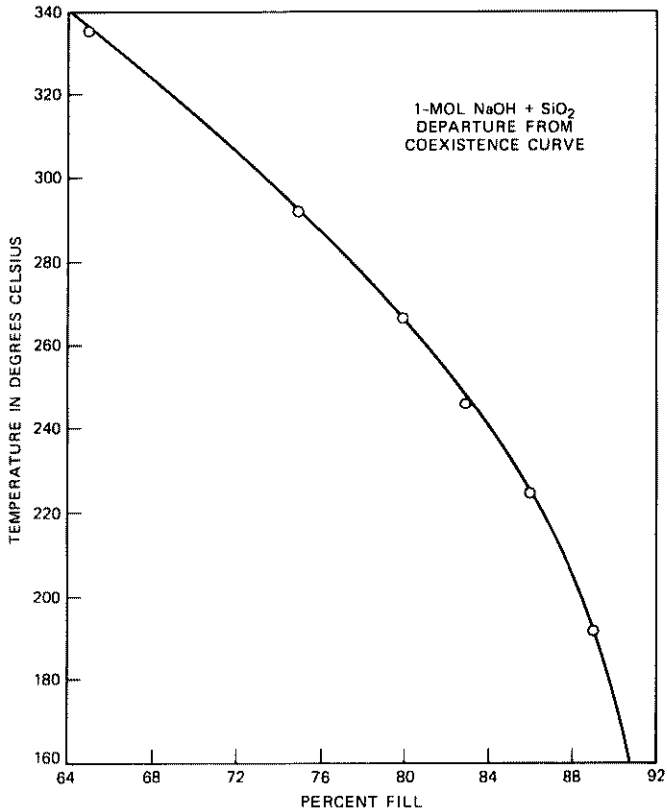



Fig. 4—Temperature at which autoclave fills for 1.0-mol NaOH saturated with quartz as a function of initial fill.

consistent with those of Refs. 15 and 16 at that concentration, suggesting a systematic error in (9).

As we can see in Fig. 6 our NaOH data are consistent with Refs. 15 and 16 obtained at slightly different NaOH concentrations. NaOH depresses the pressure along the coexistence curve and beyond, and $(\partial p/\partial T)_{\%f}$ is a constant in NaOH solutions above the coexistence curve.

When quartz is added to 1-mol NaOH, as shown in Fig. 6, the pressure is increased about 600 psi (40 bar) at 83-percent fill and more than 2000 psi (140 bar) at 65-percent fill. In previous work[2] we suggested silicate formation such as

$$2(OH)^- + SiO_2 \leftrightarrows SiO_3^{-2} + H_2O \quad (1)$$

for dissolving in $(CO_3)^{-2}$, and

$$2(OH)^- + 3SiO_2 \leftrightarrows Si_3O_7^{-2} + H_2O \quad (2)$$

for dissolving in $(OH)^-$. In eq. (1) the $(OH)^-$ is produced by $(CO_3)^{-2}$

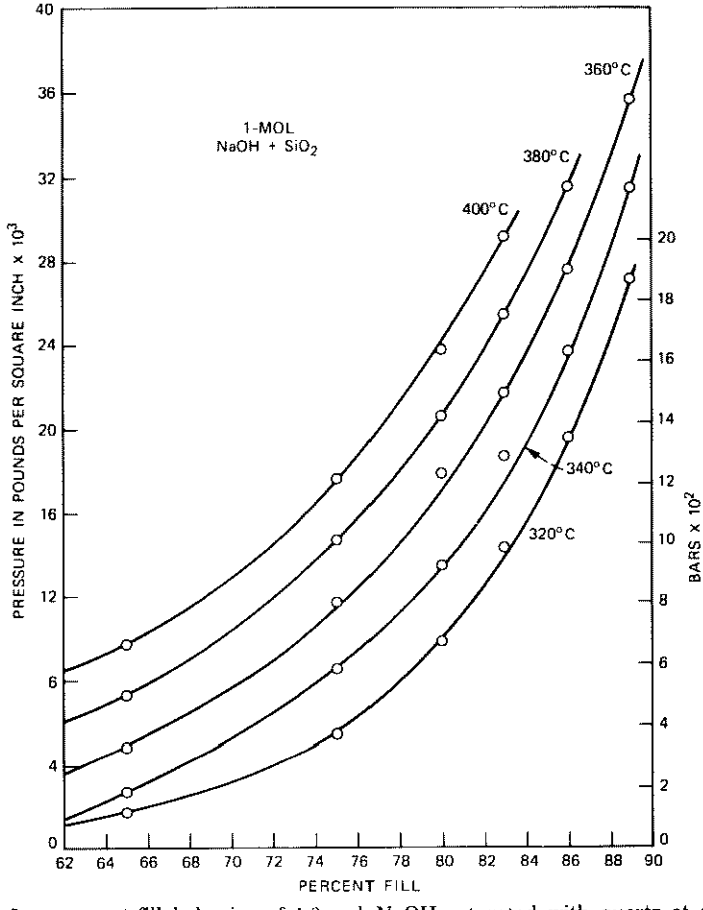



Fig. 5—*p*-percent-fill behavior of 1.0-mol NaOH saturated with quartz at several temperatures.

hydrolysis. Regardless of the details of such reactions it is likely that more than one $(OH)^-$ is consumed for each silicate ion formed. Consequently, the number of negative ions is decreased. If we make the reasonable assumption that the number and/or strength of the bonds of water molecules held in the first coordination shell by $2(OH)^-$ is greater than by a single $(SiO_3)^{-2}$, $(Si_3O_7)^{-2}$ or similar silicate then, as we observe in Fig. 6, the pressure will increase when $SiO_2$ dissolves in $(OH)^-$. The effect might be larger at lower fills because the $H_2O$ shell about an $(OH)^-$ is compressed more relative to bulk $H_2O$ when the $H_2O$ density is less.

## IV. RELATIONSHIP TO AUTOCLAVE EMBRITTLEMENT

Figure 7 shows the brittle-ductile transition temperature of some autoclave steels as a function of exposure time at quartz production

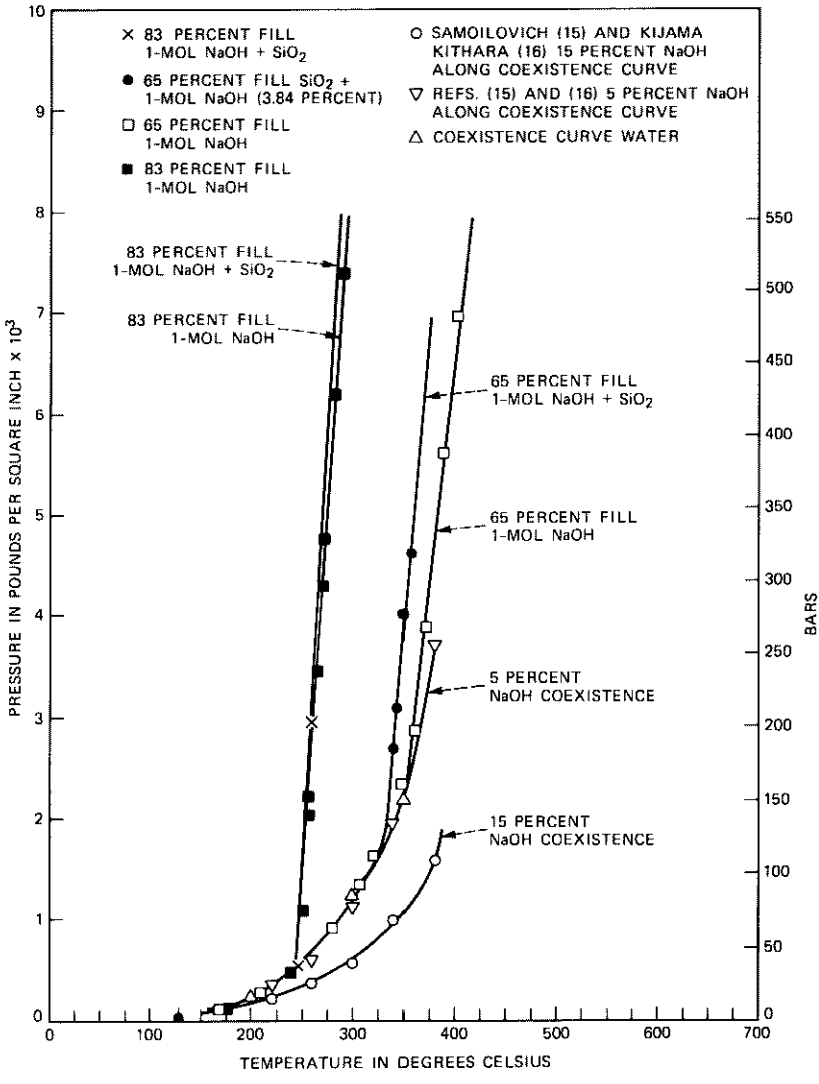



Fig. 6—p-T behavior of 1.0-mol NaOH compared to pure water and 1.0-mol NaOH saturated with quartz.

conditions (350°C crystallization temperature, 400°C nutrient temperature with the usual ~18-hr warm-up and ~24-hr cool-down and typical run time of ~30 days). Further identification of the steels is given in Ref. 12. The data of Fig. 7 were obtained by Charpy impact measurements.[12]

The autoclave operating conditions, primarily fill, should be chosen such that high pressure is not developed until the temperature is above the brittle range. On a practical basis, this requires that conditions be

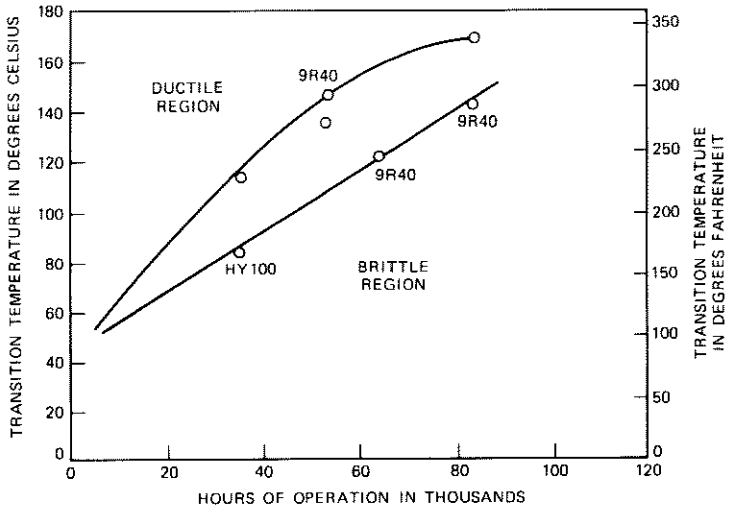



Fig. 7—Brittle-ductile transition of autoclave steels as a function of time at 350 to 400°C.

chosen using data in Figs. 4 and 7 so that pressures do not depart from the coexistence curve until the autoclave temperature is in the ductile region. For example, from Fig. 7, an autoclave with 80,000 hours of exposure (about 10 years at a typical 90-percent duty cycle) will be brittle for temperatures below 140 to 175°C. From Fig. 4, this suggests that fill should be limited to 90 percent. Thus, p-V-T data are very useful in establishing safe operating conditions and should be obtained for other mineralizers and conditions used in commercial hydrothermal processes.

## V. EFFECT OF TEMPERATURE DIFFERENTIALS

For practical crystal growth saturation occurs in a hotter, lower region of the autoclave containing relatively finely divided quartz nutrient, and growth takes place in an upper, supersaturated, cooler region containing seeds. To obtain information about the effect of temperature differentials on measured pressures a series of experiments in non-isothermal vessels were conducted. All experiments were done in 1.0-mol NaOH saturated with quartz at various fills. Figure 8 summarizes these results. We designate temperatures as follows:

$T_1$ = upper, cooler temperature region of autoclave

$T_2$ = midpoint temperature of autoclave

$T_3$ = lower, hotter temperature region of autoclave

$\Delta T_1 = T_3 - T_1$ ($T_2 \sim$ constant)

$\Delta T_2 = T_2 - T_1$ ($T_3 \sim$ constant).

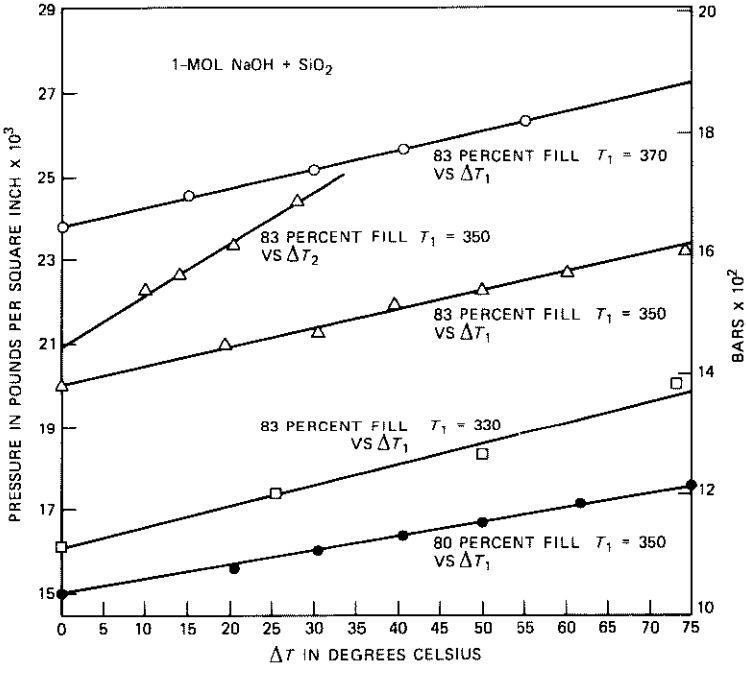



Fig. 8—p-T behavior of 1.0-mol NaOH saturated with quartz as a function of temperature as defined:

$$\Delta T_1 = T_{\text{Upper cooler region}} - T_{\text{Lower hotter region}} \quad \text{and}$$
$$\Delta T_2 = T_{\text{Mid}} - T_{\text{Upper cooler region}}.$$

As we can see, pressure is approximately linear with $\Delta T_1$ and $\Delta T_2$. Figure 8 can be used to estimate the correction necessary for converting isothermal data to $\Delta T$ data. It is interesting to point out that neither the hotter, lower region, upper, cooler region, nor average temperature determines pressure, but a fair fit can be obtained using an average weighted in favor of the upper, cooler temperature.

For example, consider Table IV, made from the data of Fig. 2 and Fig. 8, for 83-percent fill, 1-mol NaOH saturated with $SiO_2$. Row (a) is the upper, cooler temperature region of the autoclave ($T_1$); Row (b) the measured pressure; Row (c) the temperature difference, $\Delta T_1$; Row (d) the temperature, $T_4$, which gives the same pressure in an isothermal autoclave; Row (e) $\Delta T_3 = T_4 - T_1$; and Row (f) $\Delta T_3/\Delta T_1$. If the average temperature determined the pressure in a non-isothermal autoclave, then $\Delta T_3/\Delta T_1 = 0.5$. As we can see from Row (f), the average is weighted toward the upper, cooler temperature region and the weighting factor is reasonably constant.

The pressure is of course uniquely determined by a knowledge of local density and temperature anywhere in the vessel. For example, in

Table IV—Effect of temperature differentials

| | | | | | |
|---|---|---|---|---|---|
| (a) Upper Cooler Temperature Region of Autoclave ($T_1$) (°C) | 350 | 350 | 350 | 350 | 350 |
| (b) Pressure $P$ (psi) | 23,250 | 23,000 | 22,000 | 21,000 | 20,000 |
| (c) $\Delta T_1$ (°C) | 75 | 67.5 | 45.0 | 22.5 | 0 |
| (d) $T_4$ to Give $P$ in Isothermal Autoclaves (°C) | 368 | 367 | 361 | 356 | 350 |
| (e) $\Delta T_3$ (°C) | 18 | 17 | 11 | 6 | 0 |
| (f) $\Delta T_3/\Delta T_1$ | 0.240 | 0.252 | 0.244 | 0.267 | |

a vessel whose average percent fill is 83 percent, if the $\Delta T$ is 50°C with a top temperature of 350°C, the observed pressure is 22,180 psi (1530 bar). Using isothermal data for percent fill and taking density as percent fill/100 (Fig. 5), this suggests a local density of 0.85 g/cc for the region at 350° and 0.79 g/cc for the region at 400°. These densities are clearly subject to some errors since they are based on initial percent fills (Fig. 5) and do not include contributions attributable to dissolved solutes. Procedures of this sort might be used to calculate density gradients and hence be used to calculate the driving force for convective circulation in the system.

## VI. COMPARISON OF LABORATORY AND FACTORY MEASUREMENTS

Measurements of p-T were made on a production crystal-growth vessel filled with nutrient, 5-percent baffle, and seeds, using the heater placement and power inputs normal for commercial growth. The volume of the vessel was determined by filling with water to the corrosion mark indicating where the seal ring had rested in the previous run. Volume corrections were made for the leads in the cover. Capillary lead tubing and Bourdon pressure gauge were filled with water using procedures similar to those of the laboratory. The vessel was filled to a nominal fill of 83 percent with 1-mol NaOH and warmed from room temperature to operating conditions: 350°C (crystallization temperature), 400°C (nutrient region), $\Delta T$ 50°C in a period of 18 hours. During the warm-up temperatures and pressures were recorded and are shown in Fig. 9.

The results are compared with isothermal laboratory data for 83- and 86-percent fills. Careful examination of the ring-seating region at the conclusion of the run and calculations of the volume from the autoclave dimensions lead us to conclude that the true percent fill in the vessel was 84 ± 0.5 percent. On this assumption we can see that the pressures obtained are in reasonable agreement with isothermal laboratory data.

## VII. CONCLUSIONS

A reasonably rapid and convenient method for the measurement of the pressures of saturated hydothermal solutions of the sort used in

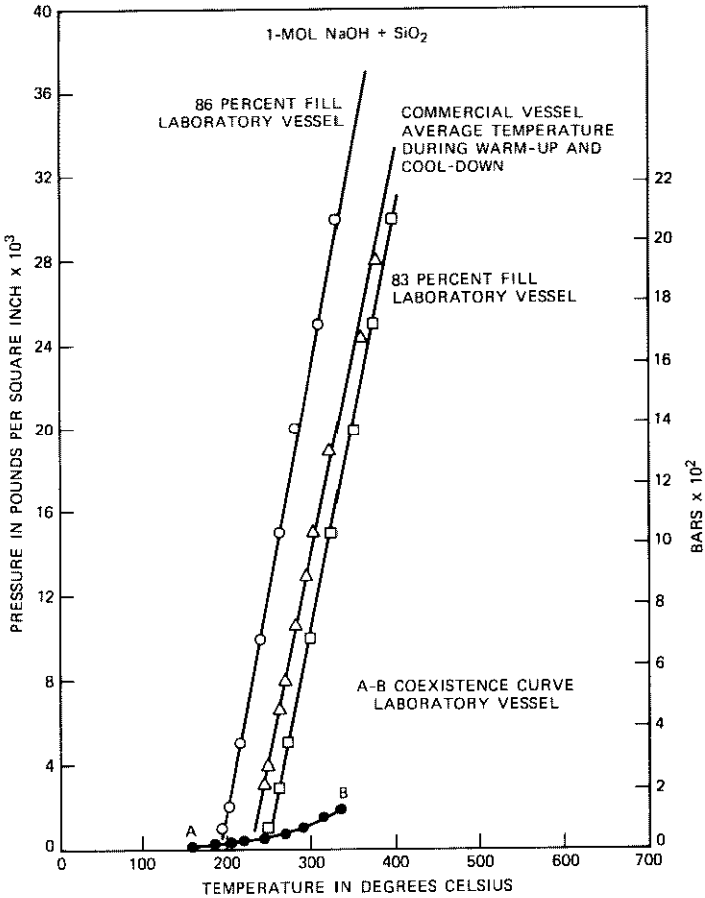



Fig. 9—Pressure-temperature of 1.0-mol NaOH plus quartz in commercial autoclaves during warm-up and cool-down.

crystal growth has been developed and applied to the determination of p-V-T relations in 1.0-mol NaOH and 1.0-mol NaOH saturated with quartz at degrees of fill from 65 to 89 percent, pressures up to ~30,000 psi (2070 bar), and temperatures up to ~450°C. The accuracy of our experimental procedures was tested in measurements on pure water where published pressure data were reproduced. Temperature and pressure dilatation effects are estimated.

For both 1-mol NaOH and NaOH saturated with quartz, $(\partial p/\partial T)_{\%f}$ is constant above the coexistence curve. Along the coexistence curve and above, the pressure of 1-mol NaOH is less than the pressure of water but the pressure of 1-mol NaOH saturated with quartz is greater than the pressure of 1-mol NaOH. The increase of pressure when quartz dissolves in 1-mol NaOH is explained on the basis of a decrease

in the number of ions in solution when $(OH)^-$ reacts with $SiO_2$ to form silicates.

After about 10 years of service the brittle-ductile transition temperature of autoclave steel increases to ~170°C so that substantial pressures should be avoided below that temperature. The p-V-T data are used to show the relationship between initial fill and temperature at which the pressure departs from the coexistence curve and $(\partial p/\partial T)_{\varphi_f}$ becomes large. These data may be used as a guide for the choice of safe operating conditions.

Pressure measurements were made on autoclaves with temperature differentials and the data were related to pressure measurements on isothermal vessels. A weighted average temperature (weighted in favor of the upper, cooler region) for the temperature differential reproduced the isothermal measurements. On the assumption that local density and temperature determine pressure, p-V-T data can be used to estimate density differences associated with typical temperature differences ($\Delta T$). A $\Delta T$ of 50° at 350°C growth temperature for 83-percent fill 1-mol NaOH saturated with quartz produces a density differential of ~0.06 g/cc.

Finally, pressure measurements were made on large-scale factory autoclaves under conditions of commercial growth and these measurements are compared to laboratory results.

## VIII. ACKNOWLEDGMENTS

We would like to thank A. J. Caporaso for suggestions and assistance and especially for help with the design and fabrication of Pt equipment.

## REFERENCES

1. G. W. Morey and J. M. Hesselgesser, Amer. J. Sci., Bowen Volume, *343* (1952); see also, "Phase Diagrams for Ceramists," Ed. by E. M. Levin, H. F. McMurdie, and F. P. Hall, Columbus, Ohio: American Ceramic Society, 1956, p. 259.
2. R. A. Laudise and A. A. Ballman, "The Solubility of Quartz Under Hydrothermal Conditions," J. Phys. Chem., *65* (1961), pp. 1396–1400.
3. R. A. Laudise, "Kinetics of Hydrothermal Quartz Crystallization," J. Am. Chem. Soc., *81* (1959), p. 562.
4. N. Lias, E. Grudenski, E. Kolb, and R. A. Laudise, "The Growth of High Acoustic Quartz at High Growth Rates," J. of Crystal Growth, *18* (1973), pp. 1–6.
5. R. L. Barns, P. E. Freeland, E. D. Kolb, R. A. Laudise, and J. R. Patel, "Dislocation-Free and Low-Dislocation Quartz Prepared by Hydrothermal Crystallization," J. of Crystal Growth, *43* (July 1978), pp. 676–86.
6. Proc. 35th Annual Symp. on Frequency Control, Electronic Industries Assoc., 2061 Eye St., Washington, D.C., May 27–29, 1981.
7. J. Libertz, "P, T, f-Werte von 4 n $K_2CO_3$-, $Na_2CO_3$-und NaOH-Losungen unter Hydrothermalen Bedingungen," Chem. Eng. Tech., *41* (1969), p. 1231.
8. L. A. Samoilovich, "The Pressure of Hydrothermal Solutions," Trudy VIII, Soveschaniya, Experimental. Tekhnich. Mineral. i Petrografi, Moscow (1971), p. 195 (See also Refs. 15 and 16).
9. S. Kanahara, N. Yamasaki, and K. Matsuoka, "Effect of Various Solute Concentration on Hydrothermal P-T Curves. II. NaOH," Rept. Res. Lab. Hydrothermal Chemistry, Kochi, Univ. Japan, *2* (8–15), (1978), p. 30.

10. R. A. Laudise and R. A. Sullivan, "Pilot Plant Production Synthetic Quartz," Chemical Engineering Progress, *55,* No. 5 (May 1959), pp. 55–9.
11. E. D. Kolb and R. A. Laudise, "Pressure-Volume-Temperature Behavior in the System $H_2O$-$H_3PO_4$-$AlPO_4$ and Its Relationship to the Hydrothermal Growth of $AlPO_4$," Journal of Crystal Growth, *56,* No. 1 (January 1982), pp. 83–92.
12. P. L. Key, E. D. Kolb, R. A. Laudise, and E. E. Bresnahan, "Embrittlement of Steels Used in Autoclaves for Hydrothermal Crystallization," J. of Crystal Growth, *21* (1974), pp. 164–6.
13. G. C. Kennedy, "Pressure-Volume-Temperature Relations in Water at Elevated Temperatures and Pressures," American Journal of Science, *248* (1950), p. 540.
14. Landolt-Börnstein, New Series, K. H. Hellwege, Ed., Group IV, Vol. 4 "High Pressure Properties of Matter," Kl. Schafer, New York: Springer, 1980, p. 89.
15. L. A. Samoilovich, "Relationship Between Pressure, Temperature, and Density of Aqueous Sodium Hydroxide Solutions," Acad. of Sci USSR, Inst. of Geol. of Ore Deposits, Petrography, Mineralogy and Geochemistry, VIII Conf. on Exptl. and Appl. Mineralogy and Petrography, Proceedings, Moscow: Nauka Publishing House, 1971, pp. 195–200.
16. R. Kijama and S. Kitahara, "S - P-F-T Relations of Sodium Hydroxide and Its Causticity at High Temperatures and Pressures," Review Phys. Japan, *27* (1957).

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# Variable Rate ADPCM Based on Explicit Noise Coding

By N. S. JAYANT

(Manuscript received July 15, 1982)

*This paper discusses a variable bit rate speech coding system based on explicit coding of the reconstruction noise in ADPCM (differential pulse code modulation with adaptive quantization). If the ADPCM bit rate is* R *bits/sample, PCM coding of its noise using an average bit rate of* $R_n$ *bits/sample provides the receiver with the possibility of operating at any bit rate in the range* R *to* $R + \max\{R_n\}$. *Using* R *values in the range 2 to 5, and* $R_n$ *values in the range 0 to 3, we compare the performance of the* $(R + R_n)$*-bit system with that of conventional* $(R + R_n)$*-bit ADPCM. If noise coding is based on instantaneous* $R_n$*-bit quantization of its samples with an optimized step size, the signal-to-noise ratio performance is comparable to that of conventional ADPCM for* $R_n = 1$, *but it deteriorates significantly for* $R_n > 1$. *With non-instantaneous noise coding, the performance can exceed that of conventional ADPCM for any* $R_n > 1$, *if* $R > 2$. *This is due to a variable bit allocation algorithm that quantizes noise samples with differing resolutions, while maintaining a constant total bit rate in every block of 4 ms. The algorithm does not require the transmission of any extra side information. It can also be regarded as a way of improving the performance of ADPCM coding at a single bit rate of* $R + R_n$ *bits/sample.*

## I. INTRODUCTION

Multiple-stage coding, where the reconstruction noise from an initial stage is itself coded for transmission in a subsequent stage, is known to provide substantial gains over single-stage coding in the context of deltamodulation using oversampled inputs.[1,2] In this paper, we consider two-stage systems for multibit differential pulse code modulation with adaptive quantization (ADPCM) coding of Nyquist-sampled speech inputs. Unlike systems that permit oversampling, signal-to-noise ratio

(s/n) gains in our systems will be seen to be either slightly negative, or positive but nondramatic. However, the proposed systems have a feature that is common to all noise-coding systems, the property of embedded coding: the output bit sequence of the coder contains a subsequence that can be used in a straightforward manner to provide lower bit-rate operation with an output speech quality very close to that of conventional operation at the lower bit rate; as a result, the channel or receiver can switch, as needed, between low-rate and high-rate modes. The possibility of *variable-rate* operation is a very desirable feature in digital communication systems such as packet-switched voice networks.[3] A PCM coder is inherently an embedded coder. Least significant bits in a PCM codeword can be progressively dropped, with a graceful loss of quality that is no greater than about 6 dB/bit. Conventional differential pulse code modulation (DPCM) is not an embedded coding system in a similar sense because of the presence of a feedback loop in coder and decoder.

Explicit noise coding is not the only way of designing an embedded ADPCM system. Coarse feedback in the DPCM predictor loop[4,5] is known to provide a very robust basis for embedded DPCM, with very little s/n degradation compared to conventional DPCM at a given bit rate; and the results are also expected to extend to DPCM with an adaptive quantizer. In the coarse-feedback approach, the encoder performs an appropriate quantization of predictor input in anticipation of a similar quantization that may be forced at the receiver as a result of bit-dropping. The coarse feedback embedded system can also drop more than one bit, in a progressive fashion, to provide a wide range of bit rates. The noise-coding approach provides zero degradation of quality at the lower bit rate, $R$. More important, explicit noise coding offers the possibility of complex versions of $(R + 1)$-bit ADPCM that can provide positive performance gains over conventional $(R + 1)$-bit coding. ADPCM with variable bit allocation (Section V) is one example of such a complex system. The noise-coding system with variable bit allocation can also be used as a *single-rate* coder in which the coding process is split into two steps (conventional ADPCM followed by noise coding) to permit a simple form of time-domain bit allocation for the improvement of ADPCM performance.

Figure 1 is a block diagram of a variable-rate coder employing optional coding of ADPCM noise, with an average noise-coding rate of $R_n$ bits/sample. The special case of $R_n = 1$ is treated at length in Sections IV through VII. When the dashed boxes for bit allocation are eliminated, instantaneous noise-coding results, with a coding rate of exactly $R_n$ bits for every noise sample. When the parts of the system within boxes $A$ or $B$ are eliminated, $R_n = 0$, and conventional single-rate ADPCM results, with a total bit rate of $R$ bits/sample. The

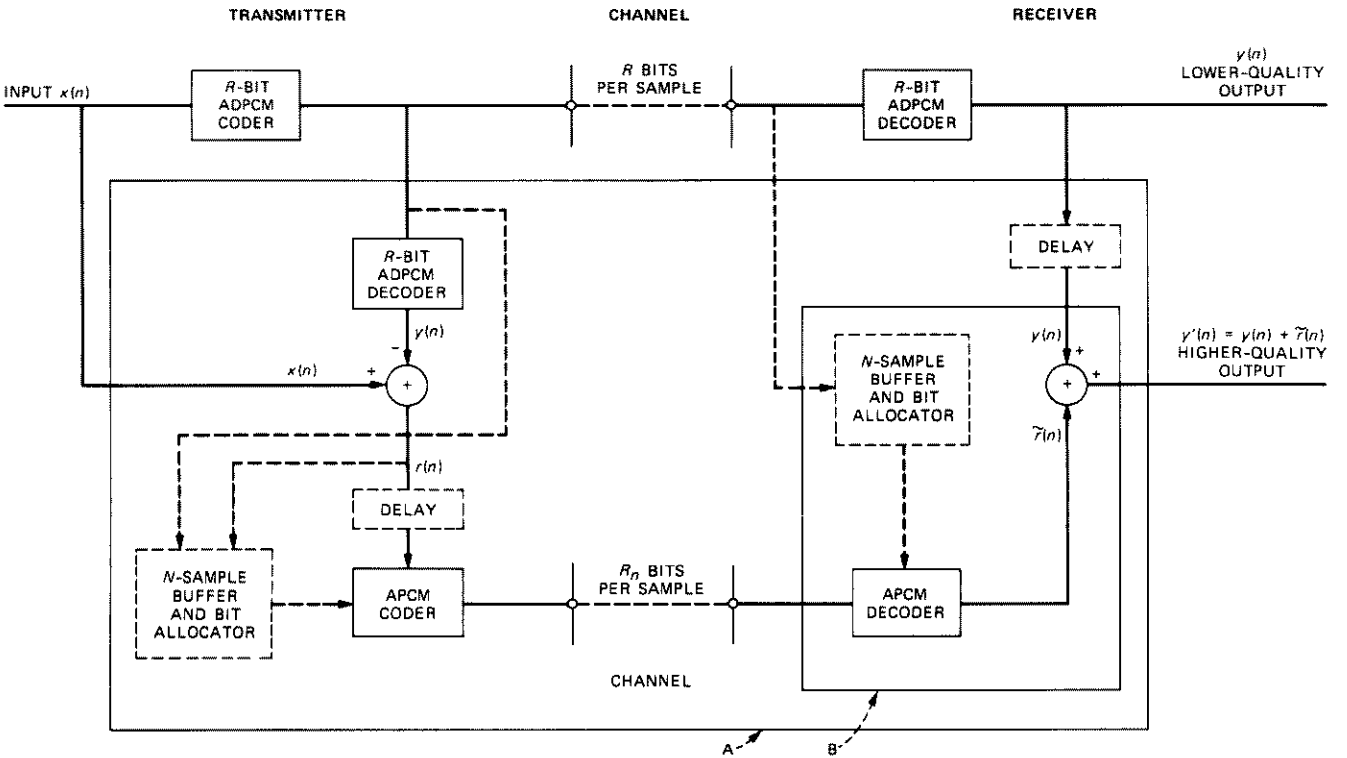



Fig. 1—Block diagram of a variable-rate ADPCM system. Variation of the average noise-coding bit rate $R_n$ results in a variable-rate system with a total bit rate that ranges from $R$ to $R + \max\{R_n\}$ bits/sample.

extreme upper part of the figure (outside of box $A$) shows a conventional $R$-bit ADPCM coder-decoder.[6] The rest of the diagram (the part included in box $A$) shows the blocks that perform $R_n$ bit/sample coding of the reconstruction noise samples

$$r(n) = x(n) - y(n), \tag{1a}$$

where $x(n)$ and $y(n)$ are the input and the output of an $R$-bit/sample ADPCM system. When the system of Fig. 1 includes noise coding, the final decoded value is $y'(n)$, a refinement of the conventional value $y(n)$:

$$y'(n) = y(n) + \tilde{r}(n). \tag{1b}$$

The total bit rate of the system in Fig. 1 is $R + R_n$ bits/sample. Variable-rate coding results from the use of different values of $R_n$. Examples in this paper cover the range of $0 \leq R_n \leq 3$. The case of $R_n = 1$ is discussed at length before generalization to $R_n > 1$. With $R_n = 1$, the variable-rate system of Fig. 1 reduces to a dual-rate system, with a total bit rate of either $R$ or $R + 1$ bits/sample.

The noise information can be altogether eliminated by the system ($R_n = 0$) to provide conventional $R$-bit operation. Alternatively, the noise information may be eliminated, as necessary, by the channel or receiver. If the receiver does the elimination, the part of the system that is eliminated is that within box $B$.

The results of this paper are based on simulations with three sentence-length utterances: "The *c*hairman cast three votes" (*f*emale speaker); "A *l*athe is a big tool" (*f*emale speaker); and "A *l*athe is a big tool" (*m*ale speaker). These speech inputs are identified in the rest of this paper as *CF*, *LF*, and *LM*. All inputs are band-limited to the frequency range 200 to 3200 Hz.

## II. SUMMARY OF THE ADPCM AND APCM CODERS

The ADPCM coder in this paper uses first-order prediction with a time-invariant prediction coefficient of 0.85. It also uses an adaptive quantizer with a one-word memory.[7] As Fig. 2 shows for the examples of $R = 2$ and $R = 3$, the (uniform mid-rise) quantization characteristic $Q(x)$ is multiplicatively expanded or compressed at every sampling instant by a factor (step-size multiplier $M$) that depends only on the magnitude of the most recent quantizer output $y(n-1)$. If $\Delta(n)$ is the quantizer step size at time $n$,

$$\Delta(n) = M(|y(n-1)|) \cdot \Delta(n-1). \tag{2}$$

The function $M$ takes on one of $2^{R-1}$ values in $R$-bit ADPCM. Recommended multiplier sets for $R = 2$ and $R = 3$ are included in Fig. 2. Recommended multipliers for $R = 4$ and $R = 5$ are tabulated in Ref.

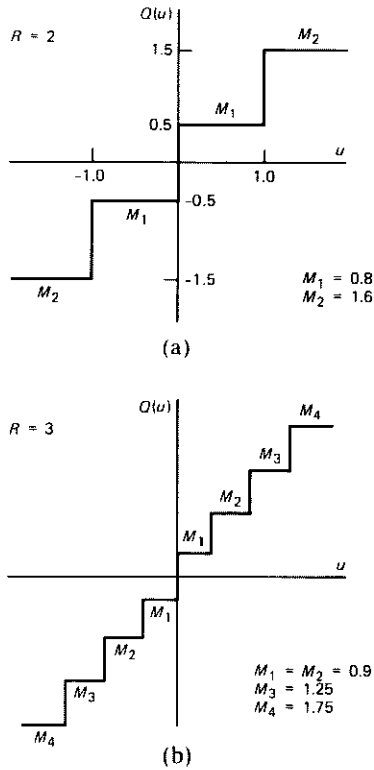



Fig. 2—Step-size multipliers used in (a) 2-bit and (b) 3-bit adaptive quantizers.

7. In the examples of Fig. 2, the use of the largest step-size multiplier also indicates the use of the outermost quantizer levels.

The adaptive PCM (APCM) coder for noise samples $r(n)$ will be described in detail in Sections IV, V, and VIII. The adaptive step size for the APCM coder will be seen to follow that of the quantizer in $R$-bit ADPCM. The purpose of the $N$-sample buffers in Fig. 1 is to permit a variable-bit-allocation procedure (Sections V and VIII) that provides a higher quality of noise quantization than what is possible with instantaneous quantization, the case of $N = 1$ (Section IV). When variable-bit allocation is employed, $R_n$ will be interpreted as the *average* bit rate for noise coding. But the total number of noise-coding bits will be guaranteed to be a constant value, $NR_n$, for every block of $N$ noise samples. The variable-bit allocation is first explained for the case of $R_n = 1$, implying noise coding with an average bit rate of 1 bit/sample (Section V). Extension to the case of $R_n > 1$ is straightforward (Section VIII).

## III. RECONSTRUCTION ERROR $r(n)$

Figure 3a is a 16-ms-long speech segment from *CF*, and Fig. 3b illustrates the reconstruction-error waveform $r(n)$ in ADPCM for the example of $R = 4$. An important property is that $r(n)$ has occasional impulsive components. These are the slope-overload error bursts typical of DPCM with non-adaptive prediction. The extent of slope overload increases with coarseness of quantization. But, as seen in Fig. 3b, slope overload is quite evident even with $R = 4$ and adaptive quantization. In voiced speech, the time separation between slope overload bursts corresponds very closely to the pitch period. In the example of Fig. 3, this separation is about 40 samples (at 8 kHz), corresponding to a pitch period of about 200 Hz. Figures 3c and 3d will be discussed in Sections IV and V.

During slope overload, the noise samples $r_O(n)$ will have magnitudes in the range

$$0 \leq |r_O(n)| < \infty. \tag{3}$$

The limit $\infty$ can be replaced by a more meaningful finite value if the input is bounded, as in band-limited speech. But this won't be necessary for the purposes of this paper.

The non-impulsive background in the $r(n)$ waveform is associated with input samples that do not cause slope overload. In this granular noise region, the maximum magnitude of noise sample $r_g(n)$ is simply half the ADPCM step size:

$$0 \leq |r_g(n)| \leq \Delta(n)/2. \tag{4}$$

## IV. THE $(R + 1)$-BIT CODER WITH INSTANTANEOUS ONE-BIT QUANTIZATION OF $r(n)$

From the theory of one-bit quantization, the reconstruction level $\delta(n)$ that provides the minimum mean square error with a one-bit noise quantizer is given by the mean absolute value of quantizer input:

$$\delta(n)_{\text{opt}} = E[|r(n)|]. \tag{5}$$

Ignoring slope-overload samples $r_O(n)$, and assuming that the magnitudes of the $r_g(n)$ samples are uniformly distributed in the range 0 to $\Delta(n)/2$,

$$\delta(n)_{\text{opt}} \sim E[|r_g(n)|] = \frac{1}{2} \cdot \frac{\Delta(n)}{2} = \frac{\Delta(n)}{4}. \tag{6}$$

Simulations have shown that the probability of slope overload is small enough for the above design to be indeed very close to the optimum. This is illustrated by the s/n versus $\delta(n)$ plots in Fig. 4 for $R = 4$, 3, and 2 bits/sample. The signal-to-noise ratio is maximum

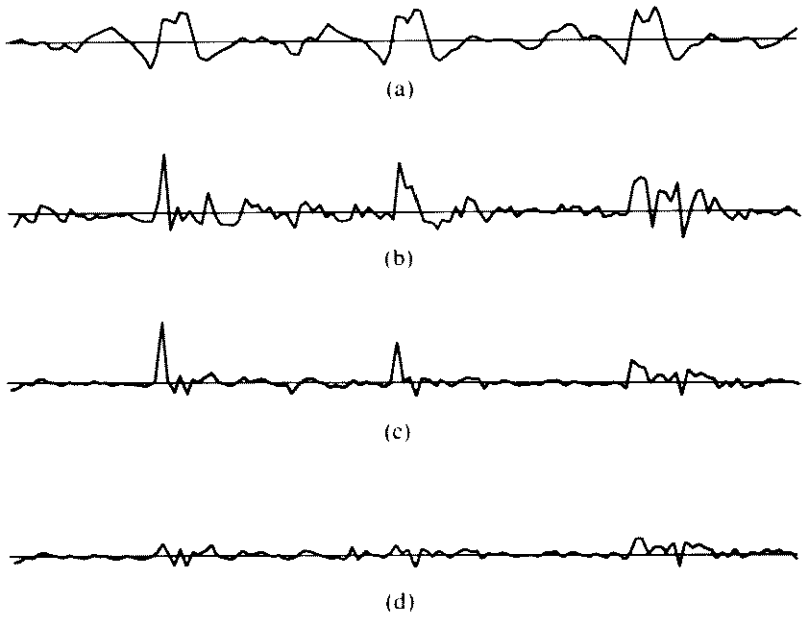



Fig. 3—(a) Input speech $x(n)$ (taken from $CF$) and reconstruction-error waveforms (b) (c) (d) in three ADPCM systems. All waveforms are 128 samples (16-ms) long. All error amplitudes are magnified by a factor of 10.

when the reconstruction level magnitude $\delta(n)$ of the 1-bit noise quantizer equals one-fourth the corresponding step-size $\Delta(n)$ in the $R$-bit ADPCM coder. When $\delta(n) = 0$, the system degenerates to the original $R$-bit ADPCM. Values at $\delta(n) = 0$ show the s/n of $R$-bit ADPCM.

Figure 3c shows the residual error after the $r(n)$ waveform (Fig. 3b) has been instantaneously quantized with a 1-bit/sample quantizer with reconstruction levels of $\pm\delta(n)_{\text{opt}}$. Note that the granular background components in $r(n)$ are uniformly reduced, but slope overload components are not.

The above step-size design implies that the noise coder is an instantaneous adaptive PCM (APCM) device that derives its step size from information that is already available in the $R$-bit ADPCM part of the $(R + 1)$-bit system. The $N$-sample buffer in Fig. 1 is not necessary for the operation of the instantaneous APCM coder.

The performance of the $(R + 1)$-bit system with instantaneous quantization is discussed at length in Section VI.

## V. THE $(R + 1)$-BIT CODER WITH NON-INSTANTANEOUS ONE-BIT QUANTIZATION OF $r(n)$

Elimination of the impulsive components in $r(n)$ requires finer quantization. We now propose an algorithm that indeed allocates

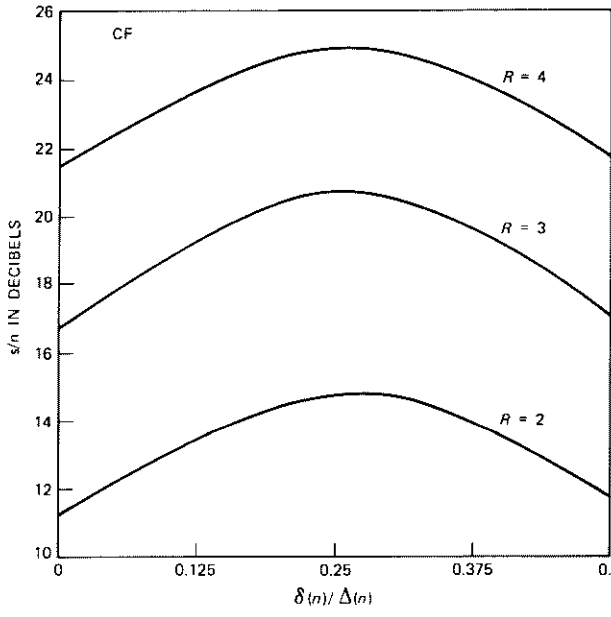



Fig. 4—Signal-to-noise ratio of $(R + 1)$-bit ADPCM system with instantaneous 1-bit quantization of reconstruction noise $r(n)$ from $R$-bit ADPCM.

$R_O > 1$ bit for slope-overload components, but still maintains an average bit rate of exactly 1 bit/sample in every block of $N$ samples of $r(n)$. This is accomplished by assigning $R_g = 0$ bit/sample for granular noise components of very low magnitude in the block. The $N$-sample buffers and $N$-sample delays in Fig. 1 will be used to effect the above variable-bit assignment.

The location of slope-overload noise samples $r_O(n)$ and that of the low-magnitude granular noise samples $r_g(n)$ are both based on information that is already available to the $R$-bit ADPCM receiver, and therefore require no further side information to be transmitted.

The slope-overload samples are determined as those for which the quantizer output in $R$-bit ADPCM reaches the highest possible values for the given value of $R$ (for example, levels associated with multiplier $M_2$ with $R = 2$ and levels associated with multiplier $M_4$ with $R = 4$; see Fig. 2).

The low-magnitude granular noise samples are located by rank-ordering $\Delta(n)$ values in the $N$-block, and by assigning zero bits to as many of these samples as necessary, in order of increasing $\Delta(n)$, until the total number of bits in the block is exactly $N$. While picking these zero-bit samples, it is very important to exclude samples associated with the use of highest output level. This precaution is needed because slope-overload errors can be associated with small values of $\Delta(n)$ as

well as large ones. In fact, as mentioned in the last paragraph, a defining cue for slope overload is not the value of the current *step size*, but rather the value of the current ADPCM quantizer *output level* (or current step-size *multiplier* if output levels and multiplier values have a one-to-one mapping, as in Fig. 2).

The net result of the above procedure will be to assign $R_O > 1$ bits where noise magnitudes are guaranteed to be the highest, and to assign $R_g = 0$ bits where noise magnitudes are guaranteed to be the smallest. The remaining samples are assigned 1 bit/sample as in Section IV.

The variable-bit-rate algorithm follows the constraint that the total number of bits per block is $N$:

$$N_O \cdot R_O + (N - N_O - N_g) \cdot 1 + N_g \cdot 0 = N; \qquad N_g = N_O(R_O - 1), \quad (7)$$

where $N_O$ and $N_g$ are the numbers of slope-overload and low-magnitude granular samples in $N$ samples of $r(n)$. Note that the constraint above also implies that

$$N_O \cdot R_O = N_O + N_g \leq N; \qquad N_O \leq N/R_O. \quad (8)$$

This latter constraint on $N_O$ is explicitly enforced even in those cases where the number of maximum multiplier samples may exceed $N/R_O$, for a chosen $R_O$.

The design of $R_O$ should reflect the probability of use of the maximum reconstruction level in the $R$-bit ADPCM coder. This probability controls the fraction $N_O/N$. As shown in Ref 7, this probability is a decreasing function of $R$; consequently, the maximum allowable value of $R_O$ that does not violate (8) is an increasing function of $R$. In fact, in our experiments, we have found that for $N$ values of interest, the s/n maximizing values of $R_O$ happen to be very close to the number of bits/sample $R$ in the basic ADPCM coder. Thus, for example, the slope-overload bursts in 3-bit ADPCM are quantized with a second stage of APCM coding with an appropriately designed 3-bit quantizer.

### 5.1 Design of noise-quantizer characteristic

Figure 5 illustrates quantizer characteristics that were experimentally found to provide nearly minimum mean square error in noise quantization. The smallest outputs in each of these characteristics are the $\pm\Delta(n)/4$ levels used in the instantaneous noise quantizer of Section IV. The largest output levels are $\pm\Delta(n)$ and $\pm 3\Delta(n)$ in the non-instantaneous quantizers for $R = 2$ and 3. For $R = 4$, the largest output levels in the noise quantizer will be $\pm 7\Delta(n)$. All these numbers obviously depend only on $\Delta(n)$, a value already available to the $R$-bit ADPCM receiver.

In one experiment with $N = 32$ and input *CF*, the number $N_O$ of $r(n)$ samples coded with $R_O > 1$ bits/sample were 2, 3, 9, and 15,

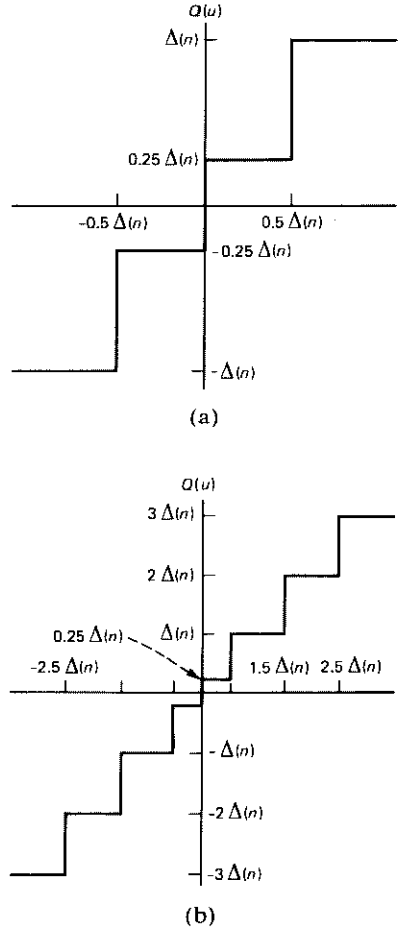



Fig. 5—Quantization characteristics used for overload noise samples $r_O(n)$ in the non-instantaneous coding of ADPCM noise when the average rate for noise coding is $R_n$ = 1 bit/sample. The ADPCM bit rate $R$ is 2 in (a) and 3 in (b).

respectively, with 5, 4, 3, and 2-bit DPCM. These numbers reflect the much higher probability of using the maximum quantizer output level as $R$ decreases. With the recommended design $R_O = R$, note that $N_O \cdot R_O < N = 32$ in all the four examples above, as required in (8). With $N = 128$ and the same input $CF$, values of $N_O$ were 5, 9, 19, and 32, respectively.

### 5.2 Design of block length N

The buffer length $N$ should be large enough so that for every noise sample coded with $R_O > 1$, there is an adequate selection of noise samples for which bit stealing ($R_g = 0$) is appropriate. However, the

quasi-periodic nature of slope-overload bursts (Fig. 3b) indicates that $N$ need be no greater than the pitch period. This is indeed demonstrated in Fig. 6, which plots the signal-to-noise ratio of the $(R + 1)$-bit system as a function of $N$. Note that the performance at $N = 32$ (which is close to the pitch period 40 in Fig. 3 for *CF*) is very close to that at $N = 128$. Note also that the gain with $N = 32$, over the instantaneous quantization scheme of Section IV (the case of $N = 1$), is over 2 dB. Gains over $N = 1$ are less in the case of $R = 2$.

Figure 3d shows the residual error after $r(n)$ has been quantized with an average rate of 1 bit/sample, with $N = 32$ (a buffer length of 4 ms, with 8-kHz samples). Note that unlike the instantaneous quantization scheme of Fig. 3c, even the impulsive components in $r(n)$ have been nearly eliminated in Fig. 3d. This is a result of quantizing these components with $R_O > 1$ bits/sample; $R_O = 4$ in this example. Since the impulsive components of the noise waveform $r(n)$ tend to occur predominantly during pitch-period onset, the system with non-instantaneous quantization can also be regarded as a form of "pitch-compensated" quantization.[8]

## VI. SIGNAL-TO-NOISE RATIO RESULTS FOR *R*-BIT AND $(R + 1)$-BIT CODERS

Figures 7 and 8 compare the performance of the coders of Sections IV and V with that of conventional single-stage ADPCM.

The signal-to-noise ratios are averages over the entire length of a given utterance. The segmental s/n is obtained by obtaining the signal-

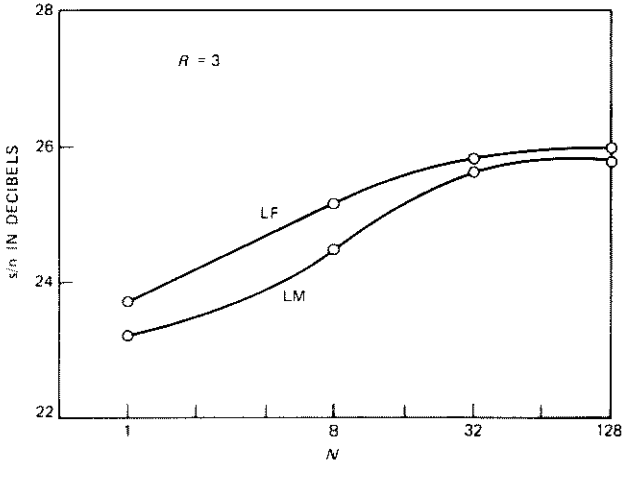



Fig. 6—Signal-to-noise ratio $(R + 1)$-bit ADPCM system with variable-rate quantization of reconstruction noise $r(n)$ from $R$-bit ADPCM. The signal-to-noise ratio reaches a value close to the maximum with a noise-buffer length of $N = 32$ (encoding delay of 4 ms). The gain over instantaneous noise quantization $(N = 1)$ is in excess of 2 dB.

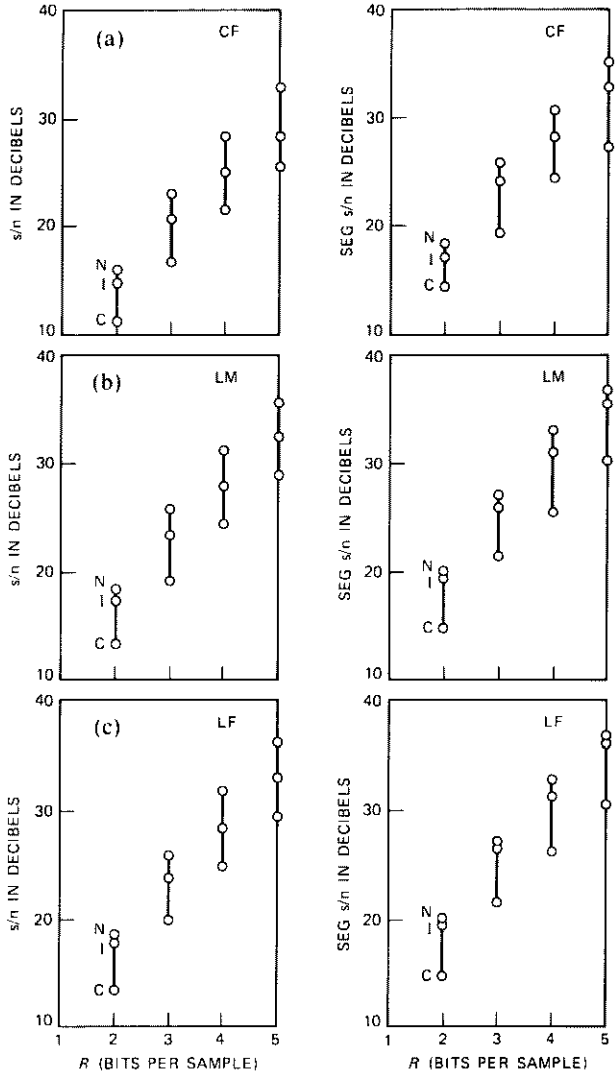



Fig. 7—Signal-to-noise ratio (s/n) and segmental signal-to-noise ratio (SEG s/n) in ADPCM systems, as a function of bit rate $R$ of the basic coder in Fig. 1. For each value of $R$, there is an ordered set of three s/n or SEG s/n values. Plots in (a), (b), and (c) refer to speech inputs *CF*, *LM*, and *LF*.

to-noise ratio in dB for each 16-ms segment of an input, and by averaging such dB values over the entire length of a given utterance.

Figure 7 shows, for each bit rate $R$ of the conventional ADPCM system ($C$), signal-to-noise ratio gains in $(R + 1)$-bit systems with *i*nstantaneous ($I$) and *n*on-instantaneous ($N$) quantization of $r(n)$, with a total of 32 bits of quantization in every 32-sample block of $r(n)$.

Note that except in the case of $R = 2$, the performance of the $(R + 1)$-bit system with instantaneous quantization (the middle point on each vertical bar) is very close to conventional $(R + 1)$-bit ADPCM (the lowest point on the next vertical bar to the right), with an s/n gap of no more than 1 dB. Note also that for $R > 2$, the $(R + 1)$-bit system with non-instantaneous quantization (the topmost point on each vertical bar) is always better than conventional $(R + 1)$-bit ADPCM, with an s/n gain of as much as 3 dB. The substantial gains at $R = 4$ and 5

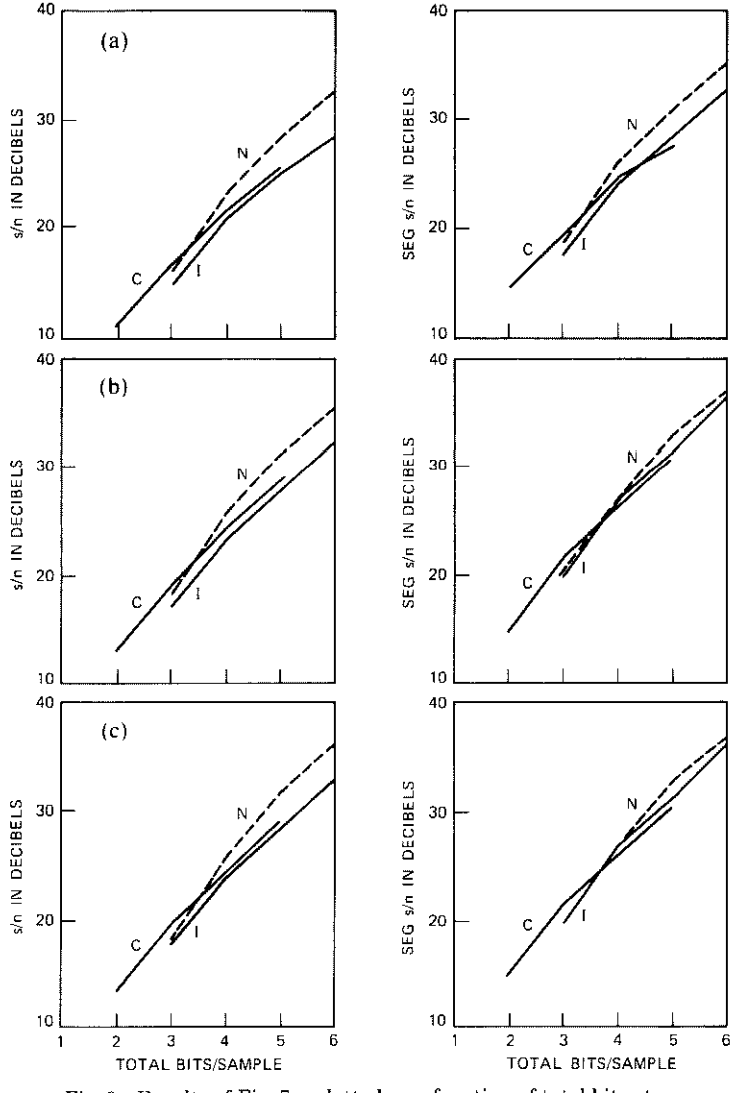



Fig. 8—Results of Fig. 7 replotted as a function of total bit rate.

may be due partly to the fact that the ADPCM quantizer in these cases is somewhat suboptimal; as $R$ increases, optimal design of the $2^{R-1}$ step-size multipliers (Fig. 2) becomes increasingly difficult, and the s/n of the conventional ADPCM coder increases by less than the expected 6 dB per additional bit.

Figure 8 replots the results of Fig. 7, and compares the three coders discussed above, for given fixed values of total bit rate. Note once again that if the overall bit rate is at least 4 bits/sample, the $(R + 1)$-bit coder with instantaneous quantization is very close to conventional $(R + 1)$-bit ADPCM; while the $(R + 1)$-bit coder with non-instantaneous quantization is consistently better than $(R + 1)$-bit ADPCM.

## VII. PERCEPTUAL EVALUATIONS OF THE CODERS OF SECTION IV AND V

Critical headphone listening reinforces the results suggested in Section VI. As expected, with $R = 2$, the outputs of 3-bit systems of Sections IV and V are both slightly worse than those of conventional 3-bit ADPCM. But with $R \geq 3$, even the output of the simpler $(R + 1)$-bit system (the system with instantaneous quantization) sounds extremely close to that of conventional $(R + 1)$-bit ADPCM. The very good perceptual performance of the instantaneous noise quantizer is very likely because much of its residual error (Fig. 3c) may be masked by the high-level speech activity in its temporal vicinity. In fact, the main motivation for the use of a non-instantaneous quantizer is not merely the increased performance with $(R + 1)$-bit coding, as demonstrated in Section VI, but also the fact that with more general $(R + R_n)$-bit coding $(R_n > 1)$, the performance of the instantaneous quantizer deteriorates rapidly, while that of the non-instantaneous quantizer maintains a 6-dB-per-bit behavior (Section VIII).

## VIII. VARIABLE-RATE CODING WITH $R_n \geq 1$ BIT/SAMPLE

Sections IV through VII discussed the design and performance of a dual-rate system with $R_n = 0$ or 1, and a total bit rate of either $R$ or $R + 1$ bits/sample. In this section, we consider a generalization to $R_n > 1$. Specifically, the average noise-coding bit rate $R_n$ will range from 0 to 3, the ADPCM bit rate $R$ will range from 2 to 5, and combinations of $R$ and $R_n$ will be such that the total bit rate $R + R_n$ will range from 2 to 6 bits/sample, the range used earlier in Fig. 8. We will note that the performance of an instantaneous noise-coding system deteriorates rapidly when $R_n > 1$, while that of a non-instantaneous noise-coding systems maintains an approximate 6-dB-per-additional-bit behavior.

### *8.1 Instantaneous noise coding*

When $R_n = 1$, the recommended output levels for the APCM noise coder were $\pm 0.25\ \Delta(n)$. These levels are in fact centered in the ranges

0 to $\pm 0.5\ \Delta(n)$, the defining ranges for granular noise amplitude. As Fig. 4 shows, the design of the instantaneous quantizer is hardly affected by the occasional incidence of overload noise magnitudes much greater than $0.5\ \Delta(n)$. Generalizations to $R_n > 1$ therefore call for sets of $2^{R_n}$ APCM output levels that are uniformly spaced in the regions $-0.5\ \Delta(n)$ to $+0.5\ \Delta(n)$. For example, with $R_n = 2$ and 3, the output levels will be

$$R_n = 2\text{: } [\pm 0.125\Delta(n),\ \pm 0.375\Delta(n)]$$

and

$$R_n = 3\text{: } [\pm 0.0625\Delta(n),\ \pm 0.1875\Delta(n),\ \pm 0.3125\Delta(n),\ \pm 0.4375\Delta(n)]. \quad (9)$$

Experiments with $R_n = 2$ and 3 show that the above design is indeed nearly optimal for instantaneous coding. However, the performance of the instantaneous coder deteriorates badly as $R_n$ increases, as we will see in Fig. 10. This is to be expected from the illustrative residual noise waveform of Fig. 3c, which shows that instantaneous coding is characterized by residual errors of very significant amplitude during periods of ADPCM overload. The situation does not improve with increasing $R_n$ because the additional output levels that become available are simply used up for finer quantization in the granular noise region, shown in eq. (9).

### *8.2 Non-instantaneous noise coding*

As we saw in the residual noise waveform of Fig. 3d for the example of average noise bit rate $R_n = 1$, non-instantaneous coding of the noise waveform can reduce the extent of granular noise as well as that of overload distortion in ADPCM coding. Slope-overload bursts are still visible in the residual noise waveform of Fig. 3d, but the waveform is much less impulsive than the original noise waveform of Fig. 3b. With $R_n > 1$, both the overload and granular components in Fig. 3d can be made increasingly smaller, provided that the bit allocation and quantizer design of Section V are properly generalized.

Recall that for an average noise bit rate of $R_n = 1$, the bit allocation (7) of Section V was as follows:

$$\begin{aligned} &R_O \text{ bits for } N_O \text{ overload noise samples} \\ &0 \text{ bits for } N_g = N_O(R_O - 1) \text{ low-amplitude noise samples} \quad (10) \\ &1 \text{ bit for } N - N_O - N_g \text{ remaining noise samples.} \end{aligned}$$

The total number of bits is then $N$ for every block of $N$ samples, as in (7). As noted in (8), the above constraint also implies that $N_O \leq N/R_O$. This condition has to be explicitly enforced even when the number of actual overload noise samples exceeds $N/R_O$ for a chosen $R_O$. A simple

generalization of (10) that works very well with $R_n > 1$ is shown below:

$R_O + (R_n - 1)$ bits for $N_O$ overload noise samples

$(R_n - 1)$ bits for $N_g = N_O(R_O - 1)$ low-amplitude noise samples (11)

$R_n$ bits for $N - N_O - N_g$ remaining noise samples.

The total number of bits is now $NR_n$ for every block of $N$ samples. Furthermore, (11) is a straightforward generalization of (10); and as in the case of (10), it requires no transmission of any side information for bit-allocation purposes, but only encoding and decoding delays in the order of $N = 32$ (4 ms, assuming 8-kHz samples).

Critical to the success of the bit-allocation algorithm (11) is a proper design of individual quantization characteristics. Unlike the instantaneous design of (9), the variable-bit allocations in (11) permit finer quantizer resolutions both in the overload range, $|r(n)| > 0.5\Delta(n)$, and in the granular noise range, $|r(n)| < 0.5\Delta(n)$. A systematic way to design these quantizers is to start with the designs in Section V (for a given $R$, and for an average-noise-bit rate of $R_n = 1$). Recall that each such design involves a set of three characteristics, for 0-bit, 1-bit, and $R_O = R$-bit quantization, as in (10). As the value of $R_n$ increases, each of these sets evolves into corresponding sets of three characteristics, for $(R_n - 1)$-bit, $R_n$-bit, and $(R_O + R_n - 1)$-bit quantization, as in (11). Resolutions improve by a factor of two for each stage of increase of $R_n$, and this improvement benefits the overload as well as granular regions of coding noise. Figure 9 illustrates the quantizer evolution for the example of $R = 3$ and $R_n = 1$ and 2. The illustration includes only one of the set of three quantizers involved in the coding process. This is the $R_O$-bit characteristic (Fig. 9a, which is the same as Fig. 5b) used for quantizing the $N_O$ overload noise samples in the $R_n = 1$ system. When $R_n = 2$, the above $R_O$-bit (in this case, 3-bit) characteristic evolves into a $R_O + R_n - 1 = 4$-bit characteristic (Fig. 9b).

Figure 10 shows the benefits of increasing $R_n$ in a non-instantaneous noise-coding system, for the example of $R = 4$ and for average-noise-coding rates of $R_n = 1$, 2, and 3 bits/sample. All error waveforms are magnified by a factor of 50. The waveform in (b) is the same as that in Fig. 3d, but is magnified by a factor of 5. In Fig. 10 we see a significant reduction in residual noise level for each stage of increasing $R_n$. We will note presently that the improvement is very close to 6 dB per additional bit in $R_n$.

### *8.3 Signal-to-noise ratios*

Figures 11a and b show s/n and segmental s/n results for explicit noise-coding systems with $R_n \geq 1$. The range of total bit rate $R + R_n$ is 2 to 6, the same as that in Fig. 8. The solid curves show the

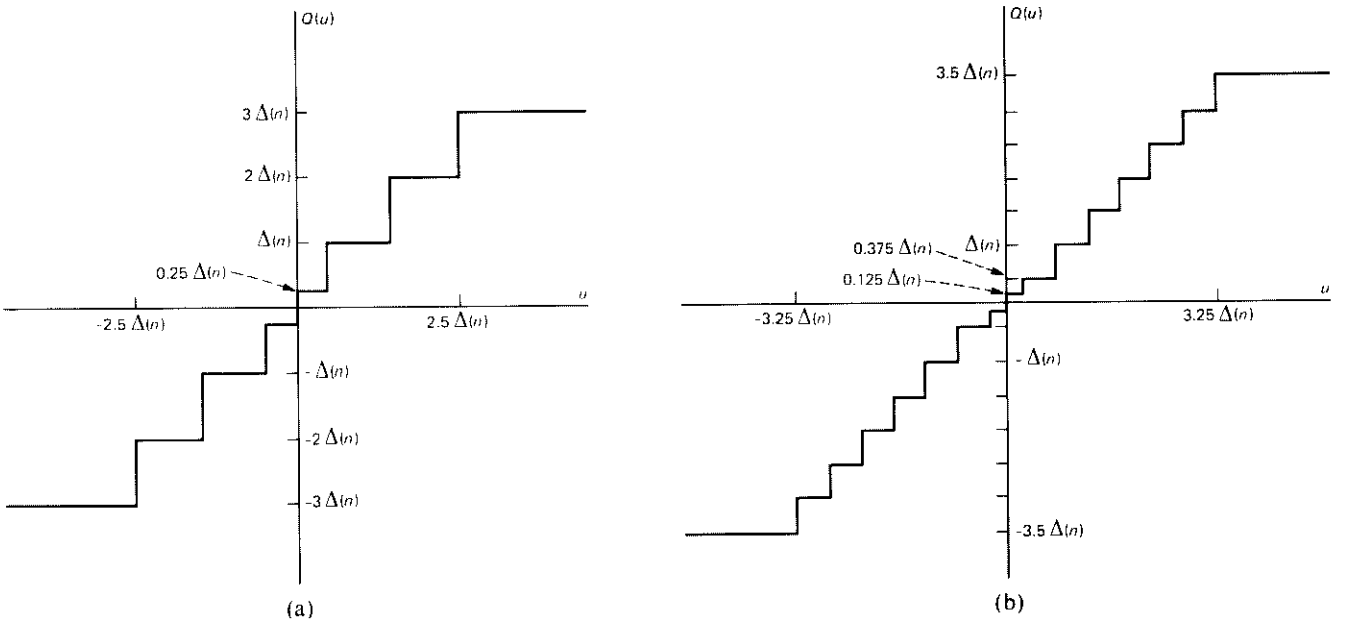



Fig. 9—Examples of quantization characteristics used for overload noise samples $r_O(n)$ in the non-instantaneous coding of 3-bit ADPCM noise with an average rate of (a) $R_n = 1$ bit/sample and (b) $R_n = 2$ bits/sample.

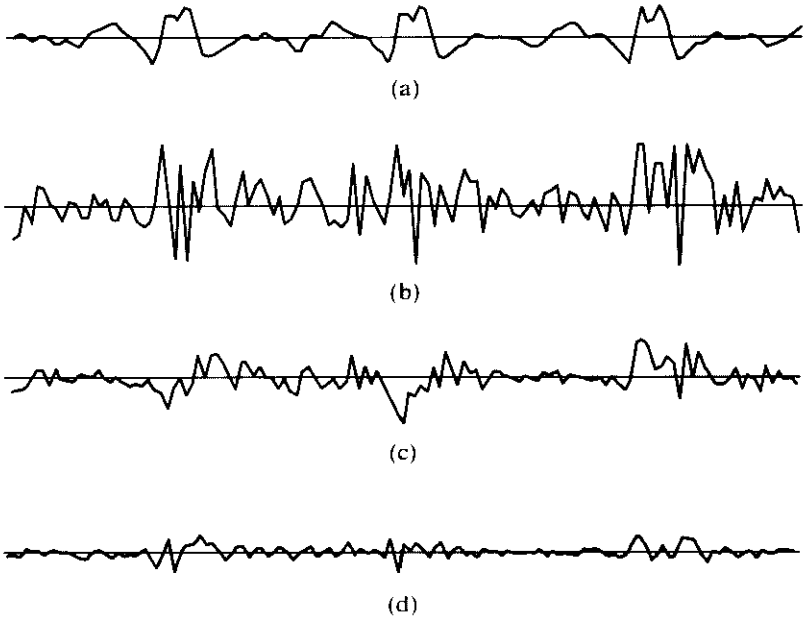



Fig. 10—(a) Input speech $x(n)$ and reconstruction error waveforms in variable-rate coding with 4-bit ADPCM, non-instantaneous noise coding and average-noise-coding rates of (b) $R_n = 1$, (c) $R_n = 2$, and (d) $R_n = 3$ bits/sample. All error waveforms are magnified by a factor of 50. The waveform in (b) is the same as that in Figure 3(d), but magnified by a factor of 5.

performance of conventional ADPCM. The circles labelled 3 show the performance of a variable-rate system based on instantaneous coding of ADPCM noise, for the example of $R = 3$. We can see that with $R_n > 1$, the s/n performance of the instantaneous coding system deteriorates fairly rapidly, compared with that of $(R + R_n)$-bit ADPCM, with increasing bit rate, but its segmental s/n performance is competitive with that of conventional ADPCM at all bit rates. Non-instantaneous coding systems, on the other hand, maintain a 6-dB per additional bit behavior, provided only that $R > 2$. This is shown by the sets of solid black dots labelled 3, 4, and 5. The performances of these systems also exceed that of conventional ADPCM at any given total bit rate, a result already noted in Section VI for the special case of $R_n = 1$.

## IX. EFFECTS OF TRANSMISSION ERRORS

Bit errors in transmission can affect the noise-coding system in two ways: they may produce effects attributable to errors in the transmission of the bits from the basic DPCM coder, and effects attributable to errors in the bits from the noise coder. Effects of both types are expected to be more severe in the case of the non-instantaneous coder.

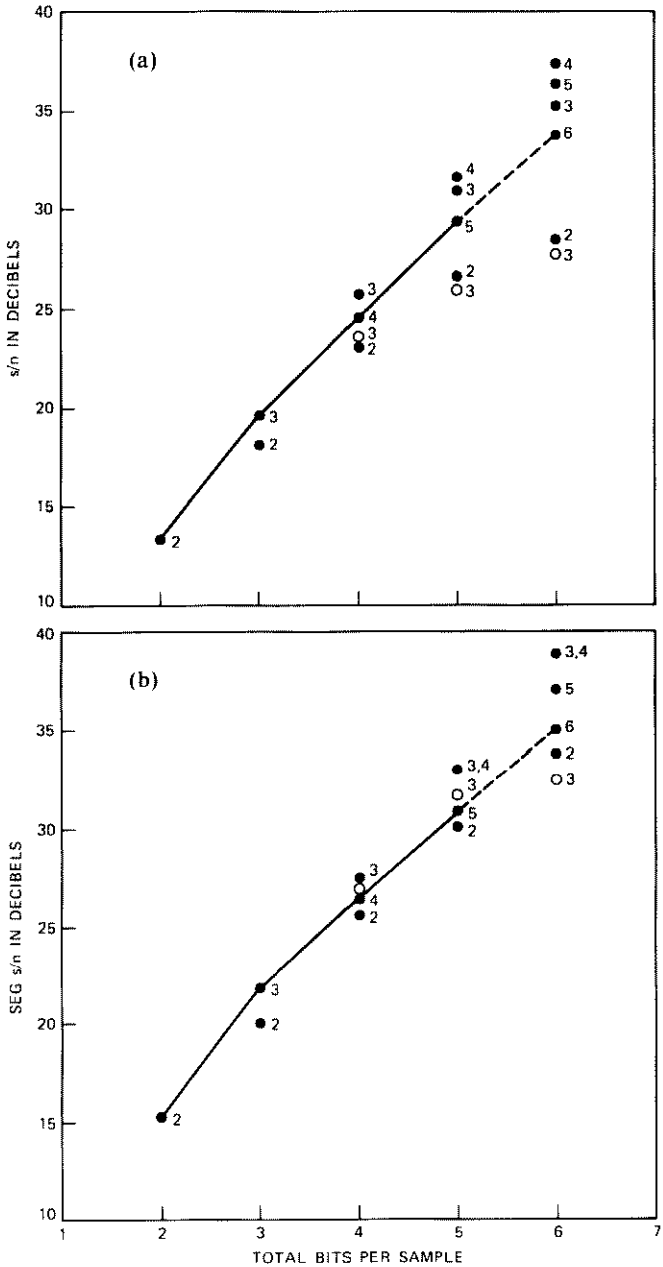



Fig. 11—(a) Signal-to-noise ratio (s/n) and (b) segmental signal-to-noise ratio (SEG s/n) in ADPCM systems, as a function of total bit rate. The solid curves refer to conventional ADPCM ($R_n = 0$) and the black dots refer to non-instantaneously quantized noise-coding systems with $R = 2, 3, 4$, and 5 and $R_n \geq 1$ bits/sample. The circles refer to an instantaneously quantized noise-coding system with $R = 3$, and $R_n = 1, 2$, and 3 bits/sample.

The greater error sensitivity of this system is due first to the presence of quantizers with larger step sizes, and hence, proportionally larger channel-error effects. More important, the increased error sensitivity of the non-instantaneous system is a result of the variable-bit-allocation algorithm, which will be miscalculated at the receiver if one or more bits from the basic ADPCM coders are received in error. Errors of this type do not propagate beyond a given $N$-sample block, but their effects can be severe enough to warrant the complete disabling of the noise-coding part of the system when errors are detected. A simple example of an error-detection system is one where the odd-even parity of the number $N_O$ of overload samples is explicitly transmitted to the receiver. A change in the parity of $N_O$, as computed at the receiver, is a good detector of perceptually significant single-bit errors in the given block. The single bit needed to transmit the parity information, or the multiplicity of bits needed to transmit the information in an error-protected format, can be incorporated in the coder output by a bit-stealing procedure based on increasing the number of zero-bit noise samples from $N_\#$ to an appropriately greater number.

Irrespective of the noise-coding method and the procedures that may be used to protect the noise-coding system from errors, the basic ADPCM coder can be made error-robust, at least for independent error rates of up to $10^{-3}$, by using robust adaptive-quantizer algorithms such as the leaky-adaptation algorithm in Ref. 9.

## X. CONCLUSIONS

We have demonstrated simple systems for variable-rate, embedded ADPCM coding of speech based on explicit coding of reconstruction noise. These systems do not require the transmission of any side information other than what is already available in a conventional ADPCM decoder. The simpler of two systems proposed in this paper uses instantaneous coding of the noise, and provides a performance very close to that of the conventional ADPCM at any given value of total bit rate $R_T = R + R_n$, for the simple but non-trivial case of dual-rate operation ($R_n = 0$ or 1 bit/sample). But its s/n performance deteriorates significantly with more widely variable operation ($R_n > 1$ bits/sample). The more complex system uses non-instantaneous noise-coding, with coding and decoding delays in the order of 4 ms to realize positive gains over conventional ADPCM at any given total bit rate $R + R_n$ bits/sample. The performance of this system has been demonstrated for $R_n = 0$, 1, 2, and 3 bits/sample, and for $R = 2$, 3, 4, and 5 bits/sample. The system with non-instantaneous noise coding can also be regarded as an $(R + R_n)$-bit ADPCM coder with a quantizing system that is better than conventional adaptive quantization with a one-word memory.

## REFERENCES


1. J. Das and P. K. Chatterjee, "Optimized Δ − Δ Modulation System," Elect. Lett., *3*, No. 6 (June 1967), pp. 286–7.
2. R. Steele, "Peak Signal-to-Noise Formulas for Multistage Delta Modulation with *RC*-Shaped Gaussian Input Signals," B.S.T.J., *61*, No. 3 (March 1982), pp. 347–62.
3. T. Bially, B. Gold, and S. Seneff, "A technique for adaptive voice flow control in integrated packet networks," IEEE Trans. Commun. (March 1980), pp. 325–33.
4. Y.-C. Ching, "Differential Pulse Code Communications System Having Dual Quantization Schemes," U.S. Patent 3,781,685, December 25, 1973.
5. D. J. Goodman, "Embedded DPCM for Variable Bit Rate Transmission," IEEE Trans. Commun. (July 1980), pp. 1040–46.
6. P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech," B.S.T.J., *52*, No. 7 (September 1973), pp. 1105–18.
7. N. S. Jayant, "Adaptive Quantization With a One-Word Memory," B.S.T.J., *52*, No. 7 (September 1973), pp. 1119–44.
8. D. L. Cohn and J. L. Melsa, "The Residual Encoder—An Improved ADPCM System for Speech Digitization," IEEE Trans. Commun. (September 1975), pp. 935–41.
9. D. J. Goodman and R. M. Wilkinson, "A Robust Adaptive Quantizer," IEEE Trans. Commun. (November 1975), pp. 1362–5.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# Random Processes With Specified Spectral Density and First-Order Probability Density

By M. M. SONDHI

(Manuscript received August 6, 1982)

*The procedure for generating a Gaussian process with a specified spectral density is well known. It is harder to generate a process with a specified spectral density and a specified first-order probability distribution. In this paper we explore, by simulation, the possibility of generating a process with such a dual specification by passing a Gaussian process with an appropriately chosen spectral density through an appropriately chosen zero-memory nonlinearity. Several applications are cited where such a dual specification is desirable.*

## I. INTRODUCTION

The procedure for generating a Gaussian random process whose power spectral density (psd) is a specified function of frequency, $S(\omega)$, is well known. As we see in Fig. 1, let $H(j\omega)$ be the transfer function of a linear time-invariant filter, and let the input to the filter be a Gaussian random process $\{x(t)\}$ with psd $\Phi_x(\omega)$. Then the psd of the output process $\{y(t)\}$ is given by

$$\Phi_y(\omega) = H(j\omega)H^*(j\omega)\Phi_x(\omega) \tag{1}$$

where * denotes complex conjugation. Since the filter $H$ is linear, $\{y\}$ is also a Gaussian random process, and if $\{x\}$ is a white noise[†] with unit psd, then $\{y\}$ has the desired psd provided

$$H(j\omega)H^*(j\omega) = S(\omega). \tag{2}$$

$H$ is then called a shaping filter (see Ref. 1) for the psd $S(\omega)$. The spectral factorization for eq. (2) can be accomplished analytically when

[†] A high degree of mathematical rigor is not intended here. For our purpose we define white noise as a noise whose psd is constant over a wide bandwidth $(-W, W)$ and zero outside. The bandwidth is assumed wide enough so that any desired psd is negligible outside it.

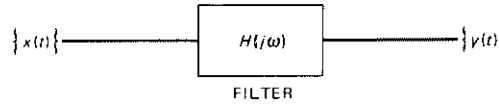



Fig. 1—Generating a Gaussian random process with specified-power spectral density.

$S(\omega)$ is a rational function of $\omega^2$.[1] In such cases the resulting shaping filter too can be synthesized by a standard procedure as a lumped parameter filter (see Ref. 2). The appropriate shaping filter can of course be derived for more general spectral densities as well as for various nonstationary processes. Numerical approximations to the shaping filter can be obtained for any reasonably well-behaved spectral density, as we shall see in Section III.

This method of generating the random process, however, leaves no choice as to the first-order probability density function (pdf) of the process, i.e., the pdf of the random variable $Y = y(t)$. Since $\{y(t)\}$ is a Gaussian process, all joint probabilities of the random variables $Y_i = y(t_i)$, $i = 1, 2, \cdots$ are Gaussian. Hence, the pdf of $Y$ too is Gaussian. (Of course, if *only* the pdf is specified, then one does not have to follow the above procedure. It is always possible to generate a white noise with any specified pdf by passing, for instance, a uniform white noise or a Gaussian white noise through a zero-memory nonlinearity. The procedure is quite analogous to the one discussed in Section 2.1.)

Frequently it is desirable to specify both the spectral density and the first-order pdf of the process. One situation where such a specification would be useful is in the simulation of speech-waveform coders. The performance of such coders can depend significantly upon both the spectral density and the pdf of the signal to be coded. Measurements have shown[3] that the pdf for speech signals is markedly different from Gaussian, and is in fact much better represented by a "gamma" distribution[4] (see Section II). At present, simulations with such coders are carried out on Gaussian processes with appropriately shaped spectra, or on sequences of uncorrelated samples with a gamma pdf. The behavior of the coders on speech signals is not well predicted by either of these; hence, tests are also performed on a variety of speech sentences. Perhaps these tests could be standardized and their predictive value improved by the use of random processes with a gamma pdf and a selection of typical spectral shapes.

Another area where this dual specification can be important is in perceptual studies. One such application is to Julesz's experiments on texture perception.[5] The independent control of spectral density and pdf of random dot patterns would enable one to decide between competing theories of texture discrimination.

Finally, such control of pdf and spectral density would be useful in

studying input-output properties of nonlinear systems, which can be represented by a zero-memory nonlinearity followed by a linear filter.

The problem of synthesizing a random process to approximate the pdf and power spectral density of a given process has been addressed in the literature.[6] However, to the best of our knowledge no exact procedure is known at present for generating a random process with specified pdf and spectral density. In the next two sections we explore, by simulation, the capabilities and limitations of one simple attack on the problem.

## II. GENERATION OF THE RANDOM PROCESS

The method of generation that we wish to study is shown schematically in Fig. 2. For simplicity we will assume throughout that the desired random process has zero mean and that the desired pdf has even symmetry. These restrictions are by no means essential for our analysis, but only simplify our presentation.

The basic idea of the proposal is as follows: We start with a "white" Gaussian noise [see the footnote in Section I] and pass it through a filter $H(j\omega)$ and a scale factor $K$ such that the random process $\{x(t)\}$ is a zero-mean-Gaussian process with unit variance. Let $q(.)$ be the desired pdf of the random variable $Y = y(t)$. Then it is straightforward to find the zero-memory nonlinearity $F(.)$ such that $Y$ has the desired pdf. The problem then is to find $H$ such that *after the nonlinear distortion by* $F(.)$ the spectral density at the output is the desired function $S(\omega)$. It is easy to come up with examples for which this problem has no solution. For instance it can be shown that $\{v(t)\}$ cannot be a strictly band-limited process for *any* choice of limiting nonlinearity $F(.)$.[7] Nevertheless, as we will show, in a variety of cases of interest the problem has a solution, or an approximate solution.

Before proceeding to a detailed description of the method, we may emphasize the reason for the choice of a Gaussian process for the input. This is the property of the Gaussian process that it stays Gaussian after a linear transformation. The Gaussian process is the only well-behaved process that has this property. The same reason also dictates the order of operations in Fig. 2. Thus, interchanging the filter and zero-memory nonlinearity would be equivalent to generating the output process by a linear transformation of a non-Gaussian white

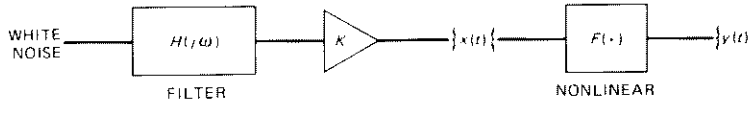



Fig. 2—Method for generating a random process with specified-power spectral density *and* specified first-order probability density.

noise. While it is possible to generate processes with non-Gaussian pdf's this way, it is not easy to compute (or control) the pdf of the output process.

### 2.1 Derivation of the function $F(.)$

Let us begin by deriving the required function $F(.)$. It is convenient to think of $F(.)$ as a composition of two functions, $P(.)$ and $f(.)$, as shown in Fig. 3. In view of the assumptions about the process $\{x(t)\}$, the random variable $X = x(t)$ has the pdf

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \tag{3}$$

If $\hat{X} = P(X)$ where $P(x)$ is the cumulative distribution

$$P(x) = \int_{-\infty}^{x} p(\lambda) d\lambda, \tag{4}$$

then the pdf of the random variable $\hat{X}$ is uniform on the interval (0, 1). Similarly, if $Y = f(\hat{X})$ and $Y$ has the desired pdf $q(y)$, then the cumulative distribution $Q(y)$ is related to the function $f^{-1}$ by the equation

$$Q(y) = \int_{-\infty}^{y} q(\lambda) d\lambda = f^{-1}(y). \tag{5}$$

Thus, in order for $Y$ to have the desired pdf, $f(.)$ must be chosen to

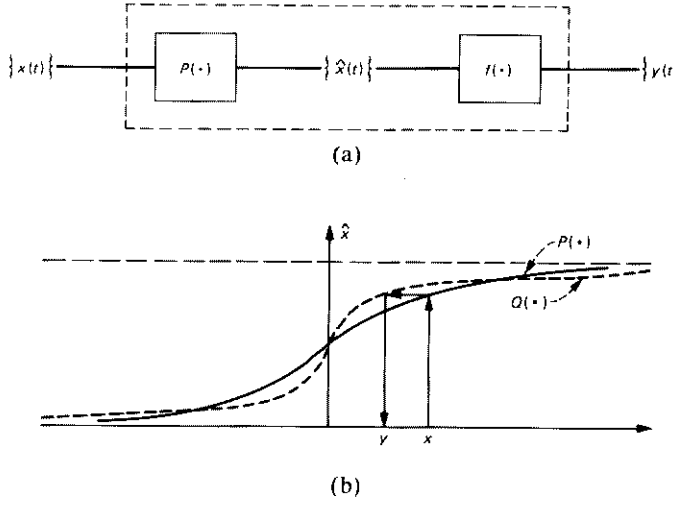



Fig. 3—Computing the nonlinearity as the composition of a Gaussian cumulative probability and the inverse of the desired cumulative distribution.

satisfy eq. (5). As long as $q(y)$ is not identically zero over an interval, $Q$ is an invertible function and eq. (5) defines $f$. The technical difficulty arising when this condition is not met is resolved in an obvious manner. Thus, if $q(y)$ is nonzero everywhere except on the interval $a \leq y \leq b$, then $Q(y)$ is constant from $a$ to $b$. Hence, $\hat{X}$ as a function of $Y$ has a jump at $Y = Q(a)$ of magnitude $b - a$. The limit of the function $f$ can be defined from above and below this value. At $Q(a)$ the function may be specified arbitrarily.

Finally, the function $F(.)$ is given by

$$F(x) = f[P(x)]. \tag{6}$$

From the assumed symmetry of $q(y)$ it is evident that $F(.)$ must turn out to be an odd function of its argument.

By way of illustration in this paper we will consider three output pdf's:
the uniform pdf:

$$q(y) = \frac{1}{2\sqrt{3}}, \qquad |y| \leq \sqrt{3}; \tag{7a}$$

the gamma* pdf

$$q(y) = \frac{3^{1/4}}{\sqrt{8\pi}} \frac{1}{\sqrt{|x|}} e^{-|x|\sqrt{3}/2}; \tag{7b}$$

and the binary pdf

$$q(y) = \tfrac{1}{2}[\delta(y-1) + \delta(y+1)]. \tag{7c}$$

In each case the constants have been chosen to normalize the variance of the pdf to be 1. Our choice of examples is of course arbitrary. However, the uniform and the binary are obvious simple examples one expects to be useful, and the gamma has the relevance to speech signals mentioned above. The function $F(.)$ turns out to be quite simple for each of these cases:
for the uniform pdf

$$F(x) = 2\sqrt{3}[P(x) - \tfrac{1}{2}], \tag{8a}$$

for the gamma pdf

$$F(x) = \frac{1}{\sqrt{3}} x^2 \text{sign}(x), \tag{8b}$$

and for the binary pdf

$$F(x) = \text{sign}(x). \tag{8c}$$

* Strictly speaking, a gamma 1/2 distribution.

### 2.2 The covariance function of $\{y\}$

Having derived the nonlinearity $F$, we must now find the relationship between the spectral properties of $\{x\}$ and $\{y\}$. Because of the nonlinear transformation, the relationship is best derived in the time domain, i.e., between the autocovariance functions of the two processes. [Recall that the autocovariance function (acf) is just the Fourier inverse transform of the spectral density.] Let $\rho(\tau)$ be the acf of the process $\{x\}$, i.e., $\rho(\tau) = E[x(t)x(t+\tau)]$. Let $g(u, v, \rho)$ be the unit variance, zero-mean, two-dimensional Gaussian pdf given by

$$g(u, v, \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{1}{2}(u^2+v^2-2\rho uv)}. \tag{9}$$

Then the acf of the $\{y\}$ process is given by

$$R = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u)F(v)g(u, v, \rho)dudv, \tag{10}$$

where, of course, $\rho$ (and hence $R$) is a function of the lag, $\tau$. A general method of evaluating the integrals in eq. (10) is to use Mehler's expansion

$$g(u, v, \rho) = \frac{1}{2\pi} e^{-\frac{1}{2}(u^2+v^2)} \sum_0^\infty \frac{\rho^n}{n!} \mathbf{He}_n(u)\mathbf{He}_n(v) \tag{11}$$

of the two-dimensional Gaussian pdf in terms of Hermite polynomials.[8] Then if $F(.)$ has an expansion in terms of the same polynomials

$$F(x) = \sum_0^\infty f_n \mathbf{He}_n(x)/\sqrt{n!}, \tag{12}$$

it can be shown[7] that

$$R = \sum_1^\infty f_n^2 \rho^n. \tag{13}$$

When $F$ is an odd function, as is the case here, the even coefficients in eqs. (12) and (13) vanish, and it is seen that $R$ is an odd function of $\rho$. [This can of course be seen directly from eq. (10) by noting that $g(u, v, -\rho) = g(u, -v, \rho)$, and changing the sign of one of the variables of integration.]

For the examples that we have chosen, there is no need to use this general procedure. In these cases the integration is easily carried out in closed form. As shown in the appendix, the relationship between $R$ and $\rho$ is as follows:

$$\text{Uniform: } R = \frac{6}{\pi}\sin^{-1}\frac{\rho}{2}. \tag{14a}$$

$$\text{Gamma: } R = \frac{2}{3\pi}\left[(1 + 2\rho^2)\sin^{-1}\rho + 3\rho\sqrt{1 - \rho^2}\right] \tag{14b}$$

$$\text{Binary: } R = \frac{2}{\pi}\sin^{-1}\rho. \tag{14c}$$

The three functions are shown in Fig. 4 for $0 \leq \rho \leq 1$; the odd symmetry gives the function for negative $\rho$ as well.

### *2.3 Derivation of the Filter H*

The functions in Fig. 4 are monotonic increasing functions over the whole range $-1 \leq \rho \leq 1$. [From eqs. (12) and (13) this property is seen to be true whenever $F$ has odd symmetry.] Therefore, the relationship between $\rho$ and $R$ is invertible.

Suppose now that $R(\tau)$ is the Fourier transform of the desired spectral density $S(\omega)$. Then the function

$$\phi(\tau) = \rho[R(\tau)] \tag{15}$$

can be computed, where $\rho(R)$ is the function corresponding to the desired pdf. For the examples chosen we get

$$\text{Uniform: } \phi_u(\tau) = 2\sin\left[\frac{\pi}{6}R(\tau)\right] \tag{16a}$$

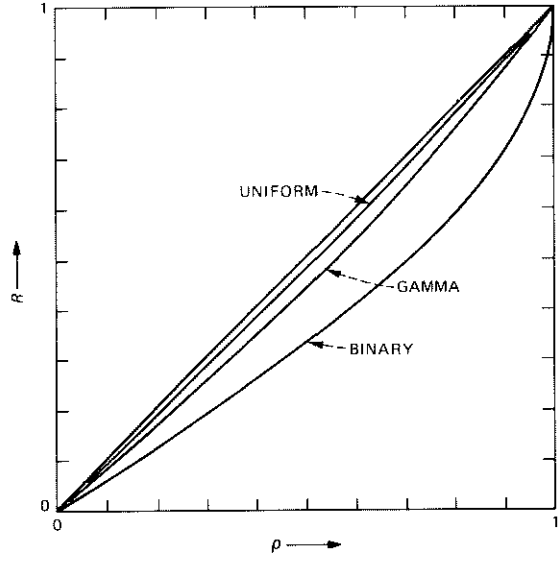



Fig. 4—The relationship between the autocovariance ($R$) at the output and ($\rho$) at the input of the nonlinearity appropriate to the generation of the three desired probability distributions. The unmarked line is $R = \rho$. Since the functions have odd symmetry, only the range of positive covariance values is shown.

$$\text{Gamma: } \phi_g(\tau) = \rho_g[R(\tau)] \tag{16b}$$

$$\text{Binary: } \phi_b(\tau) = \sin\left[\frac{\pi}{2} R(\tau)\right], \tag{16c}$$

where $\rho_g$ is the inverse of the relation (14b). We are unable to give $\rho_g$ explicitly; however, its numerical computation is trivial.

The function $\phi(\tau)$ is continuous and symmetric so that its Fourier transform, $\Phi(\omega)$, is real. For many specified acf's $R(\tau)$, $\Phi(\omega) \geq 0$ for all $\omega$. In such cases we can obviously obtain an exact solution to the problem. All we need to do is to synthesize a filter, $H(j\omega)$, such that

$$K^2 H(j\omega) H^*(j\omega) = \Phi(\omega) = \mathbf{F}[\phi(\tau)], \tag{17}$$

where $\mathbf{F}$ denotes the Fourier transform. Unfortunately, $\Phi(\omega)$ is not guaranteed to be nonnegative at all frequencies for every specified acf $R(\tau)$. This is because when $\rho(\tau)$ in eq. (10) ranges over all possible acf's, $R(\tau)$ ranges only over a subset of possible acf's.

If the desired spectral density is such that $\Phi(\omega)$ becomes negative at some frequencies, the best we can do with our method is to give an approximate solution as follows: Define the function $\Phi^+(\omega)$ such that

$$\begin{aligned} \Phi^+(\omega) &= \Phi(\omega) \quad \text{if } \Phi(\omega) \geq 0 \\ &= 0 \qquad \text{otherwise.} \end{aligned} \tag{18}$$

We then synthesize $H(j\omega)$ such that

$$K^2 H(j\omega) H^*(j\omega) = \Phi^+(\omega). \tag{19}$$

In the next section we will use eq. (19) [or its special case, eq. (17)] to generate random processes with a variety of pdf's and spectral densities.

## III. SIMULATIONS

In this section we will describe the numerical procedures required to generate a random process with a pdf $q(.)$ and a spectral density $S(\omega)$, based on the theoretical discussion of the previous section. We have already shown how the nonlinearity $F(.)$ is to be computed. It remains to be shown how to numerically approximate the shaping filter.

### *3.1 The shaping filter*

We will approximate the shaping filter as a transversal filter (FIR filter). Let $R(\tau)$ be the desired acf. Unless $R(\tau)$ happens to be of finite duration, it must first be truncated. To ensure that the truncated function $\hat{R}(\tau)$ is a legitimate acf, the truncation must be done by multiplying the given acf by an acf of finite duration. (Recall that the

product of two acf's is an acf). We chose a Hamming window convolved with itself as the truncating window. Thus let

$$w(\tau) = \int_{-\infty}^{\infty} h(t)h(t-\tau)dt, \tag{20}$$

where $h(t)$ is the Hamming window defined as

$$h(t) = 0.54 + 0.46 \cos\frac{\pi|t|}{T}, \quad |t| \leq T \tag{21}$$

$$= 0 \qquad \text{otherwise.}$$

Then we define the truncated acf as

$$\hat{R}(\tau) = w(\tau)R(\tau). \tag{22}$$

For $R(\tau)$ decaying rapidly enough as $\tau \to \infty$, $\hat{R}(\tau)$ can be made to approximate $R(\tau)$ as closely as desired by choosing $T$ sufficiently large. For the rest of the paper, therefore, we will regard $\hat{R}(\tau)$ as the desired acf, and $\hat{S}(\omega) = \mathbf{F}[\hat{R}(\tau)]$ as the desired spectral density.

The filter is synthesized from $\hat{R}(\tau)$ by the following sequence of steps:

(*i*) Discretize $\hat{R}(\tau)$ to give the sequence $\hat{R}_n$, $-N + 1 \leq n \leq N$.*

(*ii*) Compute the sequence $\phi_n = \phi(\hat{R}_n)$, where $\phi$ is the function defined by eq. (15) for the appropriate desired pdf.

(*iii*) Find the FFT of the sequence $\phi_n$ and set any negative values to zero. Let $\mu_n$ denote the resulting, adjusted Fourier transform.

(*iv*) The desired impulse response is then the inverse FFT of the sequence $\sqrt{\mu_n}$.

### *3.2 Examples*

For each of the pdf's (a), (b), and (c) we have generated several examples of random processes with acf's selected from the following types:

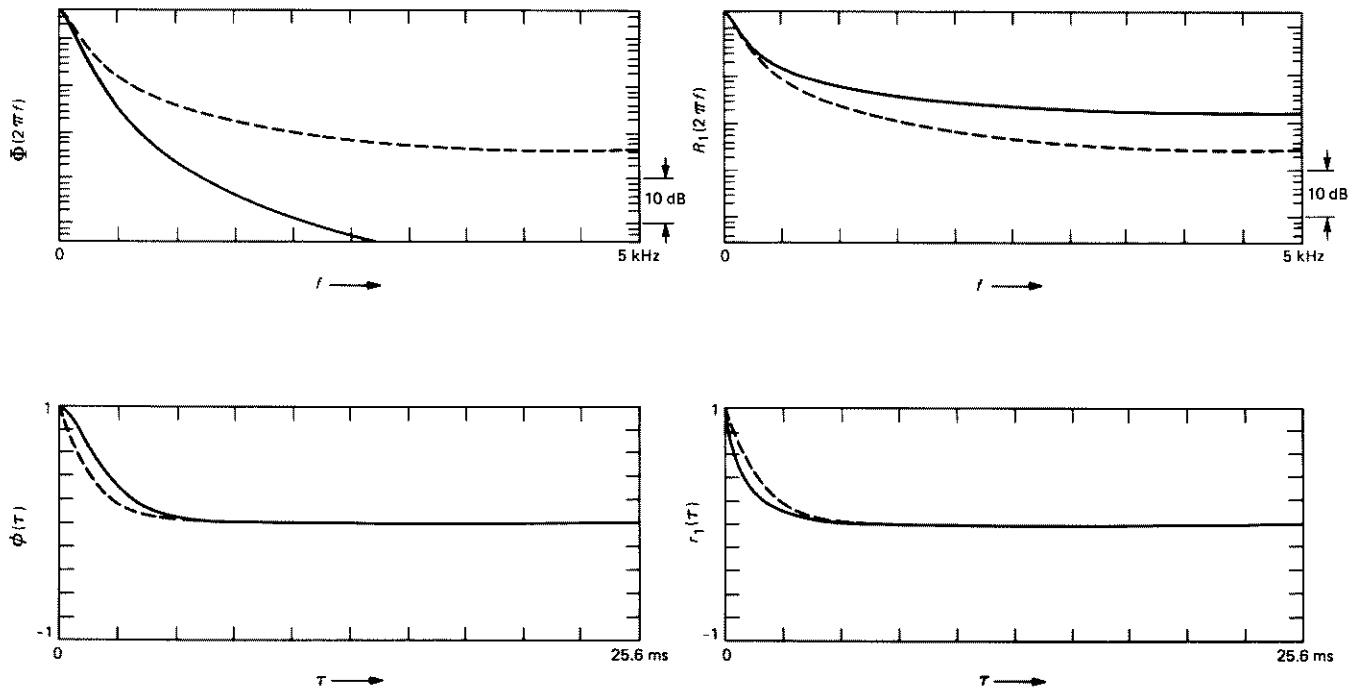
$$R(\tau) = e^{-\alpha\tau} \tag{23a}$$

$$R(\tau) = \sum_{1}^{4} a_i e^{-\alpha_i\tau}\cos(\beta_i\tau), \tag{23b}$$

where $a_i \geq 0$ in the last equation. By proper choice of parameters in eq. (23b), the spectrum can be made to approximate that of any vowel sound up to about 4 kHz. The $\beta_i$'s are the formant frequencies and the $\alpha_i$'s the half bandwidths.

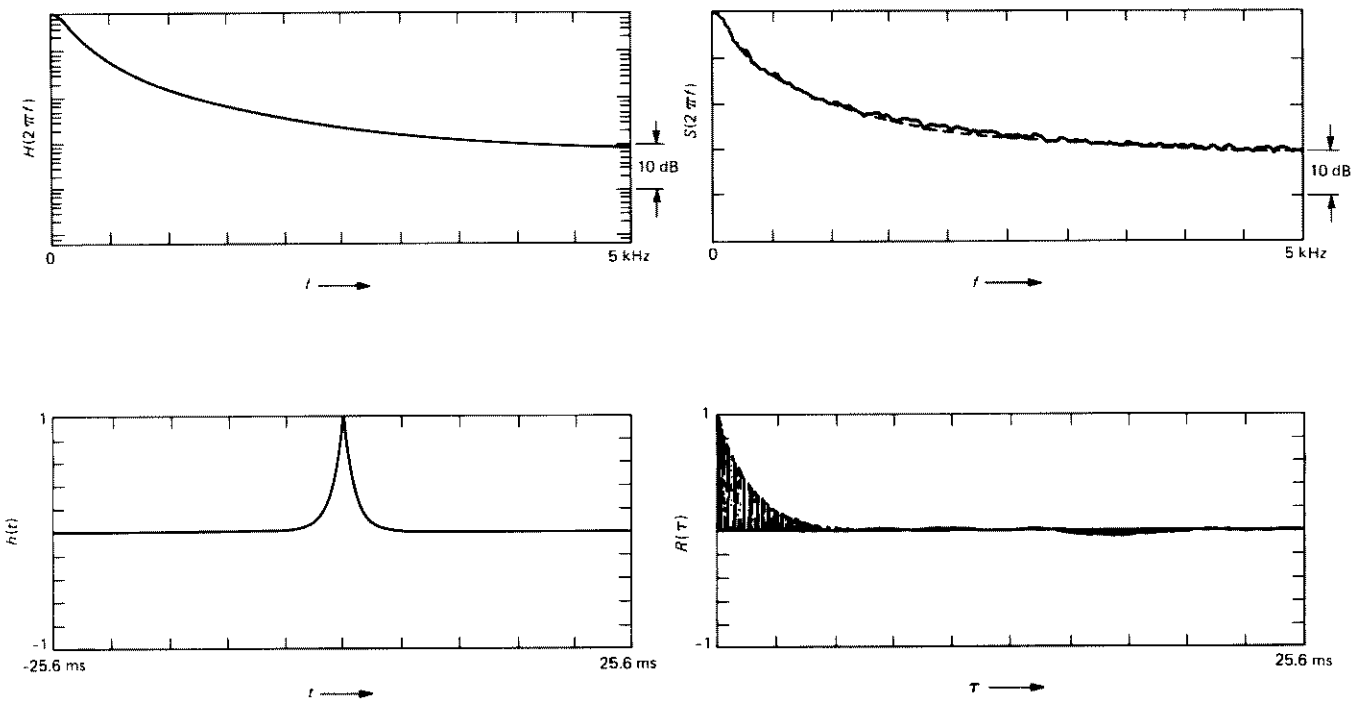
---

* The asymmetry of the range of values for $n$ makes the number of values even. Choosing $N$ to be a power of 2 allows the use of efficient FFT (Fast Fourier Transform) routines.




(a)

Fig. 5a—Generating a binary process with an exponential acf with 100-Hz cutoff frequency. On each graph the dotted line is the desired or prescribed function for the output process. The acf and spectral density required for the $\{x\}$ process are $\phi(\tau)$ and $\Phi(\omega)$, respectively. $r_1(\tau)$ and $R_1(\omega)$ are what the corresponding functions would be for the output process $\{y\}$ if the input $\{x\}$ had the specified acf.




(b)

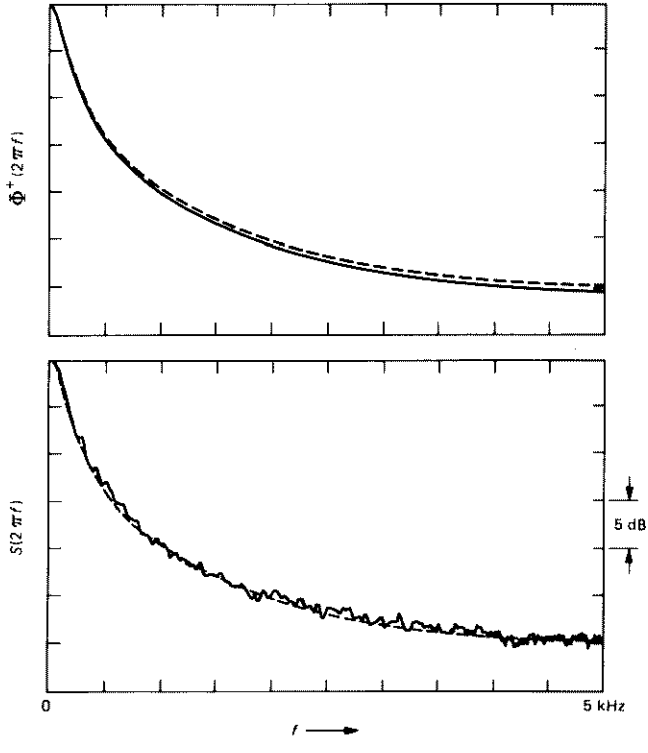
Fig. 5b—On the left are shown the time and frequency domain plots of the shaping filter required to produce the $\{x\}$ process of Fig. 5a. On the right are the acf and spectral density of an actual binary process generated by the method described in the text.

The white noise input to the shaping filter was just a sequence of independent, identically distributed Gaussian random variables. The Gaussian distribution was truncated to ±6 standard deviations.

In Figs. 5 through 9 we show several examples of acf's and spectra (both as predicted theoretically and as measured from the actual outputs) that can be generated by our method.

Figure 5 shows in detail various covariances and spectra associated with the generation of a binary process with an exponential covariance function. Whenever a dotted graph is displayed it is the desired or specified function.

In Fig. 5a the left side shows the acf and spectral density $\phi(\tau)$ and $\Phi(\omega)$ of the $\{x\}$ process, which when passed through the clipping nonlinearity gives the process $\{y\}$ with the specified properties. The right side of Fig. 5a shows the same plots for the process $\{y\}$ that would result if $\{x\}$ were a Gaussian process that had the *specified* acf.



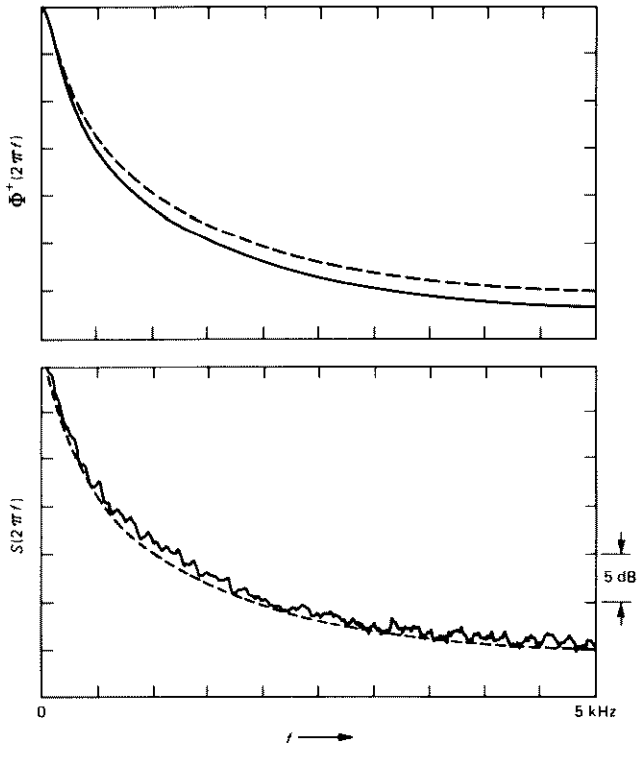

(a)

Fig. 6a—On the top the spectral density $\Phi(\omega)$ required for the $\{x\}$ process and on the bottom the spectral density estimated from a 25,000-sample sequence. The spectral density specified is the same as in Fig. 5.

In Fig. 5b the left side shows the time and frequency domain plots of the shaping filter to produce the required $\{x\}$, and the right side shows the acf and spectral density estimated from a 25,000-sample sequence generated by the method discussed above for the binary case.

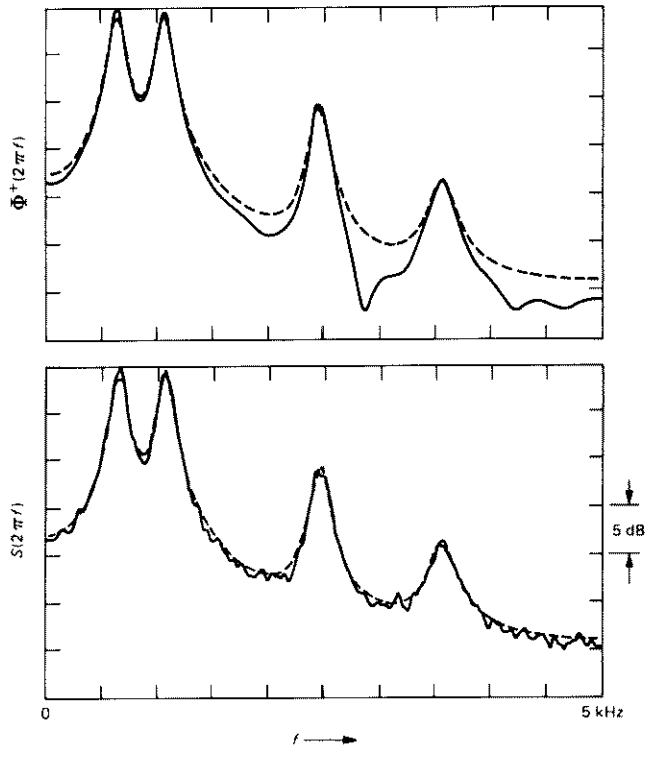
In Figs. 6 through 9 we show spectral density plots only. Further, we show only the plot for $\Phi^+(\omega)$, as defined in eq. (19), and the plot of the output spectral density estimated from 25,000-sample sequences generated by our method. (Of course, $\Phi^+$ is often identical to $\Phi$.) Again, in each case the dotted graphs represent the desired or specified function. The selections shown are an exponential acf for the uniform and gamma distributions in Fig. 6; spectra of vowels /a/ and /e/ with a uniform distribution in Fig. 7; the same vowels with a gamma distribution in Fig. 8; the same vowels with a binary distribution in Fig. 9.

The probability distributions of the generated processes are, of course, not approximated; they are therefore exact except for fluctuations because of finite sample size. Figure 10 shows the actual and




(b)

Fig. 6b—Same as Fig. 6a but for the gamma distribution.



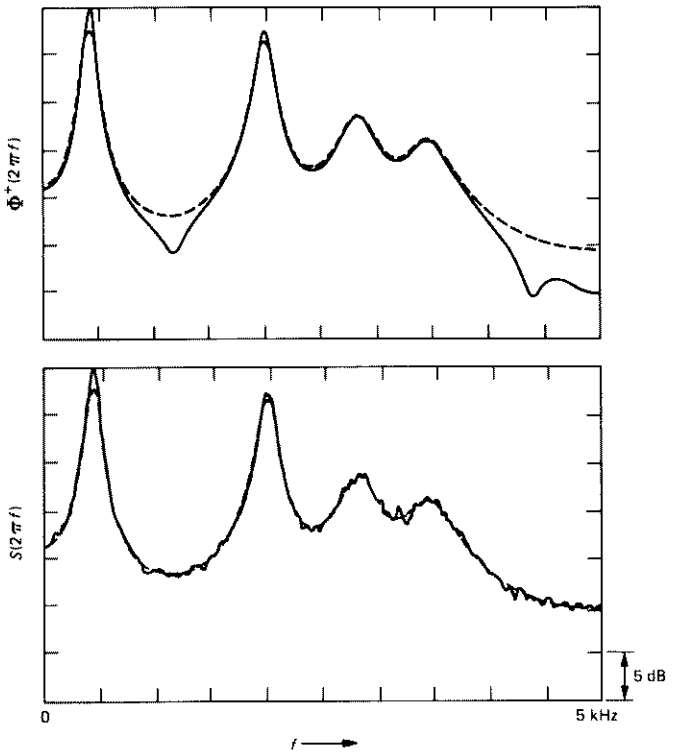

(a)

Fig. 7a—The specified spectral density is a typical spectral shape for the vowel /a/. On the top is $\Phi(\omega)$ for the required $\{x\}$ process. On the bottom is the plot estimated from a 25,000-sample sequence of the generated process. The dotted line in each case is the specified spectral density.

expected distributions for a 25,000-sample sequence with a uniform and a gamma distribution.

Figures 5 through 10 are self-explanatory and demonstrate the capabilities and limitations of the method. We may summarize the main features as follows:

(*i*) When the specified acf is given by eq. (23a), the problem can be solved exactly for any $\alpha$ for each of our examples of pdf. For the uniform and the binary pdf this can be proved analytically, by expanding the sin function in eqs. (16a) and (16c) in powers of $R$. If we group terms of these expansions in pairs, we can show that the Fourier transform of $\phi(\tau)$ is nonnegative in each of these cases. For the gamma pdf we cannot prove this analytically; however by simulation over a wide range of $\alpha$'s we conclude that $\phi(\tau)$ has a nonnegative Fourier transform in this case too.




(b)

Fig. 7b—Same as Fig. 7a but for the vowel /e/. The plots are for a process with a uniform probability density.

(*ii*) Vowel spectra can be well approximated by random processes with a uniform or gamma distribution. (In the two cases shown in Figs. 7 and 8, the spectrum is realised exactly with the uniform pdf, but only approximately with the gamma pdf; however, the error in the spectrum in the latter case is not much larger than the statistical fluctuations in a 25,000-sample segment. So from a practical point of view the approximation error might not be serious.)

(*iii*) The nonlinearity required for the binary pdf is too severe to allow a well-defined formant structure, especially at high frequencies.

## IV. CONCLUDING REMARKS

We have described a method for generating a random process with specified spectrum and first-order probability density. The method is successful for many combinations of spectrum and pdf of interest. However, with a given probability density, the method cannot achieve

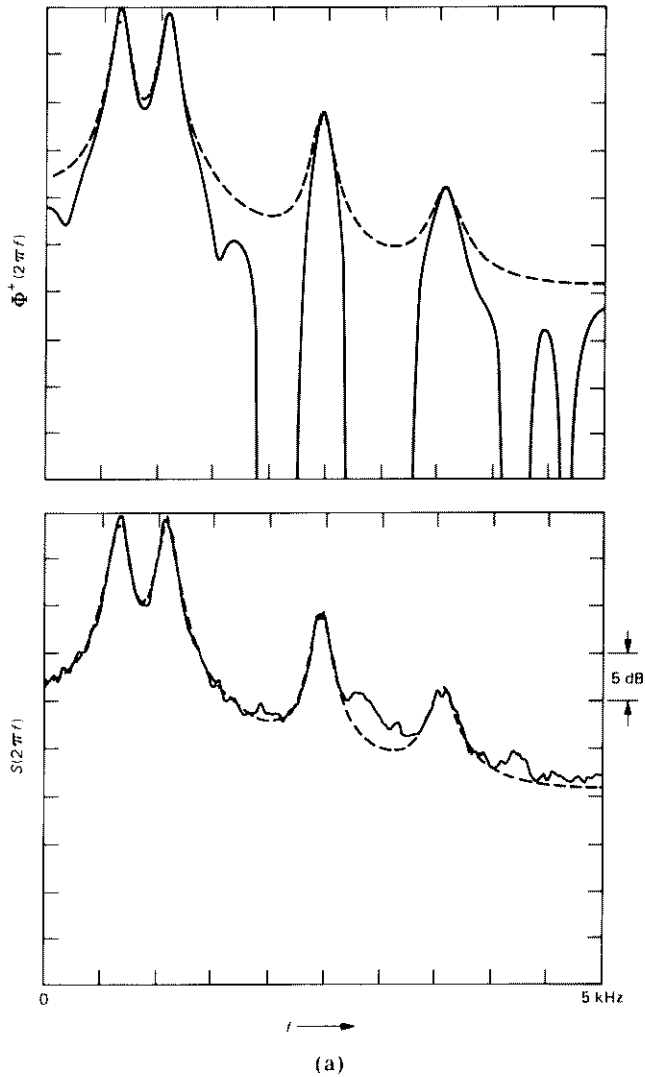



Fig. 8a—The specified spectral density is a typical spectral shape for the vowel /a/ for a process with a gamma probability distribution. On the top is $\Phi(\omega)$ for the required $\{x\}$ process. On the bottom is the plot estimated from a 25,000-sample sequence of the generated process. Note that $\Phi(\omega)$ had to be corrected to $\Phi^+(\omega)$, as shown in eq. (15).

every arbitrarily specified spectral shape. One general way to characterize an achievable spectrum is to say that the corresponding covariance function must be representable in the form given in eq. (13). Another way is to say that the function $\phi(\tau)$ in eq. (15) must have a nonnegative Fourier transform.

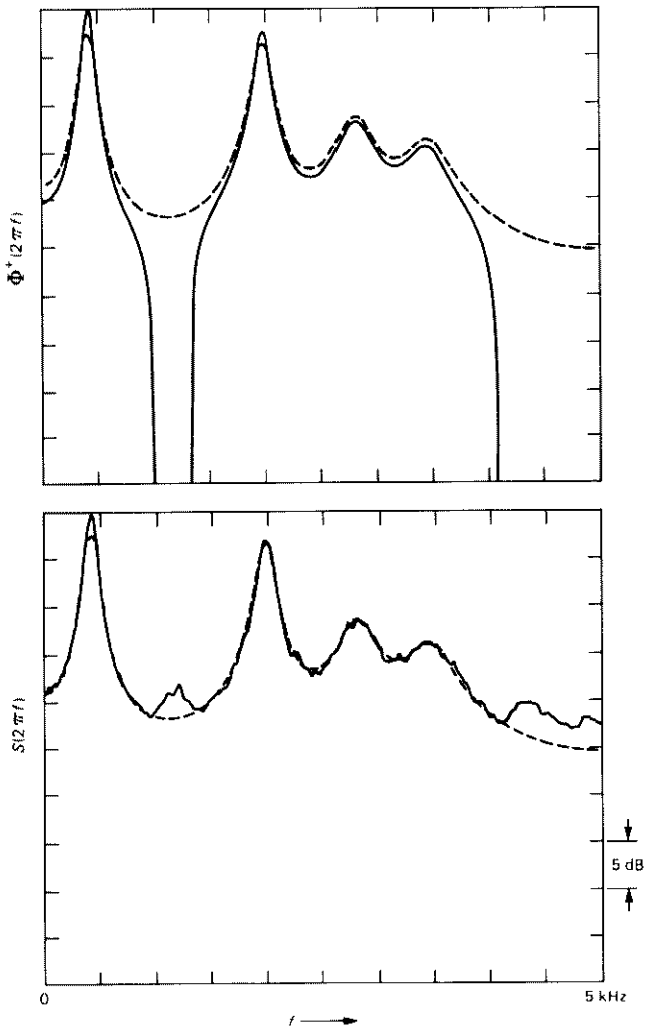
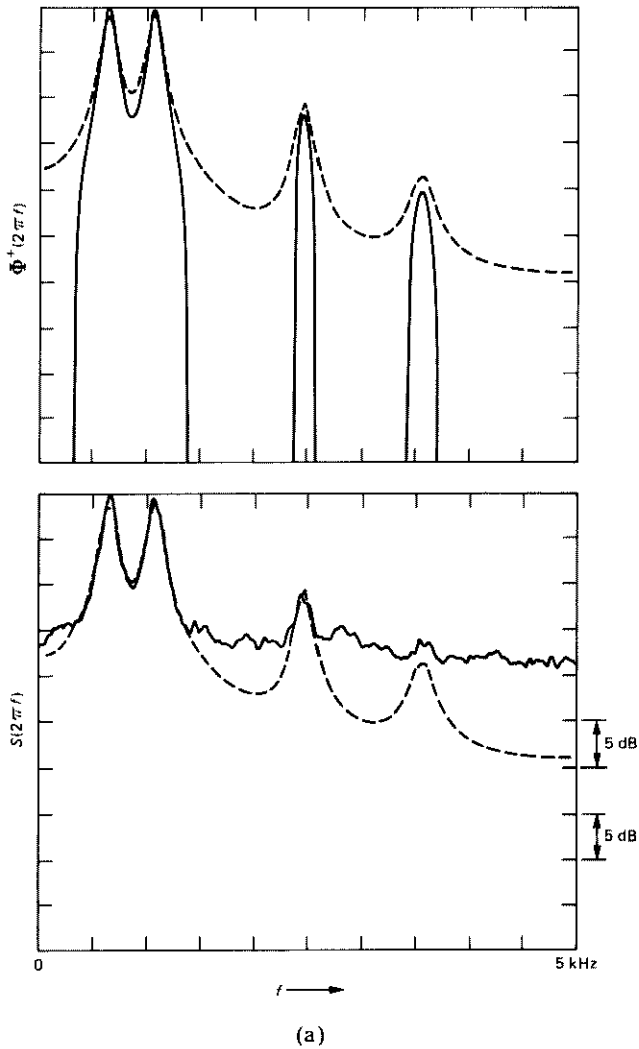



Fig. 8b—Same as Fig. 8a but for /e/.

The class of achievable spectra may be extended by other methods of generating the random process. One interesting method has been suggested by E. N. Gilbert. Rather than nonlinearly distorting a Gaussian process, the suggestion is to *modulate* a Gaussian process by an appropriately chosen nonnegative random process. This method has its own limitations. For instance, it cannot generate a process with a uniform pdf. On the other hand it appears to be very promising for

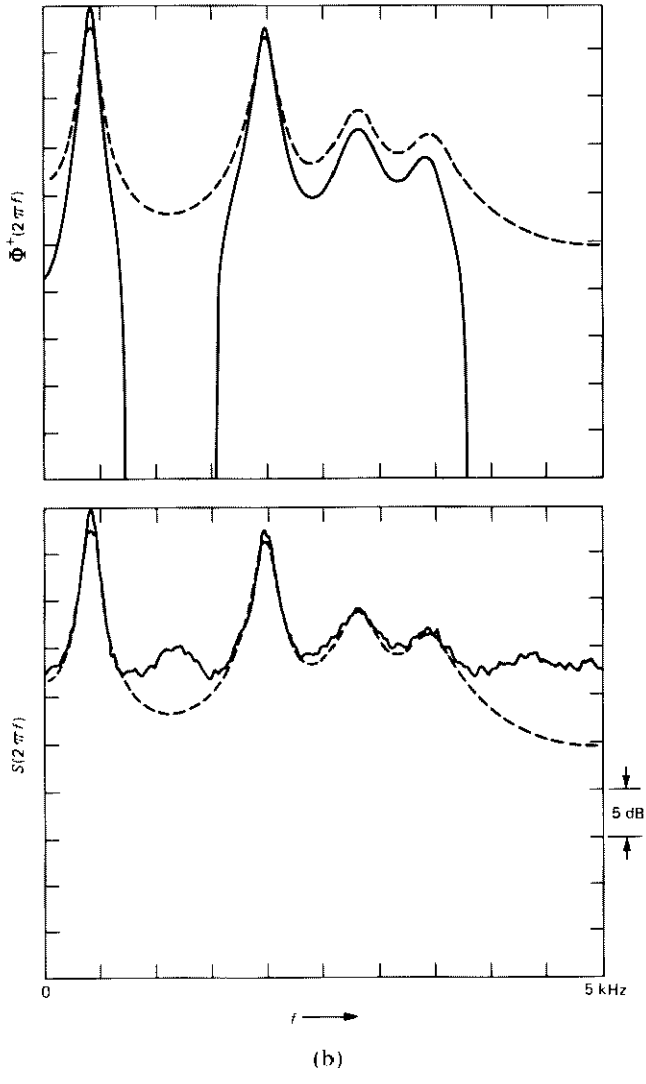



(a)

Fig. 9a—The specified spectral density is a typical spectral shape for the vowel /a/ for a process with a binary distribution. On the top is $\Phi(\omega)$ for the required $\{x\}$ process. On the bottom is the plot estimated from a 25,000-sample sequence of the generated process. Note that $\Phi(\omega)$ had to be corrected to $\Phi^+(\omega)$, as shown in eq. (15).

generating processes with speech-like pdf's (e.g., the gamma pdf discussed above). We are currently investigating this possibility.

To some extent the pdf and covariance function constrain each other regardless of the method used for generating the random process. For example, the covariance function of a time-continuous binary process must have a cusp at the origin. If the specified acf does not




(b)

Fig. 9b—Same as Fig. 9a but for the vowel /e/.

have this property, then the specifications are inconsistent and the problem has no solution. Unfortunately, to the best of our knowledge, no tractable procedure is known to decide whether a covariance function is consistent with a given pdf. L. A. Shepp has drawn our attention to some of his unpublished work in which he investigated the class of covariance functions of processes with a given pdf. He showed that this class is convex and that any such acf is a convex

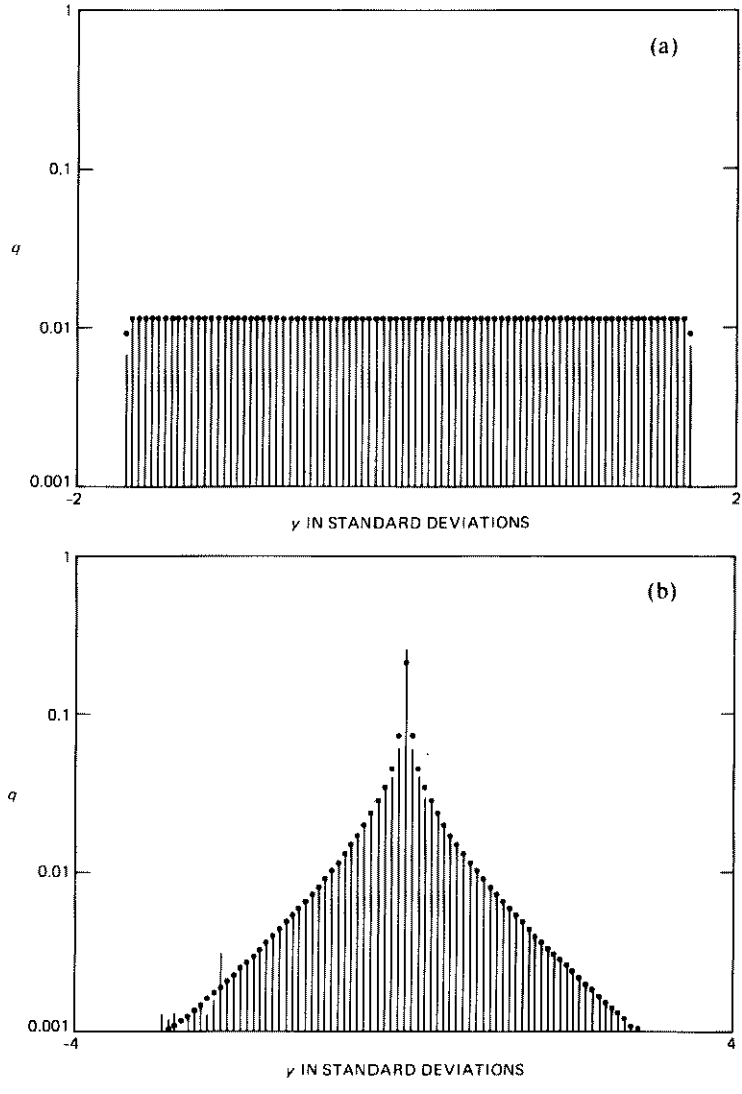



Fig. 10—A check on the probability distributions of the output processes. (a) The expected ( · ) and measured (bars) pdf estimated from a 25,000 sample when the specified distribution was uniform. The fit is approximately as good as this for every uniform-pdf process generated. (b) Same as for Fig. 10a but for the gamma distribution.

combination of the extremal acf's of the class. Unfortunately, no general method is known to determine the extremal covariances. Another relevant work is a paper[9] by Brockway McMillan, in which he considers covariances of binary processes, and gives a test for such

a covariance in terms of the class of "corner-positive" matrices. This test too is very difficult to apply in practice.

In some situations (for example, the experiment on texture perception mentioned above) it might not be important to specify the spectrum and the pdf precisely. Instead, it might suffice to be able to generate random processes with (*i*) different spectra and exactly the same pdf, or (*ii*) different pdf's and exactly the same spectrum. The first of these objectives is accomplished by keeping the nonlinearity fixed and varying the spectrum of the shaping filter. Inspection of eqs. (12) and (13) shows how the second objective may be accomplished.[7] Since the acf in eq. (13) depends only upon the squares of the expansion coefficients, it is evident that two nonlinearities for which one or more coefficients differ in sign (but not in magnitude) will yield identical spectra. The pdf will in general be quite different in the two cases.

## REFERENCES


1. J. H. Laning and R. H. Battin, *Random Processes in Automatic Control*, New York: Mc-Graw Hill Book Company, 1956, p. 217.
2. E. A. Guillemin, *Synthesis of Passive Networks*, New York: John Wiley and Sons, Inc., 1957, Chapter 3.
3. W. B. Davenport, "An Experimental Study of Speech-Wave Probability Distributions," J. Acous. Soc. Am., *24* (July 1952), pp. 390–9.
4. D. L. Richards, "Statistical Properties of Speech Signals," Proc. IEE, *111*, May 1964.
5. Bela Julesz, "Textons, the Elements of Texture Perception, and Their Interactions," Nature, *290*, No. 5802 (12 March 1981), pp. 1–7.
6. Y. Linde and R. M. Gray, "A Fake Process Approach to Data Compression," IEEE Trans. Commun., *COM-26* (June 1975), pp. 840–7.
7. G. L. Wise, A. P. Traganitis, and J. B. Thomas, "The Effect of a Memoryless Nonlinearity on the Spectrum of a Random Process," IEEE Trans. Information Theory, *IT-23* (January 1977), pp. 84–8.
8. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, Washington, D.C., Chapter 22.
9. Brockway McMillan, "History of a Problem," Jour. SIAM, *3* (September 1955), pp. 119–28.


## APPENDIX

### *Derivation of $R(\tau)$*

In this appendix we will derive the three equations (14a), (14b), and (14c) of the text. All three derivations depend on a well-known property of the bivariate Gaussian pdf $g(u, v, \rho)$, namely,

$$\frac{\partial g}{\partial \rho} = \frac{\partial^2 g}{\partial u \partial v}. \tag{24}$$

This follows trivially from the representation of $g$ as the Fourier transform of its characteristic function, i.e.,

$$g(u, v, \rho) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{j(\omega u + \sigma v)} e^{-\frac{1}{2}(\omega^2 + \sigma^2 + 2\rho\omega\sigma)} d\omega d\sigma. \tag{25}$$

Differentiating this expression with respect to $\rho$ and then with respect to $u$ and $v$ verifies eq. (24).

[In passing we note that Mehler's expansion eq. (11) also follows trivially from eq. (25). For if

$$g(u, v, \rho) = \sum_0^\infty a_n(u, v) \frac{\rho^n}{n!} \tag{26}$$

it follows that

$$\begin{aligned} a_n(u, v) &= \frac{\partial^n g}{\partial \rho^n}\bigg|_{\rho=0} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (-\omega\sigma)^n e^{j(\omega u+\sigma v)} e^{-\frac{1}{2}(\omega^2+\sigma^2)} d\omega d\sigma \\ &= \frac{\partial^n p(u)}{\partial u^n}\,\frac{\partial^n p(v)}{\partial v^n}. \end{aligned} \tag{27}$$

Here $p(.)$ is the univariate Gaussian pdf, eq. (3). However, the Hermite polynomials are *defined* by the relation

$$\frac{d^n p(x)}{dx^n} = (-1)^n p(x) \mathrm{He}_n(x), \tag{28}$$

which thus gives Mehler's expansion.]

Let us differentiate eq. (10) of the text with respect to $\rho$ and use eq. (24) to evaluate the right-hand side. Thus,

$$\frac{dR}{d\rho} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u)F(v) \frac{\partial^2 g}{\partial u \partial v}\, du dv. \tag{29}$$

Integrating eq. (29) by parts and assuming that $g(u, v, \rho)F(u)F(v)$ vanishes when $u, v \to \pm\infty$, we get

$$\frac{dR}{d\rho} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F'(u)F'(v)g(u, v, \rho) du dv. \tag{30}$$

For the binary case (8c), $F'(u) = 2\delta(u)$. In this case

$$\frac{dR}{d\rho} = 4g(0, 0, \rho) = \frac{2}{\pi\sqrt{1-\rho^2}}. \tag{31}$$

Since $R = 0$ when $\rho = 0$ (because $F$ is antisymmetric) this immediately gives

$$R = \frac{2}{\pi} \sin^{-1} \rho, \tag{32}$$

which is eq. (14c).

For the nonlinearity in the case of the gamma pdf,

$$F'(u) = \frac{2}{\sqrt{3}} |u| \tag{33a}$$

$$F''(u) = \frac{2}{\sqrt{3}} \operatorname{sign}(u). \tag{33b}$$

Using eq. (24) twice we get

$$\frac{dR}{d\rho} = \frac{4}{3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |u||v| g(u, v, \rho) du dv \tag{34a}$$

and

$$\frac{d^2R}{d\rho^2} = \frac{4}{3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \operatorname{sign}(u)\operatorname{sign}(v) g(u, v, \rho) du dv.$$

$$= \frac{8}{3\pi} \sin^{-1}\rho, \tag{34b}$$

where the last step follows from the result just derived for the binary case. Now at $\rho = 0$, $R = 0$ and, from eq. (34a) $dR/d\rho = 8/3\pi$. With these initial values eq. (34b) can be integrated easily to give eq. (14b) of the text.

Finally, for the nonlinearity (8a) $F'(u) = 2\sqrt{3}p(u)$ and, therefore,

$$\frac{dR}{d\rho} = 12 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(u)p(v)g(u, v, \rho) du dv$$

$$= \frac{3}{\pi^2} \frac{1}{\sqrt{1-\rho^2}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{(2-\rho^2)u^2 + (2-\rho^2)v^2 - 2\rho uv}{2(1-\rho^2)}} du dv. \tag{35}$$

The integrand in eq. (35) is of the same form as $g(u, v, \rho)$ with a somewhat different quadratic form in the exponent. It is a standard integral whose value is $2\pi/\sqrt{\Delta}$, where $\Delta$ is the determinant of the quadratic form. Simple algebraic manipulation then gives

$$\frac{dR}{d\rho} = \frac{6}{\pi} \frac{1}{\sqrt{4-\rho^2}}. \tag{36}$$

Again since $R = 0$ when $\rho = 0$, this gives

$$R = \frac{6}{\pi} \sin^{-1} \frac{\rho}{2}, \tag{37}$$

which is eq. (14a).

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# A Method to Characterize the Mechanical Properties of Undersea Cables

By T. C. CHU

(Manuscript received January 22, 1982)

*A method has been developed to evaluate the mechanical properties of cables containing stranded-strength members in both linear elastic and nonlinear plastic regions. The method extends Cannon and Santana's general system of equations describing the cable mechanical characteristics. In the formulation of the method, the fundamental assumptions made by Cannon and Santana are first examined and justified. Next, instead of using elastic constants for the constituent cable materials in the system of equations, the regression analysis is applied to the tensile and torsional test data of dominating high-strength cable components to obtain least-squares-fit polynominals approximating the stress versus strain and shearing-stress versus shearing-strain curves. By differentiating the polynominals, the tensile and torsional moduli of these cable components as functions of their axial strain and twist are derived. The relations describing the mechanical properties of the cable in both elastic and plastic regions are obtained by substituting the tensile and torsional moduli of the high-strength cable components and the constant moduli for the low-strength cable components into the system of equations in differential form and integrating them. Application of the method to the present experimental undersea-lightguide cable yields excellent agreement with the tensile test results of the cable.*

## I. INTRODUCTION

Because of its long suspended length in deep ocean, undersea cable normally experiences high tension and strain during its installation and recovery.[1] High-strength stranded members are employed in the undersea cable design to support the tension and keep the cable strain below a prescribed level. The problems of high tension and strain cause even greater concerns for the undersea lightguide cable because of the

static and dynamic fatigue of optical fibers.[2] Adequate protection of the fibers is essential to system reliability and requires accurate evaluation of the mechanical properties of the cable in development.

Mechanical characterization of cable containing helically wrapped or stranded members has been done by Cannon and Santana[3] using linear elasticity theory, which assumes that the tensile modulas $E$ and Poisson's ratio $\nu$ are constant and independent of the cable load. Using two fundamental assumptions relating the external applied force and moment at the cable ends to the tension developed in individual helically stranded-strength members, a system of equations was developed that describes the mechanical characteristics of a cable having one or more layers of helically stranded members. Application of this theory to the present undersea lightguide cable yields excellent agreement with the experimental data up to a strain level of approximately 0.5 percent. Beyond this strain level, the theory predicts lower strain than the experimental data. This is not surprising because, at a strain level higher than 0.5 percent, the cable has been stretched beyond its elastic region to the plastic region, where the linear elasticity theory does not apply. Since undersea cable normally experiences high tension and strain, an understanding of the mechanical properties in both regions is essential to the design of undersea lightguide cable.

In this paper, a method is presented to accurately predict the mechanical properties of the cable in both elastic and plastic regions. Since the present method uses the system of equations developed in Ref. 3, the fundamental assumptions on which the system of governing equations is based will be examined and justified in Section II. The formulation of the new method is presented in Section III. Then the method is applied to the experimental undersea lightguide cable in Section IV. The results of the theory are compared with experimental data in Section V.

## II. FUNDAMENTAL ASSUMPTIONS

The mechanical properties of cables containing helically stranded members have been analyzed in Ref. 3 using a model consisting of $n$ identical wires of radius $b$ parallel to one another in a circular array and twisted into helices with a common helix angle $\alpha$ and radius $r$, as shown in Fig. 1. The fundamental equations relating the applied force $T_t$ and moment $M_t$ along the cable axis to the tension $T$ developed in individual wire were assumed to be

$$T_t = nT \sin \alpha \tag{1}$$

and

$$M_t = nTr \cos \alpha. \tag{2}$$

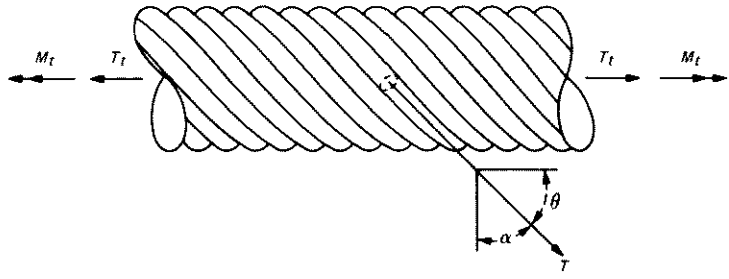



Fig. 1—Cable loads and wire tension.

From these assumptions, a system of equations was developed, which describe the mechanical properties of a cable containing helically wrapped elements.

However, from the equations of equilibrium of a bent and twisted thin rod, Costello and Phillips[4] have obtained a complete solution for the same problem illustrated in Fig. 1. The solution relates the applied tension and torque to the individual wire tension as

$$T_t = nT \sin \alpha + n \frac{\cos^3 \alpha}{r} \left| C\tau - A \frac{\sin \alpha \cos \alpha}{r} \right| \tag{3}$$

and

$$M_t = n \left| T\tau \cos \alpha - \cos^2 \alpha \sin \alpha \left| C\tau - A \frac{\sin \alpha \cos \alpha}{r} \right| + C\tau \sin \alpha + A \frac{\cos^3 \alpha}{r} \right|, \tag{4}$$

where A and C are constants, depending on the elastic properties of the material and the shape and dimensions of the rod cross-sectional area. $E$ is the tensile modulus, $\nu$ is Poisson's ratio, and, for a circular cross section of radius $b$,

$$A = \frac{\pi E b^4}{4} \quad \text{and} \quad C = \frac{\pi E b^4}{4(1 + \nu)}.$$

Notice that eqs. (3) and (4) can be reduced to (1) and (2) if the first term on the right of the equations is retained while neglecting the other terms. To neglect the second and third terms on the right-hand side of eq. (3) requires that

$$\frac{b^2 \tau \cos^3 \alpha}{4(1 + \nu) r \epsilon_s \sin \alpha} \ll 1 \tag{5}$$

and

$$\left|\frac{b^2\cos^2\alpha}{2r\sqrt{\epsilon_s}}\right|^2 \ll 1, \tag{6}$$

where $\epsilon_s$ is the strain of helical wire along its own axis. Similarly, to neglect the four terms on the right-hand side of eq. (4) requires that

$$\frac{\tau b^2\cos\alpha\sin\alpha}{4(1+\nu)rE_s} \ll 1, \tag{7}$$

$$\left|\frac{b\cos\alpha\sin\alpha}{2r\sqrt{\epsilon_s}}\right|^2 \ll 1, \tag{8}$$

$$\frac{\tau b^2\sin\alpha}{4(1+\nu)r\epsilon_s\cos\alpha} \ll 1, \tag{9}$$

and

$$\left|\frac{b\cos\alpha}{2r\sqrt{\epsilon_s}}\right|^2 \ll 1. \tag{10}$$

Inequalities (5), (6), (8), and (10) can easily be satisfied if $\alpha \simeq \pi/2$, i.e., for the large helix angle or small lay angle. However, to satisfy inequalities (7) and (9), the additional condition $\tau \simeq 0$ or $b^2/r \simeq 0$ is required.

In undersea cable design and manufacture, the helix angle $\alpha$ of the stranded members is usually close to $\pi/2$, and the members are actually made twist-free as they are formed into helices by a planetary-stranding machine. The condition $\tau \simeq 0$ can usually be satisfied. Therefore, the approximate relations (1) and (2) assumed in Ref. 3 and the resulting system of equations are justified.

## III. FORMULATION OF THE NEW METHOD

For a cable containing multiple layers of helically stranded strength members plus the cable core, the total axial load carried by the cable is the sum of the core load and the strand load, according to the following equation[3]

$$T_t = T_c + \sum_{i=1}^{m} n_i T_{si}\sin\alpha_i, \tag{11}$$

and the total cable moment equals the sum of the moment contributions from each of the strand layers plus that of the core, as shown by

$$M_t = -\psi J_c G_c + \sum_{i=1}^{m} |n_i T_{si} r_i\cos\alpha_i - \psi(n_i J_{si} G_{si}\sin^2\alpha_i)|, \tag{12}$$

where $n_i$ is the number of stranded wires in the $i$th layer; $T_c$ is the load experienced by the cable core; $T_{si}$ is the tension supported by the $i$th strand layer; $m$ is the number of strand layers in the cable; $\alpha_i$ is the common helix angle of the wires in the $i$th layer; $\psi$ is the untwist turns experienced per unit length of the cable under applied tension; $J_c$ and $G_c$ are the cable core's moment of inertia and torsional rigidity; $r_i$ is the radial location of the $i$th layer; $J_{si}$ and $G_{si}$ are the $i$th strand's moment of inertia and moment of rigidity. Note that eqs. (11) and (12) use the approximate relations (1) and (2) for the stranded wires.

Equations (11) and (12) can be transformed to the abbreviated forms

$$T_t = C_3\epsilon_c - C_4\psi \tag{13}$$

and

$$M_t = C_1\epsilon_c - C_2\psi, \tag{14}$$

where the coefficients $C_i$ are defined by

$$C_1 = \sum_{i=1}^{m} |\sin^2\alpha_i - \nu_i^*\cos^2\alpha_i \| n_i E_{si} A_{si} r_i \cos\alpha_i|, \tag{15}$$

$$C_2 = J_c G_c + \sum_{i=1}^{m} |n_i J_{si} G_{si} \sin^2\alpha_i + \pi r_i^2 \sin 2\alpha_i | n_i E_{si} A_{si} \cos\alpha_i \|, \tag{16}$$

$$C_3 = E_c A_c + \sum_{i=1}^{m} |\sin^2\alpha_i - \nu_i^*\cos^2\alpha_i \| n_i E_{si} A_{si} \sin\alpha_i|, \tag{17}$$

and

$$C_4 = \sum_{i=1}^{m} |\pi r_i \sin 2\alpha_i | n_i E_{si} A_{si} \sin\alpha_i. \tag{18}$$

Here, $\nu_i^*$ is a pseudo-Poisson's ratio for the $i$th layer and is a measure of its diametric contraction; $E_{si}$ and $A_{si}$ are the tensile modulus and cross-sectional area of the stranded member in the $i$th layer. For a cable with its ends twist-restrained, $\psi = 0$, and eqs. (13) and (14) become

$$\epsilon_c|_{\psi=0} = \frac{T_t}{C_3} \tag{19}$$

and

$$M_t|_{\psi=0} = C_1\epsilon_c. \tag{20}$$

For a cable with its ends free to rotate, $M_t = 0$, and eqs. (13) and (14) become

$$\epsilon_c|_{M_t=0} = \frac{T_t}{C_3 - \dfrac{C_1 C_4}{C_2}} \tag{21}$$

and

$$\psi|_{M_t=0} = \frac{C_1}{C_2}\epsilon_c. \tag{22}$$

If the cable characterization parameters $C_1$, $C_2$, $C_3$, and $C_4$ are known, the cable strain $\epsilon_c$, cable moment $M_t$, and cable twist can be evaluated from eqs. (19) through (22) for a given cable tension, $T_t$. If the elastic properties of cable components are all constant and the twist $\psi$ is zero or very small, the cable characterization parameters $C_1$, $C_2$, $C_3$, and $C_4$ are all constant and the relations among the cable strain, moment, twist, and tension from eqs. (19) to (22) are all linear. This is the case illustrated in Ref. 3.

In reality, however, the tensile moduli $E$'s and moduli of rigidity $G$'s of the cable components, either stranded or unstranded, are nonlinear functions of the cable strain $\epsilon_c$ and twist $\psi$, i.e., $E = E(\epsilon_c, \psi)$, $G = G(\epsilon_c, \psi)$, and thus the cable characterization parameters $C_1$, $C_2$, $C_3$, and $C_4$ are also functions of the cable strain $\epsilon_c$ and twist $\psi$.

Thus, eqs. (13) and (14) can only be applied in differential forms

$$dT_t = C_3 d\epsilon_c - C_4 d\psi \tag{23}$$

and

$$dM_t = C_1 d\epsilon_c - C_2 d\psi. \tag{24}$$

The above differential equations are meaningful only if the coefficients $C_1$, $C_2$, $C_3$, and $C_4$ can be expressed in terms of the cable strain $\epsilon_c$ and twist $\psi$. This can be accomplished in two steps. First, apply the regression analysis to the tensile and torsional test data of dominating cable components to obtain least-squares-fit polynominals, which approximate the stress versus strain and shearing-stress versus shearing-strain curves. By differentiating the polynominals, we obtain the tensile moduli and the moduli of rigidity of the cable components in terms of axial strain and twist:

$$\begin{aligned} E_{si} &= E_{si}(\epsilon_{si}) \\ E_c &= E_c(\epsilon_c) \\ G_{si} &= G_{si}(\psi_{si}) \\ G_c &= G_c(\psi), \end{aligned} \tag{25}$$

where $\epsilon_{si}$ and $\psi_{si}$ are the strain and twist of the stranded member in the direction of its own axis. Second, transform the independent variables $\epsilon_{si}$ and $\psi_{si}$ in (25) into the cable strain $\epsilon_c$ and twist $\psi$ using the following relations:[3]

$$\epsilon_{si} = |\sin^2\alpha_i - \nu_i^* \cos^2\alpha_i|\epsilon_c - |l_i \cos^2\alpha_i|\psi \tag{26}$$

$$\psi_{si} = \psi \sin \alpha_i, \tag{27}$$

where $l_i$ is the lay length of the $i$th stranded layer. Substitution of the above equations into (23) and (24) yields

$$dT_t = C_3|\epsilon_c, \psi|d\epsilon_c - C_4|\epsilon_c, \psi|d\psi \tag{28}$$

and

$$dM_t = C_1|\epsilon_c, \psi|d\epsilon_c - C_2|\epsilon_c, \psi|d\psi. \tag{29}$$

For a cable with its ends twist-restrained, $\psi = 0$, and eqs. (28) and (29) become

$$T_t = \int C_3(\epsilon_c)d\epsilon_c \tag{30}$$

and

$$M_t = \int C_1(\epsilon_c)d\epsilon_c. \tag{31}$$

Since $C_1(\epsilon_c)$ and $C_2(\epsilon_c)$ are polynominals, the above equations can readily be integrated. For a cable with its ends free to rotate, $M_t = 0$, and eqs. (28) and (29) become

$$T_t = \int \left| C_3(\epsilon_c, \psi) - \frac{C_1(\epsilon_c, \psi)C_4(\epsilon_c, \psi)}{C_2(\epsilon_c, \psi)} \right| d\epsilon_c \tag{32}$$

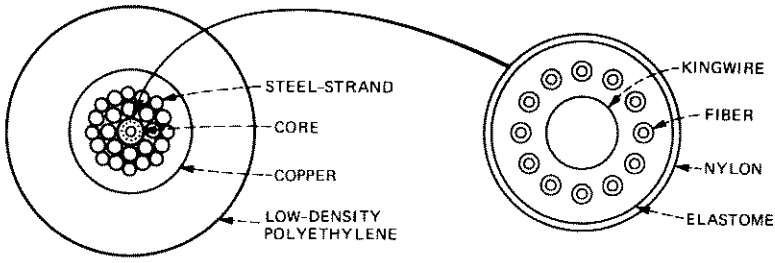
and

$$\frac{d\psi}{d\epsilon_c} = \frac{C_1(\epsilon_c, \psi)}{C_2(\epsilon_c, \psi)}. \tag{33}$$

In this case, the differential eq. (33) should be solved first to obtain a relation between $\psi$ and $\epsilon_c$. This relation is applied to eq. (32) to eliminate $\psi$. Then, eq. (32) can be integrated to give a relation between $T_t$ and $\epsilon_c$.

## IV. APPLICATION

The recovery operation induces higher tensions in undersea cable than any other handling operation. During a steady-state recovery, the cable tension is highest at the overboard sheave, where the cable twist is practically zero. Therefore, only the case of twist restraint, $\psi = 0$, will be studied here. The method is applied to an experimental undersea lightguide cable to characterize its tensile and torsional properties. The cable structure and component dimensions are shown in Fig. 2. The cable consists of a lightguide core protected by two layers of high-strength steel-strand wires, a copper conductor, and a low-density polyethylene (LDPE) jacket for high-voltage insulation and environmental protection. The lightguide core consists of a copperplated-steel




**CABLE STRUCTURE:**
STRAND DIAMETER = 7.9 mm
CONDUCTOR OD (COPPER) = 10.5 mm
INSULATION OD = 21.0 mm

**CABLE CORE:**
CENTER WIRE OD = 0.8 mm
NUMBER OF FIBERS = 12
FIBER OD (COATED) = 250 m
SHEATH THICKNESS = 0.1 mm
CORE OD = 2.6 mm

Fig. 2—Undersea cable design.

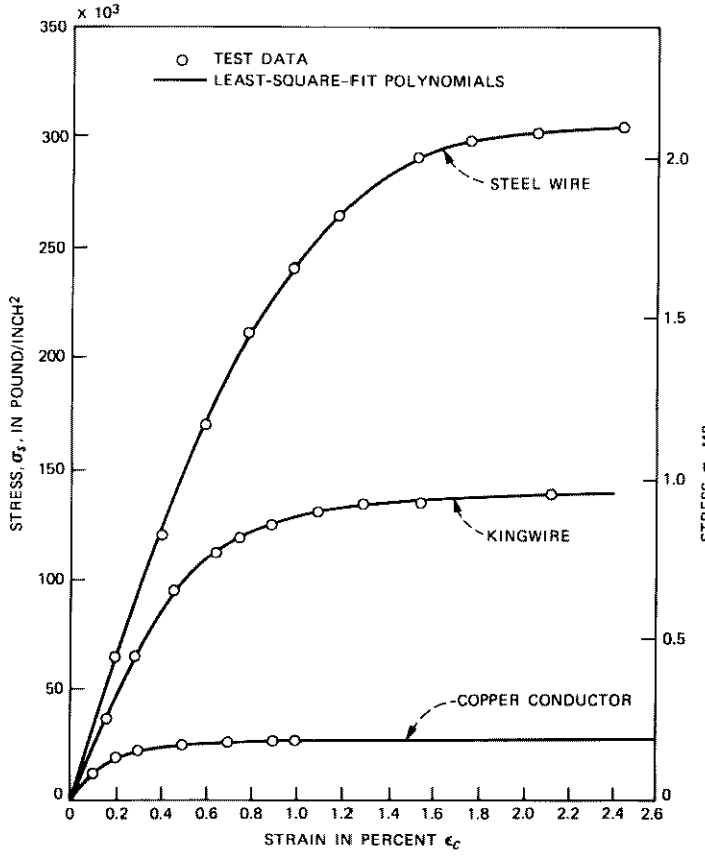



Fig. 3—Tensile testing results and the least-squares-fit polynomials of the high-strength steel-wire, copper sheath, and kingwire.

conductor called the kingwire, up to 12 helically wound fibers embedded in an elastomer material, and a thin outer nylon sheath.

From the cable structure, the metallic components, i.e., the steel wires, the copper conductor, and the kingwire, will carry most of the tensile load. Several tensile tests were conducted on individual steel wires of different sizes and on the kingwire. The test results of a representative 1.6-mm-diameter steel wire and a 0.81-mm-diameter kingwire are shown in Fig. 3. A thorough experimental study on the tensile behavior of the copper conductor was previously done in an unpublished work by R. C. Mondello of Bell Laboratories. His result on newly made copper conductors is also shown in Fig. 3. The nonlinear behavior of each component is evident at high load. Application of the regression analysis to each set of test data in Fig. 3 yields a least-squares-fit polynominal of sixth degree, approximating the stress versus strain relation for each material. The polynomials with the stresses $\sigma_s$ in $MP_a$ and $\epsilon_c$ in mm/mm are given below.

### 4.1 Steel wire

The polynomial for the steel wires is

$$\sigma_s = 0.663587607686 + 214542.346398\ \epsilon_s + 3198478.98934\ \epsilon_s^2 - 1617068166.73\ \epsilon_s^3 + 111346531878\ \epsilon_s^4 - 3.45797334075 \times 10^{12}\ \epsilon_s^5 + 4.21128175626 \times 10^{13}\ \epsilon_s^6\ (MP_a), \quad (34)$$

where the standard error for estimate "CHI" is CHI = 5.523 $(MP_a)$.

### 4.2 Kingwire

The polynomial for the kingwire is

$$\sigma_s = 1.15682945607 + 162675.763595\ \epsilon_s + 7250437.88515\ \epsilon_s^2 - 4367711589.93\ \epsilon_s^3 + 459662005052\ \epsilon_s^4 - 2.03575240495 \times 10^{13}\ \epsilon_s^5 + 3.3234268429 \times 10^{14}\ \epsilon_s^6\ (MP_a), \quad (35)$$

where the standard error for estimate "CHI" is CHI = 5.871 $(MP_a)$.

### 4.3 Copper conductor

The polynomial for the copper conductor is

$$\sigma_s = 0.299934161316 + 110226.149362\ \epsilon_s - 28633487.4233\ \epsilon_s^2 + 3792414452.98\ \epsilon_s^3 - 264983411326\ \epsilon_s^4 + 9.25874279974 \times 10^{12}\ \epsilon_s^5 - 1.27105877316 \times 10^{14}\ \epsilon_s^6\ (MP_a), \quad (36)$$

where the standard error for estimate "CHI" is CHI = 1.780 $(MP_a)$.

Table I—Mechanical properties of cable components

| Cable Components | Tensile Modulus, $E(GP_a)$ | Shear Modulus, $C(GP_a)$ | No. of Strand Members, $N$ | Helix Radius, $r$ (mm) | Helix Angle, $\alpha$ (degree) | Cross-Sectional Area ($mm^2$) | Polar Moment of Inertia ($mm^4$) |
|---|---|---|---|---|---|---|---|
| Kingwire | 158.58 | 61.02 | — | — | — | $16.58 \times 10^{-2}$ | $2.14 \times 10^{-2}$ |
| Elastomer | 0.05 | 0.03 | — | — | — | 3.87 | 2.19 |
| Glass fiber | 71.71 | 31.51 | 12 | 0.79 | 88.8 | $0.95 \times 10^{-2}$ | $7.2 \times 10^{-6}$ |
| Nylon | 1.24 | 0.44 | — | — | — | $26.32 \times 10^{-2}$ | 0.68 |
| 1st layer steel wires (1.6-mm dia.) | 206.84 | 79.29 | 8 | 2.08 | 86.8 | 2 | 0.32 |
| 2nd layer steel wires (1.48-mm dia.) | 206.84 | 79.29 | 8 | 3.27 | 83.9 | 1.74 | 0.24 |
| 2nd layer steel wires (1.135-mm dia.) | 206.84 | 79.29 | 8 | 3.51 | 83.9 | 1.03 | 0.16 |
| Copper sheath | 117.21 | 44.13 | — | — | — | 12.64 | 116.54 |
| LDPE | 0.24 | 0.08 | — | — | — | $2.82 \times 10^2$ | $9.24 \times 10^3$ |

The tensile moduli $E(\epsilon_s)$'s of each component can be obtained by taking the derivative of the corresponding polynomial

$$E(\epsilon_s) = \frac{d\sigma_s}{d\epsilon_s}. \tag{37}$$

For the stranded-steel wires and fibers, the independent variables $\epsilon_s$ can be transformed to $\epsilon_c$ using eq. (28); for the copper conductor and kingwire, $\epsilon_s$ can be directly replaced by $\epsilon_c$ since their axial direction coincides with that of the cable. Substitutions of the tensile moduli of the components into eq. (15) and (17) give

$$\begin{aligned} C_1(\epsilon_c) &= |\sin^2\alpha_f - \nu_f^* \cos^2\alpha_f| n_f E_f A_f r_f \cos\alpha_f \\ &\quad + \sum_{i=1}^{4} |\sin^2\alpha_i - \nu_i^* \sin^2\alpha_i| n_i E_{si}(\epsilon_c) A_{si} r_{si} \cos\alpha_i| \\ C_3(\epsilon_c) &= E_k(\epsilon_c) A_k + E_e A_e + E_n A_n + E_c(\epsilon_c) A_c + E_p A_p \\ &\quad + |\sin^2\alpha_f - \nu_f^* \cos^2\alpha_f| n_f E_f A_f \sin\alpha_f|, \\ &\quad + \sum_{i=1}^{4} |\sin^2\alpha_i - \nu_i^* \cos^2\alpha_i| n_i E_{si}(\epsilon_c) A_{si} \sin\alpha_i|, \end{aligned}$$

where the subscripts $k$, $e$, $n$, $c$, $p$, and $f$ refer to the kingwire, elastomer, nylon, copper, LDPE, and fibers, respectively. The tensile moduli of the fibers, elastomer, nylon, and LDPE are all assumed constant and are listed in Table I. Substitution of the above expressions for $C_1(\epsilon_c)$ and $C_3(\epsilon_c)$ into eqs. (29) and (30) and integrating give the tensile and torsional properties of the cable under twist-restraint condition. These functions are calculated and the results are shown as solid lines in Figs. 4 and 5. The dashed lines in these figures are the results using constant-elastic moduli listed in Table I.

## V. EXPERIMENTAL RESULTS

Tensile tests were conducted for five cable specimens, 100.5-meter-length each. The cable ends of the cable specimen were potted into tapered-plug epoxy terminations. The cable specimen with the terminations was installed in the tensile test bed in a straight line configuration with one end fixed to the anchor and another end attached to the hydraulic ram. Both ends were twist-restrained. The cable tension was increased in steps from 8.9 to 53.4 kN, then in steps from 4.5 to the maximum load of 62.3 kN. The cable tension and moment were measured by a load cell. The cable elongation was measured by using a 10-turn, 10-kΩ linear potentiometer. All data were monitored and displayed in digital form on a control panel and then recorded.

The experimental results are also shown in Figs. 4 and 5. The

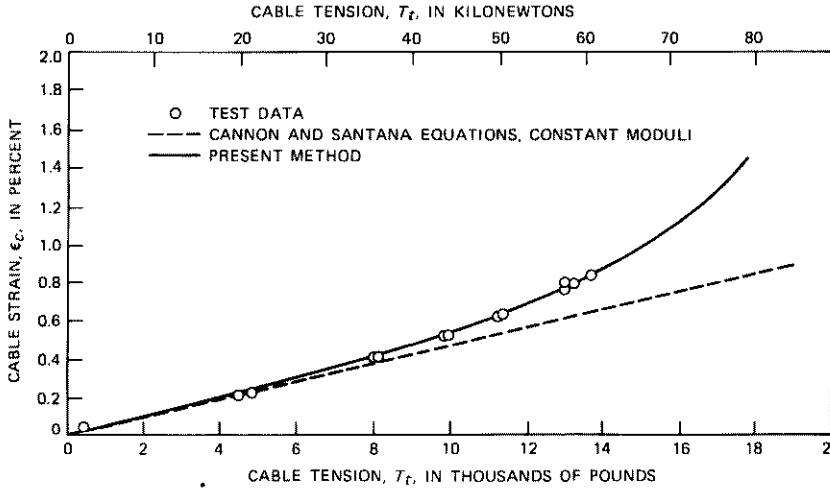



Fig. 4—Comparison of theoretical and experimental results for cable strain versus cable tension.

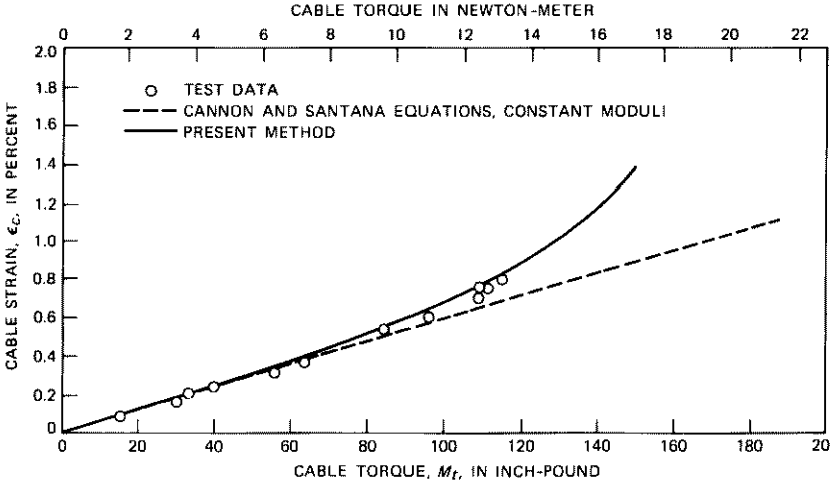



Fig. 5—Comparison of theoretical and experimental results for cable strain versus cable torque.

calculated results from the present method and the experimental results are in excellent agreement.

## VI. CONCLUSION

A method has been formulated to evaluate the mechanical properties of cables containing stranded-strength members, in both linear elastic and nonlinear plastic regions. Application of the method to the exper-

imental undersea lightguide cable shows excellent agreement to the tensile test results of the cable. This method can be applied to future cable design to evaluate the cable breaking strength, cable tension versus strain, cable moment versus strain, and cable twist versus strain in both linear elastic and nonlinear plastic regions.

## VII. ACKNOWLEDGMENTS

The author would like to thank his former teacher, Professor W. R. Sears, for his guidance during the graduate school years at Cornell University.

The author also thanks T. C. Cannon, M. R. Santana, and R. D. Tuminaro for useful discussions during the course of the study, and H. M. Brinser and D. A. Meade, for conducting the tests.

## REFERENCES

1. E. E. Zajac, "Dynamics and Kinematics of the Laying and Recovery of Submarine Cable," B.S.T.J. *35*, No. 5 (September 1957), pp. 1129–207.
2. *Optical Fiber Telecommunications*, S. E. Miller and A. G. Chynoweth, eds., New York: Academic Press, 1979, Chapter 12.
3. T. C. Cannon and M. R. Santana, "Mechanical Characterization of Cable Containing Helically Wrapped Reinforcing Elements," Twenty-fourth Int. Wire and Cable Symp., Cherry Hill, New Jersey, 1975.
4. G. A. Costello and J. W. Phillips, "Contact Stresses in Thin Twisted Rods," J. Appl. Mech., *40* (June 1973), pp. 629–30.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# Performance of a Fast Algorithm for FIR System Identification Using Least-Squares Analysis

By S. L. MARPLE, JR.,* and L. R. RABINER

(Manuscript received August 31, 1982)

*A wide variety of procedures have been proposed for identifying a finite impulse response (FIR) linear system from the input and output of the system. Most recently, a fast, efficient, least-squares method was proposed by Marple, and was shown to require less computation and storage than any other known procedure for identifying moderate to large FIR systems. In this paper we measure the actual performance of the newly proposed fast system identification algorithm by using it to estimate a variety of FIR systems excited by either white noise or a speech signal. It is shown that essentially theoretically ideal performance is achieved for white noise inputs; however, for speech signals poor performance was obtained because of the lack of certain frequency bands in the excitation. A simple modification to the estimation procedure is proposed and is shown to provide substantial performance improvements. Using the spectrally modified speech signal, the performance of the fast system identification algorithm was found to be acceptable for a wide variety of applications.*

## I. INTRODUCTION

In previous papers,[1-4] two system identification methods, the classical least-squares analysis (LSA) and a short-time spectral analysis (SSA) procedure, had their performance compared and contrasted in the presence of high noise levels and in situations where the input signal was band-limited (nonwhite noise and speech). This earlier work found that, while the LSA method produced better performance than the SSA procedure, the computational burden of the Cholesky solution of the equations of the LSA method became prohibitive when com-

* Schlumberger Well Services, Houston, Texas.

pared to the SSA method as the system order became large (as it can be in speech processing, where the filter order can be on the order of 1000). This factor weighed in favor of the SSA method. However, the development of fast algorithms for the solution of the LSA normal equations[5-8] has greatly reduced the computational complexity. This paper evaluates the performance of the LSA procedure in the context of these fast algorithms.

The fast algorithm that we will consider for solving normal equations for the least-squares FIR system identification has computational complexity proportional to $M^2$, where $M$ is the filter order, and storage that grows linearly (rather than quadratically) with $M$. A byproduct of the computation is an estimate of the linear prediction coefficients of the input process. The fast algorithm also has simple, built-in, numerical ill-conditioning checks. If the model order is uncertain, the fast algorithm recursively provides all optimum least-squares solutions from filter order 1 up to some user-selected maximum order, $M$, thereby providing a built-in search capability for finding the appropriate system order without having to start over.

The outline of this paper is as follows. In Section II we review the normal equations of the least-squares system identification and show how a fast algorithm can be derived to solve these equations. In Section III we present an evaluation of the performance of the fast algorithm for several different FIR systems with both broadband noise and speech inputs. In Section IV we review the results and compare them to those obtained previously using short-time spectral analysis methods.

## II. REVIEW OF THE LEAST-SQUARES NORMAL EQUATIONS

Figure 1 shows a block diagram of the finite-impulse-response system identification model. The discrete input signal, $x(n)$, drives the unknown system to produce the discrete sequence, $y(n)$, where $n$ is an integer index. The unknown system was modeled as an FIR filter, assumed to have an impulse response duration of $M + 1$ samples, so that the estimated impulse response, $\hat{h}(n)$, is zero for $n < 0$ and $n > M$. The order $M$ of the FIR system is defined here as the highest index of the impulse response, $\hat{h}(M)$.

The approach used is to determine the impulse-response coefficients, $\hat{h}(n)$, and the system order $M$ that minimize the squared error based on a finite number of measurements of the input process, $x(n)$, and the output process, $y(n)$. Denoting the linear estimate of $y(n)$ by $\hat{y}(n)$, then

$$\hat{y}(n) = \sum_{m=0}^{M} \hat{h}(m)x(n-m), \tag{1}$$

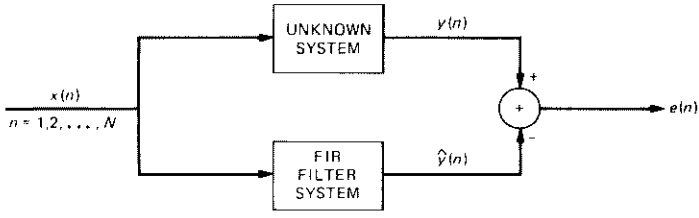



Fig. 1—Block diagram of the FIR system identification.

which is the familiar convolution expression of a finite-impulse-response filter. The estimation error, $e(n)$, is then

$$e(n) = y(n) - \hat{y}(n). \tag{2}$$

We wish to minimize the total squared error, $P_M$, of an $M$th-order model

$$P_M = \sum_n e^2(n) \tag{3}$$

with respect to $\hat{h}(0), \cdots, \hat{h}(M)$ and based on the block of finite data sets $x(1), \cdots, x(N)$ and $y(1), \cdots, y(N)$. Note that the index range for $n$ in eq. (3) has purposely been left unspecified. To operate only on the available data samples, the range must be selected to be $n = M + 1$ to $N$, which is the index range selected for this paper because it provides the best performance for relatively short data segments. However, by defining the unobserved data to be zero for $n \leq 0$, then a so-called "prewindowed" index range of $n = 1$ to $N$ can be used [the data $x(n)$ and $y(n)$ for $n \leq 0$ is "windowed" to zero]. These two cases are illustrated in eq. (4) using a matrix structure to describe the error terms:

$$\begin{Bmatrix} 2 \\ 1 \end{Bmatrix} \left\{ \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{M+1} \\ \vdots \\ e_N \end{bmatrix} \right\} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{M+1} \\ \vdots \\ y_N \end{bmatrix} - \begin{Bmatrix} 2 \\ 1 \end{Bmatrix} \left\{ \begin{bmatrix} x_1 & 0 \\ x_2 & x_1 \\ \vdots & \vdots \\ x_{M+1} & x_M \cdots x_1 \\ \vdots & \vdots \\ x_N & x_{N-1} \cdots x_{N-M} \end{bmatrix} \right\} \begin{bmatrix} \hat{h}_0 \\ \hat{h}_1 \\ \vdots \\ \hat{h}_M \end{bmatrix} \tag{4a}$$

or

$$E_M = Y_M - X_M \hat{H}_M \tag{4b}$$

and

$$P_M = E_M^T E_M, \tag{5}$$

where $E_M$, $Y_M$, $\hat{H}_M$ are column vectors and $X_M$ is a rectangular Toeplitz data matrix. The $T$ denotes matrix transpose. If $X_M$ includes only that portion within the 1 brace, this corresponds to the index range $n = M + 1$ to $N$. If $X_M$ includes that portion indicated within the 2 brace, this corresponds to the prewindowed index range $n = 1$ to $N$.

If $P_M$ is now minimized by setting the derivatives with respect to $\hat{h}(0), \cdots, \hat{h}(M)$ to zero, then the resulting least-squares solution can be expressed in matrix form as

$$\Phi_M^{xx} \hat{H}_M = \Phi_M^{yx}, \tag{6}$$

where

$$\Phi_M^{xx} = X_M^T X_M = \text{an } (M+1) \times (M+1) \text{ matrix}$$

$$\Phi_M^{yx} = X_M^T Y_M = \text{an } (M+1) \text{ column vector.}$$

This is the discrete Wiener-Hopf equation. The minimum squared error is

$$\min P_M = Y_M^T Y_M - (\Phi_M^{yx})^T \hat{H}_M. \tag{7}$$

Note that this solution is applicable no matter what index range is selected. Also note that while the matrix $\Phi_M^{xx}$ is not, in general, Toeplitz, it is the product of two rectangular Toeplitz data matrices. This will prove to be a key factor in developing a fast algorithm solution for eq. (6). For the index range of interest here, individual elements of $\Phi_M^{xx}$ are

$$\phi_M^{xx}(i, j) = \sum_{n=M+1}^{N} x(n-j)x(n-i) \qquad \text{for } 0 \le i, j \le M. \tag{8}$$

The elements of $\Phi_M^{yx}$ for the unwindowed index range are

$$\phi_M^{yx}(i) = \sum_{n=M+1}^{N} y(n)x(n-i) \qquad \text{for } 0 \le i \le M. \tag{9}$$

Also, we have

$$Y_M^T Y_M = \sum_{n=M+1}^{N} y^2(n). \tag{10}$$

Note that the number of data samples must be at least twice the system order plus one, $N \ge 2M + 1$, in order for $\Phi_M^{xx}$ to be nonsingular.

It is also possible to perform a linear-phase FIR system identification by forcing symmetry in the filter. The linear-phase estimate is then

$$\hat{y}(n) = \hat{h}(0)x(n) + \sum_{m=1}^{M} \hat{h}(m)[x(n+m) + x(n-m)] \tag{11}$$

with error

$$\epsilon(n) = y(n) - \hat{y}(n). \tag{12}$$

Note that the total impulse-response duration of this filter is $2M + 1$. Forming the sum of squared errors over all valid error terms

$$Q_M = \sum_{n=M+1}^{N-M} \epsilon^2(n) \tag{13}$$

and minimizing leads to the normal equations

$$\Psi_M^{xx} \hat{H}_M = \Psi_M^{yx},$$

where

$$\hat{H}_M = \begin{bmatrix} \hat{h}(M) \\ \vdots \\ \hat{h}(1) \\ \hat{h}(0) \\ \hat{h}(1) \\ \vdots \\ \hat{h}(M) \end{bmatrix}, \qquad \Psi_M^{yx} = \begin{bmatrix} r^{yx}(M) \\ \vdots \\ r^{yx}(1) \\ r^{yx}(0) \\ r^{yx}(1) \\ \vdots \\ r^{yx}(M) \end{bmatrix},$$

$$\Psi_M^{xx} = \begin{bmatrix} r_M^{xx}(0,0) & \cdots & r_M^{xx}(0,2M) \\ \vdots & & \vdots \\ r_M^{xx}(2M,0) & \cdots & r_M^{xx}(2M,2M) \end{bmatrix}$$

$$r_M^{xx}(j,k) = \sum_{n=1}^{N-2M} [x(n+j)x(n+k) + x(n+2M-j)x(n+2M-k)]$$

$$r_M^{yx}(k) = \sum_{n=M+1}^{N-M} y(n)[x(n+k) + x(n-k)] \qquad 0 \le k \le M$$

with minimum squared error

$$Q_M = \sum_{n=M+1}^{N-M} y^2(n) - \frac{1}{2}\hat{h}(0)r_M^{yx}(0) - \sum_{m=1}^{M} \hat{h}(m)r_M^{yx}(m).$$

As before, in order for the normal equations to be nonsingular, the number of data samples must be at least twice the system order ($2M$ in this case) plus one, or $N \ge 4M + 1$.

### 2.1 Basis for a fast algorithm solution

In this section, a brief outline of the basis for a fast algorithm that solves eq. (6) with a number of operations proportional to $M^2$ is presented. For details of the full algorithm, consult Ref. 5. A fast algorithm also exists for the linear-phase FIR system identification,[8]

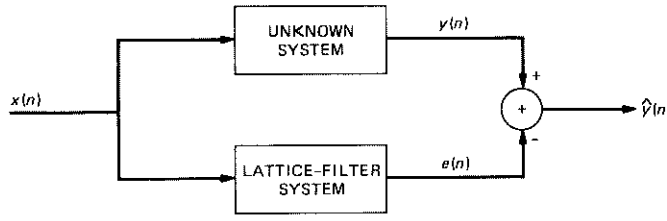



Fig. 2—Block diagram of the lattice system identification.

but will not be discussed here since the focus is on the general FIR system identification algorithm.

The fast algorithm presented here processes the available data as a block. The algorithm recursively generates all solutions from order $m = 1$ to $M$, where $M$ is the maximum order. This algorithm is just one of a wider class of fast algorithms for solutions of least-squares prediction problems. If processing data on a sequential sample-by-sample basis is preferred to block processing of the data (such as for adaptive equalizer applications), then an alternative, but numerically equivalent, solution to eq. (6) is to use a lattice filter in lieu of the FIR filter. The lattice-filter output is $e(n)$ rather than $\hat{y}(n)$ (the FIR output), as shown in Fig. 2. An example of such a sequential fast algorithm for the "prewindowed" data range is presented in Ref. 6.

The key to a fast algorithm for solution of eq. (6) is the recognition that the matrix $\Phi_M^{xx}$ appears in the context of the linear prediction problem. The linear prediction filter tries to make the best estimate of the current value of $x(n)$ based on past samples of $x(n)$,

$$\tilde{x}(n) = -\sum_{m=1}^{M} a(m)x(n-m). \tag{14}$$

The prediction error $e^f(n)$ is then

$$e^f(n) = x(n) - \tilde{x}(n) = \sum_{m=0}^{M} a(m)x(n-m), \tag{15}$$

where $a(0) = 1$ and the superscript $f$ denotes this as the "forward" linear prediction filter error. A "backward" estimate $\tilde{\tilde{x}}(n)$ can also be defined as

$$\tilde{\tilde{x}}(n-M) = -\sum_{m=1}^{M} b(m)x(n+m-M) \tag{16}$$

with "backward" prediction error

$$e^b(n) = x(n-M) - \tilde{\tilde{x}}(n-M) = \sum_{m=0}^{M} b(m)x(n+m-M),$$

where $b(0) = 1$. If we now minimize the forward squared prediction error

$$P_M^f = \sum_{n=M+1}^{N} [e^f(n)]^2$$

and the backward squared prediction error

$$P_M^b = \sum_{n=M+1}^{N} [e^b(n)]^2;$$

then we obtain the usual least-squares linear-prediction normal equations

$$\Phi_M^{xx} A_M = \begin{pmatrix} P_M^f \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \Phi_M^{xx} B_M = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ P_M^b \end{pmatrix}, \tag{17}$$

where vectors $A_M$ and $B_M$ are defined as

$$A_M = \begin{bmatrix} 1 \\ a(1) \\ \vdots \\ a(M) \end{bmatrix}, \qquad B_M = \begin{bmatrix} b(M) \\ \vdots \\ b(1) \\ 1 \end{bmatrix}.$$

Equation (17) is the so-called "covariance" equation of linear-prediction speech analysis. An efficient, recursive algorithm for their solution has been previously presented[7] and is incorporated as part of the FIR fast algorithm without further discussion. Equation (17) can be rewritten as

$$\Phi_M^{xx} \mathscr{A}_M = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \Phi_M^{XX} \mathscr{B}_M = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \tag{18}$$

where

$$\mathscr{A}_M = \begin{bmatrix} 1/P_M^f \\ a(1)/P_M^f \\ \vdots \\ a(M)/P_M^f \end{bmatrix}, \qquad \mathscr{B}_M = \begin{bmatrix} b(M)/P_M^b \\ \vdots \\ b(1)/P_M^b \\ 1/P_M^b \end{bmatrix}.$$

Thus, $\mathscr{A}_M$ and $\mathscr{B}_M$ form the first and last columns of the inverse of the $\Phi_M^{xx}$ matrix. Since the solution to eq. (12) involves the inverse $[\Phi_M^{xx}]^{-1}$,

$$\hat{H}_M = (\Phi_M^{xx})^{-1} \Phi_M^{xy}, \tag{19}$$

then we would suspect that the linear-prediction solution is an integral part of the system identification solution [eq. (19)]. This is indeed the case. In fact, only the first and last columns of the inverse (or alternatively, the vectors $A_M$ and $B_M$) of $\Phi_M^{xx}$ are required to obtain a recursive solution for $\hat{H}_M$.

Three fundamental relationships govern the ability to obtain a recursive algorithm. To illustrate these relationships, it is first necessary to combine eqs. (5) and (6) into a single augmented expression as follows:

$$\bar{\Phi}_M \bar{H}_M = \bar{P}_M, \tag{20}$$

where

$$\bar{\Phi}_M = \underbrace{\left[\begin{array}{c|c} \phi_M^{yy}(0) & (\Phi_M^{yx})^T \\ \hline \Phi_M^{yx} & \Phi_M^{xx} \end{array}\right]}_{M+2} \Bigg\} M+2$$

$$\bar{H}_M = \left[\begin{array}{c} 1 \\ \hline -\hat{H}_M \end{array}\right] \Bigg\} M+2$$

$$\bar{P}_M = \left[\begin{array}{c} P_M \\ 0 \\ \vdots \\ 0 \end{array}\right] \Bigg\} M+2, \qquad \phi_M^{yy}(0) = Y_M^T Y_M.$$

An alternative relationship for $\bar{\Phi}_M$ is

$$\bar{\Phi}_M = \sum_{n=M+1}^{N} \bar{X}_M(n) \bar{X}_M^T(n), \tag{21}$$

where $\bar{X}_M(n)$ is the vector

$$\bar{X}_M(n) = \left[\begin{array}{c} y(n) \\ x(n) \\ \vdots \\ x(n-M) \end{array}\right].$$

We may obtain the three basic partitions:

$$\bar{\Phi}_M = \underbrace{\left[\begin{array}{c|c} \phi_M^{yy}(0) & (\Phi_M^{yx})^T \\ \hline \Phi_M^{yx} & \Phi_M^{xx} \end{array}\right]}_{\begin{array}{c|c} 1 & M+1 \end{array}} \begin{array}{l} \}1 \\ \}M+1 \end{array} \tag{22}$$

$$\Phi_M = \underbrace{\left[\begin{array}{c|c} \bar{\Phi}'_{M-1} & W_M \\ \hline W_M^T & \phi_M^{xx}(M,M) \end{array}\right]}_{\begin{array}{c|c} M+1 & 1 \end{array}} \begin{array}{l} \}M+1 \\ \}1 \end{array} \tag{23}$$

$$\bar{\Phi}'_{M-1} = \bar{\Phi}_{M-1} - \bar{X}_{M-1}(M) \bar{X}_{M-1}^T(M), \tag{24}$$

where the definition of $W_M$ is

$$W_M = \sum_{n=M+1}^{N} x(n-M)\bar{X}_{M-1}(n).$$

All the recursive relationships in the algorithm have their roots in eqs. (22) through (24). Using these equations, it is possible to derive the following key recursive relationships for the system identification parameters

$$\bar{H}'_{M-1} = \bar{H}_{M-1} + \frac{e(M)}{(1-\delta_{M-1})}\begin{pmatrix} 0 \\ D_{M-1} \end{pmatrix} \text{ time update} \tag{25}$$

$$\bar{H}_M = \begin{pmatrix} \bar{H}'_{M-1} \\ 0 \end{pmatrix} + \frac{\alpha_M}{P_M^b}\begin{pmatrix} 0 \\ B_M \end{pmatrix} \text{ order update,} \tag{26}$$

where $\alpha_M$, $\delta_M$ are scalar gain values, $B_M$ is the vector of backward linear-prediction coefficients, and $D_M = (\Phi_M^{xx})^{-1}X_M(M+1)$ is a vector, all of which are obtained as part of the linear-prediction-algorithm solution. Equations (25) and (26) highlight the intertwined relationship of the linear-prediction fast algorithm with that of the fast algorithm for the system identification solution. One also can see how all lower-order solutions are obtained recursively along the way to the final selected order.

Counting only second-order terms, the number of multiplications required in the fast algorithm is $2NM + 12M^2$, the number of adds is $2NM + 9M^2$, the number of divides is $8M$, and a total storage of $2N + 7M + 20$ parameters is required (including input and output data sequences). Here $N$ is the number of data values and $M$ is the system order. Roughly $NM^2$ operations are required to directly solve eq. (6) by Cholesky decomposition if the fast algorithm described here is not used. The Cholesky technique also requires storage proportional to $M^2$, whereas the fast algorithm requires storage that only increases linearly with increasing $M$. Exact operation counts for the Cholesky method are given in the appendix. A simple numerical ill-conditioning check in the algorithm is to verify that the scalar variable $\delta_M$ in eq. (26) is in the range $0 \leq \delta_M < 1$, which is required by the mathematics of the solution. If it is not in this range, then the fast algorithm recursion has become ill-conditioned.

FORTRAN computer code of the general FIR system identification algorithm may be found in Ref. 5. Code for the linear-phase FIR algorithm may be found in Ref. 8.

## III. PERFORMANCE EVALUATION OF THE FAST ALGORITHM

To evaluate the performance of the fast, least-squares, FIR system identification algorithm that solves eq. (6), the system of Fig. 3 was

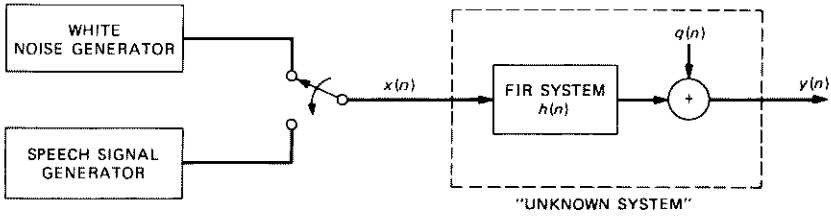



Fig. 3—Block diagram of excitation generation and signal generation for identifying an unknown system.

used to generate the required input and output signals. The signal source was either a white noise signal or a speech signal. The signal source was passed through a given $M$th-order FIR system with known impulse response, $h(n)$, $n = 0, 1, \cdots, M$. An additive, random, white noise $q(n)$ was added to the output of the FIR system, at a specified signal-to-noise (s/n) ratio, giving the output signal, $y(n)$. The system was assumed to be in steady state whenever $y(n)$ and $x(n)$ were sampled for system identification purposes [i.e., no initial transients were present in $y(n)$].

The performance measure used in this study was the logarithmic misalignment error defined in Ref. 1 as

$$Q(N, \text{s/n}) = 10 \log_{10} \left[ \frac{\sum_{m=0}^{M} [h(m) - \hat{h}(m)]^2}{\sum_{m=0}^{M} h^2(m)} \right], \tag{27}$$

where $\hat{h}(m)$ is a function of $N$ and s/n, and $M$ is the true FIR system order. Note that $h(m)$ are the true FIR coefficients and $\hat{h}(m)$ are the estimated FIR coefficients obtained from the fast algorithm. For nonwhite input signals, a weighted $Q$ function can also be defined using the estimated linear-prediction coefficients, $a(n)$, available as part of the fast algorithm. See Ref. 2 for details.

To fully evaluate the performance of the fast algorithm, five different FIR filters were used, including:

(*i*) Filter 1—A 7-point, nonlinear-phase filter with about 10 dB of spectral deviation across the frequency range. Figure 4 shows plots of the impulse and frequency responses of this filter.

(*ii*) Filter 2—A 25-point, linear-phase, equiripple low-pass filter with about 54 dB of stopband rejection for frequencies above 0.2 times the sampling frequency. Figure 5 shows the impulse and frequency responses of this filter.

(*iii*) Filter 3—A 64-point, nonlinear-phase, reverberation filter with about 30 dB of spectral variation. This filter had an impulse response

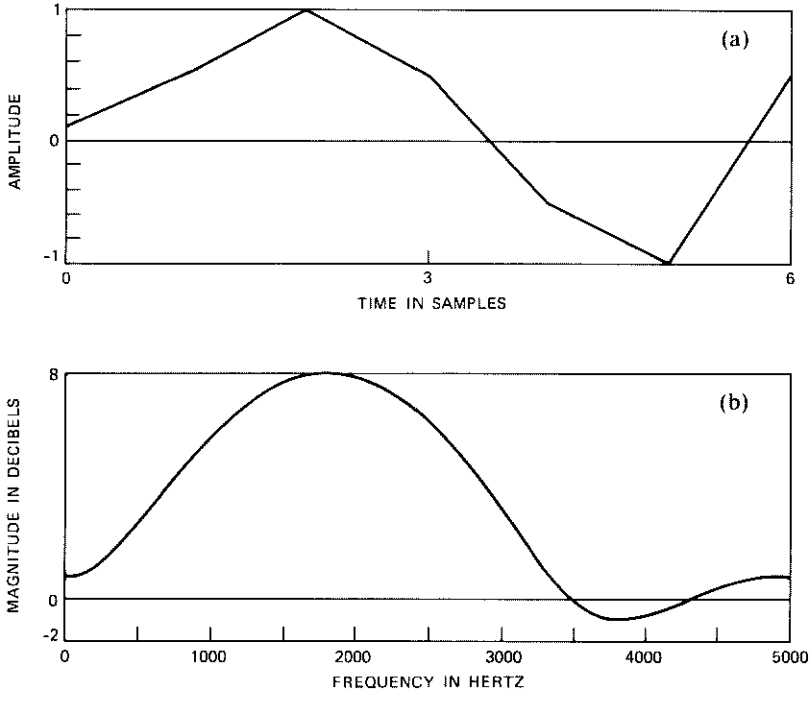



Fig. 4—(a) Impulse response and (b) frequency response of 7-point FIR filter.

that was zero except for $n = 0, 1, 3, 7, 15, 31$, and 63, at which $h(n)$ was 1.0. Figure 6 shows the impulse and frequency responses of this filter.

(*iv*) Filter 4—A 255-point, linear-phase, bandpass filter with 48 dB of rejection in both stopbands. Figure 7 shows the impulse and frequency responses of filter 4.

(*v*) Filter 5—A 256-point, nonlinear-phase, reverberation filter with impulse response that was zero except for $n = 0, 1, 3, 7, 15, 31, 63, 127, 255$, at which $h(n)$ was 1.0.

This set of five filters spans a broad range of impulse-response durations, spectral properties, and temporal properties, and it was felt that it would provide an adequate test of the fast identification algorithm.

### *3.1 Performance with noise excitation signals*

The first set of tests used as the excitation signal the white noise source of Fig. 3. Figure 8 shows plots of the long-time average autocorrelation and power spectrum of the source. The noise spectrum is essentially flat to within ±3 dB.

The white-noise signal was used to drive the system of Fig. 3 for each of the five FIR filters discussed above. For filter 1, data lengths of $N$ from 50 to 1950 were used, and values of s/n from −6 dB to ∞ (no

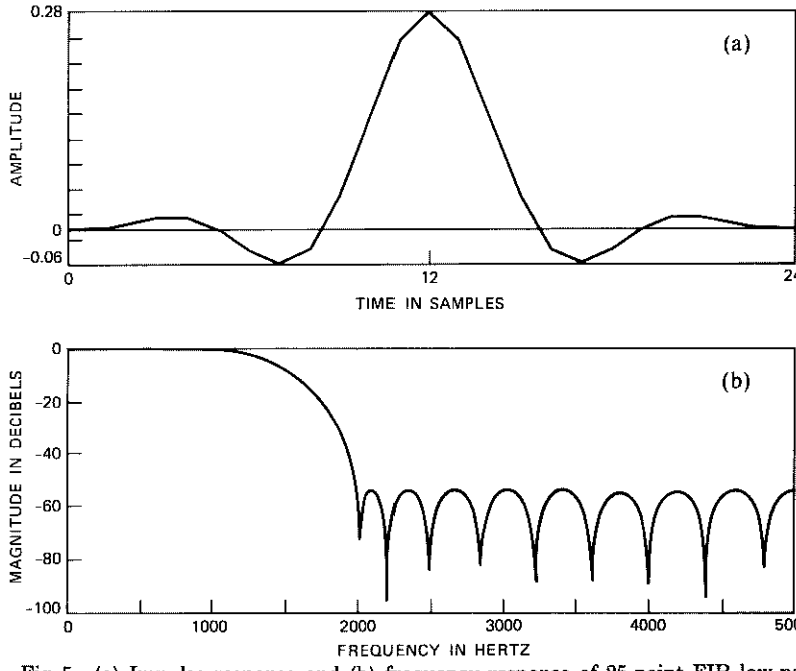



Fig. 5—(a) Impulse response and (b) frequency response of 25-point FIR low-pass filter.

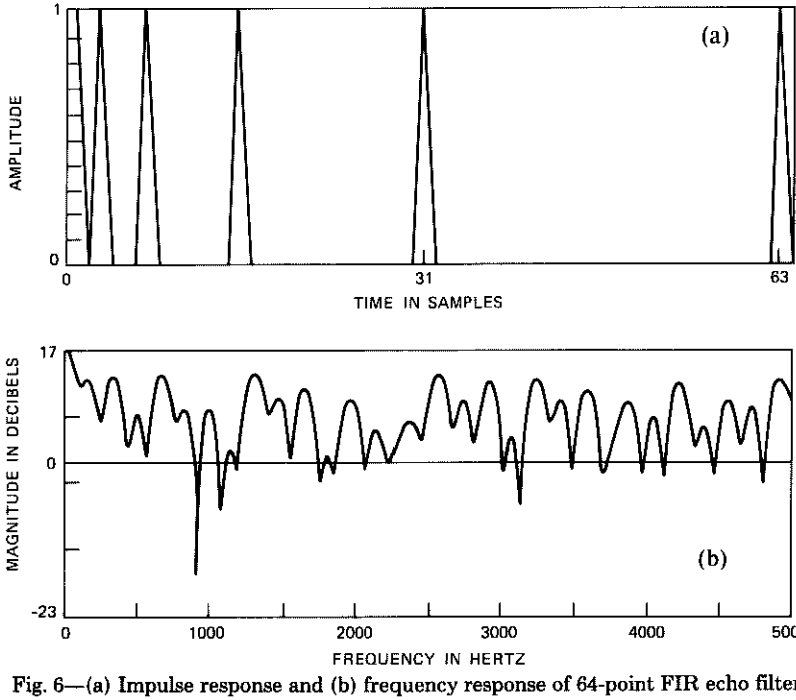



Fig. 6—(a) Impulse response and (b) frequency response of 64-point FIR echo filter.

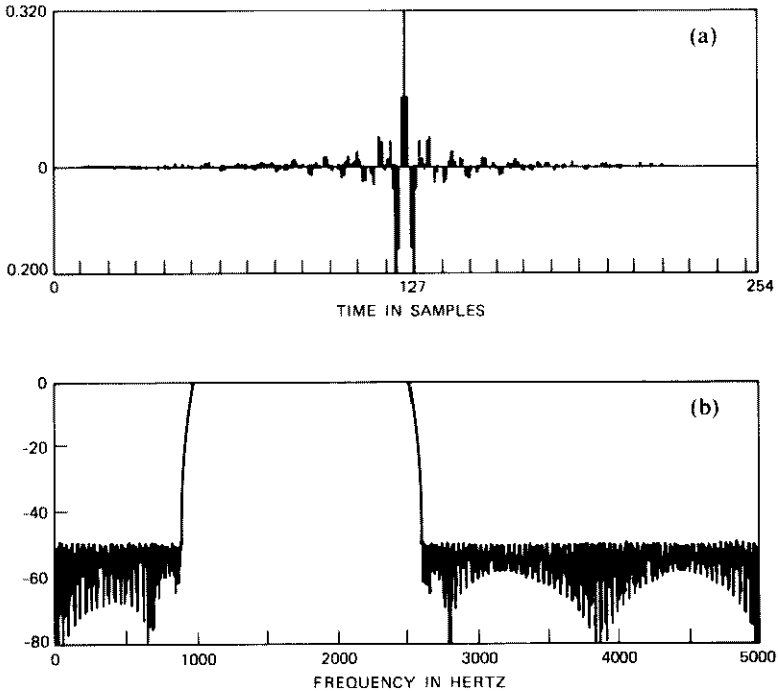



Fig. 7—(a) Impulse response and (b) frequency response of 255-point FIR bandpass filter.

additive noise) were used. For each of the other filters, values of $N$ from the minimum possible $(2M + 1)$ to 1950 were used, along with s/n = $\infty$ and s/n = 30 dB.

The results for filter 1 performance scores are shown in Fig. 9, which gives a series of curves of $Q(N, \text{s/n})$ versus log $N$ for several values of s/n. Also shown in the figure are the theoretical expected values for $Q(N, \text{s/n})$ for a white-noise input,[1] which are of the form

$$Q(N, \text{s/n})|_{\text{white input}} = 10 \log_{10}(M/N) - \text{s/n(dB)}. \tag{28}$$

It can be seen from Fig. 9 that the measured values of $Q$ are very close to the theoretical expectations for s/n's in the range −6 to 42 dB, and for all $N$. For s/n = $\infty$, the measured values of $Q$ (from −88 to −103 dB) reflect the obtainable single-precision accuracy of the computation.

Figures 10 and 11 show similar results for filters 3 and 4, the 64-point reverberation filter and the 255-point bandpass filter. For s/n = $\infty$, the algorithm had $Q$ values of from −100 dB to −109 dB for the 64-point filter, and from −98 dB to −104 dB for the 255-point bandpass filter. The small degradation in performance is due to the higher roundoff errors for the longer impulse responses. For the case where

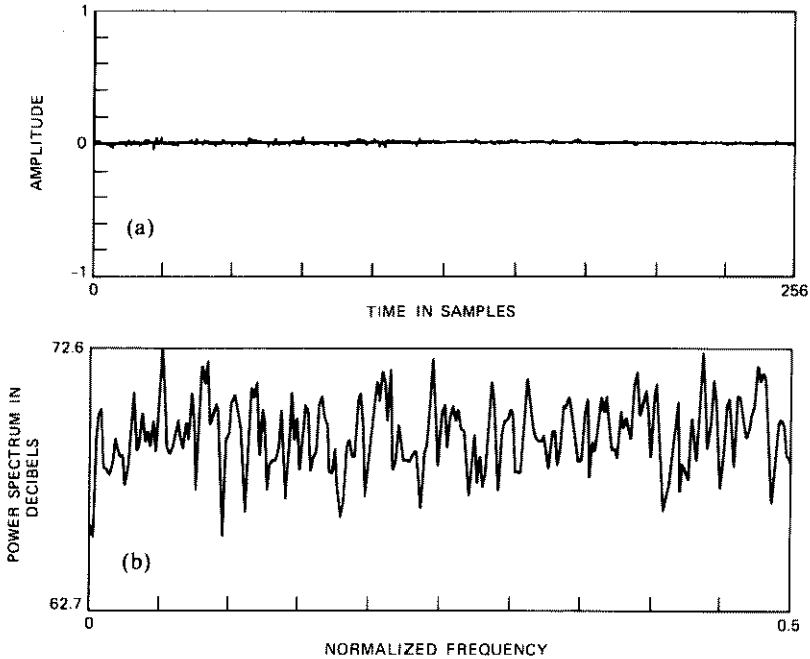



Fig. 8—Long-time average (a) autocorrelation and (b) power spectrum of white-noise source.

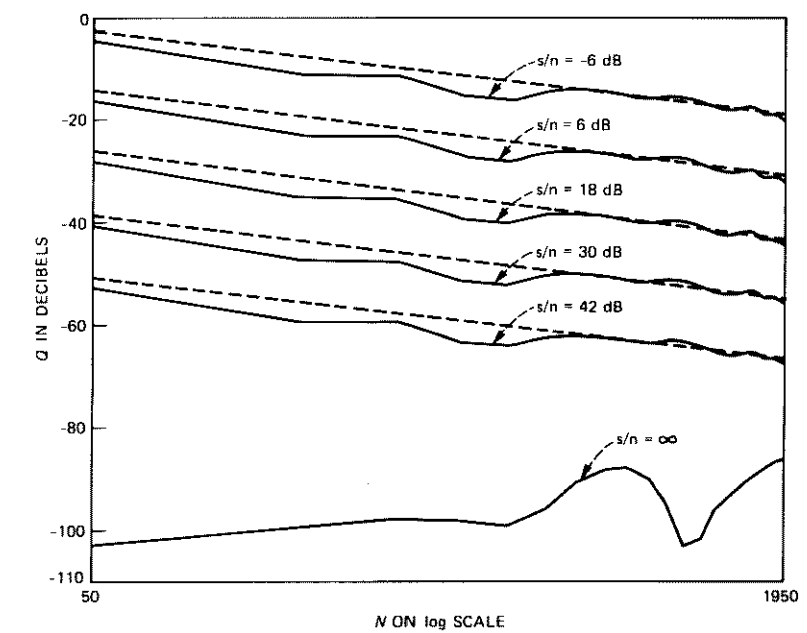



Fig. 9—Plots of $Q$ versus $N$ for several values of s/n for noise input and 7-point FIR filter.

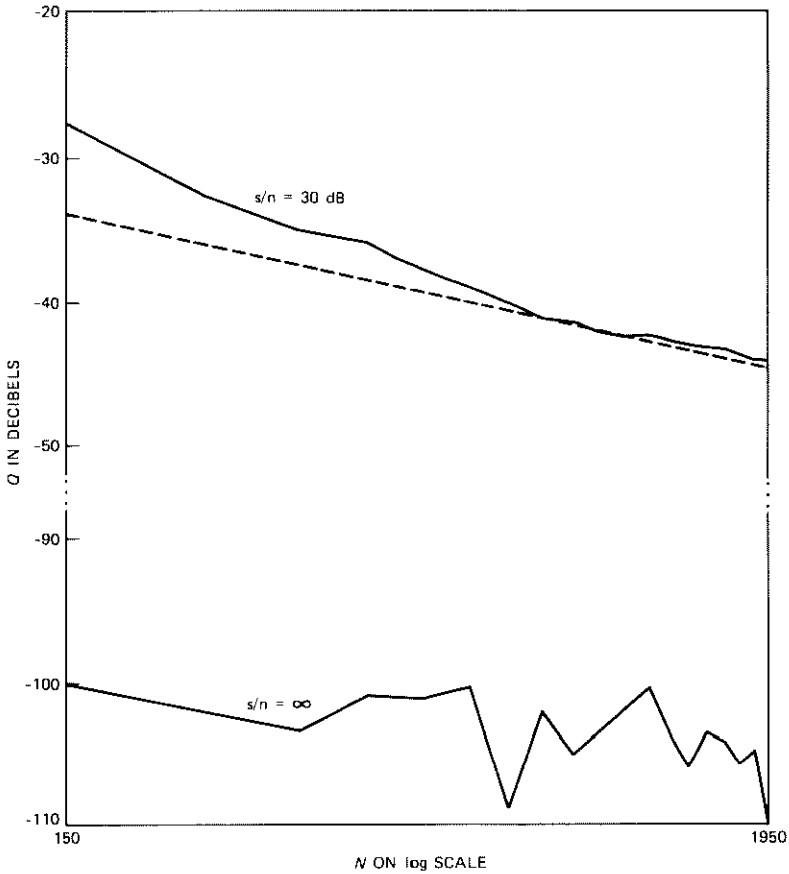



Fig. 10—Plots of $Q$ versus $N$ for s/n = 30 dB and s/n = ∞ for noise input and 64-point FIR echo filter.

s/n = 30 dB the $Q$ curves of Figs. 10 and 11 show about 10-dB worse performance than the theoretical average when $N$ is on the order of $2M + 1$ (the minimum $N$ required to ensure nonsingularity), whereas the performance of the fast algorithm approaches the theoretical estimate as $N$ becomes much larger than $2M + 1$.

The performance of the fast, least-squares estimation algorithm for filters 2 and 5 was essentially identical to that for filters 3 and 4.

As a final example of the noise-excited results, Fig. 12 shows a plot of $Q(N, \text{s/n})$ versus log $N$ for filter 2 (the 25-point low-pass filter) cases, where $N$ is varied from 30 to 70 in steps of 1 and s/n = ∞. The values of $Q$ are very poor for $N \leq 2M$; however, once $N$ exceeds this critical value, the values of $Q$ fall below −75 dB, indicating excellent algorithm performance.

In summary, the results on the white Gaussian noise excitation show

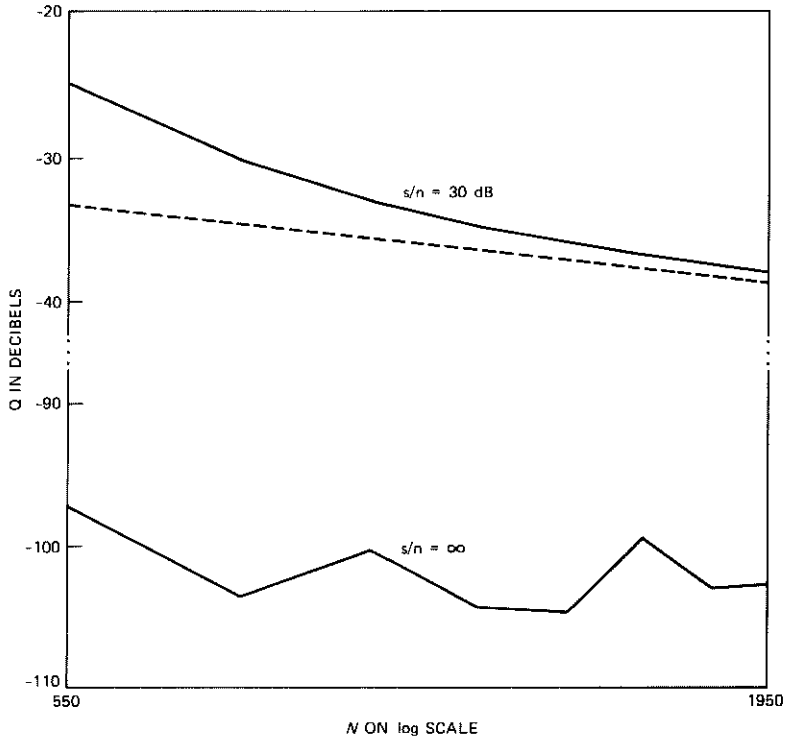



Fig. 11—Plots of $Q$ versus $N$ for s/n = 30 dB and s/n = ∞ for noise input and 255-point FIR bandpass filter.

that the fast, least-squares system identification algorithm performed exceedingly well for all FIR filters, signal-to-noise ratios, and data lengths (so long as $N \geq 2M + 1$).

### *3.2 Performance on speech excitation*

The second series of tests of the performance of the fast system identification algorithm used speech samples as the excitation. Figure 13 shows plots of the long-term average autocorrelation and power spectrum of the speech signal used in our tests. It can be seen that the long-term average power spectrum exhibits a 60- to 70-dB variation in spectral magnitudes, and noticeable gaps in energy throughout the frequency range. Thus, system identification based on speech inputs and outputs is significantly more difficult than it was for white-noise inputs.

The speech excitation was used as input to each of the five FIR filters of Section III. In this section we will concentrate on results obtained using filter 2, the 25-point linear-phase low-pass filter. Results on the other filters were more or less comparable, considering the problems that were encountered.

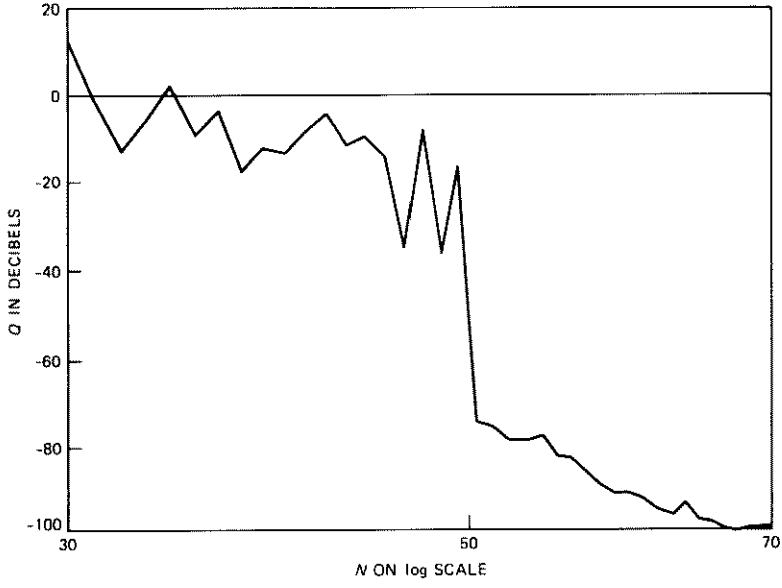



Fig. 12—Plot of $Q$ versus $N$ for 25-point FIR low-pass filter showing abrupt change in $Q$ for $N = 50$.

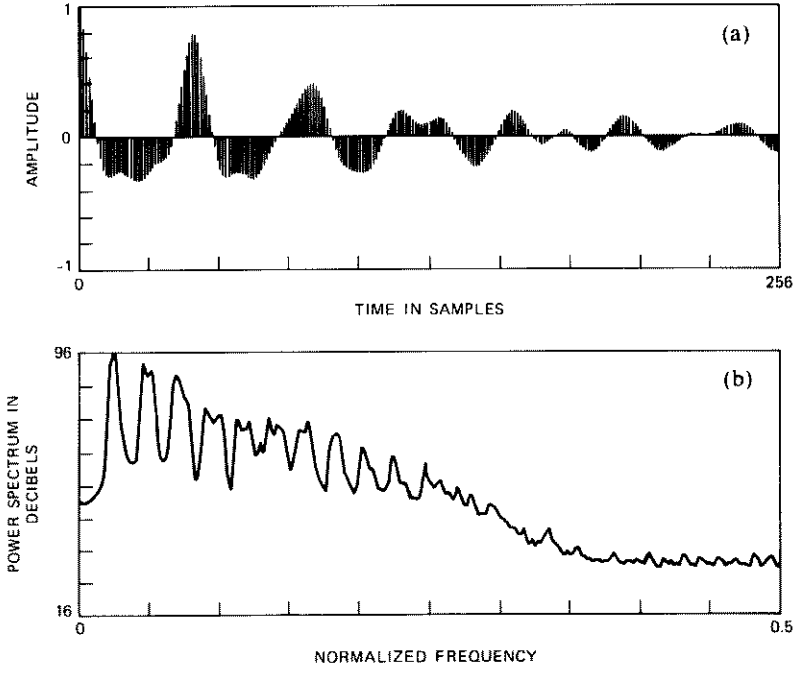



Fig. 13—Long-time average (a) autocorrelation and (b) power spectrum for speech input signal.

Figure 14 shows plots of $Q$ versus log $N$ for s/n = $\infty$ and s/n = 40 and 60 dB for the filter 2, and for values of $N$ from 100 to 1000. [These results were obtained using the fast algorithm designed for linear-phase systems. Results using the nonlinear-phase system algorithm of eq. (6) were consistently worse than for the linear-phase algorithm, and will not be presented here.] For s/n = $\infty$ the values of $Q$ range from −30 to −46 dB. These results, for the case with no additive noise,

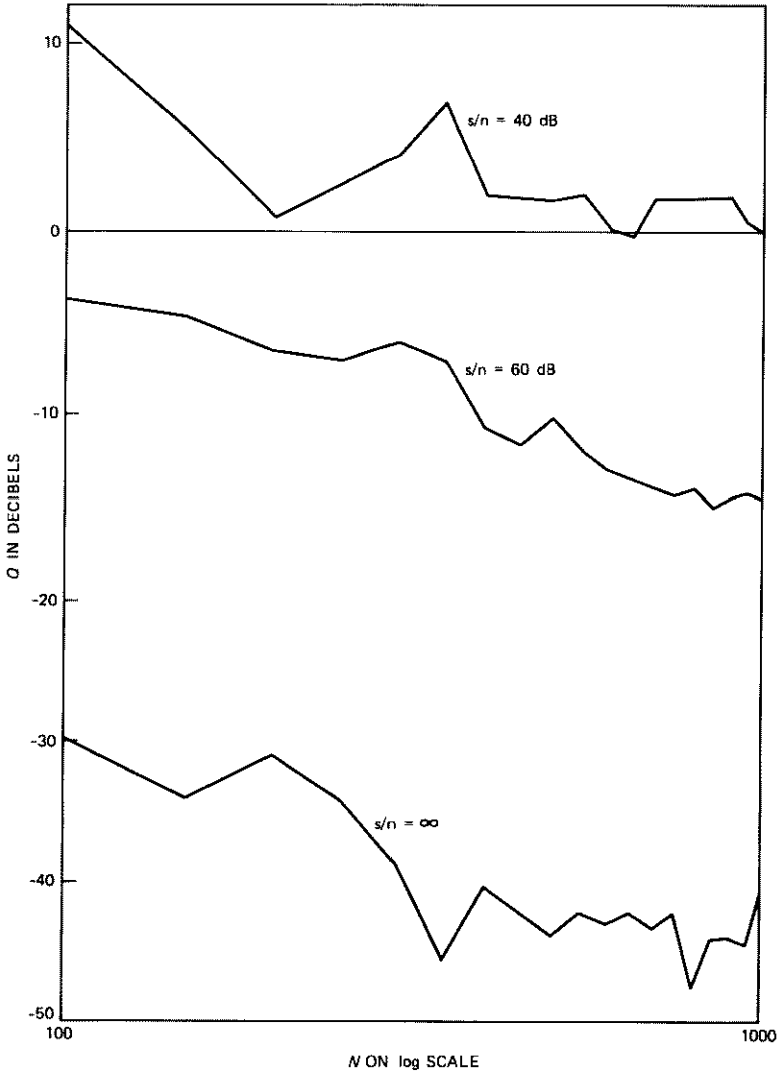



Fig. 14—Plots of $Q$ versus $N$ for several values of s/n for speech input and 25-point FIR low-pass filter.

are significantly worse than those for the white-noise inputs. In the s/n = 40-dB case, all values of $Q$ were about 0 dB or higher, indicating very poor performance. For s/n = 60 dB, the results showing values of $Q$ from −43 to −14 dB indicate marginal performance at best.

The results shown in Fig. 14 are anticipated from previous studies[1-4] and our understanding of the mechanisms of the system identification algorithm. The problem is illustrated in Fig. 15, which shows an estimated impulse response and the resulting frequency response for one set of conditions. It can be seen that the frequency-response estimate is quite good for frequencies in the passband of the low-pass filter (i.e., frequencies less than 0.2 times the sampling frequency). However, the combination of the nonwhite source and spectral gaps with the 54-dB rejection in the stopband leads to extremely poor spectral estimates in the stopband.

To eliminate the spectral gaps in the speech spectrum, a small modification was made to the system block diagram of Fig. 3. A random white noise was added to the speech signal prior to the linear filtering operation to guarantee that all frequencies were present (to some extent) at the input to the linear system. The white noise was

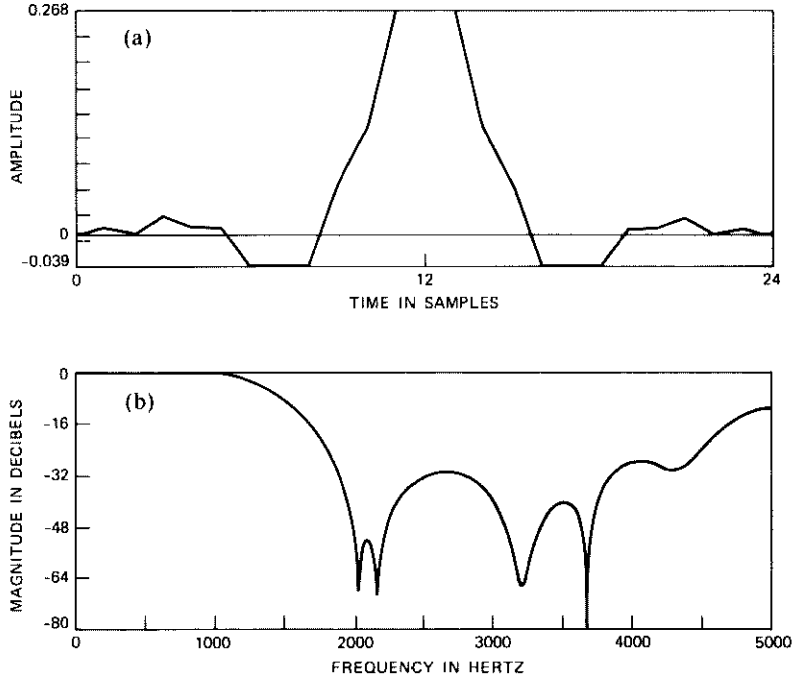



Fig. 15—(a) Plots of estimated impulse response and (b) frequency response for speech input to a 25-point FIR low-pass filter.

added to give an effective signal-to-noise ratio (at the source) of 40 dB. The speech plus noise signal was considered the new input to the system identification procedure.

Figure 16 shows the results obtained using the modified-speech signal input. Shown in this figure are plots of $Q$ versus log $N$ for s/n = $\infty$, 80, 60, and 40 dB, and for values of $N$ from 100 to 1000. For s/n = $\infty$ values of $Q$ as high as −78 dB are obtained, showing the vast improvement in system estimation. Similarly, for s/n = 40, 60, and 80 dB vast improvements in $Q$ values are obtained, leading to useful system estimates in most cases.

In summary, the results of system identification using the fast, least-squares algorithm on speech signals indicate that without some spec-

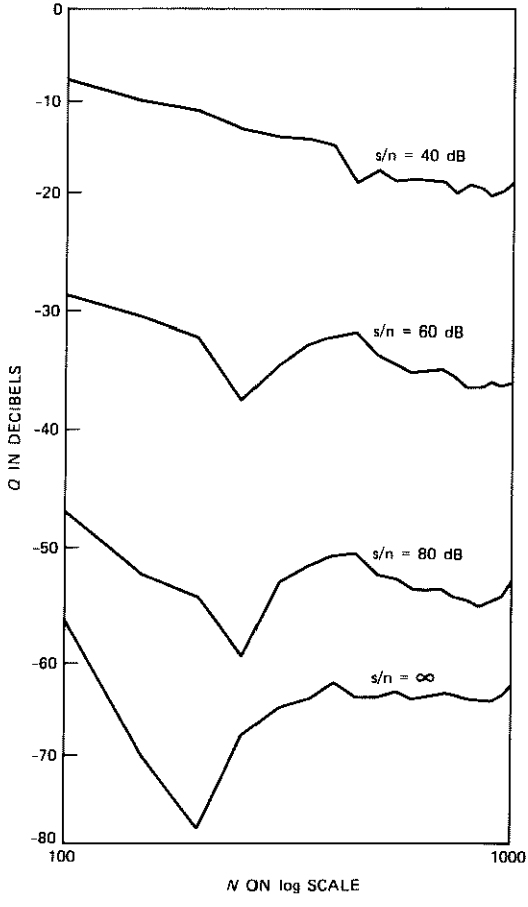



Fig. 16—Plots of $Q$ versus $N$ for several values of s/n for frequency-stabilized speech input to a 25-point FIR low-pass filter.

tral stabilization to guarantee some excitation at all frequencies of interest, the performance of the system identification algorithms is unacceptable in most cases of interest. However, it was shown that even a fairly trivial form of spectral stabilization produced greatly improved system estimates in all cases, and hence made the estimation procedure viable.

## IV. DISCUSSION

The results presented in Section III show the following:

(*i*) For white-noise input signals the fast, nonlinear-phase, least-squares system identification procedure performed extremely well over all filter types, filter durations, and signal-to-noise ratios. For such cases the proposed algorithm appears to have significant computational and storage advantages over all other proposed methods.

(*ii*) For nonwhite input signals with spectral gaps (e.g., speech signals) the fast, linear-phase, least-squares system identification procedure was at best barely adequate (for infinite signal-to-noise ratio) and inadequate for signal-to-noise ratios in the practical range of interest (15 to 50 dB). This poor performance was shown to be due primarily to the spectral gaps in the source and a simple modification was proposed whereby a white noise was added to the speech signal to provide a stabilized spectral magnitude at all frequencies of interest.

(*iii*) When the spectral stabilization procedure was used on speech signals the fast, linear-phase, least-squares system identification procedure performed fairly well over a broad range of signal-to-noise ratios and is useful for a wide range of applications.

To appreciate just how well the fast, least-squares algorithm performed, it is worthwhile comparing the results of Section III with those obtained for alternative FIR system identification algorithms. The two main alternative procedures are the short-time spectral analysis (SSA) methods of Rabiner and Allen[1-4] and the classical "slow," least-squares analysis (LSA) solution obtained by the Cholesky method. For the SSA methods it has been found that for white-noise inputs one can approach the theoretical bounds for sufficiently long data sequences (large $N$). Hence the new fast, least-squares method performs significantly better than the SSA method for noise inputs, except when $N$ becomes very large. In such cases the performances are similar, but the SSA method is still computationally more expensive than the fast, least-squares method.

In the case of speech inputs, the SSA method (which is essentially doing a spectral divide on two power spectrum estimates) runs into extremely bad sensitivity problems because of the missing frequency bands in the input signal. The spectral stabilization method proposed here does not entirely solve the problem for the SSA method because

of the large dynamic range variations in both the input and output signals. The SSA method tends to need a fairly constant spectral input level to perform at its best.

The classical least-squares method, on the other hand, performs as well as the new, fast, least-squares method. However, the computation grows as $NM^2$ (rather than $NM$) and thus the classical LSA method is greatly restricted to small values of $M$ and $N$. In addition to the computational advantage, the fast least-squares algorithm has a significant storage advantage over the classical least-squares approach, growing linearly with the assumed filter order rather than quadratically. Furthermore, the classical Cholesky decomposition used for the LSA method is notoriously sensitive to numerical ill-conditioning and often yields invalid (improper) solutions for cases such as the speech signal input. Hence, except for small values of $N$ and $M$, the fast, least-squares algorithm appears to be preferable to the classical LSA method.

Another important advantage of using the fast, least-squares algorithm is that it provides information about the squared prediction and identification errors [linear-prediction coefficient (LPC) and FIR] as a function of filter order. These squared-error parameters, which are computed directly in the algorithm at no additional computational cost, are important to monitor for the following reasons:

(*i*) They provide good indications of the required filter order.

(*ii*) The linear prediction and FIR system identification squared errors are monotonically decreasing functions of the filter order (at least for the nonlinear-phase algorithm).

(*iii*) When the linear prediction squared errors fall off more rapidly than the FIR system identification squared error, then the input to the filter is not spectrally rich, and therefore the solution may be ill-conditioned, and the FIR quality, $Q$, may be poor.

This latter case is illustrated in Fig. 17, which shows plots of FIR system identification squared error on a log scale, and LPC forward and backward squared errors (also on log scales). For this example the 25-point low-pass filter was excited by nonspectrally rich, voiced speech. The LPC prediction errors decrease rapidly for low filter orders, indicating strong spectral coloration in the input signal. In this case a very poor FIR estimate was obtained.

Figure 18 shows a similar set of squared prediction and identification error plots for the case of a white-noise excitation of the same 25-point low-pass filter. In this case the LPC squared errors are essentially flat (to within 0.6 dB) for all FIR filter orders, indicating a spectrally rich input signal. A very good estimate of the FIR filter was obtained in this case.

It should be noted that the squared error of the fast, linear-phase, system identification estimate is not guaranteed to be a monotonically

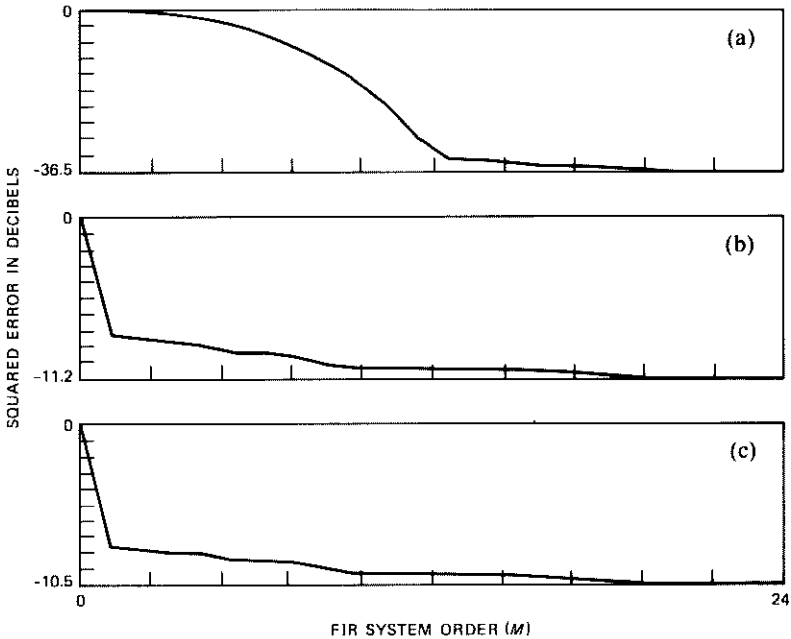



Fig. 17—Plots of squared prediction error versus FIR filter order, $M$, for a speech signal excitation of a 25-point low-pass filter. (a) FIR error. (b) LPC forward error. (c) LPC backward error.

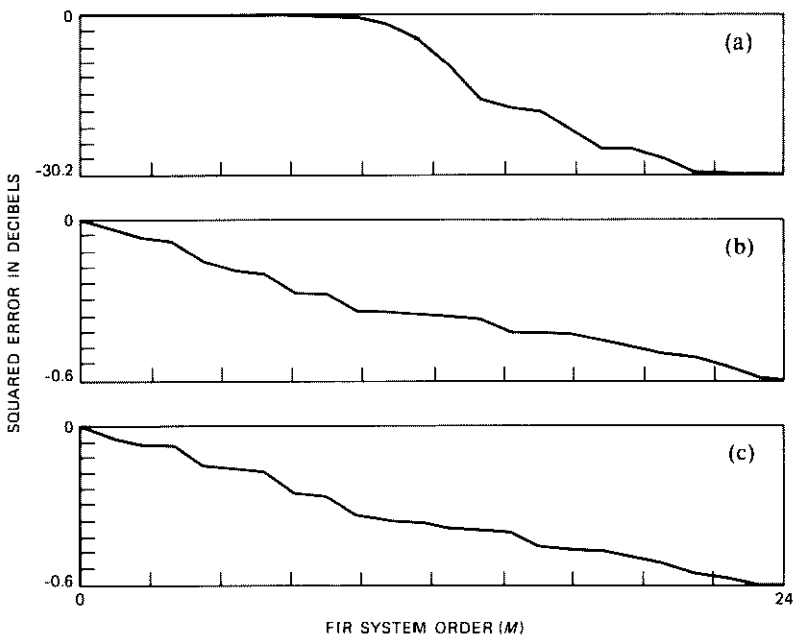



Fig. 18—Plots of squared prediction error versus FIR filter order, $M$, for a noise-signal excitation of a 25-point low-pass filter. (a) FIR error. (b) LPC forward error. (c) LPC backward error.

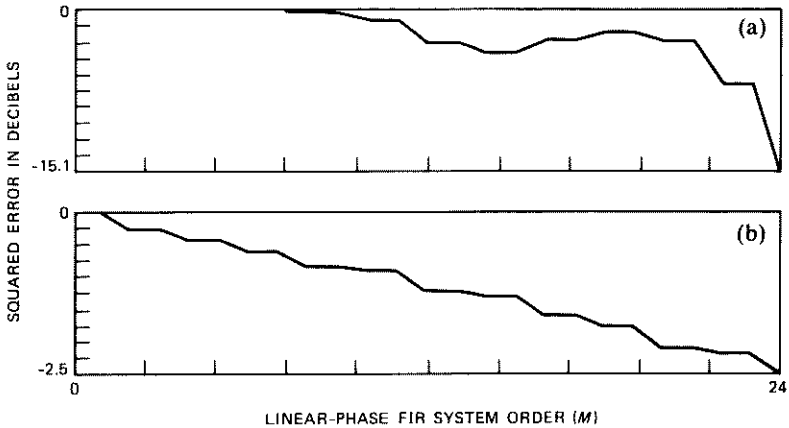



Fig. 19—Plots of squared prediction error versus FIR filter order, $M$, for the linear-phase algorithm with noise excitation of a 25-point low-pass filter. (a) Linear-phase FIR error. (b) LPC forward and backward error.

decreasing function of filter order. This point is illustrated in Fig. 19, which shows the normalized squared error versus filter order for a noise-signal input to the 25-point low-pass filter. The reader will note that for $M$ in the neighborhood of 19, an increase in squared error occurs. The non-monotonicity of the squared error in the linear-phase algorithm is due to the fact that the algorithm is solving a linear *smoothing* problem (taking future and past samples) rather than a linear *prediction* problem (taking only past samples). Also, separate forward and backward linear prediction squared errors are *not* obtained in the linear-phase, fast algorithm; only a combined forward plus backward linear prediction squared error is obtained.

One final point is worth reiterating. In all cases where the FIR system to be estimated is known to be a linear-phase system, the linear-phase, fast, least-squares method is preferable to the general (nonlinear-phase) procedure in that it requires less computation and yields more accurate results.

## V. SUMMARY

We have shown that the fast, least-squares FIR system identification algorithm originally proposed by Marple performs essentially perfectly for white-noise inputs for a wide range of FIR system responses and signal-to-noise ratios. For input signals whose spectrum is highly colored (e.g., speech signals) it was shown that the simple expedient of adding a low-level white noise to the input of the linear system provided a high degree of spectral stabilization and enabled the fast, least-squares algorithm to work well over a wide variety of conditions.

## REFERENCES


1. L. R. Rabiner, R. E. Crochiere, and J. B. Allen, "FIR System Modeling and Identification in the Presence of Noise and with Band-Limited Inputs," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-26* (August 1978), pp. 319–33.
2. J. B. Allen and L. R. Rabiner, "Unbiased Spectral Estimation and System Identification Using Short-Time Spectral Analysis Methods," B.S.T.J., *58* (October 1979), pp. 1743–63.
3. L. R. Rabiner and J. B. Allen, "On the Implementation of a Short-Time Spectral Analysis Method for System Identification," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-28* (February 1980), pp. 69–78.
4. L. R. Rabiner and J. B. Allen, "Short-Time Fourier Analysis Techniques for FIR System Identification and Power Spectrum Estimation," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-27* (April 1979), pp. 182–92.
5. S. L. Marple, Jr., "Efficient Least Squares FIR System Identification," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-29* (February 1981), pp. 62–73.
6. E. Shichor, "Fast Recursive Estimation Using the Lattice Structure," B.S.T.J., *61* (January 1982), pp. 97–115.
7. M. Morf, B. Dickinson, T. Kailath, and A. Vieira, "Efficient Solution of Covariance Equations for Linear Prediction," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-25* (October 1977), pp. 429–33.
8. S. L. Marple, Jr., "Fast Algorithms for Linear Prediction and System Identification Filters with Linear Phase," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-30*, No. 6 (December 1982), pp. 942–53.


## APPENDIX

### *Count of Operations to Solve Classical LSA Normal Equations Using Cholesky Decomposition*

Solve $\Phi\alpha = \Psi$ for vector $\alpha$.

Elements of matrix $\Phi$:

$$\phi_{ij} = \sum_{t=M+1}^{N} x(t-j)x(t-i) \qquad \begin{matrix} 0 \le i \le M \\ 0 \le j \le i. \end{matrix}$$

Elements of vector $\Psi$:

$$\psi_i = \sum_{t=M+1}^{N} y(t)x(t-i) \qquad 0 \le i \le M.$$

In Step 1 we form elements of $\Phi$ and $\Psi$:

| | |
|---|---|
| + | $\frac{1}{2}NM^2 + \frac{5}{2}NM + 2N - \frac{1}{2}M^3 - 3M^2 - \frac{9}{2}M - 2$ |
| × | $\frac{1}{2}NM^2 + \frac{5}{2}NM + 2N - \frac{1}{2}M^3 - \frac{5}{2}M^2 - 2M$ |
| Storage | $\frac{1}{2}M^2 + \frac{5}{2}M + 2$ (does not include $x$ and $y$ data). |

In Step 2 we let $\Phi = VDV^T$. Solve for $V$ and $D$:

| | |
|---|---|
| + | $\frac{1}{6}M^3 + \frac{1}{2}M^2 + \frac{1}{3}M$ |
| × | $\frac{1}{6}M^3 + \frac{1}{2}M^2 - \frac{2}{3}M$ |
| Storage | $M$. |

In Step 3 we solve for vector $Y$ in $VY = \Psi$:

| | |
|---|---|
| + | $\frac{1}{2}M^2 + \frac{1}{2}M$ |
| × | $\frac{1}{2}M^2 + \frac{1}{2}M$ |
| Storage | Store in place. |

In Step 4 we solve for $\alpha$ in $DV^T\alpha = Y$

| | |
|---|---|
| + | $\frac{1}{2}M^2 - \frac{1}{2}M$ |
| × | $\frac{1}{2}M^2 + \frac{1}{2}M + 1$ |
| Storage | Store in place. |

In Step 5 we compute residual squared error $E_m = \sum_{t=M+1}^{N} y^2(t) - \alpha^T\Psi$

| | |
|---|---|
| + | $N + 1$ |
| × | $N + M + 3$ |
| Storage | 2. |

As a result our total computations are:

| | |
|---|---|
| Adds | $\frac{1}{2}NM^2 + \frac{5}{2}NM + 3N - \frac{1}{3}M^3 - \frac{3}{2}M^2 - \frac{25}{6}M - 1$ |
| Multiplies | $\frac{1}{2}NM^2 + \frac{5}{2}NM + 3N - \frac{1}{3}M^3 - M^2 - \frac{2}{3}M + 4$ |
| Storage | $\frac{1}{2}M^2 + \frac{7}{2}M + 4$ (not including $x$ and $y$ data). |

The fast, least-squares algorithm requires

| | |
|---|---|
| Adds | $2NM + 9M^2$ |
| Multiplies | $2NM + 12M^2(+8M$ divides) |
| Storage | $7M + 20$ (not including $x$ and $y$ data), |

which only counts terms of squared powers.

Running some various numbers for filter order $M$ and data lengths $N$ shows that the Cholesky method is more efficient than the "fast" algorithm when $N < 25$ and $M < 10$. If the Cholesky decomposition must be applied many times to determine the correct order to select, then the fast algorithm is more efficient since all lower solutions are obtained recursively.

## CONTRIBUTORS TO THIS ISSUE

**Andres Albanese,** Ingeniero Electricista, 1970, Universidad Central de Venezuela; M.Sc., 1972, University of Texas at Austin; Ph.D., 1976, Stanford University. Instituto Venezolano de Investigaciones Cientificas, 1969–1970; Bell Laboratories, 1975–1982; American Bell, 1983—. At Bell Laboratories, Mr. Albanese did research in systems and components for lightwave communications and local-area networks. His current research interest is architectures and applications for local-area networks. He is Supervisor, Local Networks Application Group.

**T. C. Chu,** B.S., 1964, Cheng Kung University (Taiwan); M.S., 1967, Syracuse University; Ph.D., 1971 (Aerospace Engineering), Cornell University; Bell Laboratories, 1972—. Mr. Chu has worked on the physical design of T4M digital transmission system and the development of an optical fiber connector. Since 1981 he has been working on the design of undersea lightguide cable.

**Patrick J. Fitzgerald,** B.S. (Electronic Engineering Technology), Capitol Institute of Technology, 1973; Bell Laboratories, 1973—. Mr. Fitzgerald has been involved in various research studies including atmospheric microwave and optical propagation and digital modulation techniques. He has also been responsible for several computer-controlled, real-time, data-acquisition systems.

**Nuggehally S. Jayant,** B.Sc. (Physics and Mathematics), 1962, Mysore University; B.E., 1965, and Ph.D. (Electrical Communication Engineering), 1970, Indian Institute of Science, Bangalore; Research Associate at Stanford University, 1967–1968; Bell Laboratories, 1968—. Mr. Jayant was a Visiting Scientist at the Indian Institute of Science from January–March 1972 and August–October 1975. He has worked in the field of digital coding and transmission of waveforms, with special reference to speech communications. He is also editor of an IEEE Reprint Book, *Waveform Quantization and Coding.*

**P. Leland Key,** B.A., B.S. (Mechanical Engineering), 1960, Rice University; United States Navy, 1960–1964; M.S. (Engineering Science), 1964, George Washington University; Ph.D. (Materials Science), 1969, University of California, Berkeley; Bell Laboratories, 1969—. Mr. Key spent his naval service working on materials for nuclear reactors for naval ship propulsion. Since joining the Metallurgy and Engineering Science Department of Bell Laboratories, Mr. Key has been

involved in a broad range of materials activities from whisker growth on electrodeposited films to steels for pressure vessels. Currently, Mr. Key is studying reliability aspects of optical fibers. Member, American Society for Metals, Tau Beta Pi, Sigma Xi, Sigma Tau.

**Bernard George King,** B.S. (Electrical Engineering), 1944, USC; Ph.D., 1955, University of Wisconsin; Bell Laboratories, 1955—. Mr. King taught electrical engineering at USC from 1944 through 1948, and at the University of Wisconsin from 1948 through 1955. He is presently a Supervisor in the Communications Methods Research department, and since 1981 he has been the Editor of The Bell System Technical Journal.

**Ernest D. Kolb** graduated from Curtiss-Wright Technical Institute in 1939 and worked for the Wright Aeronautical Corporation in the Flight Research Section in Caldwell, New Jersey, until 1944. Mr. Kolb then joined the National Advisory Committee for Aeronautics in their Flight Research Section and was concerned with research on aircraft engines until entering the Air Force in 1945. He joined Bell Laboratories in 1951 and was initially involved in the Czochralski growth of single crystals of germanium and silicon from the melt. This work led to J. A. Burton, R. C. Prim, and W. P. Slichter's theory of the effective distribution coefficients of donor and acceptors in the growth of semiconductor materials. He then became involved in the hydrothermal synthesis of inorganic compounds and the growth of single crystals of compounds such as zinc oxide, zinc sulphide, yttrium iron garnet, lithium gallate, and alpha quartz. More recently, he has resumed work with semiconducting compounds with the growth of single crystals from the melt.

**Robert A. Laudise,** B.S. (Chemistry), 1952, Union College, Schenectady, New York; Ph.D. (Inorganic Chemistry), MIT, 1956; Bell Laboratories, 1956—. Mr. Laudise has done research on crystal growth and the preparation and properties of electronic materials. He is particularly interested in the physical chemistry of high-pressure synthesis and in pizeoelectric, semiconductor, and optical materials. He is presently Director of the Physical and Inorganic Chemistry Research Laboratory. He was a recipient of the Sawyer Award for Pizeoelectric Materials and of the International Crystal Growth Award in 1981. President, International Organization for Crystal Growth; member, National Academy of Engineering.

**S. Lawrence Marple, Jr.,** B.A. and M.E.E., 1970 (Electrical Engineering), Rice University, Houston, TX; Engr. Deg., 1976 (Electrical

Engineering), Stanford University, Stanford, CA. From 1971 to 1977, Mr. Marple was employed by ArgoSystems, Inc., Sunnyvale, CA, where he was involved in hardware design for and digital processing of radar and communication signal analysis. From 1978 to 1979, he was a senior member of the technical staff of both Signal Science, Inc. and Advent Systems, Inc., of Mountain View, CA. From 1980 to 1982 he was with the McLean Operations of The Analytic Sciences Corporation (TASC), McLean, VA. He is presently with Schlumberger Well Services in Houston, Texas. His research interests include applications of fast, digital signal processing techniques to spectrum analysis, communications, radar, and well logging problems. Member, Tau Beta Pi.

**Lawrence R. Rabiner,** S.B. and S.M., 1964, Ph.D. (Electrical Engineering), The Massachusetts Institute of Technology; Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975) and *Digital Processing of Speech Signals* (Prentice-Hall, 1978). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication of the Acoustical Society, ASSP Technical Committee on Speech Communication; Member, IEEE Proceedings Editorial Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

**R. A. Semplak,** B.S. (Physics), 1961, Monmouth College; Bell Laboratories, 1955—. Mr. Semplak's main research interest is in studies of atmospheric effects on micro- and millimeter-wave propagation. Currently, he is involved in experimental studies of astigmatic feeds. Member, Sigma Xi and Commission F of the International Union of Radio Science (URSI/USNC).

**Earle E. Simpson,** B.S. (Electronic Engineering), 1957, University of Maine; Western Electric, 1957—. Mr. Simpson was involved in engineering quartz crystal units, crystal filters, and crystal oscillators from 1963 to 1973. Since 1973, he has been responsible for the quartz-growing operation at the Merrimack Valley Works. He is a member of the American Association for Crystal Growth, and is a registered Professional Engineer in Massachusetts.

**Man Mohan Sondhi,** B.Sc. (Physics), Honours degree, 1950, Delhi University, Delhi, India; D.I.I.Sc. (Communications Engineering), 1953, Indian Institute of Science, Bangalore, India; M.S., 1955, Ph.D., 1957 (Electrical Engineering), University of Wisconsin, Madison, Wisconsin; Bell Laboratories, 1962—. Before joining Bell Laboratories, Mr. Sondhi worked for a year at the Central Electronics Engineering Research Institute, Pilani, India and taught for a year at the University of Toronto. At Bell Laboratories his research has included work on speech signal processing, echo cancellation, adaptive filtering, modeling of auditory and visual processes, and acoustical inverse problems. From 1971 to 1972 Mr. Sondhi was a guest scientist at the Royal Institute of Technology, Stockholm, Sweden.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Calderbank, R., **On Uniformly Packed [n,n-k,4] Codes Over GF (q) and a Class of Caps in PG(k-1,q).** J Lond Math 26(Oct):365–384, 1982.

Egan, D. E., Gomez, L. M., **Characteristics of People Who Can Learn to Use Computer Text Editors—Hints for Future Text Editor Design and Training.** P Am S Info 19:75–79, 1982.

Lagarias J. C., Odlyzko, A. M., Shearer, J. B., **On the Density of Sequences of Integers the Sum of Number Two of Which is a Square. 1. Arithmetic Progressions.** J Comb Th A 33(2):167–185, 1982.

Lund, R. E., Linnell, M. G., **Description and Evaluation of a Nested Cube Experimental-Design.** Comm Stat A 11(20):2297–2313, 1982.

## ENGINEERING

Alferness, R. C., et al., **Efficient Single-Mode Fiber to Titanium Diffused Lithium-Niobate Wave-Guide Coupling for Gamma=1.32-μm.** IEEE J Q El 18(10):1807–1813, 1982.

Anthony, P. J., **Alteration of Diffusion Profiles in Semiconductors Due to p-n-Junctions.** Sol St Elec 25(10):1003–1009, 1982.

Boenke, M. M., Wagner, R. E., Will, D. J., **Transmission Experiments Through 101 km and 84 km of Single-Mode Fiber at 274 mbit/s and 420 mbit/s.** Electr Lett 18(21): 897–898, 1982.

Bosik, B. S., Kartalopoulos, S. V., **A Time Compression Multiplexing System for a Circuit Switched Digital Capability.** IEEE Commun 30(9):2046–2052, 1982.

Boyer, G. R., **Administration of New Feeder Technology.** IEEE Commun 30(9):2029–2033, 1982.

Byrne, T. P., Coburn, R., Mazzoni, H. C., Aughenbaugh, G. W., Duffany, J. L., **Positioning the Subscriber Loop Network for Digital Services.** IEEE Commun 30(9):2006–2011, 1982.

Campbell, L. W., Cline, R. R., Litofsky, B., Martz, L. M., Varma, G. K., **Maintenance Operations for the Circuit-Switched Digital Capability.** IEEE Commun 30(9):2053–2056, 1982.

Clarke, R. H., **Theoretical Performance of an Antireflection Coating for a Diode-Laser Amplifier.** Int J Elect 53(5): 495–499, 1982.

Cohen, L. G., Marcuse, D., Mammel, W. L., **Radiating Leaky-Mode Losses in Single-Mode Lightguides With Depressed-Index Claddings.** IEEE J Q El 18(10):1467–1472, 1982.

Coldren, L. A., Furuya, K., Miller, B. I., Rentschler, J. A., **Etched Mirror and Groove-Coupled GaInAsP/InP Laser Devices for Integrated-Optics.** IEEE J Q El 18(10):1679–1688, 1982.

Dale, O. B., Rubin, H., Vetter, R. W., **A Highly Distributed Mechanized Loop Testing System.** IEEE Commun 30(9):2038–2045, 1982.

Dutta, N. K., Wright, P. D., Nelson, R. J., Wilson, R. B., Besomi, P. R., **InGaAsP Laser With High $T_0$ (Letter).** IEEE J Q El 18(10):1414–1416, 1982.

Ebeling, K. J., Coldren, L. A., Miller, B. I., Rentschler, J. A., **Generation of Single-Longitudinal-Mode Sub-Nanosecond Light-Pulses by High-Speed Current Modulation of Monolithic Two-Section Semiconductor-Lasers.** Electr Lett 18(21):901–902, 1982.

Furuya, K., Miller, B. I., Coldren, L. A., Howard, R. E., **A Novel Deposit Spin Wave-Guide Interconnection (DSWI) for Semiconductor Integrated-Optics.** IEEE J Q El 18(10):1783–1789, 1982.

Joy, D. C., **Deconvolution for ELS Quantitation.** Ultramicros 9(3):289–294, 1982.

Khanarian, G., **Non-Linear Dielectric Effect of Flexible Polar Polymers.** Macromolec 15(5):1429–1432, 1982.

Korotky, S. K., Minford, W. J., Buhl, L. L., Divino, M. D., Alferness, R. C., **Mode Size and Method for Estimating the Propagation Constant of Single-Mode Ti-$LiNbO_3$ Strip Wave-Guides.** IEEE J Q El 18(10):1796–1801, 1982.

Lee, T. P., Burrus, C. A., Marcuse, D., Dentai, A. G., Nelson, R. J., **Measurement of Beam Parameters of Index-Guided and Gain-Guided Single-Frequency InGaAsP Injection-Lasers.** Electr Lett 18(21):902–904, 1982.

Liu, P. L., **$LiNbO_3$ Wave-Guide Modulator With 1.2-$\mu$m Thick Electrodes Fabricated by Lift-Off Technique.** IEEE J Q El 18(10):1780–1782, 1982.

Liu, P. L., Lee, T. P., Burrus, C. A., Kaminow, I. P., Ko, J. S., **Observation of Transient Spectra and Mode Partition Noise of Injection-Lasers.** Electr Lett 18(21):904–905, 1982.

Marcuse, D., **Quantum-Mechanical Explanation of Spontaneous Emission K-Factor.** Electr Lett 18(21):920–922, 1982.

Minford, W. J., Korotky, S. K., Alferness, R. C., **Low-Loss Ti-$LiNbO_3$ Wave-Guide Bends at Gamma=1.3-$\mu$m.** IEEE J Q El 18(10):1802–1806, 1982.

Philen, D. L., White, I. A., Kuhl, J. F., Mettler, S. C., **Single-Mode Fiber OTDR—Experiment and Theory.** IEEE J Q El 18(10):1499–1508, 1982.

Prakash, M. N., Fam, A. T., **The Theory of Stable Minimal Partial Realizations—The Scalar Case (Letter).** IEEE Circ S 29(9): 646–647, 1982.

Santana, M. R., Yanizeski, G. M., **Bending Performance of Fiberoptic Cable Sheaths—Application of New Apparatus and Techniques.** J Mech Des 104(3):578–586, 1982.

Wyndrum, R. W., **Special Issue on Subscriber Loops—The Evolution of Electronic Loop-Systems in the 1980s (Editorial).** IEEE Commun 30(9):2005, 1982.

## MANAGEMENT/ECONOMICS

Jovanovic, B., **Truthful Disclosure of Information.** Bell J Econ 13(1):36–44, 1982.

Perry, M. K., **Oligopoly and Consistent Conjectural Variations.** Bell J Econ 13(1):197–205, 1982.

Sharkey, W. W., **Suggestions for a Game-Theoretic Approach to Public Utility Pricing and Cost Allocation.** Bell J Econ 13(1):57–68, 1982.

## PHYSICAL SCIENCES

Anderson, P. W., **The 1982 Nobel-Prize in Physics.** Science 218(4574):763–764, 1982.

Antler, M., **Fretting of Electrical Contacts—An Investigation of Palladium Mated to Other Materials.** Wear 81(1):159–173, 1982.

Bachelet, G. B., Hamann, D. R., Schluter, M., **Pseudopotentials That Work—From H to Pu.** Phys Rev B 26(8):4199–4228, 1982.

Bally, J., **Energetic Activity in a Star-Forming Molecular Cloud Core—A Disk Constrained Bipolar Outflow in NGC-2071.** Astrophys J 261(2):558–568, 1982.

Bjorkholm, J. E., Liao, P. F., Wokaun, A., **Distortion of On-Resonance 2-Photon Spectroscopic Line-Shapes Caused by Velocity-Selective Optical-Pumping.** Phys Rev A 26(5):2643–2655, 1982.

Bokor, J., Freeman, R. R., Cocke, W. E., **Photo-Autoionization Pumped Ba Ion Laser.** Aip Conf PR1982(90):153–162, 1982.

Bondybey, V. E., English, J. H., Shiley, R. H., **Pyridine Radical Cation and Its Fluorine Substituted Derivatives.** J Chem Phys 77(10):4826–4831, 1982.

Bowden, M. J., Thompson, L. F., Robinson, W., Biolsi, M., **Thermal-Degradation of Poly(1-Butene Sulfone).** Macromolec 15(5):1417–1422, 1982.

Bridges, T. J., Chraplyvy, A. R., Bergman, J. G., Hart, R. M., **Broad-Band Infrared Generation in Liquid-Bromine-Core Optical Fibers.** Optics Lett 7(11):566–568, 1982.

Broughton, J. Q., Gilmer, G. H., Jackson, K. A., **Crystallization Rates of a Lennard-Jones Liquid.** Phys Rev L 49(20):1496–1500, 1982.

Brown, W. L., Lanzerotti, L. J., Johnson, R. E., **Fast Ion-Bombardment of Ices and Its Astrophysical Implications.** Science 218(4572):525–531, 1982.

Chan, K., **A Simple Proof That the Unstable (CO-)Homology of the Brown-Peterson Spectrum Is Torsion Free.** J Pure Appl 26(2):155–157, 1982.

Chin, A. K., Zipfel, C. L., Dutt, B. V., **The Reliability of Schottky-Barrier Restricted GaAs/GaAlAs LEDs.** Jpn J A P 1 21(9):1308–1312, 1982.

Chu, S., Wong, S., **Pulse-Propagation in an Absorbing Medium—Reply (Letter).** Phys Rev L 49(17):1293, 1982.

Cohen, R. L., Feldman, L. C., West, K. W., Kincaid, B. M., **Displacement Around a Dissolved Impurity Atom in a Metal-Sn in Cu.** Phys Rev L 49(19):1416–1419, 1982.

Corruccini, L. R., **Second Sound in Superfluid $^3$He.** Physica B&C 110(1–3):1590–1595, 1982.

Cross, M. C., **Magnetic-Properties of Solid $^3$He—What Do We Know and What Do We Learn.** Physica B&C 110(1–3):1796–1804, 1982.

DuBois, L. H., **Vibrational-Spectra of Atomic Adsorbates—Carbon, Oxygen, and Sulfur on Rh(100).** J Chem Phys 77(10):5228–5233, 1982.

Even, U., Magen, J., Jortner, J., Friedman, J., Levanon, H., **Isolated Ultracold Porphyrins in Supersonic Expansions. 1. Free-Base Tetraphenylporphyrin and Zn-Tetraphenylporphyrin.** J Chem Phys 77(9):4374–4383, 1982.

Even, U., Magen, J., Jortner, J., Friedman, J., **Isolated Ultracold Porphyrins in Supersonic Expansions. 2. Zn-Tetrabenzoporphyrin.** J Chem Phys 77(9):4384–4390, 1982.

Fisher, D. S., **Shear Moduli and Melting Temperatures of Two-Dimensional Electron Crystals—Low-Temperatures and High Magnetic-Fields.** Phys Rev B 26(9):5009–5021, 1982.

Fox, D. L., Golding, B., Haemmerl, W. H., **Optically Induced Gap in the Atomic Tunneling Spectrum of $AS_2S_3$ Glass.** Phys Rev L 49(18):1356–1360, 1982.

Freeman, R. R., Jopson, R. M., Bokor, J., **Generation of Light Below 100nm in Hg Vapor.** AIP Conf PR1982(90):422–430, 1982.

Friedman, J. M., Rousseau, D. L., Ondrias, M. R., **Time-Resolved Resonance Raman Studies of Hemoglobin (Review or Bibliog.).** Ann R Ph Ch 33:471–491, 1982.

Geerk, J., Gurvitch, M., McWhan, D. B., Rowell, J. M., **Electron-Tunneling into Nb/Al Multilayers and into Nb with Al Overlayers.** Physica B&C 110(1–3):1775–1784, 1982.

Gibson, J. M., Bean, J. C., Poate, J. M., Tung, R. T., **Direct Determination of Atomic-Structure at the Epitaxial Cobalt Disilicide on (111) Si Interface by Ultrahigh Resolution Electron-Microscopy.** Appl Phys L 41(9):818–820, 1982.

Greywall, D. S., Paalanen, M. A., **Second Sound in Spin-Polarized $^3$He-$^4$He Solutions.** Physica B&C 110(1–3):1575–1582, 1982.

Haque, C. A., Antler, M., **Atmospheric Corrosion of Clad Palladium and Palladium Silver Alloys—Film Growth and Contamination Effects.** Corros Sci 22(10):939+, 1982.

Hohenberg, P. C., **Critical Phenomena in -$^4$He.** Physica B&C 110(1–3):1436–1446, 1982.

Hutton, R. S., Roth, H. D., Schilling, M. L., Trozzolo, A. M., Leslie, T. M., **Magnetic-Field Dependence of F-19 Nuclear-Spin Polarization—The System Tetrafluorobenzoquinone Tetrafluorohydroquinone.** J Am Chem S 104(22):5878–5883, 1982.

Johari, G. P., Goodby, J. W., **Dielectric Relaxations in a Supercooled Liquid and Glassy Smectic Phase.** J Chem Phys 77(10):5165–5172, 1982.

Johari, G. P., **Effect of Annealing on the Secondary Relaxations in Glasses.** J Chem Phys 77(9):4619–4626, 1982.

Jopson, R. M., Freeman, R. R., Bokor, J., **Laser-Produced Plasmas in Hg—A Source of Pulsed XUV Radiation.** AIP Conf PR1982(90):303–304, 1982.

Kuo, C. Y., Vieira, M. M. F., Patel, C. K. N., **Infrared Spectroscopy of Phonons in Hexagonal-Phase Solid Hydrogen.** Phys Rev L 49(17):1284–1287, 1982.

Lee, P. A., et al., **Magnetoresistance of Weakly Disordered Electrons.** Phys Rev B 26(8):4009–4012, 1982.

Lindsay, S. M., Halawith, B., Patterson, G. D., **Observation of Hypersonic Shear-Waves in Polystyrene and Poly(Methyl Methacrylate) by Brillouin-Scattering.** J Pol Sc Pl 20(11):583–588, 1982.

Littlewood, P. B., Varma, C. M., **Amplitude Collective Modes in Superconductors and Their Coupling to Charge-Density Waves.** Phys Rev B 26(9):4883–4893, 1982.

McBrierty, V. J., Douglass, D. C., Wudl, F., Aharon-Shalam, E., **Nuclear-Resonance and Relaxation in Ditetramethyltetraselenafulvalenium Salts.** Phys Rev B 26(9):4805–4809, 1982.

McCrory, J. C., Rosamilia, J. M., **Differential Pulse Polarography of Germanium(IV), Tin(IV), Arsenic(V), Antimony(V), Selenium(IV) and Tellurium(VI) at the Static Mercury Drop Electrode in Catechol Perchlorate-Media.** Analyt Chim 142(OCT):231–238, 1982.

Miller, T. A., **Light and Radical Ions (Review or Bibliog.).** Ann R Ph Ch 33:257–282, 1982.

Mitchell, J. W., **Purification of Analytical Reagents.** Talanta 29(11):993–1002, 1982.

Nakada, Y., **Metallurgical Characterization of Arc-Conditioned Carbon Coating on Copper Electrodes.** Thin Sol Fi 95(2):185–193, 1982.

Nassau, K., Chadwick, D. L., **Class-Forming Systems Involving $GeO_2$ With $Bi_2O_3$, $Ti_2O$, and PbO.** J Am Ceram 65(10):486–491, 1982.

Osheroff, D. D., **Spin Ordering in Solid $^3He$—A Current Perspective.** Physica B&C 110(1–3):1461–1473, 1982.

Perriere, J., Siejka, J., Chang, R. P. H., **Observation of Short-Range Oxygen Migration and Oxygen-Exchange During Low-Temperature Plasma Anodization of Silicon Through Thin $ZrO_2$ Films.** Thin Sol Fi 95(4):309–314, 1982.

Sammon, M. J., **Numerical Three-Dimensional Relaxation of Liquid-Crystal Director Fields.** Molec Cryst 89(1–4):305–317, 1982.

Santoro, A., Cava, R. J., Murphy, D. W., Roth, R. S., **Use of the Pearson Type-VII Distribution in the Neutron Profile Refinement of the Structures of $LiReO^3$ and $Li_2ReO_3$.** AIP Conf PR1982(89):162–165, 1982.

Shepp, L. A., **Citation Classic—The Fourier Reconstruction of a Head Section.** CC/Eng Tech 1982(46):14, 1982.

Turner, D. R., **Modern Techniques for Electroplating Gold Contacts.** Thin Sol Fi 95(2):143–149, 1982.

Vanbenthem, M. H., Gillispie, G. D., Haddon, R. C., **Intramolecular Hydrogen-Bonding. 2. Enormous Deuterium-Isotope Effect on the Phosphorescence of 6-Hydroxybenzanthrone (Letter).** J Phys Chem 86(22):4281–4283, 1982.

Venkatesan, T., **Species and Deposition Angle Dependence of Ion-Beam Induced Densification of Germanium Selenide Films.** Appl Phys L 41(9):839–841, 1982.

Walstedt, R. E., **Numerical-Simulation Studies of RKKY Spin-Glasses.** Physica B&C 110(1–3):1924–1935, 1982.

White, J. C., Henderson, D., **Threshold and Dispersion Effects in the Anti-Stokes Raman Laser.** Optics Lett 7(11):517–519, 1982.

Young J. S., et al., **Physical Conditions and Carbon-Monoxide Abundance in the Dark Cloud B5.** Astrophys J 261(2):513–531, 1982.

## SOCIAL AND LIFE SCIENCES

Finkel, R. A., Fishburn, J. P., **Parallelism in Alpha-Beta Search.** Artif Intel 19(1):89–106, 1982.

Patel, D. J., **Antibiotic-DNA Interactions—Intermolecular Nuclear Overhauser Effects in the Netropsin-d (C-G-C-G-A-A-T-T-C-G-C-G) Complex in Solution.** P Nas Biol 79(21):6424–6428, 1982.

Seldner, M., Uson, J. M., **The Cross-Correlation of the Zwicky and Shane-Wirtanen Catalogs of Galaxies.** Astrophys J 261(2):L65–L69, 1982.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Overview and Objectives

By R. E. STAEHLER and J. I. COCHRANE

(Manuscript received June 30, 1982)

*This paper presents an overview and introduction to the detailed technical papers that describe the Traffic Service Position System No. 1B. The objectives and design philosophy are discussed and the overall organization of the system is described.*

## I. INTRODUCTION

### *1.1 Background*

In January 1969 the Bell System's first stored program controlled operator services system was introduced into the field. This system, named the Traffic Service Position System (TSPS) No. 1, employed a hardware/software architecture designed to permit the addition of new features to further automate operator functions as those features became technologically and economically viable.

#### *1.1.1 Initial capabilities*

The initial design[1] of TSPS No. 1 was developed to be used in conjunction with most local and toll switching systems in the Bell System. The system enabled customers to dial a number of calls heretofore only possible with operator assistance and thereby relieved the operators of many tedious operations demanded by cord switchboards. It automated routine operator functions for coin and noncoin calls, such as call timing and recording, recording of originating directory number, and recording and transmission of customer-dialed numbers.

Customers benefited from the advantages of the higher speed and increased accuracy of a stored program controlled system. The auto-

mation of operator functions was a great improvement over manual methods of number recording, timing, charge calculations, and billing.

Operating companies benefited because the automation and administrative features enabled operators to handle calls more efficiently and effectively, thereby reducing costs. The operator groups could be remotely located at convenient sites, and the entire work force could be managed more effectively.

Operators benefited from the more attractive working environment, the automatic equitable distribution of calls to their positions, and the satisfaction of improving their services to the customers.

#### *1.1.2 Technological advances*

The decade following the introduction of TSPS No. 1 brought major advances in technology in the areas of larger scales of integration in circuits, semiconductor memories, automated machine speech response techniques, microprocessors and miniprocessors, and advanced operating systems. A large number of these innovations have been incorporated into the operator services system as need and economic viability have been demonstrated.

#### *1.1.3 Additions to initial system capability*

Additional firmware and software[2-5] were continuously developed and introduced into the system (with new technology when applicable) to provide new and improved features. These include automation of hotel/motel charge quotation; partial automation of international call handling; time and charge quotations for noncoin, nonhotel calls; recording of charges for directory assistance calls; provision of service to remote locations; and automated charge quotation and coin collection for coin toll calls.

Two of the newest features introduced in 1979 and 1980, respectively, are automated verification of busy lines (with ensured privacy) and Automated Calling Card Service, which allows credit card calling without operator assistance from telephones using dual-tone multifrequency signaling.

#### *1.1.4 Rapid nationwide deployment*

Since its introduction in 1969, TSPS No. 1 has been rapidly deployed throughout the Bell System's nationwide telecommunications network to the point where there are more than 157 systems installed. Over 95 percent of Bell System customers and a large number of customers served by other companies are served by TSPS No. 1, providing almost universal availability of stored program controlled toll and assistance operator services.

### 1.1.5 TSPS No. 1 capability

The continuing growth of operator services system traffic, plus the continuing addition of new features, have steadily reduced the remaining real-time capacity of the TSPS No. 1 processor, the Stored Program Control No. 1A (SPC 1A) Processor, in sites in the field. Since real-time processor capacity is a function both of the hardware configuration and of the call mix at an individual installation, a program called TSPS Real-Time Capacity Program (TSPSCAP) was developed to run on an off-line computer and permit the operating companies to verify the remaining capacity at each specific site.

Site-by-site surveys of the TSPSCAP results conducted in the mid-1970's indicated that some of the TSPS No. 1 sites were at their real-time capacity. Furthermore, a large number of additional sites would soon reach their real-time capacity.

Finally, this same growth of traffic and addition of features has exhausted the memory capacity at many sites because of the increased size of the software programs and data tables required.

### 1.1.6 Contemplated new network services

Plans to utilize TSPS in the future[5] as an action control point in the emerging stored program controlled network dictated a need for new processor peripherals, such as a mass memory disk that cannot be provided by the current SPC 1A Processor.

## II. TSPS NO. 1B DEVELOPMENT

For the reasons given in the previous section, it was projected in the mid-1970's that a major evolution of TSPS No. 1 was necessary to provide continuing operator services for the 1980's and beyond. This major evolution was designated TSPS No. 1B.

### 2.1 TSPS No. 1B capability objectives

The basic objective in the TSPS No. 1B development was to increase capability compared to TSPS No. 1.

Studies of network growth and economics led to the conclusion that an appropriate call capacity objective for TSPS No. 1B would be a design with an initial objective of 1.6 times that of the TSPS No. 1. Studies also indicated that to provide adequate office data and program storage, a reasonable objective for the physical memory of TSPS No. 1B would be four times that provided by TSPS No. 1.

The TSPS No. 1B also had to eliminate those functional restrictions in TSPS No. 1 that limited its evolution as an action control point in the stored program controlled network. This could be accomplished

by the provision of processor peripherals such as random access bulk storage and general-purpose data links.

### 2.2 TSPS No. 1B architecture objectives

The new features demanded by the utilization of TSPS in the stored program controlled network also required that TSPS No. 1B provide a flexible architecture for future developments.

The TSPS No. 1B design had to incorporate a high-level language and an advanced operating system not only to expedite introduction of new features, but also to maximize the productivity of the software developers.

Since there are over 150 TSPS No. 1 sites in the field, the size of the investment in TSPS No. 1 software programs and peripheral hardware is very large. To preserve the majority of this large investment, the existing software and peripheral hardware must be retained. At the same time, new high-level software and additional peripheral hardware must be added to the system for new features. Thus, the TSPS No. 1B design must permit the use of the new high-level language in parallel with the continued availability of the existing assembly language development environment.

An important objective was the ability to upgrade the TSPS No. 1 sites in service that were limited by capability exhaust without interrupting call processing.

### 2.3 Other objectives

Besides additional capabilities and a flexible architecture, other basic objectives were also set for the TSPS No. 1B development.

The TSPS No. 1B must provide dependable operator service 24 hours a day. This requirement was converted to a specific design objective of reducing service interruptions that result in a total loss of service to an average of less than 3 minutes per year per system.

To be cost effective, the TSPS No. 1B should take advantage of the newest technology and reduce energy consumption and floor space usage.

### 2.4 TSPS No. 1B design approach

To meet these objectives for the TSPS No. 1B, the basic design approach implemented was to replace the existing processor with a new processor while retaining both peripheral hardware and existing software. The most efficient way to retain the software was for the new processor to provide emulation capability. Further, peripheral interface hardware was needed in order for the new processor to execute existing software and control the existing peripheral hardware.

Where required, software could be written in the new high-level language.

## III. 3B20D PROCESSOR

The 3B20D Processor[6,7] was under development at the time the TSPS No. 1B development was initiated. It is one of a family of general-purpose processors designed not only to meet the real-time demands and dependability requirements of switching systems, but also to be versatile enough for a broad spectrum of present and future telecommunications applications.

A task force formed in 1976 to evaluate different processors concluded that the 3B20D Processor would meet all of the objectives stated in Section II and recommended that the 3B20D be utilized in the TSPS No. 1B design.

In brief, the 3B20D Processor is significantly faster than the SPC 1A Processor and has four times the physical memory capacity of the TSPS No. 1 processor. It provides additional processor peripherals such as random access bulk storage disks and general-purpose data links. The 3B20D Processor is designed with a flexible architecture and an advanced operating system, the Duplex Multi-Environment Real-Time (DMERT) operating system, and supports the high-level C programming language and a robust software environment. Finally, it provides switching system dependability, and is designed with large-scale integrated circuit technology, thus reducing energy consumption and size.

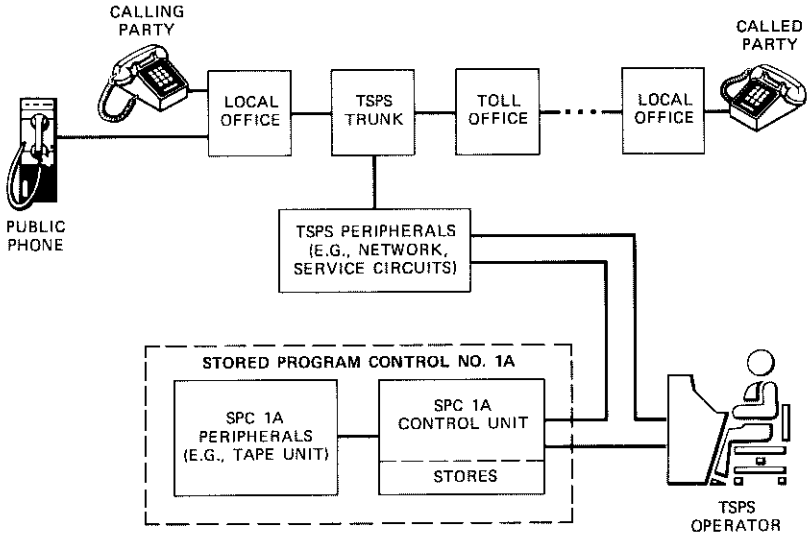
## IV. ORGANIZATION OF TSPS

### *4.1 TSPS No. 1 organization*

As we see in Fig. 1, TSPS No. 1 bridges an operator or service circuits onto a trunk connecting local and toll offices. TSPS No. 1 has dedicated specialized trunk circuits and peripherals (including the link network, service circuits, signal distributors, and scanners), a Stored Program Control No. 1A (SPC 1A) Processor, and operator consoles. Once the connection between customers is established, connections to operators or service circuits are released. Such connections are reestablished if additional services are required.

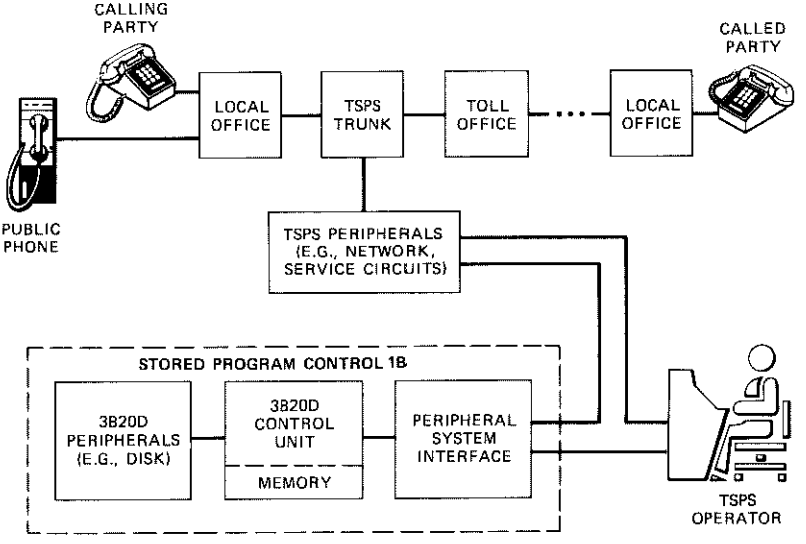
### *4.2 TSPS No. 1B organization*

As we see in Fig. 2, the general architecture of TSPS No. 1B is the same as TSPS No. 1, except that the SPC 1A is replaced by the Stored Program Control 1B (SPC 1B). The SPC 1B includes a 3B20D Processor, a Peripheral System Interface (PSI) to interface the existing TSPS peripherals to the 3B20D Processor, and various 3B20D Proc-




SPC – STORED PROGRAM CONTROL
TSPS – TRAFFIC SERVICE POSITION SYSTEM

Fig. 1—TSPS No. 1 organization.




TSPS – TRAFFIC SERVICE POSITION SYSTEM

Fig. 2—TSPS No. 1B organization.

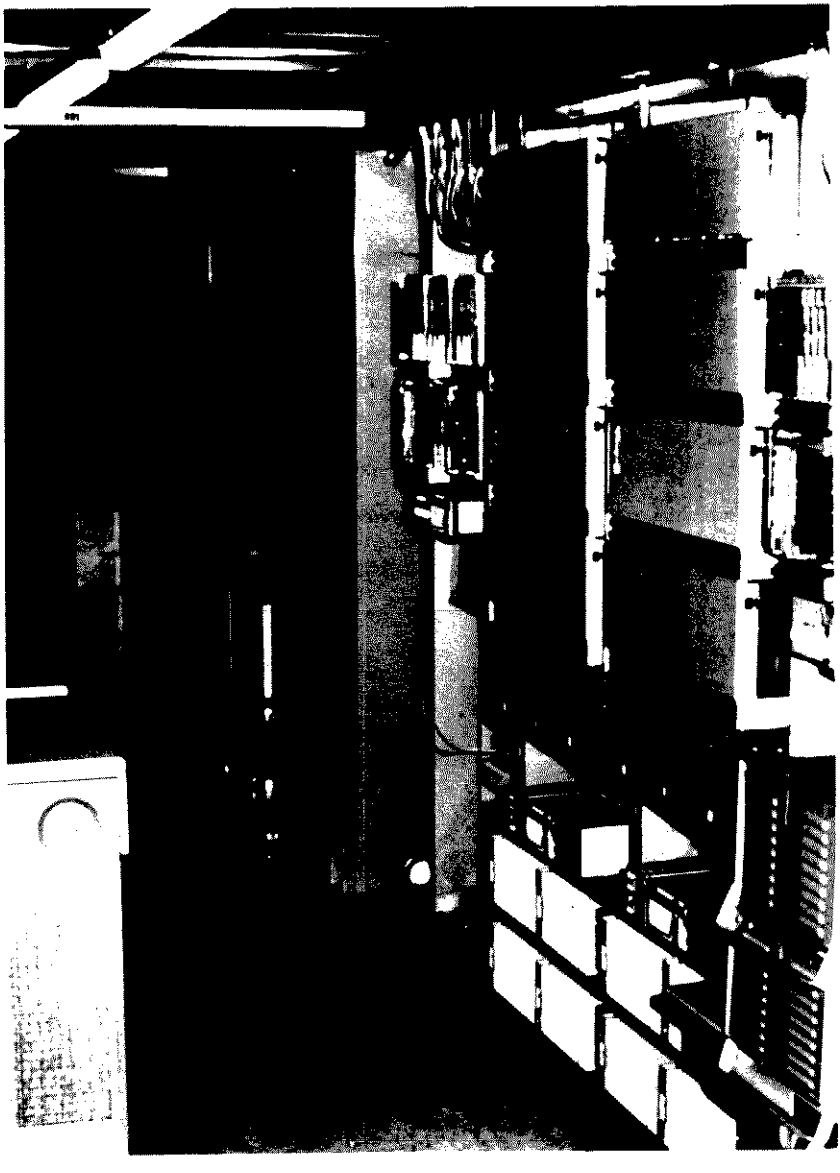


Fig. 3—3B20D Processor in the new TSPS No. 1B service installation at Fresno, California.

essor microcode and native-mode software to emulate the TSPS No. 1 environment.

The articles in this issue covering the TSPS No. 1B[8-16] include detailed descriptions of the system, the software architecture and

hardware configuration, the software development system, integration and testing of the TSPS No. 1B, retrofit procedures, capacity and reliability evaluation, Switching Control Center System interface, and long-range planning for TSPS No. 1B installation and growth.

## V. TSPS NO. 1B STATUS

### *5.1 New service installations (cordboard replacements)*

The first TSPS No. 1B was placed in service in Fresno, California, in November 1981. The 3B20D Processor in the TSPS No. 1B in Fresno is shown in Fig. 3. The Operator Services Center at Fresno is shown in Fig. 4.

The second TSPS No. 1B was placed in service in San Antonio, Texas, in January 1982.

### *5.2 Retrofit service installations (processor replacement in an in-service system)*

The first TSPS No. 1B retrofit or processor replacement in an in-service TSPS No. 1 took place in Redwood City, California, in March 1982.

### *5.3 Continuing deployment*

As of December 1982, 37 TSPS No. 1B's were in service. All but two of the TSPS No. 1B installations were retrofit installations to provide urgently needed expansion of either call capacity or memory capacity. Continuing retrofit deployment is scheduled for 1983 and beyond.

## VI. PERFORMANCE

Performance data from all TSPS No. 1B sites indicate that all design objectives have been achieved. More details are covered in later articles.

## VII. FUTURE TRENDS

The increased capacity of TSPS No. 1B will accommodate anticipated growth in operator services traffic for a number of years. In addition, an advanced operating system and high-level programming language will allow TSPS No. 1B to evolve readily to: (1) support the evolution of the stored controlled network,[5] (2) continue to improve operator efficiency, and (3) respond to changes caused by the evolving restructure of the telecommunications industry.

## VIII. SUMMARY

This paper has presented a general background for TSPS No. 1B as an introduction to the technical papers that follow. While all design

Fig. 4—Operator Service Center in Fresno, California.

details could not possibly be included, the papers in this issue provide a comprehensive overview of TSPS No. 1B.

## IX. ACKNOWLEDGMENTS

The development of TSPS No. 1B required the participation of hundreds of people in many organizations in Bell Laboratories, Western Electric, AT&T, and the operating companies. All of the authors in this issue are indebted to these organizations for their cooperation and the team effort that culminated in the successful completion of the TSPS No. 1B project. The authors of this paper also wish to acknowledge the contributions of all the team members whose work is summarized here, as well as the support of D. J. Leonard, K. E. Martersteck, and C. M. Rubald and L. C. Stecher, the coordinating editors.

## REFERENCES

1. R. J. Jaeger and A. E. Joel, Jr., "TSPS No. 1—System Organization and Objectives," B.S.T.J., special issue on TSPS No. 1, *49* No. 10 (December 1970), pp. 2417–43.
2. R. E. Staehler and W. S. Hayward, Jr., "TSPS No. 1 Recent Developments: An Overview," B.S.T.J., special issue on TSPS No. 1, *58*, No. 6 (July 1979), pp. 1109–18.
3. E. M. Prell, V. L. Ransom, and R. E. Staehler, "The Changing Role of the Operator," 9th Int. Switching Symp. Rec. (May 1979), pp. 697–703.
4. J. C. Kylin, E. M. Prell, and R. P. Weber, "Benefits of Integrating Data Bases into the SPC Network," Internal Conf. on Commun. Rec. (June 1979), pp. 3.4.1–4.
5. S. Horing, J. Z. Menard, R. E. Staehler, and B. J. Yokelson, "SPC Network: Overview," B.S.T.J., special issue on SPC Network, *61*, No. 7, Part 3 (September 1982), pp. 1579–88.
6. N. X. DeLessio, J. R. Kane, M. W. Rolund, J. M. Scanlon, and R. E. Staehler, "The 3B Processor System and Its Application to TSPS," 10th Int. Switching Symp. Rec. *3* (September 1981), Session 33A, Paper 4, pp. 1–9.
7. J. M. Scanlon, et al., "3B20D Processor and DMERT Operating System: Prologue," B.S.T.J., special issue on 3B20D Processor, *62*, No. 1, Part 2 (January 1983), pp. 167–9.
8. N. X. DeLessio and N. A. Martellotto, "Traffic Service Position System No. 1B: System Description," B.S.T.J., this issue.
9. R. J. Gill, G. J. Kujawinski, and E. H. Stredde, "Traffic Service Position System No. 1B: Real-Time Architecture Utilizing the DMERT Operating System," B.S.T.J., this issue.
10. H. A. Hilsinger, J. H. Tendick, and R. A. Weber, "Traffic Service Position System No. 1B: Hardware Configuration," B.S.T.J., this issue.
11. T. G. Hack, T. Huang, and L. C. Stecher, "Traffic Service Position System No. 1B: Software Development System," B.S.T.J., this issue.
12. R. Ahmari, R. S. DiPietro, S. C. Reed, and J. R. Williams, "Traffic Service Position System No. 1B: Integration and System Testing," B.S.T.J., this issue.
13. J. C. Dalby, D. Van Haften, and L. A. Weber, "Traffic Service Position System No. 1B: Retrofitting the Processor," B.S.T.J., this issue.
14. B. A. Crane and D. Suk, "Traffic Service Position System No. 1B: Capacity and Reliability Evaluation," B.S.T.J., this issue.
15. J. J. Bodnar, J. R. Daino, and K. A. VanderMeulen, "Traffic Service Position System No. 1B: Switching Control Center System Interface," B.S.T.J., this issue.
16. P. L. Bastien and B. Wycherley, "Traffic Service Position System No. 1B: TSPS No. 1/1B Long-Range Planning Tools," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

# *Traffic Service Position System No. 1B:*

# System Description

By N. X. DeLESSIO and N. A. MARTELLOTTO

(Manuscript received June 30, 1982)

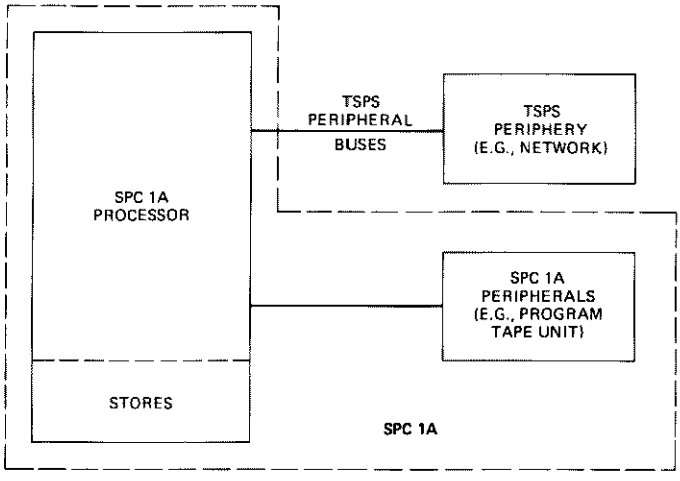
*The Traffic Service Position System No. 1B (TSPS No. 1B) is the first field application of the Bell System's new 3B20 Duplex Processor (3B20D) in the emulation mode. The 3B20D Processor replaces the existing Stored Program Control No. 1A (SPC 1A), while retaining the existing TSPS periphery and software. A key factor is the ability to switch between the emulated software and the 3B20D native software within a single process with a single instruction. This allows the flexibility of adding software in either environment as appropriate.*

## I. INTRODUCTION

The 3B20 Duplex Processor (3B20D), with its associated Duplex Multi-Environment Real-Time (DMERT) operating system, meets the objectives of Traffic Service Position System No. 1B (TSPS No. 1B).[1-3] The 3B20D Processor is significantly faster than the Stored Program Control No. 1A (SPC 1A) and provides the required increase in processor capability. The ability of the 3B20D Processor to emulate allows the retention of most of the existing TSPS No. 1 software and peripheral hardware. In addition, the 3B20D Processor consumes much less energy and is significantly smaller in physical size than the SPC 1A.

## II. SYSTEM STRUCTURE

The 3B20D Processor replaces the existing SPC 1A (Fig. 1) while retaining the existing TSPS No. 1 periphery and—through emulation—preserving the existing TSPS software. This software preservation is accomplished by defining (through microcode) one of the four




SPC – STORED PROGRAM CONTROL
TSPS – TRAFFIC SERVICE POSITION SYSTEM

Fig. 1—Existing TSPS system structure.

3B20D instruction sets to be that of the SPC 1A, thus emulating the old processor and allowing existing TSPS software to be transported to the 3B20D Processor almost intact. The ability exists to switch between the emulated instruction set and the 3B20D native instruction set within a single process with a single instruction. This allows the flexibility of adding new software into either environment, as appropriate. For example, some new TSPS No. 1B software, generated to take advantage of the new disk capability and to provide a new interface to maintenance personnel, is written in the C programming language,[4] which compiles into 3B20D native-mode assembly language. Both emulated and native-mode software are run under the DMERT operating system, allowing operating system services to be available to both forms of software. The emulated SPC 1A assembly language code is structured as a single process executing under DMERT.

The existing TSPS periphery is retained and interfaced with the 3B20D Processor through the use of a Peripheral System Interface (PSI) circuit. This unit is designed to interface the TSPS peripheral buses with the 3B20D Central Control Input/Output (CCIO) bus via an Application Channel Interface (ACHI). The PSI duplicates the signals, timing, and error checking of the SPC 1A. This enables TSPS peripheral units to remain unchanged. Future hardware can be added to the existing TSPS peripheral buses or can utilize the I/O processor of the 3B20D Processor to off-load the main processor and to provide fast block data transfers through direct memory access.

The combination of the 3B20D, PSI, and microcode required to

emulate the SPC 1A is called the SPC 1B (Fig. 2) and is the entity that replaces the SPC 1A. Because the processor replacement is economically attractive both for new installations and for retrofits, techniques have been developed to replace an SPC 1A with an SPC 1B in an in-service office. The resultant system provides significant increases in call processing and main memory capacities, allows the preservation and future growth of existing software and peripheral hardware, and adds modern software and hardware architectures to facilitate future feature introduction.

## III. PROCESSOR HARDWARE DESCRIPTION

The 3B20D Processor has been developed as the first member of a family of processors designed for a broad range of Bell System applications. The DMERT operating system provides a comprehensive set of functions associated with management of system resources such as the real-time, memory, input/output, and software processes.

The control unit of the 3B20D (Fig. 3) uses a 32-bit architecture throughout, including the memory buses to main store and an 8K-byte cache. Extensive self-checking logic is employed to ensure immediate detection of errors, thus supporting quick and graceful recovery measures. Hamming correction of all single-bit errors and detection of all double-bit errors are performed by the main memory controller. In addition, data parity is checked on every refresh operation required for the dynamic random access memories (RAMs) used in main

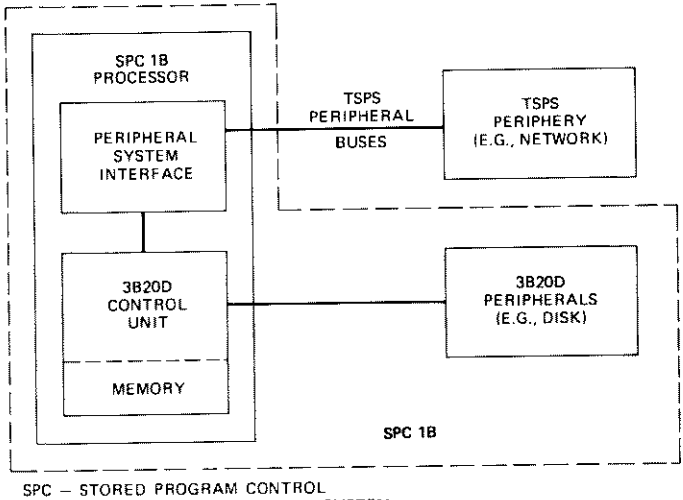



Fig. 2—TSPS No. 1B system structure.

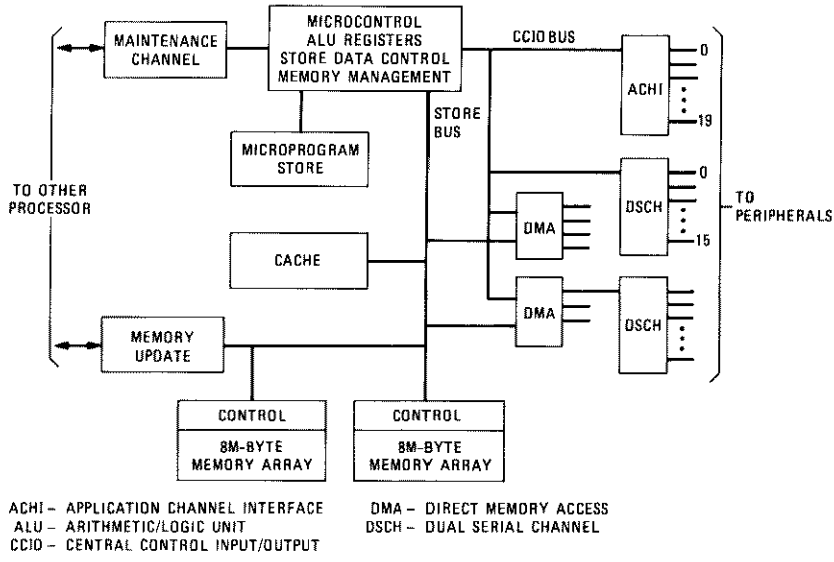



Fig. 3—3B Processor block diagram.

memory, thus ensuring that even infrequently used addresses are periodically checked for integrity. The 3B20D Processor uses a 24-bit virtual address, which is converted to a 24-bit physical address using a paged segmentation scheme. The 16M-byte address space is divided into 128 segments, each having up to 64 pages of 2K bytes each. Memory protection can be provided on either a segment or a page basis. Physical memory is growable in 512K-byte increments, to a maximum of 16M bytes. The main-memory access time is 525 nanoseconds, while the cache access time is 250 nanoseconds. Memory communication provides a byte-addressing capability with byte, half-word, full-word (32-bit), and move block options as part of the instruction repertoire. A memory management unit performs virtual-to-physical address mapping and main-store access protection. A high-speed, two-way, set-associative memory called the Address Translation Buffer (ATB) is provided to reduce the overhead associated with the address translation function. The ATB is divided into eight sections that are assigned to processes by software.

The Central Control (CC) is microprogrammable with the capability of executing a variety of instruction sets. Up to four instruction sets can be selected dynamically. The microstore uses a 64-bit word length with up to 16K words of high-speed, bipolar programmable read-only memory (PROM) or RAM available. A variable microcycle ranging from 150 to 300 nanoseconds is employed to optimize execution times. The native instruction set of the 3B20D Processor was designed to be

compatible with the C programming language. It optimizes the execution and memory-space utilization of the language while including instruction-level support for all C-language data types and control structures.

Figure 4 is a general block diagram of the 3B20D Processor. Two basic connections exist between the duplicated Control Units (CUs). One is an update connection that serves to keep the off-line CU's memory completely up to date. The second connection is a maintenance channel over which diagnostics of the off-line CU are performed. The Central Control and memory are duplicated and grouped as a switchable entity. The I/O and disk systems can be accessed by either CU through duplex intelligent controllers. The Disk File Controllers (DFCs) are normally both active in order to keep the data on the disks identical. Thus, under trouble conditions, either disk can support system operation. Unlike the SPC 1A Processors, the CUs are not run in a synchronous matching mode. Instead, both stores (on-line and standby) are kept up to date by the memory-update hardware concurrent with instruction execution. This is achieved by having the on-line memory-update circuit write into both memories simultaneously when memory data are written by the CC. Under trouble conditions, when control is switched to the standby CU, its memory will contain up-to-

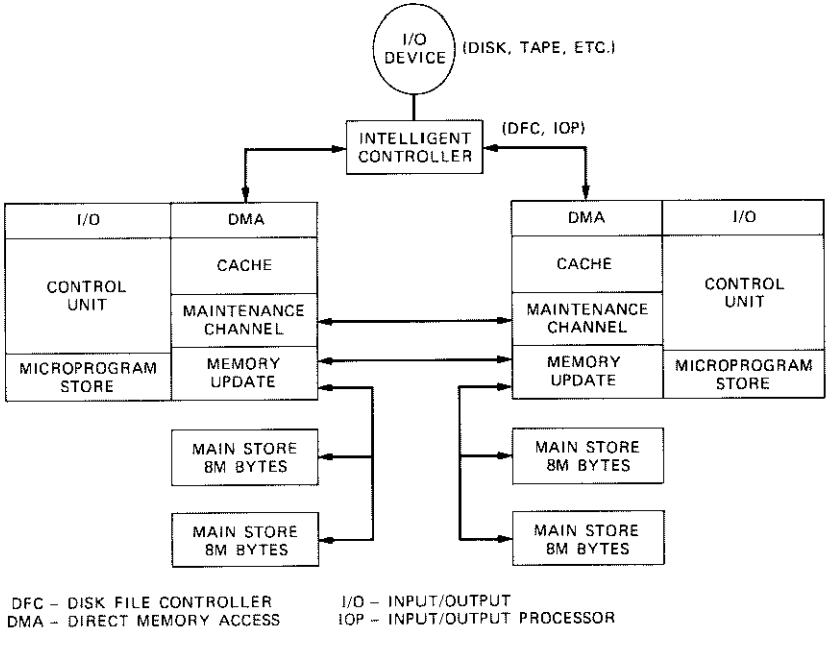



Fig. 4—General block diagram of the 3B20 Duplex Processor.

date information without performing a complete transfer from one CU to another.

3B20D Processor peripheral units are connected to the CC via the Direct Memory Access (DMA) unit. The DMA does not interface directly with peripheral units, but rather communicates with two intelligent subsystems, a Disk File Controller, and an Input/Output Processor (IOP). The CC builds job blocks for a peripheral unit that in turn notifies the CC upon job completion. The parallelism afforded by the autonomous processing capabilities of the DFC and IOP frees the CC for other work. Communication to both the DFC and the IOP is via Dual Serial Channels (DSCH) that allow any peripheral to operate with either CC of a duplex pair. Each DFC is capable of supporting 16 movable-head disk drives of 300M-byte capacity. Each IOP is capable of supporting a wide variety of peripherals, such as nine-track tape units, printers, synchronous and asynchronous data links, maintenance terminals, scanner/signal distributors, and custom network interfaces. Peripherals also may be connected to the Central Control Input/Output (CCIO) bus via an interface such as the Application Channel Interface (ACHI) used to communicate with the TSPS No. 1 periphery that was retained.

## IV. OPERATING SYSTEM DESCRIPTION

DMERT is the real-time operating system for the 3B20D Processor. This operating system was developed concurrently with the hardware and uses a multiple-environment approach where time-critical code coexists with time-shared software.[5] That is, one environment supports real-time response while another environment provides a time-shared interface similar to that of the *UNIX** operating system.[6] The architecture of the operating system is process oriented; this allows applications to write software at the level most productive for each task. A process is an instance of a program executing on the processor and is characterized by a separate virtual address space. The multiple environments are implemented via a kernel that supports three levels of processes, described below. The DMERT kernel provides the most primitive virtual machine as it handles hardware interrupts, timer interrupts, and operating system traps. In all cases, the kernel saves the state of the interrupted process, provides whatever service is requested, and then restores the state of the interrupted process. The first level of process, known as a kernel process, is offered limited services by the DMERT kernel and is dispatched because of real-time events such as interrupts. The second level of process, known as

* Trademark of Bell Laboratories.

supervisor, is offered more services by the DMERT kernel in the area of I/O and dynamic memory allocation. Supervisor processes support a third level of process called a user-level process. An example of this is a *UNIX* operating system process controlled by a kernel-level process. In order for processes to cooperate in accomplishing their tasks, DMERT provides a set of interprocess communication and synchronization mechanisms including messages, events, process ports, interprocess traps, and shared memory. These interprocess communication primitives are fundamental to the DMERT structure.

I/O is accomplished by a kernel process, known as a driver. A unique driver is provided according to the type of I/O device, such as disk or IOP. Drivers receive I/O requests in the form of messages. When the I/O is completed, the message is returned to the requesting process. An intermediate supervisor process known as the file manager stands between disk users and the disk driver. This process implements a logical file system on the disk for those processes that care to use it. Files can be created, opened, closed, grown, and deallocated through the file manager. Hierarchical directories of files such as those found in the *UNIX* operating system are supported.

A set of processes and a special IOP peripheral controller provide a modern craft interface for the 3B20D and the TSPS No. 1B. A split-screen cathode ray tube (CRT) and a printer interface are utilized. The top portion of the screen is the status and display portion. Various status and display pages can be called upon demand. A special page that provides basic machine-control features such as initialization requests is provided by peripheral firmware. The lower portion of the screen is utilized for terminal I/O messages.

The continuous operation aspects of the 3B20D Processor are supported by a number of processes that are an integral part of the DMERT operating system. These processes handle error interrupts, control processor switches, and provide I/O to common peripherals such as disks; they also run equipment diagnostics and audit key data structures for consistency. Reference 7 contains a detailed description of both the 3B20D Processor and DMERT.

## V. SOFTWARE STRUCTURE

The software structure of the TSPS No. 1B system was governed by three major design goals. First, the software architecture had to maximize the call-handling capacity. Second, steps were taken to maintain the SPC 1A programming environment as closely as possible in order to allow maximum utilization of existing software support utilities. Third, the existing TSPS software was to be emulated with a minimum of modifications. This guideline precluded unnecessary redesigns or restructures of the current field-proven software.

The emulation of the SPC 1A at the instruction level by the microcoding of its instruction set and at the system level by the PSI has allowed most TSPS programs to be executed on the 3B20D Processor with minimal modifications. The TSPS emulated code has been incorporated into a single, large, high-priority kernel process under DMERT.

The single process structure was dictated primarily by the existing tightly coupled nature of the TSPS software. As on the SPC 1A, all TSPS data and programs share and have access to the entire address space and communicate through data structures that reside in memory. Retaining these structures maintained the goal of exact emulation and avoided interprocess communication overhead detrimental to achieving the required performance gain.

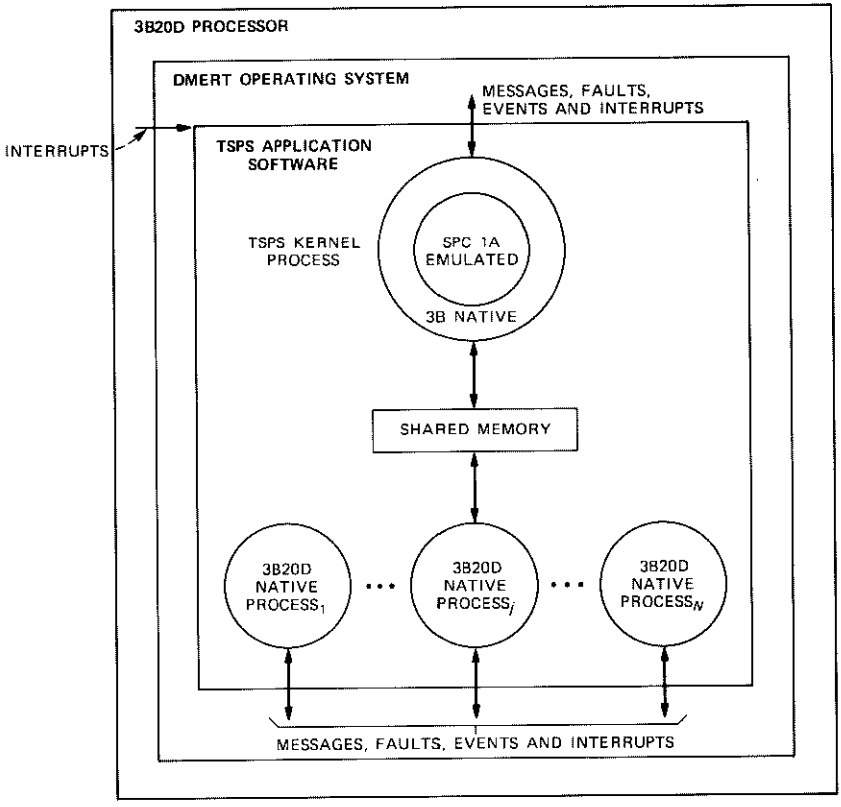
In addition to the emulated code, the TSPS kernel process contains native-mode (C-language) code that provides several capabilities. First, it provides a standard C-language data structure interface to the operating system and to other processes. Native code resident in the TSPS kernel process also works in conjunction with emulation microcode to implement system-level emulation of the SPC 1A interrupt structure within the actual interrupt structure defined by the 3B20D and DMERT. In addition, native code resident in the TSPS kernel process has eliminated the need to introduce new instructions, such as system calls, not available in the emulated instruction set.

All entries into the TSPS kernel process are through native code, which then calls the emulated code as a subroutine with a single instruction. Certain functions exist in the current TSPS software that would have required extensive modifications to emulate because of machine dependencies. In these cases, new replacement native code was written as subroutines that are called with a single instruction from the emulated code. In cases where completely new functions were implemented, native-mode processes separate from the TSPS kernel process were generated. An example of this is the TSPS File System Interface, which provides disk file system access as a replacement for functions previously implemented using the program tape unit of the SPC 1A.

The control structure of the TSPS call processing and peripheral maintenance software, being emulated, is almost identical to what exists on the SPC 1A. The major difference is that rather than running continuously as on the SPC 1A, the high-priority TSPS kernel process voluntarily must give up control of the machine periodically to allow lower priority processes to run. This is done by requesting periodic time-outs from DMERT. After the specified time has passed, the TSPS kernel process is reentered. The native code responds to the event and causes the emulated code to resume where it had left off.

The real-time breaks are not noticeable to the emulated software. The combined DMERT and TSPS priority structure is such that the TSPS kernel process dominates control of real time.

The process structure of TSPS No. 1B is summarized in Fig. 5. The DMERT operating system running on the 3B20D Processor provides a high-level, multiprocess environment for TSPS application processes. In this environment, the TSPS kernel process appears to the DMERT operating system to be identical to other native-mode processes that utilize the facilities of messages, faults, events, and interrupts in communicating with the operating system and—through the operating system—with other processes. The TSPS kernel process, in some instances, also communicates with other TSPS application processes through shared memory. An example, noted above, is the TSPS File System Interface. Within the TSPS kernel process, the combination




SPC – STORED PROGRAM CONTROL
TSPS – TRAFFIC SERVICE POSITION SYSTEM

Fig. 5—TSPS No. 1B process structure.

of emulation microcode, PSI, and native code creates an SPC 1A environment, thereby shielding the emulated code from the details of the specific machine and operating system on which it is running.

## VI. SUMMARY

The processor capability was increased by replacing the existing SPC 1A with the 3B20D Processor while retaining the existing TSPS periphery and using emulation to preserve the existing TSPS software. A key aspect in the software architecture is the ability to execute either the emulated instruction set or the 3B20D native instruction set, and to switch between the two within a single process with a single instruction. Thus, it is possible to add new software to either environment and thereby increase the future flexibility of the system. Because the processor replacement is economically attractive both for new installations and for retrofits, techniques have been developed to replace an SPC 1A in an in-service office.

## VII. ACKNOWLEDGMENT

The design of the 3B20D Processor and TSPS No. 1B required the cooperative efforts of a large number of people in Bell Laboratories, Western Electric, and AT&T. The authors wish to acknowledge the contribution of all of the team members whose work is summarized here.

## REFERENCES

1. R. E. Staehler, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.
2. N. X. DeLessio, J. R. Kane, M. W. Rolund, J. M. Scanlon, and R. E. Staehler, "The 3B Duplex Processor System and Its Application to TSPS," 10th Int. Switching Symp. Proc., September 21–25, 1981.
3. N. A. Martellotto, "An Operating System For Reliable Real-Time Telecommunications Control," Fourth Int. Conf. on Software Eng. for Telecommun. Switching Systems Proc., University of Warwick, England, July 20–24, 1981.
4. B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, Englewood Cliffs, NJ: Prentice-Hall, 1978.
5. H. Lycklama and D. L. Bayer, "*UNIX* Time-Sharing System: The MERT Operating System," B.S.T.J., *57*, No. 6, Part 2 (July 1978), pp. 2049–86.
6. D. M. Ritchie and K. Thompson, "The *UNIX* Time-Sharing System," B.S.T.J., *57*, No. 6, Part 2 (July 1978), pp. 1905–29.
7. J. M. Scanlon, "3B20D Processor & DMERT Operating System: Prologue," B.S.T.J., *62*, No. 1, Part 2 (January 1982), pp. 167–9.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Real-Time Architecture Utilizing the DMERT Operating System

By R. J. GILL, G. J. KUJAWINSKI, and E. H. STREDDE

(Manuscript received June 30, 1982)

*The Traffic Service Position System No. 1B (TSPS No. 1B) architecture was conceived to increase performance significantly for future features and traffic growth. The design preserves the TSPS No. 1 software with minimal changes. At the same time, the 3B20 Duplex Processor (3B20D) used in TSPS No. 1B provides additional processor peripherals and a modern programming environment with a real-time operating system. This paper describes how the TSPS No. 1 software, initially designed to run on the Stored Program Control No. 1A (SPC 1A), executes on the SPC 1B of the TSPS No. 1B. The SPC 1B is a 3B20D tailored for the TSPS application and provides an SPC 1A environment by directly emulating its instruction set. The paper also presents major TSPS application processes and their interaction with the emulated TSPS process and the Duplex Multi-Environment Real-Time (DMERT) operating system of the SPC 1B. In addition, the paper describes the integration of TSPS maintenance software into the DMERT maintenance structure.*

## I. INTRODUCTION

The Traffic Service Position System No. 1B (TSPS No. 1B) real-time architecture was designed to meet the project goals discussed in Ref. 1. The implementation of this architecture entailed four major developments:

(*i*) Replacement of the Stored Program Control No. 1A (SPC 1A) of TSPS No. 1 with the 3B20D Processor, the TSPS Peripheral System Interface (PSI), and microcode to execute the SPC 1A instruction set,

which together comprise the Stored Program Control No. 1B (SPC 1B)

(*ii*) Emulation of most existing TSPS No. 1 software structured as a process under the Duplex Multi-Environment Real-Time (DMERT) operating system

(*iii*) Development of additional processes to support the emulation

(*iv*) Integration of the PSI and TSPS peripheral maintenance into the overall DMERT maintenance structure.

Before discussing the TSPS No. 1B real-time architecture, this paper presents two sections of background information. Section II presents an overview of how TSPS operates using the SPC 1A. Section III reviews the fundamentals of DMERT, and Section IV describes the SPC 1B and the TSPS No. 1B software architectures. These sections enable the reader to understand the real-time architecture of the TSPS No. 1B. More detailed information can be obtained by reading the references.

## II. TSPS NO. 1 REAL-TIME ARCHITECTURE

This section presents a brief overview of the TSPS No. 1 operation on the SPC 1A and provides a base for understanding how the TSPS No. 1 was emulated on the TSPS No. 1B. A complete description of TSPS No. 1 operation on the SPC 1A can be found in Refs. 2 and 3.

### *2.1 SPC 1A programming environment*

The SPC 1A uses 20-bit addresses to reference approximately one million 20-bit words of main memory. The address spectrum consists of up to 30 store name codes with each name code containing 32K 20-bit words. Store name code 0 is not used since low memory addresses are mapped into the SPC 1A's buffer bus (see Section 2.2). Store name code 31 (037) is not implemented. Hence, the maximum physical memory size for the SPC 1A is 960K 20-bit words. The SPC 1A does not support memory management. Hence, there is no virtual addressing; all addresses are physical addresses. All of memory is equally accessible (shared) by all programs.

Write protection can be set on a 2K word boundary within a name code. Protected areas within the name code, however, must be contiguous, and there can be only one protection change boundary within a name code. Memory is unprotected from the high end of the address spectrum within each name code. For example, if one fourth of a name code is to be unprotected (read/write), then the first three quarters would be read only, and the last quarter would be unprotected. Protected areas can be unlocked to change programs or fixed data by having the processor execute a special unlocking sequence. Typically, protected areas are used for office data, read-only tables, and program

logic. Unprotected areas are used for volatile data (e.g., call information).

SPC 1A instructions are 40 bits wide (two 20-bit words). The odd-addressed word contains the operation code, while addresses or data, when present, are placed in the even-addressed word. There are relatively few instructions, but a single instruction can have many options and perform several operations. Each instruction takes from one to three 6.3-microsecond machine cycles to execute. The execution time of each instruction is fixed regardless of the number of options specified. Hence, the time a segment of code will execute can be precisely determined.

### *2.2 Buffer bus*

The SPC 1A buffer bus is an internal collection of processor and peripheral status and control registers. The buffer bus registers are used for such things as peripheral and processor configuration control and status indications, interrupt sources, and interrupt masks. The buffer bus consists of 24 registers with 20 to 24 bits each. Each register has a low-memory address (e.g., 600 or 2200) associated with it. All buffer bus registers can be read with Memory to Register (MR) instructions. Some registers can be written, set, or reset using Register(s) to Memory (RM or RRM) instructions or Constant to Buffer Bus (CBB) instructions. Other registers are read-only.

### *2.3 Interrupt structure*

#### *2.3.1 Interrupt levels*

There are nine interrupt levels in the SPC 1A. These interrupt levels are A (highest priority), B, C, E, F, G, H, J, and K. A tenth level, L (commonly referred to as base level), runs continuously in the absence of any interrupt. Normally, base level is only interrupted every 5 ms by J-level. Although, base level is the lowest-priority processing level, the bulk of TSPS software (e.g., call processing, diagnostics, audits) executes in base level.

Each interrupt level can only interrupt lower-priority levels with the sole exception being that A- and B-level can interrupt each other. An A-level interrupt is caused either by a manual action at the Maintenance Control Center (MCC) Control and Display (CD) frame or by the execution of an ANOP instruction (used to fill unused instruction space). B-level is entered as a result of a processor switch or for emergency actions required as a result of system-sanity-check failures. C- and K-level interrupts generally occur as a result of processor errors. C-level handles SPC 1A Processor fault recovery. K-level interrupts are only enabled during processor diagnostics for error-data recording. SPC 1A store errors will generate E- and G-level interrupts.

E-levels also result from software errors such as invalid addresses or protection violations. E-levels provide data for immediate problem analysis and necessary store reconfiguration. G-level is used for collecting store-error data for intermittent failures. Peripheral-unit errors will generate an F-level interrupt. F-level software is dedicated to peripheral fault recovery. J- and H-levels perform the system I/O.

#### 2.3.2 J- and H-level interactions

The J-level interrupt is generated every 5 milliseconds by a clock pulse. Because of its periodic nature, J-level provides the main time reference for all application processing. Within J-level there are two classes of jobs: high priority and low priority. The high-priority jobs are executed first. While they are running, H-level is inhibited. In making the transition to low-priority jobs, H-level is enabled. H-level programs consist of the high-priority, J-level jobs. An H-level is caused whenever J-level executes longer than 5 ms and low-priority jobs are being run (see Fig. 1*). This ensures that the high-priority jobs are executed every 5 ms, unless H-level runs longer than 5 ms. In this case the high-priority jobs would miss an execution, as H-level cannot interrupt itself.

J- and H-levels on the SPC 1A have two separate interrupt sources that are driven by the same clock pulse. The H-level interrupt source is only enabled when low-priority J-level is executing. When a return from interrupt is performed from low-priority J-level, the H-level inhibit bit is set.

#### 2.3.3 Interrupt handling

When an interrupt occurs under normal conditions on the SPC 1A, the processor saves the address of the interrupted program and passes control to the first instruction of the interrupt program for that level. The address of the interrupt program and the save area (bin) for the interrupted program address are known (i.e., hard-wired) by the processor. The interrupt-handling program for each interrupt, except J-level, first saves the contents of the seven general-purpose registers in memory and then determines what actions to take. Because of the frequency of the J-level interrupt (every 5 ms), general-register contents are copied to an auxiliary set of registers by the hardware. The overhead of saving and restoring the registers every 5 ms is greatly reduced by performing this function in hardware.

The return from interrupt is normally performed by the Execute Go Back To Normal (EGBN) instruction. The one exception is J-level. J-

* Acronyms and abbreviations used in the figures and text are defined in the Glossary.

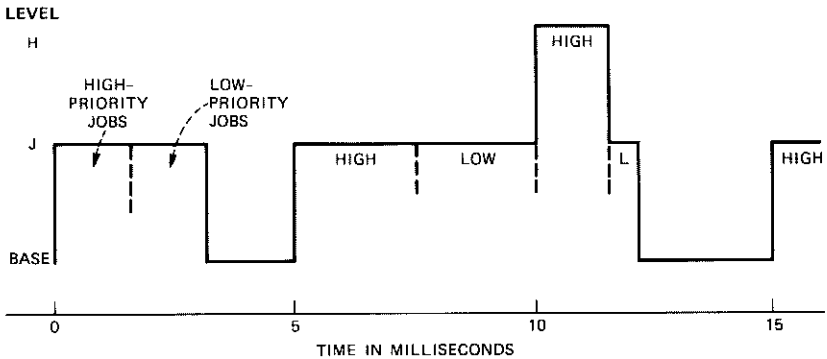



Fig. 1—5-ms input/output processing.

level uses the Go Back To Normal H- or J-level (GBNHJ) instruction. In returning, the interrupt handler (except for J-level) first restores the general-purpose register contents from memory (if it were going to return to the point of interrupt) and then executes the EGBN. The EGBN instruction determines what interrupt level is being returned from by looking at the interrupt-level activity flags in the buffer bus. It restores the address of the interrupted program from the save area for that interrupt and clears the highest interrupt bit in the buffer bus interrupt-level activity word, thus dropping to the next highest, previously active level. If an intermediate-level interrupt is pending, it would be serviced at this time. The GBNHJ is slightly different in that it causes the general-purpose registers to be restored by hardware from the auxiliary set of registers. It also disables the H-level interrupt in addition to performing the functions of the EGBN.

Any interrupt servicing routine can change its point of return by overwriting the saved address in the appropriate interrupt bin in memory. Also, an interrupt is effectively returned from by merely clearing its activity bit in the buffer bus, which essentially erases the history of the interrupt. This immediately puts the program into the next lower, previously active level. This re-enables that interrupt level as well as any intervening levels.

#### *2.3.4 Interrupt inhibiting*

Interrupts are normally only masked when the program is handling a higher-priority interrupt. The interrupt activity bits in the buffer bus inhibit lower-priority interrupts from occurring. E-, F-, H-, and J-level interrupts can also be individually inhibited by setting the specific inhibit bits in the buffer bus. The H-level inhibit should always be set except when in low-priority J-level. The J-level inhibit can be set for brief periods when executing "critical region" code in base level, which

should not be interrupted by J-level activity because of potential interference. Similarly, the H-level inhibit could be set in low-priority J-level to prevent H-level interference. Some instructions have an "inhibit I/O interrupt" option that inhibits H- and J-level interrupts until the completion of the following instruction. For some instructions this option is implicit (not an option, but always on). The E- and F-level inhibits can be set manually as well as by software. C-level interrupts can be inhibited by software, and A- and B-level interrupts can be manually inhibited from the MCC.

### 2.4 Program structure

J-level (and H-level) programs execute under control of the Executive Control for I/O (ECIO) program. For the most part ECIO passes control to a predetermined and fixed set of programs every 5 ms. The list of jobs varies from execution to execution, but is cyclical over a 300-ms interval for high-priority work and over a 200-ms interval for low-priority work. Thus, J-level jobs execute with frequencies that are multiples of 5 ms, ranging up to 200 or 300 ms. Some program executions are permanent (always active) while others are run only on demand.

Similar to J-level, base-level programs are run under control of the Executive Control for the Main Program (ECMP) routines. Base-level programs execute in one of six priority classes. These priority classes are interject (highest priority), A, B, C, D, and E. ECMP passes control to jobs in priority classes A through E with a relative frequency of 15:8:4:2:1, respectively (see Fig. 2). Jobs within each priority class can run every execution of the priority class or only as needed on demand. Each priority class has a set of task dispensers that pass control to individual programs (tasks) when work is to be performed.

Interject work is requested when certain immediate actions are required (e.g., to immediately unload a full input hopper) or at a fixed frequency on demand (i.e., J-level-initiated sanity check of interject operation). A check for interject requested work is performed at least once in every base-level loop (one E-priority class to E-priority class cycle) and after every task. Interject should be serviced quickly. Hence, all base-level tasks are designed to run in short segments. Figure 3 illustrates the base-level priority-class execution with the fixed-interject check and another representative interject job executing in the middle of a priority class.

### 2.5 System integrity

The objective of system integrity is to provide an uninterrupted call-processing environment. The reliability goal for TSPS No. 1 is to achieve less than three minutes per year of total system outage

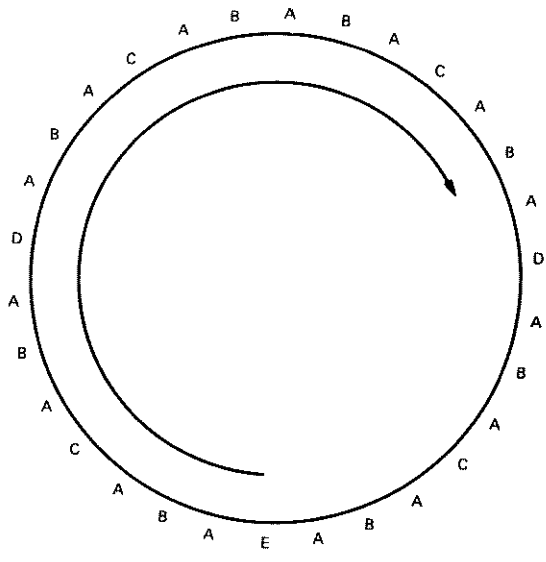



Fig. 2—TSPS No. 1 base-level loop.

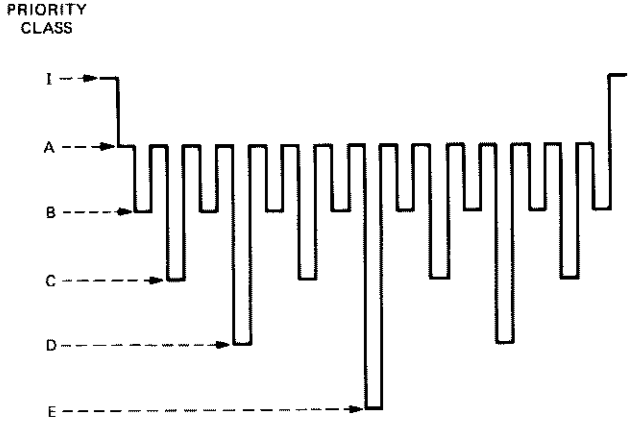



Fig. 3—TSPS No. 1 base-level, priority-class frequency.

averaged over all systems. This goal covers outages attributable to any possible cause. Hence, the term "system integrity" is used in this and other sections of this paper to encompass software stability as well as the traditional hardware reliability.

#### *2.5.1 Hardware integrity*

The maintenance strategy for TSPS hardware is based on the duplication of all critical units. This hardware redundancy allows

faulty units to be switched out of service with the load being carried by the remaining good unit. Maintenance programs are organized on priority levels such that the faulty unit can be removed from service as soon as possible, and then later, while the system is processing calls, the faulty unit is diagnosed in order to isolate the failing circuit pack. There are three main types of maintenance programs for the TSPS peripherals. These are, in order of decreasing priority, fault-recognition programs, diagnostic programs, and exercise programs.

Fault-recognition programs run when the presence of a fault is detected. They determine which of the duplicated units is in error, and reconfigure the system around the problem. Before returning to call processing, the fault-recognition program initiates a diagnostic request on the unit suspected of malfunctioning. The purpose of the diagnostic program is to provide resolution of the fault by indicating to the craft personnel the smallest replaceable unit (e.g., a circuit pack). This is accomplished by running a series of tests on the suspected hardware unit and then comparing the actual test results with a set of expected values. Another method of detecting faults employs the use of exercises. These programs are similar to diagnostics in that they run selected tests on the hardware. They differ in that they are intended to find faults in circuits not exercised by normal system operation (e.g., by call processing). Unlike diagnostics that are initiated by fault-recognition programs upon detection of a fault, the exercise programs are scheduled periodically.

### *2.5.2 Software integrity*

***2.5.2.1 Initialization and recovery.*** Whenever the state of the software is such that normal processing cannot continue, call-processing recovery actions are taken in an attempt to restore the system's sanity. The least severe actions are taken first. If these fail, the recovery attempt is escalated to the next highest level. The five TSPS recovery phases are Minor Audits, Major Audits, Selected Audits (miniphases), System Initialization A (SIA) and System Initialization B (SIB). Except for Selected Audit phases, these recovery phases can be manually requested or automatically generated by the software. Selected Audits phases can only be initiated by software, but they can be manually inhibited.

All audits that run during a recovery phase are "stitched" together. This means that they are run consecutively. Meanwhile, the normal base-level priority-class execution and all call processing is temporarily suspended. Minor Audit phases are short and, as a result, they have the least effect on call-processing activity. Major Audit phases are more extensive, and, hence, have a more disturbing effect. Selected Audit phases are run as the result of problems detected by software

sanity checks. A specific set of audits are stitched together, depending on the error found. SIAs and SIBs run a complete set of audits. An SIA also zeroes most unprotected memory and initializes the hardware. An SIB is more extensive in that it performs a more thorough hardware initialization. If an SIB fails to restore system sanity, another SIB will automatically be taken with a different hardware configuration. Looping SIBs with hardware reconfigurations will continue indefinitely until the system is recovered or manual intervention takes place. For the Minor, Major, and SIA phases, recovery actions are identical whether the phase is software generated or manually requested. On the other hand, a manually requested SIB will zero all unprotected memory, but it will not cause a hardware reconfiguration.

***2.5.2.2 Reference returns.*** Under certain conditions, maintenance-interrupt routines (levels A through K) must return to H-level, J-level, or base-level at a reference point in the job administration stream when a return to the interrupted point is not warranted. This safe transfer of control is called a reference return. The base-level reference return will result in the base-level cycle restarting at the end of E priority. The H- and J-level reference returns result in the cancellation of the job being run at the time of the interrupt, but the remaining scheduled jobs are executed.

***2.5.2.3 Sanity.*** Sane program execution is monitored via a hierarchical scheme. Base level checks itself every E-priority class by determining that the various priority classes have been run the proper relative number of times (15:8:4:2:1). Base-level sanity is checked in J-level by requesting an interject job every 500 ms and monitoring its execution. Failure to execute interject suggests a base-level loop. High-priority J-level (and, hence, H-level) monitors low-priority J-level by monitoring the execution of a fixed 100-ms job. High-priority J-level (and H-level) has the responsibility to reset a hardware sanity timer every 500 ms. Failure to reset the timer within 640 ms or resetting it too soon (less than 320 ms) will cause the timer to generate a B-level interrupt. Low-priority J-level insanity will result in a minor audit call-processing phase. Continued interject response failures or base-level priority class execution insanity will also trigger a minor phase. High-priority J-level also monitors system sanity by checking the average time of the last three E-E cycles. If this time exceeds a threshold a minor phase is taken. Lower threshold crossings will trigger overload recovery actions.

***2.5.2.4 Overload strategy.*** If at any point the elapsed time to run the last three E-E cycles exceeds a minimum threshold, phase 1 overload actions are taken. These actions consist of gradually busying trunks back to the local offices and reducing the rate at which processing of new calls is allowed to begin. If during phase 1 overload the elapsed E-E times exceed another, higher threshold, phase 2 actions are taken.

These actions busy all trunks to the local offices except for a minimum number and inhibit the processing of any new calls in the system. As E-E times return to acceptable levels, the system returns to phase 1 actions and eventually to normal. The return to normal operation (unbusying trunks and increasing the new call-processing rate) is done gradually as the overload subsides.

## III. DMERT OPERATING SYSTEM

The DMERT operating system[4] evolved from the MERT[5] operating system. While MERT was designed to operate on a simplex minicomputer, DMERT has incorporated maintenance software to control the duplex hardware in order to provide Electronic Switching System (ESS) reliability on the 3B20D Processor.

### *3.1 Processes*

An executable entity under DMERT is called a process. A process is a collection of programs and data with a distinct virtual address space (see Section 3.2) that is executed as a unit to perform a single (or set of related) function(s). Once in execution, a process controls the scheduling of its internal routines, barring any external stimulus such as an interrupt or fault. While the process is executing it appears to have an entire (virtual) machine to itself, although it may be interrupted by higher-priority processes or even swapped out to disk while waiting for some event to occur (e.g., the completion of an I/O request). The process address space may be completely protected from access by other processes or it may be shared at will. One process can communicate with another via several mechanisms supported by DMERT, as described in Section 3.6.

### *3.2 Memory management*

The basic unit of memory handled by the 3B20D's memory management system is a page: 2K 8-bit bytes (five hundred twelve 32-bit words). A segment consists of 1 to 64 pages that need not be contiguous in physical memory. A process in DMERT consists of at least three segments, where one segment contains the process stack, another contains the process control block (PCB), and at least one segment contains executable code. Each process executes in its own logical (or virtual) address space, which may be as large as 16M bytes (4M words). Memory management swaps processes between memory and disk, enabling many processes to coexist even though the sum of their memory requirements exceeds the physical memory of the processor. It also provides protection from misuse (i.e., writing into read-only memory) and unauthorized access by other processes.

The information required to perform virtual address to physical

address translation is maintained by DMERT in segment and page tables in memory. Accessing these tables for every address translation would take a considerable amount of time (6.8 microseconds each). To speed up this process a high-speed associative memory called an address translation buffer (ATB) is used to keep the most recently used translation data. There are eight 64- × 2-word-set associative memories. Four of the ATBs can be dedicated to individual processes. One other is dedicated to the kernel. The other three are shared dynamically by all other processes. There is one each for kernel, supervisor, and user processes. These processes are described in Section 3.3. Address translation via the ATB is performed in 150 nanoseconds.

### *3.3 Abstract machines*

DMERT supports four levels of software. Each level provides a different abstract view of the machine to the software. These abstract machine levels are:

(*i*) The kernel
(*ii*) Kernel processes
(*iii*) Supervisor processes
(*iv*) User processes.

The kernel is the lowest abstract software level under DMERT and the core of the operating system. It provides the basic services of the operating system, such as interrupt control, process dispatching, scheduling, and timing. It essentially extends the set of operations for kernel and supervisor processes.

The kernel-process level is used for those processes that have stringent timing constraints and must respond rapidly to real-time stimuli such as interrupts. Also, processes that must directly interact with hardware devices (such as a peripheral-unit driver) are coded as kernel processes. Similar to the kernel, kernel processes can have direct hardware access. Kernel processes also share some system data (e.g., the kernel stack and message buffers) with the kernel. Other system data are accessed via kernel services. Because of their performance requirements, kernel processes are totally memory resident and cannot be swapped out to disk. Some DMERT kernel processes (e.g., the process manager and memory manager) are referred to as special processes and share the kernel address space.

The next highest abstract machine level is for supervisor processes. This level is for those processes that neither have stringent timing constraints nor require direct access to the hardware or system data. Access to hardware (i.e., for I/O) and to system data is provided to supervisor processes by the kernel and kernel processes. The hardware and system data are completely shielded from supervisor processes.

Because of this layered software structure, errors in supervisor processes are much less likely to have catastrophic system effects. The price for this protection, however, is slower response time and performance. To improve response time supervisor processes have the option of being memory resident and not being swapped out to disk.

Both kernel and supervisor processes have the option of being nonkillable. A nonkillable process must perform its own internal fault recovery. A killable process can be terminated and recreated under certain error conditions.

The highest and most abstract software level under DMERT is the user-process level. User processes exist only in conjunction with a supervisor process, and in effect are just a unique state of that supervisor. The user portion of the process, however, does have a separate virtual address space distinct from the supervisor portion. A supervisor process can gain access to its user address space, but the reverse is not true. As a result, user processes are totally removed from the details of the actual machine and operating system under which they execute. Hence, the user level is the easiest programming level. However, user processes have poorer performance than supervisor processes.

### *3.4 Interrupt structure*

Interrupts are detected between the execution of two instructions and change the sequence of execution. More specifically, an interrupt results in the interruption of the current executing process and a transfer of control to a specific interrupt-handling process. The state of the interrupted process is saved on the interrupt stack so that it can be restored at the completion of the interrupt processing and the interrupted process can resume execution. There are 32 maskable, hardware interrupt sources contained in the interrupt source (IS) register. The 3B20D also has four unmaskable interrupt sources. Interrupts can be generated by hardware (i.e., clocks and peripheral devices), microcode, and software. Corresponding to the IS is an interrupt mask (IM) register. The IM and IS registers are "anded" together to determine which interrupts are allowed to occur between any two instructions.

Table I shows the layout of the IS as used in TSPS No. 1B. Bit 0 is the highest-priority interrupt source. That is, if more than one interrupt source is set and unmasked, the lowest-order bit position will be serviced first. Of particular interest are the Program Interrupt Request (PIR) sources (bits 17 through 31). There is one PIR per DMERT execution level 1 through 15 (see below). A PIR is set in response to a process request to send an event or fault (discussed later). The PIR corresponding to the execution level of the receiving process is set.

Table I—DMERT interrupt source register

| Bit | Use |
|---|---|
| 0–1 | Hardware errors |
| 2 | Software errors |
| 3 | Unused |
| 4 | PSI errors |
| 5 | Timer (10 ms) |
| 6 | TSPS (5 ms) |
| 7–9 | Unused |
| 10–13 | Direct Memory Access Input/Output (DMA I/O) |
| 14–16 | Non-DMA I/O |
| 17–31 | Program Interrupt Requests (PIRs) |

Since supervisor and user processes always receive their events at level 1, PIR 0 is not required. For user processes, the associated supervisor process receives events and faults and deals with them appropriately on behalf of the user process.

Currently, the only use of the unmasked interrupt sources is for Operating System Traps (OSTs). The execution of an OST by a process causes this interrupt to be set, which results in the kernel gaining control of the machine (the kernel services this interrupt source) and providing the requested service. In some instances the service is provided by a kernel process. *UNIX** operating system user-level services are provided by a DMERT supervisor process. In these latter two cases the kernel passes control to the appropriate process via its OST entry.

### *3.5 Execution levels and priorities*

DMERT prioritizes processes into sixteen execution levels (see Fig. 4). Execution level 15 is highest in priority. Kernel processes run at execution levels 2 through 15. Application kernel processes can only use execution levels 3 through 14. Level 2 is reserved for DMERT special processes, and level 15 is used by the DMERT Timer and Error Interrupt Handler. Supervisor and user processes run at execution level 0, and supervisor processes in a critical region run at level 1. Running at level 1 prevents any other supervisor process from interrupting the supervisor process while in its critical region. Within execution level 0, all supervisor and user processes are further prioritized within a 256-level priority structure. Supervisor and user processes are scheduled on a highest-priority basis by the DMERT scheduler. Processes of equal priority are scheduled among themselves using a round-robin scheme.

Kernel processes are dispatched as a result of an interrupt. This

* Trademark of Bell Laboratories.

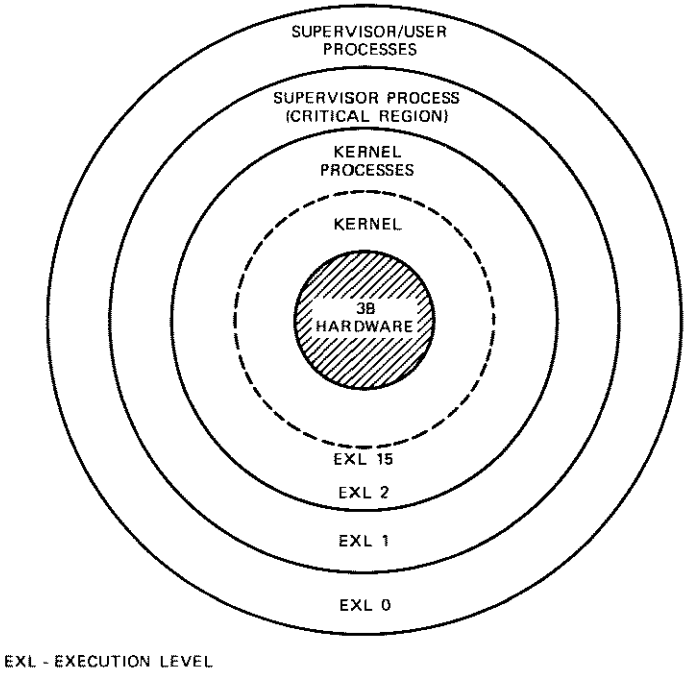



Fig. 4—Hierarchical organization of DMERT.

interrupt may be from a peripheral device or simply a PIR indicating the reception of an event or fault. Each execution level has a unique IM associated with it. The IM for each execution level inhibits all interrupts that are handled at the same or lower execution level. Since supervisor and user processes run at execution levels 0 and 1, they are always preempted by kernel processes.

### *3.6 Interprocess communication*

DMERT provides several mechanisms for interprocess communication. These include events, messages, faults, shared memory, and shared files.

#### *3.6.1 Events*

An event is a single bit of information having a predefined and agreed upon meaning between cooperating processes. When an event is sent from one process to another, DMERT will set the PIR interrupt for the level at which the receiving process executes. When that PIR is unmasked (the current execution level specifies an IM that has that PIR unmasked), DMERT will handle the PIR interrupt and dispatch the process for which the event was intended at its event entry. The type of event(s) sent is passed as a parameter.

#### 3.6.2 Messages

A message is a mechanism for transmitting multiple words of data between cooperating processes. The content of the message must be predefined and understood by sender and receiver alike. The reception of a message is indicated to the receiving process by a message event. Except for kernel processes, the contents of a message buffer (the data being passed) is copied from the sender's address space into the message buffer and then to the receiver's address space by various DMERT operations. All kernel processes include the system message buffer segments in their virtual address space. Hence, transmission of messages between kernel processes is much more efficient as the data is actually passed in shared memory.

#### 3.6.3 Faults

Faults are another mechanism for interprocess communication that are usually used to indicate an error, system initialization, or other emergency-type of communication. A fault consists of an 8-bit (byte) fault code that has a predefined meaning between sender and receiver. The reception of a fault causes a process to be dispatched at its fault entry.

#### 3.6.4 Shared memory

Processes can share memory on a segment basis. The entire segment (up to 32K words) would be mapped into the virtual address space of each process. This is the most efficient means of interprocess communication, but it may result in tight coupling between the sharing processes.

#### 3.6.5 Shared Files

Sharing a file is very similar to sharing memory except the storage is done on a secondary storage device (specifically a disk). This mechanism is obviously not as efficient as memory, but provides a media for sharing larger amounts of data.

### 3.7 DMERT system integrity

#### 3.7.1 Structure and strategy

The DMERT integrity package is based on the duplex, self-checking, nonmatching philosophy of the 3B20D Processor, and the hierarchical organization of the DMERT operating system. It was designed to provide tolerance to both hardware and software faults, so that the 3B20D Processor running under DMERT meets its reliability requirements. The functions provided by the integrity package running under DMERT appear in all abstract machine levels. Their placement in the hierarchical structure depends upon their complexity, the desired real-time response, and the services that they require.

The nondeferrable functions are activated when a hardware or software fault has been detected or a maintenance request has been made via a TTY message. They may initialize one or more system components, reconfigure the system, or just generate status reports. Some nondeferrable functions (e.g., processor initialization) cannot assume operating system sanity and require a fast response time. They are placed in the kernel and are initiated by the hardware self-checking circuits of the processor. Other functions like the recovery from a fault in an on-line peripheral can assume operating system sanity, but they must be performed in the minimum amount of time. Consequently, they are implemented as kernel processes.

Some functions such as diagnostics require services provided by lower abstract machines and their execution can be defferred. These execute under a DMERT supervisor process that provides a *UNIX* operating system environment. The deferrable integrity functions include the initiation and control of the diagnostics, administration of diagnostic requests, and requests to remove or restore a unit to service.

### *3.7.2 Software integrity*

***3.7.2.1 Overload and sanity.*** The focal point of the software integrity package of DMERT is the System Integrity Monitor (SIM). SIM is a kernel process that is responsible for, among other things, system overload control, sanity monitoring, and coordination of initialization actions. SIM coordinates its actions with an application process called the Application Integrity Monitor (AIM). The combined action of these two processes defines the overall software integrity strategy. Further discussion on overload and sanity is presented below as part of the TSPS No. 1B system integrity, which includes a description of the AIM process.

***3.7.2.2 Initialization levels.*** DMERT provides six levels of initialization (levels 0–5). The application can specify sublevels for DMERT levels 0–4. Level 0 is for application initialization only. That is, DMERT does not perform any initialization; DMERT merely notifies the application to initialize. Level 1 results in DMERT initializing and then notifying the application to initialize. In this case, all processing in the machine comes to a halt, the kernel and interrupt stacks are cleared, interrupt sources are disabled, and all kernel processes as well as the currently running supervisor process are faulted. Level 2 is a reboot of the system. At this level, however, certain protected segments of memory specified by the application are not lost, nor is the system Equipment Configuration Data (ECD). A Level-3 bootstrap reloads the ECD from disk, while a Level-4 bootstrap reinitializes all of memory including the protected application segments. Level 4 can only be requested manually. Level 5 is also manually initiated. It

reloads the system disk from tape. A manual bootstrap is then required to initialize the system.

## IV. THE SPC 1B

The SPC 1B Processor is the functional entity that replaces the SPC 1A in TSPS No. 1B. In reality, the SPC 1B is an SPC-like environment created on the 3B20D by several components. The Peripheral System Interface (PSI), emulation microcode, 3B20D hardware, native-mode software in the TSPS process, and the DMERT operating system all play a role in realizing the SPC 1B image. This section will describe the basic characteristics of the SPC 1B, briefly discuss its components, and concentrate on those capabilities required to emulate the SPC 1A at the instruction level. Other aspects of the machine and the cooperation of the components are developed in subsequent sections of this paper.

### *4.1 SPC 1B components*

The PSI provides the necessary interface between the 3B20D Central Control (CC) and the TSPS peripheral buses. Emulated programs communicate with the TSPS peripherals without hardware modifications to the peripherals themselves. The microprogrammed control and the flexibility of the 3B20D architecture make it feasible to emulate a machine of vastly different characteristics. Emulated instructions are implemented by a microprogram that co-exists with the microprogram for the native instruction set. An off-line object code post-processor complements the emulation microcode by creating instruction formats optimized for execution on the 3B20D hardware (see Section 4.5).

Finally, some functions must be emulated at the system level. Functions such as unlocking the write protection on an emulated store or requesting a processor switch are handled by native-mode software in the TSPS process. It provides services to emulated software not easily performed in the SPC 1A environment. In many cases, TSPS native-mode code interacts with other processes and the operating system to realize other services. Section 5.1.1.1 discusses TSPS native code in depth.

### *4.2 Basic characteristics*

The characteristics of the SPC 1B are quite different from those of the physical 3B20D. In the 3B20D Processor, all data paths, memory locations, and registers are 32 bits wide. Memory addressing is byte-oriented and 24 bits wide, providing a 16-million-byte capability. Invisible to both software and firmware is the memory management hardware, which provides virtual addressing. A high-speed buffer cache

is also included, which shortens the effective memory access time. The Arithmetic/Logic Unit (ALU) provides ones or twos complement arithmetic by allowing microprogram control of the carry in bit. A Rotate-Mask Unit (RMU) provides right rotates in any amount up to 31. The rotate amount also selects a mask from 16 mask classes to implement shifts and item extraction. In addition to the AND operation with a mask, the OR with the complement of a mask is possible to facilitate operations such as sign extension. There are 16 general-purpose registers in the 3B20D, although three are reserved for stack maintenance.

On the other hand, the SPC 1B, like its predecessor SPC 1A, is a 20-bit word-addressed machine. The arithmetic is done in ones complement and emulates a subtractor circuit. The significance of a subtractor is that the minus zero result is avoided in almost all cases. Seven general registers can be used for indexing, data manipulation, return addresses, and peripheral communication. A null (N) register can also be specified as a source of a zero operand. Rotates and shifts are allowed in both right and left directions. Contiguous bit masks of most sizes from 1 to 20 bits can be used for item manipulation. An insertion masking operation is also available to allow user-specified, non-contiguous bit masks.

#### *4.3 Mapping the SPC 1A image*

Since assembly-language programs are designed with an intimate knowledge of the machine they are written for, the 20-bit structure of the SPC 1A is embedded deep into the TSPS software. As a result, operations such as rotation and arithmetic represent a potential source of emulation errors. Because of this and the basic goal of emulating with minimal changes, the 20-bit architecture of the SPC 1A has been retained. Because of the difference in word size between the 3B20D and the SPC 1A and other differences described in the previous section, an image of the SPC 1B must be mapped onto the physical 3B20D hardware.

A 20-bit SPC word is contained right adjusted in a 32-bit 3B20D word. The most significant 12 bits are maintained as zeroes in both registers and memory locations. Twenty-bit word addresses are converted into 24-bit byte addresses by multiplying each address by four and forcing the uppermost two bits to be zeroes. The emulation of 20-bit addressing retains the limitation of 1 million word addressability as on the SPC 1A. Further, the one-million-word emulation address space must start at virtual address zero in the four-million-word TSPS process address space.

Registers zero through seven have been chosen as the SPC 1B registers. This assignment is identical to the numerical encoding on

the SPC 1A, thus simplifying the post-processing of SPC object code. Although the N register is assigned to be register zero, it requires special handling. When used as a source or argument operand, register zero must first be cleared. This guarantees a source of zero in the event that the N register was modified by specifying it as a destination in a previous instruction. The SPC 1B also contains an imaginary ones register, which may be used in register-to-memory and register-to-buffer bus instructions. The 'ones' register of the 3B20D is used for this purpose.

### *4.4 Instruction implementation*

The instruction set of the SPC 1B is very similar, but not identical to, that of the SPC 1A. Maintenance instructions related to the SPC 1A Processor were deleted for the SPC 1B. Other instructions that dealt closely with the SPC 1A hardware have required slight modifications. In spite of these changes, an SPC program or emulation mode programmer cannot and need not distinguish between the SPC 1A and 1B in most cases. Architectural differences described in previous sections are handled by microcode and are invisible to the programmer.

#### *4.4.1 Related 3B20D processor hardware*

To provide a framework for the description of instruction implementation, that portion of the 3B20D architecture directly affecting the emulation will be presented. A complete description of the 3B20D hardware can be found in Refs. 4 and 6.

As shown in Fig. 5, the CC is structured around a source and destination bus. The Data Manipulation Unit (DMU) accepts data from the source bus, and gates results to either an internal register or to an external register via the destination bus. The DMU contains the ALU, RMU, general registers, and parity circuits. Another circuit between the buses is the Find Low Zero (FLZ) unit. This circuit accepts 32 bits as input and yields the binary value of the bit position of the least significant zero. Finally, a direct path between the buses exists for fast data transfers between the external registers.

The store interface consists of the Store Address Register (SAR), Store Control Register (SCR), Store Data Register (SDR), and Store Instruction Register (SIR). These registers, along with a separate store operation field in each microinstruction, allow store operations to be done in parallel with 3BCC operations. To facilitate instruction fetching, hardware is dedicated to increment the Present Address (PA) register. A Program Status Word (PSW) bit specifies the PA increment amount to be 2 or 4, depending upon whether halfword or fullword mode is desired. The output of the incrementer is loaded into the SAR and back into the PA when a fetch is initiated.

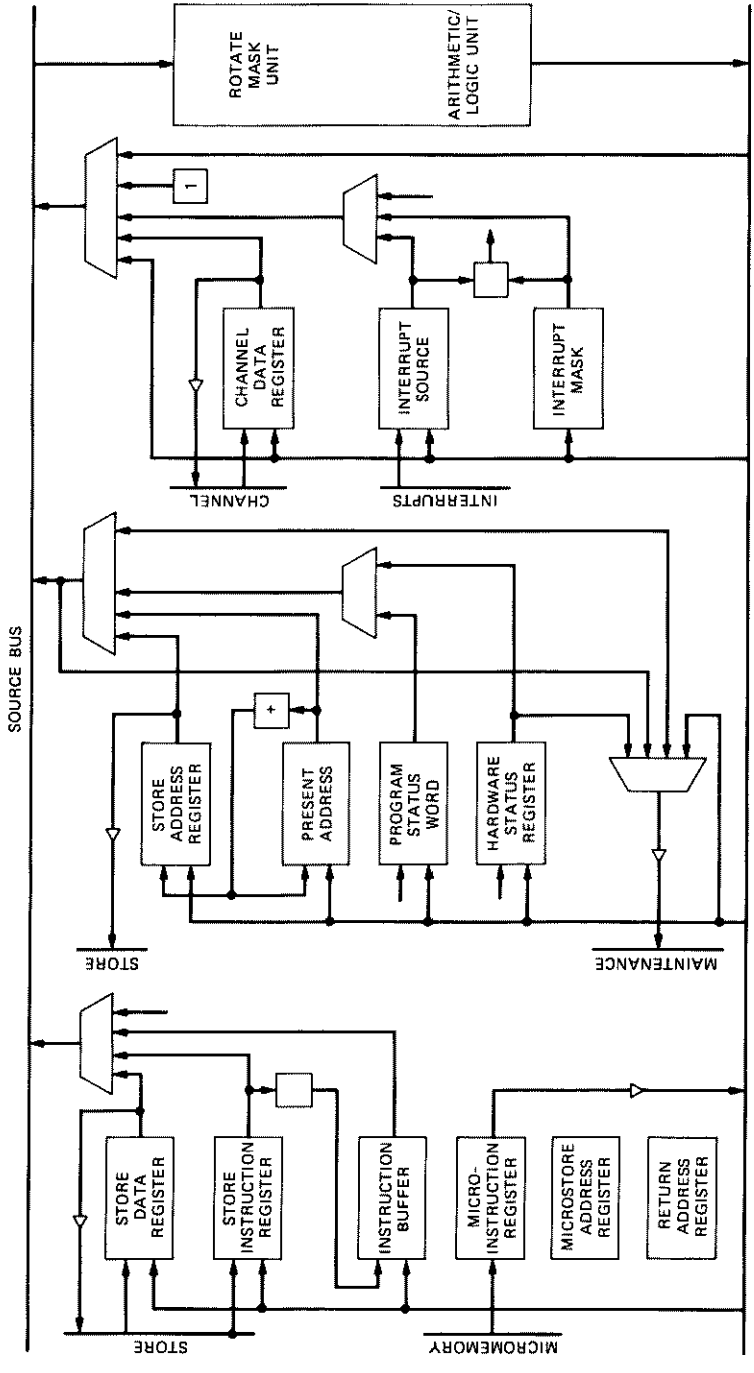



Fig. 5—Processor control architecture.

The basic I/O registers are the Channel Address Register (CAR), and the Channel Data Register (CDR). The Hardware Status Register (HSR) holds status and response information relating to channel operations. Channel operations are performed by microcode via the Pulse Point Register (PPR). Bits in the PPR are set and reset by microcode to form control pulses on the CCIO bus.

#### 4.4.2 Instruction fetching and decoding

An opcode on the 3B20D is eight bits wide. Multiple virtual machines are implemented by providing four complete sets of 256 opcodes. The instruction-initiation operation is a store-field function that operates as follows. Fetched instructions are loaded by the store into the SIR. When a new instruction is to be started, the contents of the SIR are transferred into the Instruction Buffer (IB). The opcode portion and two emulation-mode bits in the PSW are used to form a microstore address, which is the entry point into the microprogram for that instruction. Therefore, the mode bits effectively partition the microstore into four segments and, hence, four instruction sets.

#### 4.4.3 Instruction staging

As mentioned above, the instruction currently being executed is located in the IB. General registers and indices for the 16-way branch microinstruction can be indirectly specified by four-bit fields (nibbles) in the IB. For single-bit testing, the high-order bit of each nibble is available as a condition for the conditional jump microinstruction.

Similarly, there are three fields defined in the IB for rotate and mask operations. For a rotate and mask operation, a microinstruction must specify a mask class, a rotate amount, and masking operation. The rotate amount also selects which mask in the named class is to be used. The three five-bit fields in the IB can be used to specify the rotate amount and corresponding mask to be used.

### 4.5 Basic operation

The SPC 1B instruction set differs from other sets in that most basic operations can be specified as options on many instructions. Hence, the instruction set is small in number but rich in data-handling provisions. As an example, there is no explicit ADD instruction. Addition can be specified as an option on most instructions. This section will describe the implementation of the basic operations common to many instructions.

#### 4.5.1 Instruction formats

Instructions on the SPC 1A were encoded in a 40-bit double word with a four-bit basic opcode. Opcodes were extended by other bits in

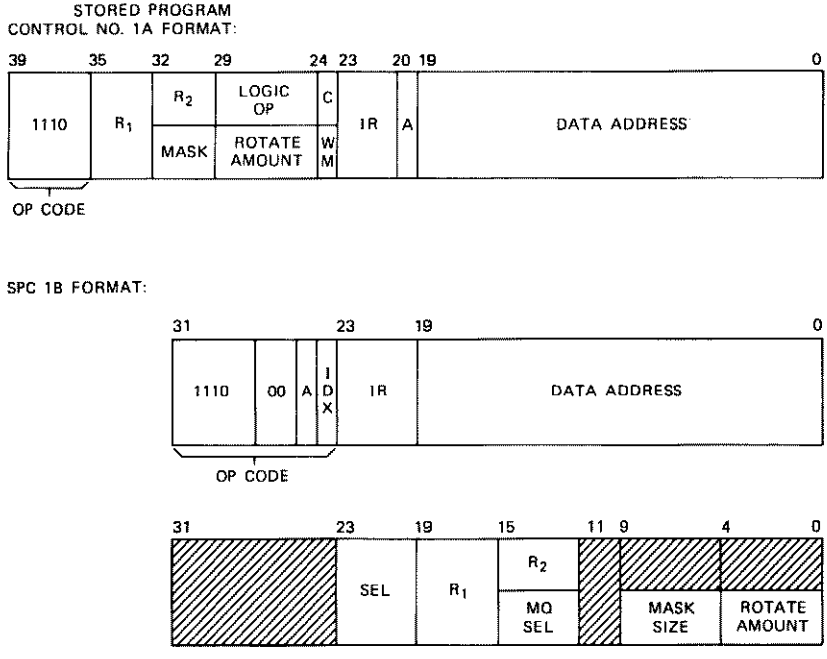



Fig. 6—Memory-to-register instruction.

some instructions, usually in a non-adjacent field. Register fields are three bits. Mask size was specified in a four-bit field that was not strictly binary encoded. The rotate amount is contained in a five-bit field. Options are encoded in multi-function fields within the instruction. In general, these fields were not aligned with the IB fields mentioned above for instruction staging. A bit-for-bit emulation would have resulted in a very inefficient result. Since one of the fundamental goals was an increase in real-time capacity, new formats were designed to provide the most efficient encodings. As an example, Fig. 6 shows the SPC 1A and SPC 1B formats for the Memory-to-Register (MR) instruction.

A post-processor is used to realign the 40-bit SPC 1A object code word into two 32-bit words suitable to execute on the 3B20D. In addition to simple bit shuffling, the post-processor adds information to aid the emulation microcode by performing those computations that can be done off-line. In the same fashion, bits have been included with the basic 4-bit opcode to take full advantage of 8-bit opcode encoding. For example, bits specifying an option can be included in the opcode to eliminate execution time required to decode that option. The unique entry point for each opcode would specify not only the instruction, but

also the occurrence of the option. The post-processor is described in more detail in Ref. 7.

#### 4.5.2 Execution protocol

As mentioned previously, SPC 1B instructions require two 32-bit words to hold the 40 bits of information contained in the SPC 1A instructions. Even in those cases where all relevant information can be held in one word, the spacing of two addresses between instructions must be maintained. As an example, indexed transfers into transfer tables, a common structure in TSPS software, would have had to be recoded had this spacing not been retained. Unlike the SPC 1A, which has a 40-bit memory bus, two separate fetches are required for each instruction. The placement of argument fields in the instruction words has to correspond to the logical execution flow for the instruction. SPC 1B formats are designed to match the flow of execution. Since instructions are a multiple of 32-bit fullwords, the mode bit in the PSW is set for a PA increment amount of 4. In contrast, the 3B20D native instruction set operates on 16-bit halfwords, with a PA increment amount of 2.

The basic fetch-execute protocol is for each instruction to fetch both words of the next instruction. When an instruction is started, the first word is in the IB and the second word is being fetched into the SIR. In this way, the processing of information in the first word can be effectively overlapped with the fetch of the second word. When the first word is no longer needed, the second word is gated from the SIR into either the IB or a scratch register as needed. The SIR is now free to accept the fetch of the first word of the next instruction. The last microinstruction, in addition to requesting that the next opcode be decoded, also starts the fetch of the second word of the next instruction.

#### 4.5.3 Arithmetic

As described previously, the SPC 1B represents negative numbers in ones complement form. A difficulty arises in the end-around carry for a 20-bit word on a 32-bit arithmetic unit. The algorithm used is to insert ones in the upper 12 bits of one of the operands to propagate the carry from bit 19 to the carry-out bit. In addition, a carry-in of one is initially assumed. If the carry-out is a one, then the carry-in assumption was correct and the operation is complete. If a carry was propagated from bit 19 to the carry-out, the upper 12 bits of the result are automatically cleared, as desired. Also, this assumption properly avoids the minus zero result that would have occurred if the carry-in was not made. This is the only case where the carry-in itself forces the carry-out. If the carry-out is a zero, the operation is repeated without the carry-in and with zeroes in the upper 12 bits of both operands. Since

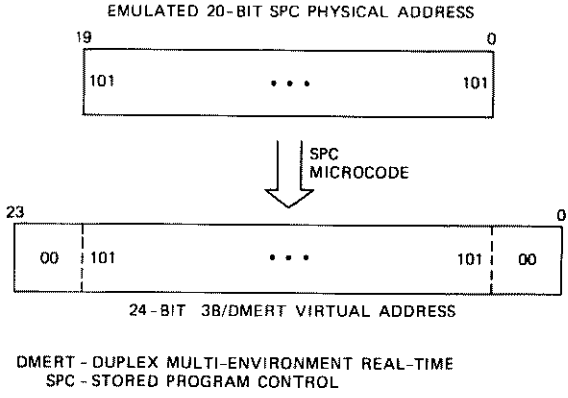



Fig. 7—SPC 1B address translation.

no carry-out of bit 19 will result, the upper 12 bits of the result will be zero.

#### 4.5.4 *Effective address generation*

For those instructions that access memory, indexing of the address is performed by adding the address and the contents of the specified index register. Any of the SPC 1B general registers, including the N register, can be used as an index register. Since negative indices are possible, the indexing operation is done in the same manner as ones complement arithmetic described above. The resulting 20-bit word address is then rotated left two bits (actually left 30 bits) and masked to clear bits 23 to 22 and 1 to 0 to form a 24-bit byte address, as shown in Fig. 7. Since indexing is costly in terms of execution time, the post-processor detects the specification of N as the index register, and sets an indexing bit to indicate that indexing is not required. The indexing bit is usually contained in the opcode to provide free decoding. The indexing bit, non-existent in the SPC 1A, is an example of information provided by the post-processor to facilitate efficient on-line execution.

#### 4.5.5 *Shifts and rotates*

In the SPC 1B, shifts and rotates are performed over 20-bit data words, even though these words physically reside in 32-bit 3B20D words. Since the 3B20D RMU only accepts right-rotate amounts, all left-direction amounts must be converted by taking their complements with respect to word size (i.e., word size minus left amount equals right amount). For shifts, the proper mask is applied to inject zeroes into vacated bit positions.

Implementation of the wrap-around from bits 0 and 19 for rotates is more difficult. Values less than 12 can be handled by copying the

upper or lower 12 bits of the 20-bit data word into the upper 12 bits of the 32-bit word. For left rotates, the upper 12 bits are used, while the lower 12 bits are copied for right rotates. Rotates greater than 12 can be converted to their equivalent value in the opposite direction before applying the above algorithm. For example, a left rotate of 15 can be effectively realized by a right rotate of 5. It should be noted that rotate and shift amounts specified in the instruction can be converted by the post-processor and encoded as an adjusted value. Rotates and shifts based on the contents of a register must be adjusted during run-time by the microcode.

#### 4.5.6 Data handling

The SPC 1B provides the capability of reading (unpacking) and writing (packing) contiguous data items of almost arbitrary size and position within a 20-bit word. A data item is defined by its size, M, and its position, Q. The unpacking operation involves reading a data word, rotating it right by Q, and applying a mask of size M. The packing operation involves applying a mask and then rotating it into position by the amount Q. This value is then inserted, using a read-modify-write sequence, into the target location.

Since the MQ option is used heavily in TSPS programs, implementation of the rotation in the manner described previously would be very costly. Luckily, in most cases much of this overhead is avoided. When the total of the rotate amount plus the mask size is less than or equal to 20, no wrap-around actually occurs. The more efficient shift operation can be used instead. In TSPS, this is true nearly every time an MQ is specified.

#### 4.5.7 Conditional transfers

The SPC 1B, like the SPC 1A, has no explicit condition flags. However, the results of data manipulations can be tested by conditional transfer instructions. Conditional transfers provide the capability of testing either the entire contents or a specific bit of any general register. The 3B20D condition codes, carry, overflow, zero, and negative, are used for these tests. It should be noted that like the SPC 1A, the minus zero value, coded as all ones, is not detected as zero in these conditionals.

A special conditional transfer instruction is the Detect Right Most One (DRMO) instruction. This instruction transfers if no ones are present in the test register. If a bit is set, the bit position in binary of the least significant one is loaded into a result register. The DRMO and its variation, the DZRMO, which zeroes the rightmost one, are used, for example, by task dispensers to initiate clients based on a job-activity word. The 3B20D FLZ is used to implement these instructions.

#### 4.5.8 Alternate entry points

There are two functions in the SPC 1B instruction set that are unique in that their operation extends across instruction boundaries. They are the Inhibit Interrupt (I) option and the Execute (EXC) instruction. The I option inhibits the J- and H-level interrupts until after the next instruction is completed. Although hardware exists in the 3B20D to unconditionally start the next instruction, blocking all interrupts if error interrupts are pending is unacceptable. In this case, the I option is implemented by raising the execution level to block J-level and H-level, starting the next instruction, and restoring the execution level during the instruction following the I-option instruction. The EXC instruction calls for the execution of an instruction at a target address, followed by an automatic return to the next sequential instruction after the EXC. The responsibility of the target instruction is to restore the PA to resume sequential execution.

Since any instruction can follow an I-option instruction or be the target of an EXC, every instruction must determine if a special operation has preceded it. Explicit tests by microcode at the start of each instruction would cause a prohibitive amount of overhead even when these operations are inactive. Instead, an alternate instruction set is used to force a different entry for the same opcode. The I option and EXC microcode forces the next instruction to be entered at its alternate entry by setting the appropriate emulation-mode bits in the PSW. The alternate entry typically determines which operation to unwind, restores the mode bits to their normal value, and transfers to the normal entry point to execute the instruction. Since the SPC 1B does not have condition codes, the PSW condition flags are available to specify whether the EXC or I option is in effect.

#### 4.5.9 New SPC 1B instructions

There are four new instructions in the SPC 1B instruction set. The Register to Buffer Bus (RBB), Buffer Bus to Register (BBR), and Ones to Buffer Bus (OBB) instructions have been added to reference the buffer bus system. These instructions, along with the buffer bus, are described in Section 5.1.4. The remaining new instruction is the Switch Mode and Transfer (SMT).

The SMT instruction allows emulated programs to transfer to the native mode and begin executing its instruction set. It is patterned after the native-mode CALL instruction to provide a calling sequence consistent with the C language. The SMT puts a stack frame, including two register arguments, on the stack, switches the mode bits in the PSW to specify native mode, and transfers to the address specified in the instruction. Two instructions in the native language are also used to support mode switching. They are the Call to Emulation (CALE),

and Return to Emulation (RETE) instructions. Except for the mode switch, these instructions are identical to the standard native CALL and RETN instructions.

The ability to transfer between native and emulated code allows C language to be incorporated into the TSPS process. In fact, the TSPS process has standard C-coded entry points, as described in Section 5.1.1.1. Mode switching eliminates the need for new instructions to perform operations not possible with the emulated instruction set. Examples are Operating System Traps (OSTs), 32-bit data manipulations, and stack accesses. Finally, designers of new features can evaluate on an individual basis which language would be most suitable. Trade-offs can be made between the necessary degree of coupling with existing emulated code and the ease of programming in C.

#### *4.6 Peripheral orders*

As mentioned in Section 4.1, the PSI provides the necessary timing and electrical interface to the TSPS periphery. The function of the microcode for peripheral instructions is simply to pass necessary information for the PSI to execute the order. At the end of the sequence, the PSI passes back an answer word and an indication of whether the periphery returned the correct reply signals. A failure indication is used to generate the emulated F-level interrupt, as discussed in Section 5.1.5.1.

The CC communicates with its channels over the Central Control Input/Output (CCIO) bus. An Application Channel Interface (ACHI) is provided with the 3B20D processor to allow the PSI limited access to the CCIO bus. In essence then, the PSI appears as a main channel to the 3B20D processor. Commands and data are sent and data received in a parallel fashion to/from the ACHI-PSI via the CDR CAR, and PPR. Status information, returned by the PSI into the HSR, is used to indicate a failed peripheral order.

The sequencer in the PSI performs four basic functions. In addition to the peripheral order, there are sequences to read a register and write a register. These functions are needed not only for buffer bus references, but also for diagnostic access. The remaining sequence is a pulse sequence, which is used to send special maintenance pulses to the periphery.

All SPC 1A peripheral orders are retained in the SPC 1B instruction set. In addition, a set of PSI instructions have been provided in the native instruction set to execute the four PSI operations. The native-mode PSI instructions were originally designed for the PSI diagnostic, discussed in Section 5.3, which is written in C and native assembly language. They are also used by the Application Integrity Monitor (AIM) process for PSI fault recovery and initialization (Section 5.2.2.1.).

One additional capability provided by the emulation microcode is a set of microcode routines to execute peripheral operations on the off-line processor. These routines accept data from the 3B20D Maintenance Channel (MCH), temporarily release the inhibit on the off-line PSI, and execute the order without making any main-store accesses. By not interfering with the main-store update mechanism, these routines can be executed while the off-line processor remains in the standby mode.

Off-line sequences for pulse and peripheral orders are used by TSPS peripheral fault recognition to retry failed orders and provide better fault resolution. An off-line PSI initialization routine is used by AIM just prior to a soft switch. It is also used periodically by AIM as a hardware audit of the off-line PSI. Should this routine indicate a failure in initializing the PSI, a diagnostic will be requested by AIM. The benefit of this audit is to reduce the latency time for detecting faults in the standby PSI.

### *4.7 Emulation-dependent software*

Although the majority of SPC 1A instructions have been emulated on the SPC 1B, there are functions that do not work in the same manner as on the SPC 1A. In general, the functions requiring modification are in the maintenance programs, which are by necessity more processor-dependent.

The most obvious source of change is the need to recode or eliminate those routines that contain instructions not carried over on the SPC 1B. The most common source of change is recoding buffer bus references to use the new instructions. Unmodified buffer bus references would simply reference a memory location in segment zero since the microcode does not trap the address as a buffer bus address in normal memory-access instructions. A third source of change is due to the new object-code formats and the difference in size between instruction and data words. Since the emulated code can only access 20 bits, instructions cannot be moved or created via data operations. Similarly, data cannot be executed as an instruction because the uppermost 12 bits of data words are zeros. On the SPC 1B, the zero opcode is interpreted as the ANOP instruction. This protective measure would trap a wild transfer into a data area by generating an A-level interrupt. Finally, usage of part of an instruction as data may not work because of the shifting of fields in the new object-code formats. An example of this in TSPS No. 1 code is to read the address field of a transfer instruction and store it to be used as an address later.

A more subtle difference is the execution time of instructions on the SPC 1B. The SPC 1A is a fixed-cycle machine, and instructions are made up of a number of these basic cycles. The execution time of an

instruction could be determined exactly, independent of the options specified. A common dependence on the execution time of an instruction is in timing loops. A precise time delay is created by executing a loop the correct number of times.

Since the 3B20D executes emulated instructions several times faster than the SPC 1A,[8] timing constants coded for the SPC 1A produce delays proportionately smaller on the SPC 1B. The modification of these constants is not quite as simple as multiplying by a speedup factor for the following reasons. First, the SPC 1B instruction set is microprogrammed, and thus every option must be accounted for when calculating the execution time for an instruction. Second, since the 3B20D employs virtual addressing, memory management delays owing to translation are incurred as a function of program flow. The cache memory also plays a role by shortening the memory access time when the access is contained in the cache. Since the cache is shared by all processes in the system, its effect is even less predictable. Finally, Direct Memory Access (DMA) activity steals memory cycles from the processor and represents an invisible form of interference.

As a result, timing loops with tight window tolerances cannot be guaranteed. In several cases, routines that performed window timing required recoding. On the other hand, minimum timing poses no problem. Best-case estimates can be made to determine the minimum execution time for a program segment by assuming no translation or DMA delays and all cache hits. Any delays actually incurred during execution serve only to lengthen the program segment time, which is acceptable for minimum timing.

## V. TSPS NO. 1B SOFTWARE

This section describes how the TSPS software was ported to run under the DMERT operating system. In addition, it describes the system integrity software and other processes required to complete the emulation.

### *5.1 The TSPS kernel process*

The emulated TSPS software has been incorporated into a single, large, nonkillable kernel process under DMERT. This is due largely to the real-time constraints of TSPS software operation and its existing structure on the SPC 1A. All TSPS code on the SPC 1A shares a single physical address space. All data and programs are equally accessible from any other program. The entire software structure is very tightly coupled. Subdividing the software into several processes would have required a restructure and redesign of the TSPS software. Also, the operational timing constraints of TSPS software (e.g., 5-ms I/O processing, interject responsiveness and base-level E-E times) require

TSPS to be memory-resident and interrupt-driven at the kernel-process level.

#### *5.1.1 Emulation environment*

Emulation of existing TSPS software on the 3B20D Processor required the creation of an SPC 1A environment. That is, in order to work properly the TSPS software had to be shielded from both the actual physical machine (3B20D) it was running on and the DMERT operating system it runs under. This environment has been established with a combination of hardware (PSI), firmware (the microcoded SPC 1A instruction set), and software (native-mode code within the TSPS process).

***5.1.1.1 Native-mode code.*** Except for the emulated code within the TSPS process, all software in the TSPS No. 1B system is coded in 3B20D native mode. Most of the software is coded with the C language. Hence, the 20-bit emulated code exists in a 32-bit universe.

Figure 8 illustrates the structure of the TSPS process. The TSPS process is dispatched (given control of the machine by DMERT for execution) at either its interrupt, event, or fault entry. These entry points are coded in 3B20D native mode. Thus, the TSPS process always begins execution in 3B20D native-mode code. These entry points determine what type of processing is to be performed and transfer, while simultaneously changing instruction sets, to appropriate routines within emulated software.

In addition to the native-mode entry points, the TSPS process also contains native-mode routines for communication with DMERT and other processes and for performing certain functions which the emulated code is incapable of doing. As a result, the native-mode code within the TSPS process completely isolates the emulated code from the 32-bit universe surrounding it. Hence, to the rest of the system the TSPS process appears to be a typical (albeit very large) native-mode kernel process.

***5.1.1.1.1 Interrupt entry.*** The only hardware interrupt source to which the TSPS process is attached is a clock-driven 5-ms interrupt for I/O processing. This single interrupt source is used for both J- and H-level processing. This same 5-ms clock pulse is transmitted to the PSI to synchronize its clock with the processor. The PSI clock transmits clock pulses to TSPS peripherals. When the interrupt source fires, the TSPS process is dispatched at its interrupt entry point. The interrupt entry is coded in 3B20D native assembly language because of its simplicity and need for efficiency: it runs every 5 ms. Its only functions are to determine whether this execution is a J- or an H-level, set the appropriate bit in the interrupt-level activity flag buffer bus word, and transfer to the appropriate emulated routine.

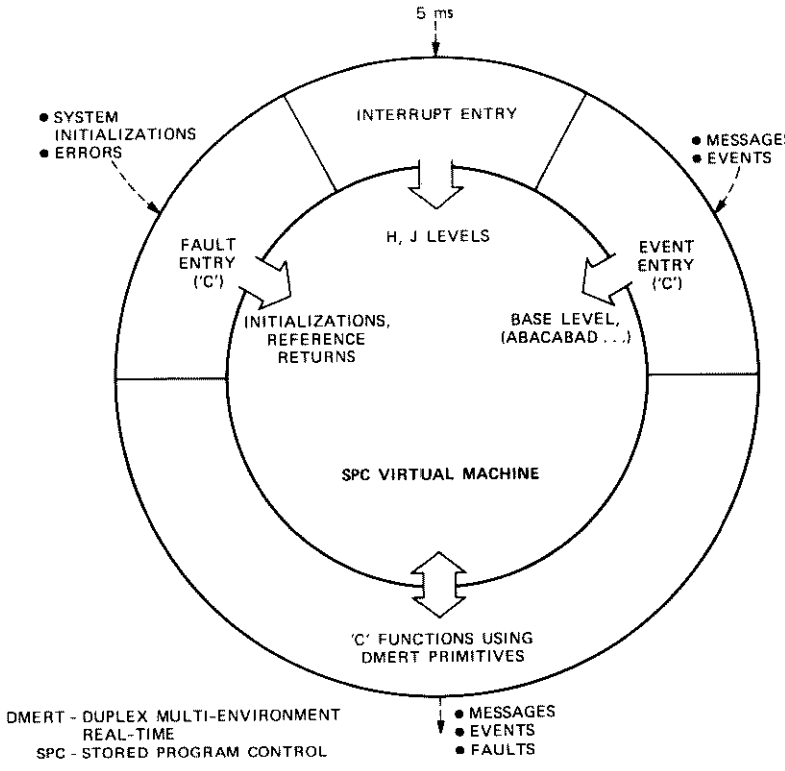



Fig. 8—TSPS Process.

Handling both J- and H-level interrupts with a single interrupt source is accomplished by using two execution levels for J-level processing. When TSPS is dispatched at the interrupt entry to handle a J-level interrupt, it runs at execution level 12. While at level 12 the interrupt source is masked off by the level-12 interrupt mask. Hence, the interrupt entry cannot be re-entered. When high-priority J-level work is completed, the H-level inhibit bit in the buffer bus is cleared. In doing this the microcode for the buffer bus write operation drops to execution level 11 for low-priority J-level work. This is possible since no other processes in the system run at execution levels 11 or 12. At level 11, the 5-ms interrupt source is unmasked and will allow the TSPS interrupt entry to be re-entered if J-level does not complete within 5 ms. The interrupt entry reads the buffer bus to determine whether a particular execution is a J- or H-level. If neither activity bit is set, then it is a J-level and the J-level activity bit is set. If the J-level activity bit is already set, then an H-level has occurred and the H-level bit is set. In the latter case low-priority J-level work has been interrupted and its state has been saved on the interrupt stack.

Although the interrupt handling begins in the native-mode interrupt entry, the return from interrupt is always performed from emulated code. At the end of H-level, the last job executes an emulated EGBN instruction. The EGBN clears the H-level activity flag in the buffer bus and performs a return from interrupt sequence, which causes the saved J-level state to be popped from the interrupt stack. J-level then resumes execution at the interrupted point. At the completion of low-priority J-level an emulated GBNHJ instruction is executed. The GBNHJ clears the J-level activity flag, sets the H-level inhibit bit, and does a return from interrupt. Hence, the process that was executing when J-level began is resumed at the point of interrupt. This interrupted process itself may be the TSPS process performing base-level work.

***5.1.1.1.2 Event entry.*** The TSPS process event entry is coded with the C language and runs at execution level 5. An event sent to the TSPS process causes the process to be dispatched and begin execution at its event entry. The primary events sent to the TSPS process are for initialization upon creation, interprocess message reception, and time-outs requested for real-time breaks in TSPS-base level processing.

***5.1.1.1.2.1. Initialization event.*** An initialization event is sent to a process upon its creation. As TSPS is a nonkillable kernel process, this will occur only after a system bootstrap. The native-mode code initializes the process (i.e., attaches to the 5-ms interrupt source, connects to a system port for message reception, etc.) and prepares to take an SIA or SIB call-processing recovery phase.

An important function that is performed during the initialization is setting the proper page protection over the emulated-code address space. DMERT creates processes with protection set on segment boundaries. The TSPS process, however, emulated the SPC 1A protection mechanism. Hence, some segments may have both protected and unprotected areas. The protection within these segments must be modified by changing the protection on the appropriate pages. TSPS uses a DMERT OST for this run-time page-protection change. This same OST is used by the emulated recent change programs when applying a modification to protected memory.

***5.1.1.1.2.2 Time-out event.*** The TSPS process is only one of many processes time-sharing the SPC 1B. The TSPS process must voluntarily take frequent real-time breaks to allow other processes at execution level 5 and below to execute. (Base-level processing is done at execution level 5, which is the lowest execution level at which the TSPS process runs. This is described in more detail below.) These real-time breaks are performed by issuing a time-out request to the DMERT timer* and then relinquishing control of the machine. (The DMERT timer is

part of the kernel that executes at execution level 15. It runs every 10 ms as a result of clock-driven interrupt source. The Timer maintains the DMERT system clock and provides general-purpose timing for all DMERT processes.) At the end of the time-out period the timer will send TSPS a time-out event. When TSPS takes the break, it remembers where in emulated code it has left off. Upon receipt of the time-out event it will resume execution within emulated code at the same place where it had left off. A related event is one sent by the scheduler when the system is idle to wake TSPS up early, before the time-out event has fired. This is described further in Section. 5.1.6.

Currently there are five areas where the TSPS process takes real-time breaks. These are during memory zeroing in an SIA or SIB, during initialization timing loops that are longer than 10 ms, during audit stitching, at the end of an SIA or SIB while waiting for the J-level portion of the phase to complete, and during the base-level E-E cycle.

***5.1.1.1.2.3 Message event.*** The TSPS process receives interprocess messages from other processes. TSPS is notified of the message reception with a message event. When TSPS fields the event, it will process all queued messages and take whatever actions are appropriate. Some uses of interprocess messages by the TSPS process are discussed in the later section describing other TSPS application processes.

***5.1.1.1.3 Fault entry.*** The TSPS process fault entry is also coded in the C language. The TSPS process is dispatched at its fault entry as a result of a processing error (i.e., a protection violation or an invalid address), a system initialization, or a fault sent by another process as a means of interprocess communication. The TSPS fault entry runs at execution level 12 so that recovery actions can be taken without interference from J-level processing.

***5.1.1.1.4 Additional routines.*** Besides the standard process entry points there are other special-purpose native-mode routines included within the TSPS process. These routines are used to interface with DMERT (e.g., to execute an OST), to use interprocess communication mechanisms such as events and messages in order to communicate with other processes, and to implement some functions that cannot be performed in emulated code. In addition, some new code added to the TSPS process was coded using the C language for development convenience.

### *5.1.2 Address space*

TSPS is currently designed as a DMERT "small" process. A small process can have up to 64 segments, and, hence, a virtual address space as large as two million words. Figure 9 shows the TSPS process virtual address space as a small process under DMERT.

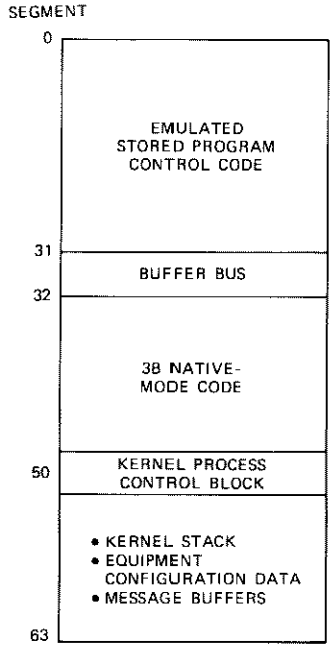



Fig. 9—Virtual address space in the TSPS process.

***5.1.2.1 Emulated code.*** The emulated code resides in the first 32 (0–31) segments of the virtual address space. Thus, the emulated code resides in the first one million words of the virtual address space. This, in fact, is the only part of the virtual address space that is directly addressable by the emulated code. As mentioned previously, all addresses in the emulated code are 20-bit word addresses as in the SPC 1A. A 20-bit address can only address one million words. Hence, the 20-bit physical word addresses on the SPC 1A are now 20-bit virtual addresses in the TSPS process on the SPC 1B.

All 32 segments are fully allocated to their maximum size of 32K words. In effect each segment emulates an SPC 1A store name code. The emulated address space, however, includes segments 0 and 31, whereas store name codes 0 and 31 do not exist on the SPC 1A. Emulated buffer bus references are mapped by the microcode into segment 32. As a result, the emulated code within the TSPS process has a full 1M word address space.

***5.1.2.2 Native-mode code.*** The remaining 31 segments are used for native-mode code within the TSPS process. Not all of these segments are currently used. Those that are used contain TSPS code and data, the process control block, and shared DMERT segments such as the ECD, the system message buffers, and the kernel stack.

### 5.1.3 Native-mode/emulated-code interface

The TSPS process contains both emulated SPC 1A and native-mode code. As all processing begins in native code, all emulation-code routines are called as functions (subroutines) from the process entry points. Similarly, native-mode routines can be called as subroutines from the emulated code.

All processes under DMERT use a program stack to save the state of a function when it calls another and to pass arguments. Kernel processes (including TSPS) share a common kernel stack with the kernel. A standard stack frame must be maintained when calling another function. Similarly, a standard return sequence is followed. To maintain sanity while transferring between emulated and native code, the TSPS process follows the same protocol. The mechanism used by the TSPS process was patterned after the structure used to transfer between native-mode assembly code and C-language code.

### 5.1.4 Buffer bus

As described in Section 2.2, the buffer bus on the SPC 1A is a software-accessible set of hardware control and status registers. Since the microcode emulates the hardware action on the SPC 1B, accessing one of these registers causes the microcode to perform a special action, emulating what the hardware would have done. Hence, firmware must be aware that a buffer bus location is being accessed, and what action is required. To eliminate the overhead required to test each memory address in memory reference instructions, new instructions have been added to explicitly reference the buffer bus.

The Register to Buffer Bus (RBB), Buffer Bus to Register (BBR), and Ones to Buffer Bus (OBB) instructions are the exact images of their SPC 1A counterparts, RM, MR, and OM. Converting an SPC 1A buffer bus reference requires only a change to the instruction mnemonic. The address and option fields are identical. The addition of these instructions has the added benefit of not having to set aside store zero as buffer bus images, thus allowing an additional 32K words of usable memory. Depending on their function, buffer bus registers are mapped into either memory locations outside of the SPC 1B address space, PSI internal registers, or 3B20D hardware registers.

Not all registers have been carried over from the SPC 1A. Locations used only in processor and store maintenance were removed along with their corresponding programs. Other locations have been added to maintain and access the PSI. Still others are partially retained, with individual bits having been removed. The set can be broken down into three components: interrupt system, craft interface, and peripheral system.

Locations related to the interrupt system are the Interrupt-Level

Activity Flags (ILAF), Maintenance Interrupt Sources (MAIS), Interrupt Inhibits (PEST), 3B20D Inhibits (3BPEST), and the Millisecond Clock State (MSEC). With the exception of the MSEC register, these are discussed in Section 5.1.5. MSEC on the SPC 1B is the value of the 3B20D Interrupt Timer. The Interrupt Timer is used as the 5-ms clock source for the TSPS J-level interrupt. Access to this timer allows programs to predict when the next J-level will occur. This function is necessary for clients such as diagnostics, which must synchronize their actions with J-level.

Buffer bus locations related to the craft interface are Maintenance Control Center Data (MCCD), MCC Interrupts (MCCI), and Emergency Recovery Display (ERD). The hardware of the SPC 1A MCC has been replaced by the 3B20D craft interface hardware, and DMERT and TSPS craft software. These locations reside in memory locations accessible by TSPS native-mode craft software. The craft interface is discussed in more detail in Section 5.3.3.1.

The remaining locations are all related to the peripheral system. The PSI hardware directly implements many of these locations. Locations emulated by the PSI hardware require that microcode send the necessary command to the PSI to access data or perform some hardware action.

#### *5.1.5 Interrupt structure*

***5.1.5.1 Emulated interrupts.*** The emulated interrupt structure consists of A-, E-, F-, H-, and J-level interrupts. Of these, only the 5-ms H- and J-level interrupts are driven by a 3B20D hardware-interrupt source. A-level interrupts are either emulated by native-mode software or generated by the microcode as the result of executing an illegal opcode or an ANOP instruction. E-levels are emulated by native code as the result of fault codes received from DMERT, and F-levels are generated either by microcode or native-mode code.

The microcode for the ANOP instruction traps to an illegal instruction routine in the native microprogram that generates an error interrupt. The DMERT kernel handles the error interrupt, and immediately faults the TSPS process. The TSPS process fault entry determines that an ANOP instruction was the cause of the fault, sets the A-level activity flag in the buffer bus and transfers control to the emulated A-level interrupt-handling routine.

An A-level interrupt can be simulated by native software through setting the A-level activity bit in the buffer bus, which raises the execution level to 12 (the level at which A-level runs), filling in the interrupt-state bin locations, and transferring to the A-level interrupt routine. This is done, for example, to indicate a manual request for a recovery phase. The TSPS process fault handler will emulate the SPC

1A hardware by "generating" an A-level interrupt in the aforementioned way. When the A-level routine is entered, it appears to the software that a hardware interrupt had occurred.

SPC 1A store-error E-level interrupts are not emulated, as this type of error is not seen by the TSPS process. They are handled completely by DMERT fault recovery. Software E-levels (i.e., protection violations and bad addresses) are fielded by TSPS as faults. When this type of error occurs, the TSPS process is immediately interrupted and entered at its fault entry with the state of the machine at the time of the error passed as parameters. The fault entry sets the E-level activity bit in the buffer bus and transfers to emulated code to handle an E-level interrupt. Again, to the emulated code it appears just as if a hardware interrupt had occurred on the SPC 1A.

During the execution of an emulated I/O instruction or buffer-bus instruction,* any failures will be detected by the microcode. The microcode 'generates' an F-level interrupt by setting the F-level activity bit in the buffer bus, saves the address of the interrupted program in the F-level bin, and transfers directly to the emulated F-level routine.

Most PSI errors are detected by the microcode also. An F-level is also generated in these cases. This is done so that F-level can perform an on-line and, if necessary, an off-line retry before letting AIM take PSI recovery actions. In the few cases where a PSI error interrupts AIM first, AIM will fault the TSPS process. The fault entry will then emulate an F-level in the same way as E-levels are handled. There is another class of errors that generate 3B20D error interrupts. The DMERT Error Interrupt Handler (EIH) will field these and fault the running process. In these cases TSPS passes control (via a fault) to AIM for recovery actions.

The microcode and native code must of course be aware of interrupt priorities. If TSPS is already in A- or E-level processing, then the microcode must only set the F-level interrupt source bit and continue processing. An F-level will be generated by the microcode if the source is still set when an EGBN is performed from A- or E-level. Also, if the F-level inhibit is set in the buffer bus the error must be ignored. Setting the J-level inhibit while in base level causes the microcode to raise the execution level of the process to 12. The interrupt mask for execution level 12 masks off the 5-ms interrupt source. Similarly, the "I" option is implemented via the microcode. The microcode guarantees that both J- and H-level interrupts are inhibited for the instruction with the I option and the next instruction executed.

---

* As some of the buffer-bus registers are contained in the PSI, a failure while trying to access these registers is considered a PSI error.

*5.1.5.2 Returning from interrupts.* The emulated EGBN instruction works the same as it does on the SPC 1A for A-, E-, and F-level interrupts. Because H- and J-level interrupts are driven by a hardware-interrupt source, the EGBN for H-level and the GBNHJ instruction used by J-level operate differently. This difference (described below) is undetectable by the programmer.

When a 5-ms interrupt occurs, the state of the interrupted process is saved on the 3B20D interrupt stack. The return from interrupt (EGBN for H-level or GBNHJ for J-level) must restore the state from the interrupt stack and not from memory-resident bin locations. Hence, these two instructions perform the same function as on the SPC 1A, but the actions taken to restore the interrupted state of the machine are different.

### *5.1.6 Emulated program structure*

The emulated program structure within the TSPS process is almost identical to that which exists on the SPC 1A; the major difference is that processing always begins in native code. The native code sets up the proper environment and passes control to the emulated software for the bulk of the processing.

H- and J-levels still run under control of ECIO. As pointed out earlier, the only difference is that a single interrupt source is used on the SPC 1B. The front-end native code determines whether an H- or J-level has occurred by the state of the buffer bus. Different execution levels are used to unmask the H-level interrupt source when going from high- to low-priority J-level.

Base-level programs still run under control of ECMP. Base level is essentially called as a large subroutine of the event handler. Rather than run continuously, as on the SPC 1A, base level must voluntarily give up control of the machine to allow lower-priority processes (processes that run at execution levels below 5) to run. Base level requests a time-out event from DMERT when at least 10 ms have elapsed, and then exits.

There are two ways for base level to be re-entered in response to its time-out request. If during the real-time break there are no processes ready to run, the DMERT scheduler will enter its idle loop at the lowest system priority. At that point, the scheduler finds that the TSPS process has requested, via an OST during its initialization event entry, that a specific event be sent by the scheduler whenever the idle loop is entered. The event sent by the scheduler causes TSPS to be entered at its event entry, thus waking base level up early. If the idle loop is never entered during the break, base level must wait until the full duration of the time-out request has expired. In this case, receiving

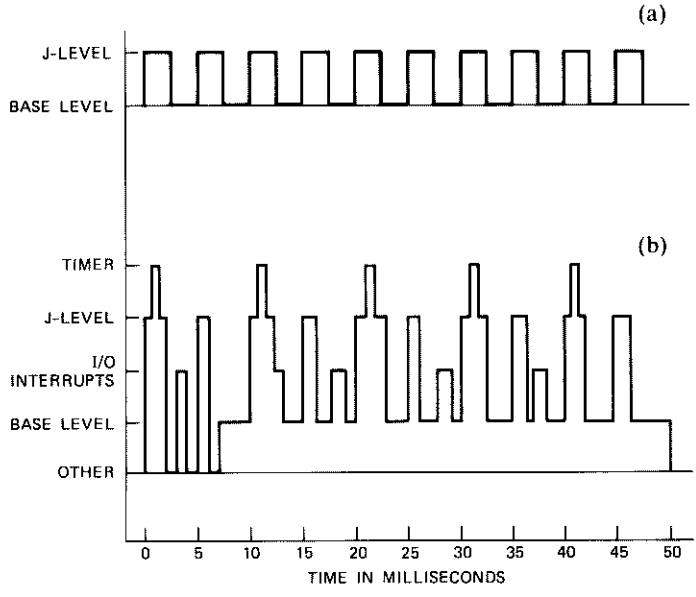



Fig. 10—Interaction between processing levels of (a) SPC 1A and (b) SPC 1B.

the time-out event from the DMERT timer causes base level to resume execution.

This early wakeup mechanism employed by TSPS has a significant impact on the real-time profile of the system. By eliminating idle system real time, base level executes more E-E cycles. This is especially true when the offered call load is light and real time is ample. A higher E-E rate improves the execution of TSPS diagnostics, audits, and other services that are run as a function of E-E rate. The full impact of the early wakeup mechanism is described in Ref. 8.

Base-level code in ECMP was modified to take its real-time breaks between priority classes. There are five breaks per E-E cycle: one after each C and E priority class. Figure 10 contrasts the interaction between various processing levels on the SPC 1A versus the SPC 1B. On the SPC 1A, base level runs continuously, only being interrupted (in the absence of errors or manual actions from the MCC) every 5 ms by J-level. On the SPC 1B, however, there are other processes executing concurrently with TSPS. DMERT I/O processes run at execution levels higher than base level, but lower than J-level. J-level executes every 5 ms whether or not base level is executing or taking a real-time break. Table II shows the execution levels of various DMERT processes, the TSPS process, and AIM.

The real-time breaks are not noticeable to the emulated software. They merely appear as an additional priority class placed in between

Table II—DMERT execution levels

| Level | Processes |
|---|---|
| 15 | Error Interrupt Handler, Timer |
| 14 | Generic utilities |
| 13 | System and Application Integrity Monitors |
| 12 | TSPS (Fault entry, A, E, F, H and high-priority J) |
| 11 | TSPS (low-priority J) |
| 10 | Disk and IOP drivers |
| 9 | — |
| 8 | — |
| 7 | File Manager |
| 6 | — |
| 5 | TSPS (Base level) |
| 4 | — |
| 3 | Plant measurements, Field update |
| 2 | Special processes (Scheduler and Memory Manager) |
| 1 | Supervisor processes (critical region) |
| 0 | Supervisor/User processes |

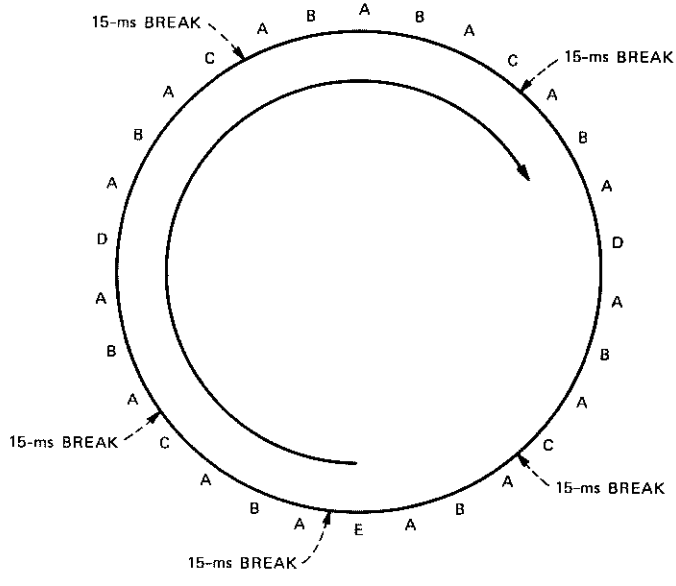



Fig. 11—TSPS No. 1B base-level loop.

the two priority classes separated by the break. Figure 11 shows the modified base-level loop with the five real-time breaks. The insertion of these breaks, though, has required one minor change to interject processing. To guarantee proper interject responsiveness additional interject checks have been inserted. A check for interject work is made after each real-time break. Hence, interject is serviced within 20 ms as a result of a base-level real-time break. Once the interject check is made, base-level processing resumes with the next scheduled priority

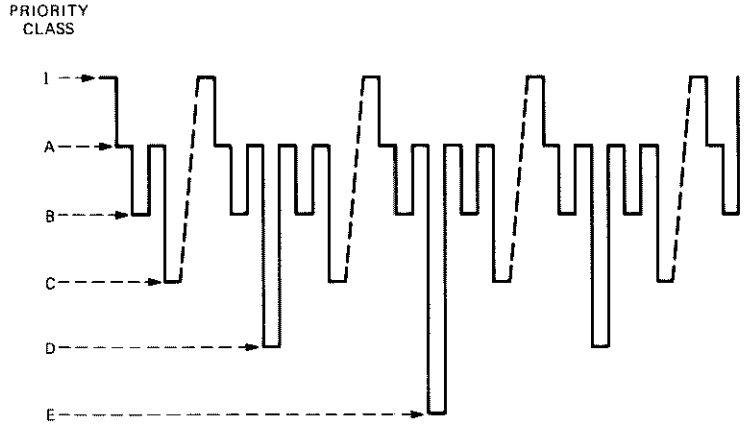



Fig. 12—TSPS No. 1B base-level, priority-class frequency.

class. Figure 12 shows the base-level priority-class execution with the added interject checks.

Another important point is the relationship between the DMERT timer and J-level execution. As the timer runs at execution level 15, it will interrupt J-level. The 10-ms timer interrupt, however, is offset from the 5-ms interrupt by 1 ms. Hence, the timer runs every 10 ms, 1 ms after J-level begins. This guarantees proper operation of the Automatic Message Accounting (AMA) data recording. The AMA data transfer routine must write data to be recorded to the AMA within 1 ms after the 5-ms interrupt occurs. (Recall that the AMA tape unit is synchronized with the 5-ms clock pulse.) Since the timer will not run until this 1-ms window has passed, it does not interfere.

### *5.2 System integrity*

The TSPS No. 1B integrity software consists of three distinct software packages. In addition to the software package provided with DMERT and the emulated TSPS No. 1 integrity package, a new integrity software package was developed. The major functions of this software are:

(*i*) Interface and coordination of integrity-related activities between DMERT and the TSPS application (e.g., initialization, overload, and processor switch)

(*ii*) Sanity and integrity of application processes

(*iii*) PSI maintenance.

These functions have been implemented in the Application Integrity Monitor (AIM), the native-mode portion of the TSPS process, and the PSI diagnostic process.

#### 5.2.1 Hardware integrity

The TSPS No. 1B hardware architecture and, in particular, the design of the PSI, made it possible to retain most of the peripherals used in TSPS No. 1. As a result, the maintenance strategy for these peripherals remained virtually unchanged and the maintenance software implementing it was emulated with only minor changes. The entire maintenance software for the TSPS peripherals resides in the kernel TSPS process and represents a distinct maintenance package.

Unlike the SPC 1A, the SPC 1B does not provide matching between the on-line and off-line processors. Instead, both the 3B20D and the PSI employ extensive self-checking hardware circuits to detect most service-affecting faults. The basic switchable entity is the simplex SPC 1B. When a fault is detected in the on-line processor, a switch to the off-line processor is performed. This switch may be followed by an initialization sequence. The faulty circuit pack is then identified with the aid of a diagnostic. The SPC 1B hardware is designed such that no single fault in either processor can cause a system outage. The TSPS No. 1B hardware architecture, including the PSI, and its maintenance, is described in Ref. 6. TSPS No. 1B reliability is covered in Ref. 8.

#### 5.2.2 Software integrity

*5.2.2.1 Application integrity monitor.* The Application Integrity Monitor (AIM) process is the sole application interface to DMERT for system integrity issues—initialization, sanity, processor switches, and DMERT overload conditions. Primary responsibility for monitoring the sanity of the TSPS application and controlling its recovery resides with AIM. In addition, AIM performs PSI fault recovery and initialization. Reports of DMERT recovery actions and resource overload are made to AIM, which in turn controls the application response to the event.

AIM is a nonkillable kernel process running at the same execution level as the DMERT System Integrity Monitor (SIM), higher than all other application processes. It interfaces with SIM via interprocess messages. AIM executes briefly every 500 ms at execution level 13 or as a result of PSI error interrupt or faults sent from the TSPS process. Since AIM is at a higher execution level than the TSPS process, it runs immediately when faulted by TSPS. This is done, for example, when TSPS requests a phase. TSPS faults AIM with its request, and AIM requests the phase of DMERT and reports back to TSPS about when to start its initialization.

*5.2.2.2 Initialization.* The TSPS call-processing recovery phases were integrated into the DMERT initialization levels to provide a coherent strategy. In the resulting initialization mechanism, between two and four TSPS initialization levels are associated with each DMERT

initialization level. Table III shows the mapping of TSPS recovery phases to DMERT initialization levels. TSPS minor and major phases are not taken after a bootstrap (DMERT initialization levels 2 through 4). The bootstrap recreates the TSPS process in memory, zeroing all its unprotected data areas. Hence, a minor or major phase will not initialize the TSPS process; an SIA or an SIB is required.

The TSPS No. 1B initialization mechanism employs a sequential escalation for most transitions. In addition, TSPS executes four SIB phases, with each using a different peripheral configuration, before escalating to the next higher DMERT level. However, in some cases, such as manual initialization or an interrupted initialization, the sequential escalation may be bypassed. Control of the escalation sequence is distributed between SIM and AIM. If the highest automatic initialization phase [DMERT level 3 and TSPS level 4 (SIB)] is reached and the system still does not recover, the system will loop continuously, rebooting the system and taking SIBs with varying peripheral configurations until the system recovers or manual action is taken.

A few minor changes were made to the initialization code to allow the TSPS process to take real-time breaks during recovery phases. Since ECMP is not cycling through priority classes during phases, it was necessary to add other breaks during the phases. Continuous running of TSPS during a several-minute-long phase would cause several DMERT sanity and overload checks to fail as lower-priority processes would not be run. Thus, during a phase the TSPS process will break during memory zeroing, for timing loops greater than 10 ms in duration, during audit stitching, and while base level waits for J-level to signal the end of initialization. The length of the phase is not affected by these real-time breaks. The length of the phase is determined by the amount of peripheral equipment, and hence the number of 5-ms interrupts (J-levels) that must occur for peripheral equipment initialization. Although each interrupt will execute faster, the total

Table III—DMERT level (EAI command)

| | 0 (Application Only) (50) | 1 (51) | 2, 3, 4 (Bootstrap) (52, 53, 54) |
|---|---|---|---|
| Application parameter | | | |
| 0 | No action | Reference return | Boot without TSPS process |
| 1 | MNA | MNA | Boot + SIA |
| 2 | MJA | MJA | Boot + SIA |
| 3 | SIA | SIA | Boot + SIA |
| 4 | SIB | SIB | Boot + SIB |
| L | Limp mode | Limp mode | Boot + limp mode |
| Other values | No action | Reference return | Boot + SIA |
| No value (null) | No action | Reference return | Boot + SIA |

elapsed time depends on the number of interrupts and not the speed at which they execute.

*5.2.2.3 Reference returns.* Taking reference returns in the TSPS process is more difficult than on the SPC 1A. All function calls, whether they be between native-code routines or between native code and emulated code (or vice versa), maintain a history of the function call on the process stack. (Transfers between emulated routines do not affect the stack. These transfers work just as on the SPC 1A. In fact, emulated code cannot explicitly access the stack and is unaware of its existence. Only the microcode for the SMT instruction manipulates the stack to maintain a proper stack frame when calling a native routine.) A function must make a normal return to the calling routine to properly unwind the stack. Direct transfers between functions would quickly result in an incongruous stack from which the process would have no way of properly returning to the calling routines. Hence, when taking a reference return, the TSPS process must properly unwind the stack.

To further explain reference returns within the TSPS process, consider a fault generated as the result of an invalid address. The TSPS process will be interrupted immediately, entered at its fault entry, and passed the state of the machine at the time the fault occurred. The fault entry then "generates" an E-level interrupt. The state information indicates the instruction in error and the contents of all the general-purpose registers at the time of the error. Because the TSPS process contains both native and emulated code, this error could have occurred in either. In addition the error could have occurred at a point many levels deep in nested function calls. If a direct transfer to a known "safe" reference point is performed, then a later return to the original calling function (i.e., the event or interrupt entry) will fail as the program stack will not have been unwound properly. To accommodate the existence of the stack, slight modifications to the handling of reference returns had to be made.

If the E-level occurs in H- or J-level, an immediate return from interrupt (using an EGBN or GBNHJ) will be executed. The return from interrupt will clear all of the interrupt's function call entries from the stack and allow the interrupted process to resume processing. This has the effect of canceling not only the current J-level job in progress (as on the SPC 1A), but also all jobs scheduled to execute that 5-ms entry.

If the E-level occurs in base level, a similar situation exists with respect to the kernel stack. In this case, however, a simple return from interrupt will not suffice. Base level must stimulate itself to be re-entered at some later point in time. (J-level, of course, is re-entered by the next 5-ms interrupt.) This is normally done with a time-out request.

This is not sufficient here, either, as base level must know that its next entry is for the purpose of taking a reference return rather than continuing its base-level loop. Hence, base level sends a fault to itself and then performs the return from interrupt. As before, the return from interrupt will clear the stack of any function call entries made by base-level processing. When the fault entry is entered, the fault code will indicate that a base-level reference return is to be made. The fault entry drops to execution level 5 and transfers to the base-level reference point.

The base-level reference-return handling is functionally equivalent to that on the SPC 1A. The only difference is that a momentary delay is experienced while base level relinquishes control of the machine to pop its function call entries from the stack and is then re-entered at its fault entry before taking the reference return.

***5.2.2.4 Sanity.*** The integrity of the system is preserved through a sanity detection mechanism implemented as a hierarchy of sanity timers, integrity checks implemented in native and emulated software, and a recovery mechanism implemented as a hierarchy of initializations.

Each application process has a built-in set of checks designed to detect any abnormal behavior that may lead to insanity. In particular, the TSPS process has preserved, through emulation, the hierarchical structure of sanity checks between base-level, low-priority J-level, and high-priority J-level. Also retained are the critical data structure checks and a monitor of the frequency of maintenance interrupts. The only sanity check that could not be emulated was the hardware long timer, previously implemented in the SPC 1A to check high-priority J-level sanity. This has been replaced by a software sanity timer implemented by AIM.

At the next higher level, the AIM process is responsible for monitoring the sanity of the TSPS process. It does so through the software sanity timer implemented as a counter shared between the two processes. This counter is incremented by high-priority J-level and decremented and then checked by AIM every 500 ms. The expected value of the counter is 0. Repeated non-zero values indicate an insane condition.

The sanity of AIM is in turn monitored by the DMERT System Integrity Monitor (SIM), a process responsible for the sanity of the entire software system. SIM monitors AIM, and implicitly the entire application, by an application sanity timer. AIM activates the application sanity timer after successful process creation in a system bootstrap. AIM must continue to indicate normal operation by sending a periodic sanity event to SIM to reset the timer.

The highest sanity check in the hierarchy is the 3B20D hardware

sanity timer. SIM periodically resets the sanity timer. Hence, the sanity of SIM is implied by the failure of the timer to fire. Should the timer fire, a stop-and-switch operation will be performed to force the off-line processor into the active role.

***5.2.2.5 Overload control.*** The main objectives of the overload control strategy in TSPS No. 1B are to:

(*i*) Preserve system sanity

(*ii*) Maintain a high level of call completions regardless of the load applied to the system.

There are two types of resources that may be exhausted and lead to overload: TSPS call-processing resources, used exclusively by the TSPS process (e.g., TSPS peripherals, software resources used for call processing within the TSPS process, TSPS real time, etc.); and DMERT resources, either used exclusively by DMERT or shared with TSPS (e.g., kernel message buffers, nonswappable main memory, disk swap space, etc).

The TSPS No. 1 overload control strategy for TSPS resources has been preserved in TSPS No. 1B, with the software implementing it being emulated within the TSPS process. In this strategy, the E-E cycle time represents the detection parameter, while the number of calls being admitted and the number of active trunks represent the load control parameters. As the load increases, the E-E cycle time also increases in value. There are three thresholds for the E-E cycle time, determining the transitions from normal state, to phase 1 (minor) overload, to phase 11 (major) overload, and system initialization. During overload, the number of calls admitted and the number of active trunks are gradually reduced. As the load decreases, the E-E cycle time also decreases in value and the system is returned to its normal state.

The overload control strategy for DMERT resources shared by DMERT and TSPS is based on overload detection by the process that administers the particular resource, overload monitoring by the SIM process, and overload control by the SIM and AIM processes. Upon determining that an overload condition exists, SIM notifies the craft, attempts to free some resources to alleviate the condition, and then notifies AIM. For some DMERT-detected overloads (e.g., kernel or supervisor and user-level lockout) an initialization is executed to restore resources. Once notified of these overload conditions, AIM requests a DMERT Level 1 and TSPS minor audit initialization (MNA) initialization. Subsequent overload indications result in escalation of initializations.

***5.2.2.6 Processor switch.*** An example of the communication between DMERT and the TSPS application required for effective control of the system is illustrated by the processor switch. The duplex SPC 1B

normally operates with one processor on-line actively processing calls and the other off-line with its memory continuously updated on every write operation. The off-line unit is thus ready to be switched on-line should the need arise. A processor-switch request can be one of three types: recovery, routine, and manual. A recovery processor switch is requested when a fault has been detected in the on-line processor and processing cannot continue on it. The routine processor switches are scheduled periodically, while the manual requests are made by the craft when needed.

Certain time-sensitive activities within the TSPS application, e.g., writing on the AMA tape, benefit from an advanced notification that a processor switch is about to take place. Such a notification is used to complete time-sensitive operations and get ready for the switch. However, only routine and manually requested switches can be postponed until the application has been notified and given its approval. Therefore, the routine and manual processor switch requests are routed through the AIM process. In turn, AIM notifies the TSPS application processes of an imminent processor switch and, after their completion of time-sensitive operations, returns a switch go-ahead to SIM. As a defense mechanism, if the application does not approve the switch within 10 seconds, DMERT will proceed with the switch.

### *5.3 Other TSPS application processes*

In addition to the TSPS process numerous other native-mode processes were developed to support the emulation. These processes provide needed functions for a complete system under DMERT. The major processes are discussed below. Figure 13 illustrates the interaction between these major processes. The AIM process has been described previously, and therefore will not be included here. The PSI Diagnostic Driver and Control processes are detailed in Ref. 6.

#### *5.3.1 Kernel processes*

***5.3.1.1 PSI diagnostic driver.*** The PSI Diagnostic Driver process is a killable kernel process that performs diagnostic tests of the on-line PSI. To perform the on-line tests the driver must synchronize its operation with TSPS J-level operation in order not to interfere with ongoing peripheral I/O. It will execute the tests at execution level 12 when a non-interfering 3-ms window between J-levels is found. Its other processing is performed at lower execution levels. The driver shares memory with AIM and the TSPS process for synchronization of activities and in order to monitor TSPS peripheral equipment equippage and status.

#### *5.3.2 Supervisor processes*

***5.3.2.1 File system interface.*** The TSPS No. 1B application has only

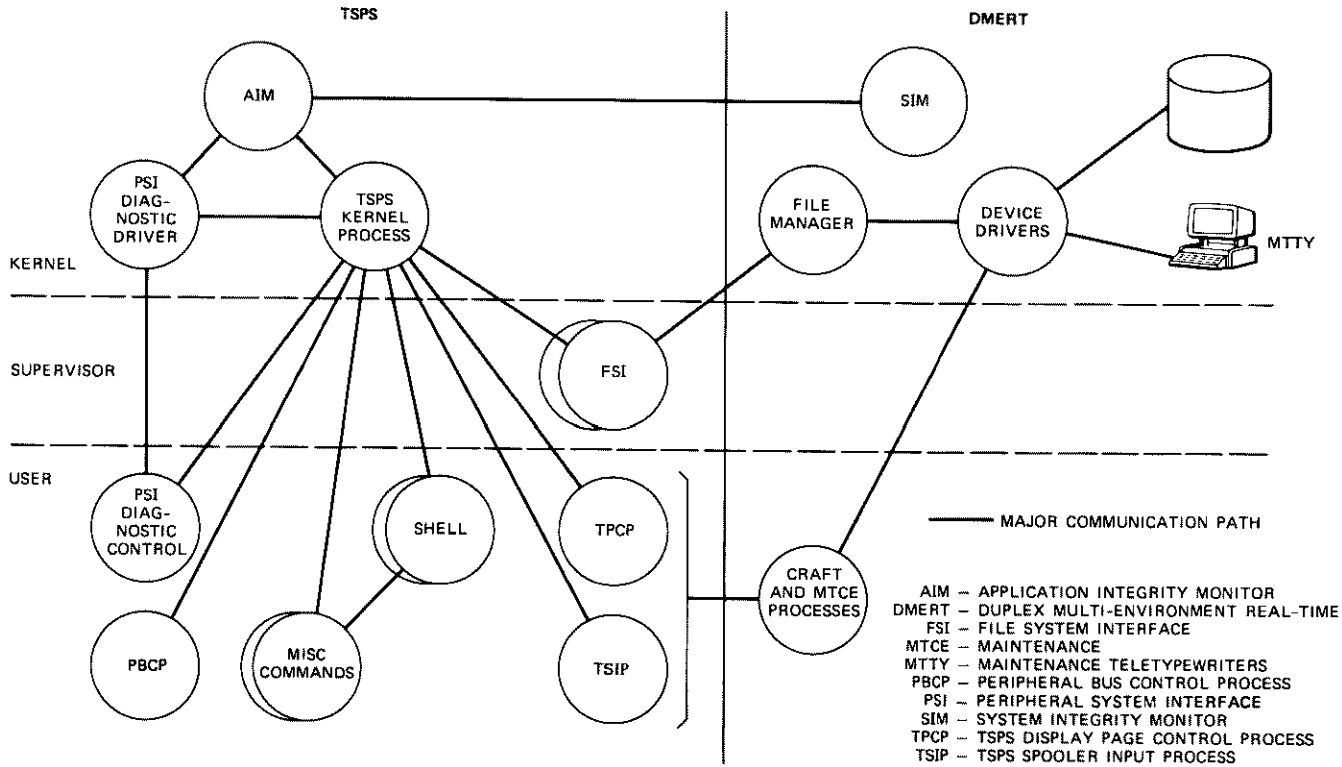



Fig. 13—Interaction between TSPS and DMERT processes in TSPS No. 1B process architecture.

one supervisor process, the File System Interface (FSI). The SPC 1A's program tape unit (PTU) was not retained in TSPS No. 1B. All required operations of the PTU were functionally reproduced with the FSI. Examples of such functions include dumping emulated office data, protected memory, or program to tape. The FSI shares the emulated-code address space and interfaces with the DMERT file manager to transfer data between the emulated address space and the DMERT disk file system. The 3B20D's nine-track tape unit and DMERT I/O facilities are used to move the data between tapes and the disk file system. None of the FSI functions are time-critical. Hence, the FSI is designed as a killable supervisor process that is created upon demand and dies when it has completed its function. Multiple instances of the FSI can simultaneously coexist as long as their respective functions do not interfere with each other.

#### *5.3.3 User processes*

The TSPS application has numerous user-level processes. Most of these perform demand tasks as the result of craft input. The process is created to perform the task and then dies when it is completed. The most significant user-level processes are described below.

***5.3.3.1 Craft interface processes.*** The maintenance center craft interface for the TSPS No. 1B consists of a video display terminal for input, output, and status displays, along with an adjacent printer for a hard copy of all output messages. The terminal's screen is split into four areas, as shown in Fig. 14. The top is for system status information and is always displayed. The variable-sized middle section is for displays which, for example, may show the status of a particular hardware subsystem. The remainder of the screen is for scrolling system output messages, and there is a single, dedicated line for input. This interface is also remoted to the Switching Control Center System (SCCS), described in Ref. 9, for remote maintenance.

Although the actual I/O to the devices is done by the DMERT I/O driver, the data read and written to the devices are formatted and interpreted by a collection of user-level processes. The major DMERT processes include the input Shell, the Controller of Output Spooler Process (CSOP), and the Display Administration Process (DAP).

The application interface to CSOP and DAP is designed to be from other user-level processes. Hence, TSPS has developed two user processes that serve as intermediaries between the TSPS kernel process and CSOP and DAP. The TSPS Spooler Input Process (TSIP) shares a memory buffer with the TSPS process. TSPS queues messages by priority in this buffer. TSIP dequeues these messages and passes them to CSOP via interprocess messages. CSOP then merges the TSPS-generated messages into a single, systemwide, prioritized queue for

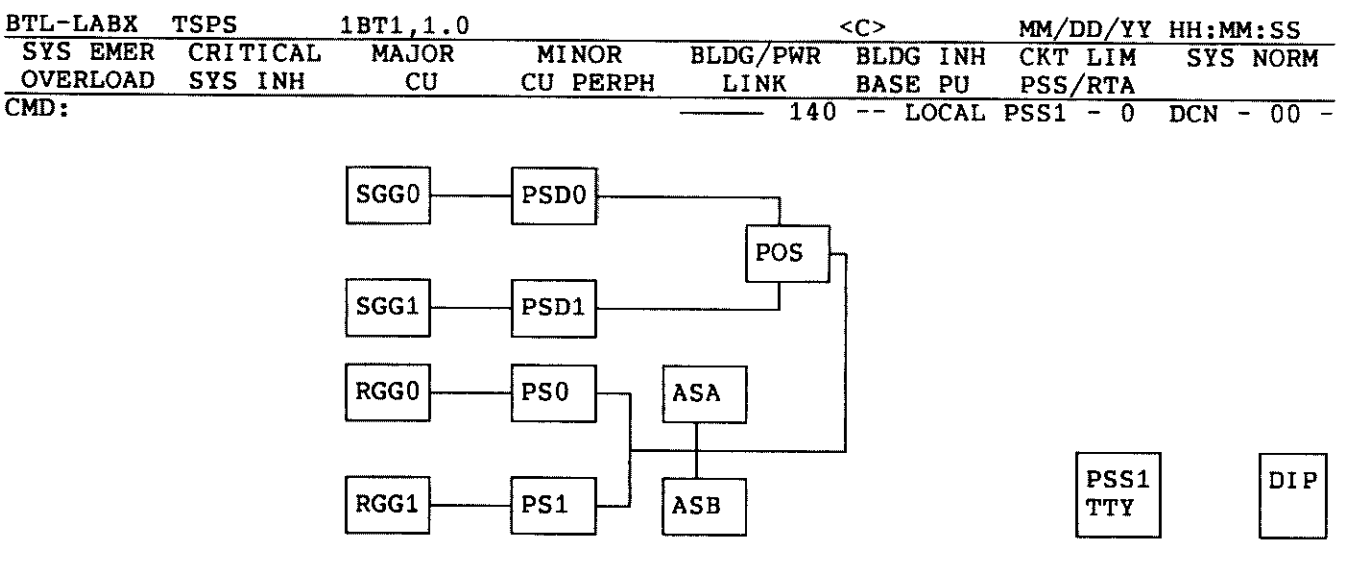



Fig. 14—Maintenance center craft interface for TSPS No. 1B.

printing to the maintenance terminals. The TSPS Display Page Control Process (TPCP) controls the TSPS application displays. TPCP receives change of state information from the TSPS process via interprocess messages. TPCP then translates this into the appropriate display page control information and directs DAP via interprocess messages to change the state of the various display page indicators.

The above-mentioned processes are potentially subject to being killed as the result of a system error or manual action. However, they provide a critical service and must have continuous operation. DMERT provides a Craft Monitor (CMON) process that monitors the operation of craft processes. CMON will immediately recreate any craft process that dies, along with any other process that must be reinitialized. CMON itself is monitored by the User-Level Automatic Restart Process (ULARP). ULARP will automatically restart any user-level process under its surveillance should it die. ULARP execution is monitored by the DMERT System Integrity Monitor process (SIM).

TSPS input message syntax is quite different from that used by DMERT. The processing of these input messages is also performed quite differently. As such, it was necessary to modify the DMERT shell to handle TSPS input as well. When an input message is entered by the craft and read by the combined shell, it first checks for a TSPS syntax message (the most likely to be input). If the input is a TSPS message, it is sent to the TSPS process in an interprocess message. The TSPS process will look up the message in its memory-resident catalog and act on it accordingly. If the input was not a TSPS message, the combined shell handles it in the standard DMERT fashion. That is, a disk directory search is performed to find the appropriate user-level process to execute in response to the command. This process is then created and executed to handle the input accordingly.

***5.3.3.2 PSI diagnostic control process.*** The PSI diagnostic control process is a user-level process that controls the execution of diagnostic tests on the off-line PSI. The PSI diagnostic was designed similar to DMERT common-system diagnostic processes and is woven into the DMERT diagnostic control structure. It is reacted and executed on demand as a result of a manual request, routine exercise or automatic diagnostic request. The PSI diagnostic creates the PSI diagnostic driver and communicates with it through interprocess messages to coordinate on-line and off-line PSI tests and TSPS process F-level interrupt recovery actions.

***5.3.3.3 Peripheral bus control process.*** Power to the PSI peripheral buses is controlled by the 3B20D Processor's scanner/signal distributor (SC/SD). The PSI bus-power control points are duplicated with a set on a SC/SD controller in each I/O processor (IOP). The Peripheral

Bus Control Process (PBCP) monitors and controls the status of the PSI peripheral buses. PBCP interfaces with the DMERT SC/SD administrator and the TSPS process via interprocess messages to coordinate actions. PBCP is a critical process that must always be active. Hence, it runs under the surveillance of ULARP.

## VI. SUMMARY

This article has described how the characteristics of the TSPS No. 1 and the 3B/DMERT systems have been combined in the TSPS No. 1B architecture. Successful emulation of the TSPS No. 1 software has been accomplished by hardware, firmware, and software in the TSPS No. 1B. Emulated code, along with associated native code, has been structured as a single kernel process running under DMERT. This kernel process cooperates with other application processes and DMERT to form integrated maintenance and craft interface packages. The TSPS No. 1B provides a modern, flexible vehicle for the future expansion of TSPS services.

## VII. ACKNOWLEDGMENTS

The TSPS No. 1B software described in this article was the result of contributions by many people. The authors wish to acknowledge their help in the preparation of this article. In particular, the authors wish to thank J. M. Aiken, P. S. Bogusz, D. L. Brown, D. L. Hofmockel, M. H. Richardson, E. S. Sachs, and M. D. Soneru for their contributions.

## REFERENCES

1. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.
2. B.S.T.J., *49*, No. 10 (December 1970).
3. B.S.T.J., *58*, No. 6, Part 1 (July-August 1979).
4. "3B20D Processor & DMERT Operating System", B.S.T.J., *62*, No. 1, Part 2 (January 1982).
5. D. L. Bayer and H. Lycklama, "*UNIX* Time-Sharing System: The MERT Operating System," B.S.T.J., *57*, No. 6, Part 2 (July-August 1978), pp. 2049–86.
6. H. A. Hilsinger, J. H. Tendick, R. A. Weber, and G. T. Clark, "Traffic Service Position System No. 1B: Hardware Configuration," B.S.T.J., this issue.
7. T. Hack, T. Huang, and L. C. Stecher, "Traffic Service Position System No. 1B: Software Development System," B.S.T.J., this issue.
8. B. A. Crane and D. S. Suk, "Traffic Service Position System No. 1B: Capacity and Reliability Evaluation," B.S.T.J., this issue.
9. J. J. Bodnar, J. R. Daino, and K. A. VanderMeulen, "Traffic Service Position System No. 1B: Switching Control Center System Interface," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Hardware Configuration

By G. T. CLARK, H. A. HILSINGER, J. H. TENDICK, and R. A. WEBER

(Manuscript received June 30, 1982)

*The Traffic Service Position System No. 1B (TSPS No. 1B) was developed by replacing the Stored Program Control No. 1A (SPC 1A) with the SPC 1B, consisting of a 3B20 Duplex Processor (3B20D) and a Peripheral System Interface (PSI). The PSI was designed to interface the 3B20D Processor technology with the existing TSPS peripheral system. This article describes the differences in technologies between the SPC 1A and SPC 1B, the hardware design required to overcome these differences, and the fault recovery and diagnostic software development required to integrate the new hardware into the maintenance structure of the TSPS No. 1B.*

## I. INTRODUCTION

The 3B20 Duplex Processor (3B20D) was developed as a general-purpose processor with a set of common system peripherals to support a wide range of applications and an instruction set optimized for a high-level language compiler.[1] To allow the 3B20D Processor to communicate with the Traffic Service Position System (TSPS) peripheral community, a special interface circuit was needed to bridge the difference in speed, timing, and control protocols between the new processor and the existing TSPS peripherals. This circuit is called the Peripheral System Interface (PSI) circuit. To bridge the software technologies without rewriting all existing TSPS operational code, the Stored Program Control No. 1A (SPC 1A) instruction set was emulated via special microcode, and 3B20D native code was used to interface the emulated code with the DMERT operating system. With these changes in hardware and software came the task of developing a

maintenance strategy that could integrate the advantages of the 3B20D Processor and Duplex Multi-Environment Real-Time (DMERT) operating system with the existing TSPS maintenance strategy. Finally, to introduce the 3B20D Processor into in-service TSPS offices, retrofit procedures had to be developed to replace the existing TSPS No. 1 processor with a 3B20D Processor.

The task of incorporating the 3B20D Processor into the TSPS system provided a unique set of challenges. This article describes the hardware and associated maintenance software required to accomplish this task. The emulation microcode, overall software architecture, and the implementation of the processor retrofit are covered in Refs. 2 and 3.

## II. TSPS NO. 1B PROCESSOR AND PERIPHERAL INTERFACE DESCRIPTION

The TSPS No. 1B configuration consists of the existing TSPS peripheral system and a new processor called the SPC 1B. The SPC 1B consists of the 3B20D central control unit, disk file community, input/output processors, and the Peripheral System Interface unit (see Fig. 1*).

The peripheral architecture for TSPS No. 1B remains essentially unchanged from TSPS No. 1. The interface between 3B20D Processor and the TSPS peripheral communication buses is provided by the Peripheral System Interface circuit. In the following sections a short description will be given of the SPC 1A, and the TSPS peripheral system architecture. These sections are included to highlight the significant differences in processor design and to identify integration requirements. For a more detailed description of the SPC 1A see Ref. 4.

### *2.1 Description of the SPC 1A*

The SPC 1A consists of a duplicated pair of central control (CC) units, a duplicated program and data store complex, a program tape unit (PTU), a master scanner, a signal distributor, a central pulse distributor (CPD), a maintenance control center, and a duplicated peripheral communications bus system (see Fig. 2).

The CC has a processor cycle time of 6.3 $\mu$s, with an instruction execution time of from 1 to 3 cycles. Most register operations require a single cycle, while memory writes require three cycles, and most peripheral orders require two cycles. The processor performs logic with a 20-bit combinational logic subtractor, which also has the capability

* Acronyms and abbreviations used in the figures and text of this paper are defined at the back of this Journal.

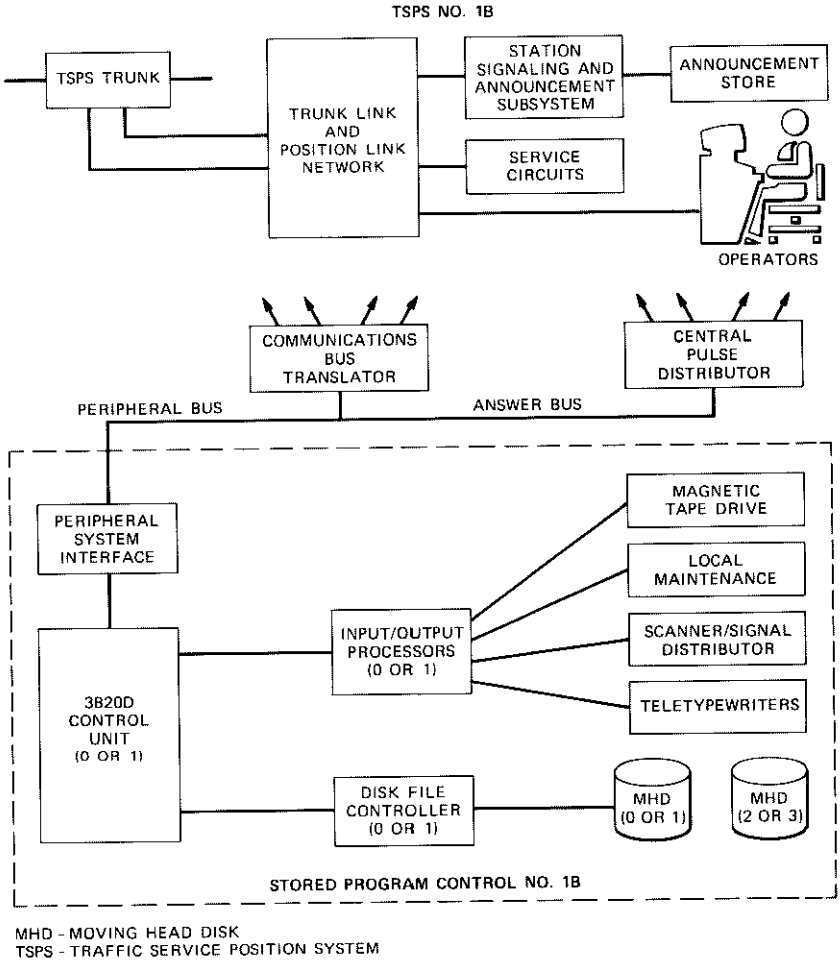



Fig. 1—Traffic Service Position System No. 1B.

of performing the logic operations AND, OR, and EXCLUSIVE-OR. Circuits are also provided to perform shift or rotate functions, both right and left. There are seven general-purpose registers each containing 20 bits, which provide internal storage for information manipulation.

Although the CC arithmetic/logic unit (ALU) operates on 20-bit words, the communication between the CC and the main stores is over a 47-bit data and instruction bus with a 19-bit address bus. Data are physically stored as 47-bit blocks, which consist of either a 40-bit instruction or two 20-bit data words, six Hamming error-correcting code bits, and an overall parity bit. An additional bit is used internal

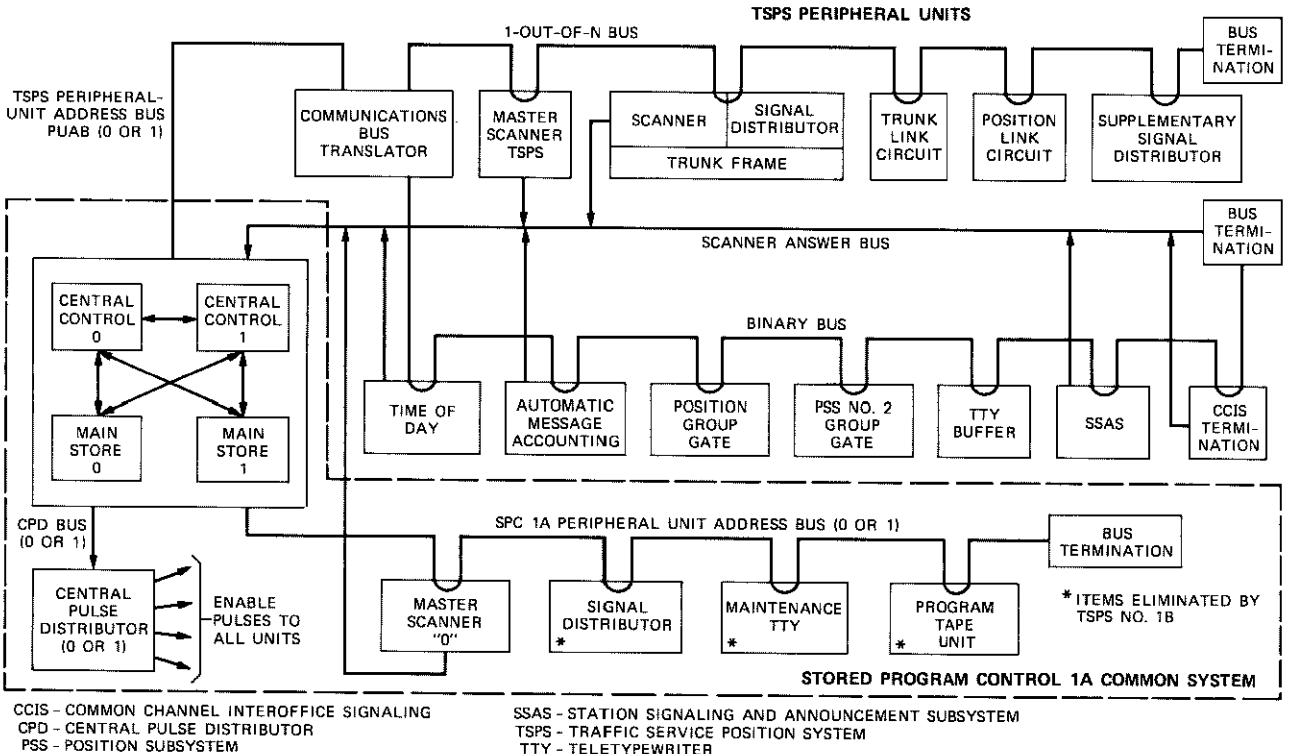



Fig. 2—Traffic Service Position System No. 1.

to the processor to select one of the 20-bit data words for read and write operations. This additional bit provides an effective 20-bit addressing capability.

The SPC 1A program and data store complex consists of duplicated store communities with each community containing up to a maximum of 30 store modules with each module containing 16,384 47-bit words. These stores are used for both program and data storage. Each processor has access to both store communities for increased reliability. Normal nonfault operation is for both store buses to be active with the active processor accessing both stores and comparing results. For processor write operations both store buses are written in order to maintain up-to-date copies in both memories.

The interface of the processor to the peripherals is provided by four AC buses. First is the SPC 1A peripheral-unit address bus, which is dedicated to the units that are common to every SPC 1A installation. The common equipment includes a master scanner, signal distributor, maintenance teletypewriter (MTTY), and a program tape unit (Fig. 2). The second bus is the Peripheral-Unit Address Bus (PUAB), which transmits data to the TSPS peripheral units. The third bus is the Central Pulse Distributor Bus (CPDB), and the fourth is the Scanner Answer Bus (SAB). A more detailed description of this peripheral interface will be given in Section 2.2, since it is this peripheral bus architecture that must be interfaced with the 3B20D Processor.

As mentioned above, a program tape unit, master scanner, signal distributor (SD), central pulse distributor (CPD), and maintenance control center are the basic set of peripherals required for all SPC 1A applications. The program tape unit is a nine-track, 200-bit-per-inch (BPI) tape unit that provides a bootstrap program loader as well as a general-purpose input/output capability. The master scanner is used throughout the TSPS No. 1 for data input from the peripheral equipment. The master scanner contains a matrix of scan points connected in such a manner that they can be interrogated in groups of 16 to determine the state of a particular scan point. The output of the scan points is transmitted to the SPC 1A on the SAB. The signal distributor is accessed via the SPC 1A Peripheral Unit Address Bus and provides a matrix of latching relay contacts that may be opened or closed on command from the processor. The central pulse distributor (CPD) is used to provide unit addressing for data transfers over the shared system buses. The CPD is a translator decoder that takes a binary address from the CPDB and provides a unipolar or bipolar pulse on one of a possible 1024 points. The maintenance control center consists of the MTTY and the Control and Display circuit. The MTTY provides the primary craft interface function. System configuration, status, and system diagnostics may be controlled via MTTY input. In

addition to the MTTY, the Control and Display circuit visually indicates the system status through indicator lamps and allows manual control of the system by means of keys and switches.

### *2.2 Description of the TSPS peripheral interface*

The primary hardware development for the TSPS No. 1B was the PSI, which interfaces the 3B20D to the existing TSPS periphery. The TSPS peripheral interface is described at this time to provide background for description of the PSI.

The SPC 1A Peripheral-Unit Address Bus shown in Fig. 2 has been eliminated for TSPS No. 1B. The units associated with this bus, the master scanner, the signal distributor, the MTTY, and the program tape unit, have either been eliminated or moved to the TSPS Peripheral-Unit Address Bus. The signal distributor was used primarily for SPC 1A maintenance and control and is no longer needed. A few signal distributor points that were not associated with the SPC 1A were reassigned to signal distributor points in the TSPS periphery. The MTTY and the program tape unit were replaced by equivalent functions in the 3B20D Processor, and the master scanner was moved to the PUAB because many of the scan points were used by TSPS peripheral circuits.

The peripheral bus structure retained after replacement of the SPC 1A consists of the Central Pulse Distributor Bus, the Peripheral-Unit Address Bus and the Scanner Answer Bus. These three peripheral communication buses retained for TSPS No. 1B contain four types of leads: data, control, error detection, and diagnostics.

The CPD bus includes 32 enable address bits that select one of the possible 1024 enable pulse outputs from the CPD. Control leads include a bus sync and four execute pulses that activate the CPD decoding logic. Each execute pulse is dedicated to one CPD unit. An additional control lead is the "we-really-mean-it" (WRMI) pulse, which is used to provide additional signaling redundancy, thereby increasing noise immunity. The error-detection leads include parity check bits across the data bus, an execute complete signal, an "all-seems-well" (ASW) response bit, a bit that indicates single or multiple CPD output pulses, and two enable-verify reply signals that indicate reception of an enable pulse by the selected peripheral unit. Diagnostic and maintenance leads include a master reset, and a bit to diagnose error-checking circuits.

The PUAB leads pass through a Communications Bus Translator (CBT) circuit before being transmitted out to the TSPS peripherals. The PUAB has 20 bits of data and three control leads. The control leads include a CBT reset pulse to clear CBT input registers, a latch pulse to load the CBT input registers, and an execute bit that causes

the CBT to send translated data to the peripheral units. Error-checking leads include a parity bit over the data field, and an ASW bit, which is a summary of error checks performed by the CBT.

The SAB is the input bus to the processor; it includes 20 bits of data. Bit twenty is used by some peripheral units as a parity bit; the processor provides a parity check on the scanner answer data. Control functions for the scanner bus are all provided internal to the processor and consist of a window during which peripherals must load answer data onto the bus.

The error-checking leads described above are connected to check circuits in the processor or connected directly to a set of hardware registers called the Peripheral-Unit Maintenance Summary (PUMS) registers. The outputs of the error-checking circuits are also loaded into the PUMS register. The data in the PUMS register is compared with expected data, and if any errors are detected, the processor circuit generates a maintenance interrupt that results in the scheduling of TSPS fault-recovery programs.

Other peripheral interface leads include clock leads and processor-controlled pulse leads. The clock leads include 91-$\mu$s, 0.5-$\mu$s, and 5-ms output pulses, which drive the Automatic Message Accounting tape units, teletypewriter buffers, and other circuits that need them. The pulse leads are direct processor-controlled leads called "P-position" pulse sources. The P-position pulses were provided in the SPC 1A by special processor hardware called Buffer Bus Registers.[4] The purpose of these pulse sources is to provide direct processor control of critical peripheral units for hardware recovery in the event that the normal peripheral communication path is failing. These pulses include: a CPD reset; a network reset for trunk-link and position-link circuits, universal trunk frames, etc; and an auxiliary reset pulse, which is presently used for the TSPS Position Group Gate and the Service-Observing Gate transmitter circuit.

In essentially all cases, the characteristics of the signals transmitted and received on these buses are bipolar AC pulses of 0.5-$\mu$s duration and approximately 6-volt amplitude.

All peripheral orders—both distributes(DIST) and scans (SCAN)—operate with the same basic sequence. An address is loaded onto the CPD bus, which activates one of a possible 1024 enable-pulse points out of the CPD; immediately after the CPD address is pulsed, binary data may be placed onto the PUAB. The enable pulse from the CPD activates receiving circuits in the selected peripheral unit and primes that circuit to receive data over the PUAB. The selected circuit responds based on the data received and the CPD pulse received. For example, if a scanner were being accessed, the binary address would specify a group of 16 scan points to be interrogated, and the state of

the scan points would be gated onto the SAB and returned to the processor. By requiring each circuit in the periphery to read the binary address only after receiving a CPD enable pulse, a high degree of noise immunity and reliability is achieved in peripheral communications.

In summary, there are greater than 200 data and control leads that provide communications between the TSPS peripherals and the processor. They are AC bipolar pulses and are transformer-coupled to the peripheral buses.

## III. GENERAL CHARACTERISTICS OF THE 3B20D PROCESSOR

The central control (CC) of the 3B20D Processor[5] is microprogrammed, extensively self-checked, and handles 32-bit data words in a 24-bit address space. The control unit (CU) consists of the CC, its main memory, cache, Direct Memory Access (DMA), and input/output channels. In the standard duplex processor configuration, two control units operate in a nonmatching mode with one designated as active, while the other serves as standby-ready. A main memory update circuit maintains the consistency between main stores to ensure an up-to-date environment in the event of a processor switch. The SPC 1B Processor and peripheral architecture is shown in Fig. 3.

The 3B20D Processor supports a wide variety of peripheral devices including moving head disks, magnetic tape drives, data links, and terminals. Peripherals are controlled in programmed I/O mode directly by the processor, or with Direct Memory Access (DMA) through appropriate interface circuitry. A Disk File Controller (DFC) unit manages the information flow to and from the moving head disk drives, and an Input/Output Processor (IOP) unit provides interface control for magnetic tape drives, data links, terminals, and other customized peripheral controllers.

The central processing unit of the 3B20D utilizes a 32-bit data architecture throughout, including the main memory buses and an optional 8K-byte cache memory. Extensive self-checking is used to ensure immediate fault detection and possible correction of errors to allow graceful recovery and ensure high system availability. Hamming correction of single-bit errors and detection of all double-bit errors are performed by the main-store controller on each memory read operation. In addition, on every refresh cycle, a memory word is checked such that all words are tested periodically. This capability ensures that memory failures occurring on infrequently used addresses will be detected quickly.

Software processes on the 3B20D use a 24-bit virtual address space, which is converted by memory management hardware to 24-bit physical addresses using a paged-segmentation scheme. The memory man-

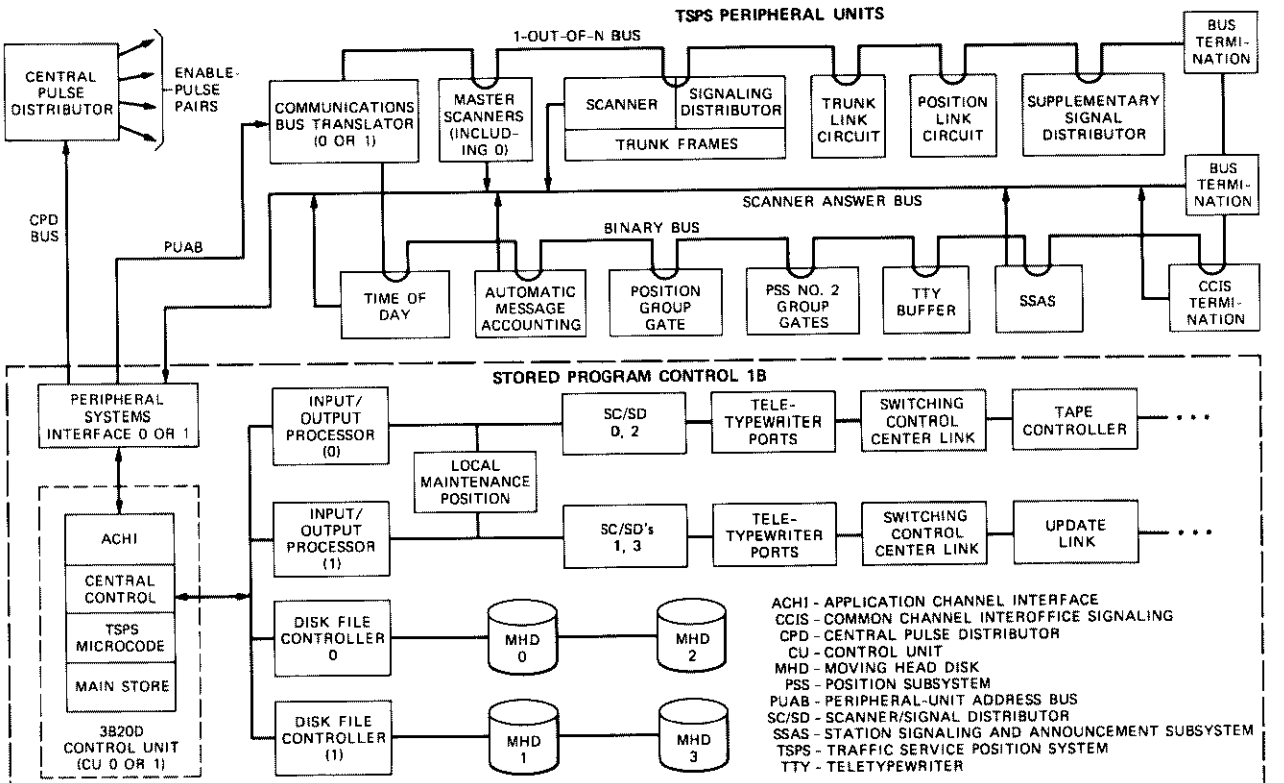


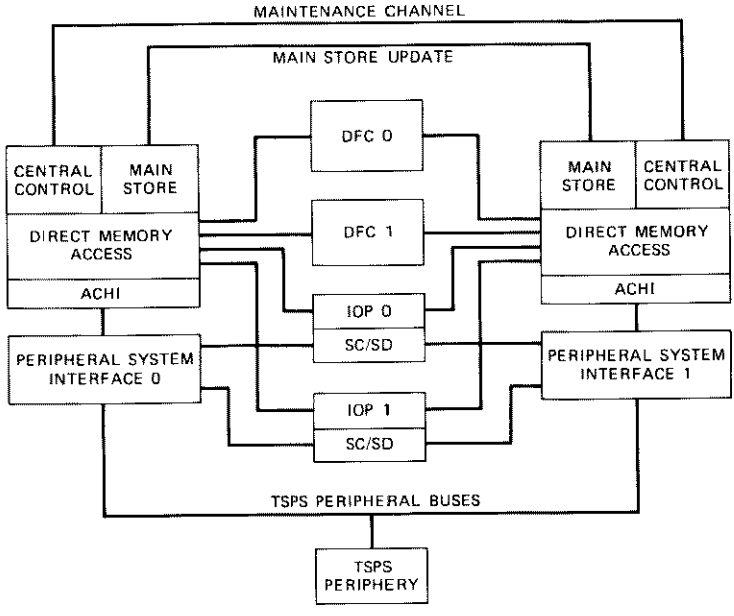

Fig. 3—The SPC 1B Processor and peripheral architecture.

agement circuitry is a high-speed, two-way, set-associative memory called the Address Translation Buffer (ATB). The ATB reduces overhead associated with the address translation function, and also provides main-store access protections on a page or segment basis. Physical memory can grow in half- or full-megabyte increments to a total of 16 megabytes. Memory instructions provide addressing capabilities with byte, half-word, full-word (32 bits), or move-block options.

The 3B20D Processor is a microprogrammed machine where up to four instruction sets can be dynamically selected. A variable microcycle ranging from 150 to 300 nanoseconds is employed to minimize instruction time. The switchable instruction sets allow an application to transport current software, via emulation, to the 3B20D while still providing a modern operating system in a high-level language (i.e., DMERT and C language). The native instruction set is optimized for the C programming language and supports all C data types and control structures. Additional features within the CC include: duplicated Arithmetic/Logic Units (ALUs) for immediate fault detection; extensive bit rotate, mask, and test capability; a real-time clock with one-millisecond resolution; and internal sanity timers.

In its duplex configuration, two basic links exist between duplicated CUs. The first link is driven by the maintenance channel circuit and serves as a communication and control link between the processors. This link is used to execute off-line diagnostics, off-line audits, and information transfers during a processor switch. The second link connects the memory update circuits, which keep both main stores synchronized by sending all writes to cache and main store to the off-line memory controller. This second link also contains a backup maintenance channel in the event of a fault in the regular link.

Peripheral units are connected to the CU via the Direct Memory Access (DMA) unit. The DMA does not interface directly with peripheral units, but rather communicates with two intelligent subsystems; a Disk File Controller (DFC), and an Input/Output Processor (IOP). Communications to both the DFC and IOP are via Dual-Serial Channels (DSCH), which allow any peripheral to operate with either CU of a duplex pair. Each DFC is currently capable of supporting 16 moving-head disk drives of 300-megabyte capacity each. An IOP is capable of supporting a wide variety of peripherals such as nine-track tape units, printers, synchronous and asynchronous data links, maintenance and general-purpose terminals, scanner/signal distributor, and custom network interfaces. Peripherals may also be connected to the processor Central Control Input/Output (CCIO) bus via an interface such as the Application Channel Interface (ACHI). The ACHI is a 32-bit, parallel-interface board that provides I/O capabilities for applications needing high-speed data transfers with minimal overhead.




ACHI - APPLICATION CHANNEL INTERFACE
DFC - DISK FILE CONTROLLER
IOP - INPUT/OUTPUT PROCESSOR
SC/SD - SCANNER/SIGNAL DISTRIBUTOR
TSPS - TRAFFIC SERVICE POSITION SYSTEM

Fig. 4—Duplex configuration of the TSPS No. 1B.

## IV. CONFIGURATION OF THE TSPS NO. 1B

The TSPS No. 1B configuration consists of the duplex SPC 1B, TSPS peripheral units, several printers, terminals, and a duplicated link to the Switching Control Center System (SCCS), which serves as the remote maintenance center.[6] The SPC 1B includes the 3B20D Central Control Unit (CU), disk file community, Input/Output Processor (IOP), TSPS emulation microcode, Application Channel Interface (ACHI), and the Peripheral System Interface. The overall TSPS No. 1B configuration is shown in Fig. 4.

The 3B20D Processor is equipped with nine megabytes of main store for the initial TSPS No. 1B program generic. Hardware added to the basic processor for the TSPS application includes two boards containing three kilobytes of Programmable Read-Only Memory (PROM) for the emulation microcode, and an Application Channel Interface (ACHI) circuit board. The ACHI plugs directly into the CCIO bus and provides a high-speed, 32-bit data transfer interface to the PSI. The PSI was designed as an interface for the different I/O signaling protocols of the 3B20D and the existing TSPS peripheral circuits. A

detailed description of the PSI is given in Section V. The TSPS periphery[4] is preserved in its same basic configuration with only minor differences.[3]

The disk file community for TSPS consists of duplicated DFCs including two Moving Head Disks (MHDs) for each disk file controller (DFC). One disk on each DFC is physically designated as the system disk and contains the DMERT and TSPS programs and data. The other disk drive is a spare that is fully diagnosable and contains a copy of the system disk. If a system disk does fail, the spare drive may be recabled to provide a full-duplex disk configuration once again.

The IOPs are equipped to provide four scanner/signal distributors plus interface boards for a 1600-bpi magnetic tape frame, Craft Interface MTTY, Receive-Only Printer (ROP), an office alarm control circuit, a recent change and verify terminal, an auxiliary printer, a data-linked Field Update interface,[7] and remote SCCS data links. The additional scanner/signal distributor monitors the PSI power switch and also provides duplicated control over +24-volt power to the PSI peripheral bus drivers. The MTTY and ROP are separately connected to one of the two IOPs through an automatically or manually controlled port switch. This 3B20D feature provides a duplicate path between the central processor and the maintenance center equipment without requiring recabling or duplicate terminals. Included with the common 3B20D equipment is a software package to receive and verify software updates at the field site over data links.[7] The SCCS interface is a duplicated 4800-baud link that provides 14 virtual channels through a BX.25 protocol. The channels are used to transmit critical status indicators, alarms, processor recovery messages, and normal maintenance channel messages. In the typical configuration, the SCCS is used to remotely monitor the TSPS sites and has complete capability to control the machine in recovery situations.[6] The local MTTY is a video display console that shows critical indicators, provides several display pages depicting system configuration information, and allows craft personnel to enter system input messages. TSPS equipment configurations have been incorporated into the MTTY displays by providing additional display pages and additional page numbers for the application-only displays. A description of the Craft Interface is given in Ref. 2.

An extensive, non-interfering, field-debugging utility is also available by inserting a Dual Utility Circuit (DUC) into the 3B20D. The DUC has a dedicated slot in the processor that has access to all the important system buses. A Field Test Set (FTS) connects to the DUC via ribbon cable to provide a program debugging environment for both high-level programming languages and TSPS emulated code. Address and data matchers are available, as well as program tracing capabilities. More

information about the debugging environment used in the TSPS No. 1B development is given in Ref. 7.

## V. PERIPHERAL SYSTEM INTERFACE

### 5.1 Basic purpose

The basic purpose of the Peripheral System Interface is to act as a data transfer interface with the TSPS periphery identical to that provided by the SPC 1A. TSPS peripheral units communicate with the SPC 1A via the PUAB, CPDB, and the SAB. The 3B20D Processor communicates with its channels over the CCIO bus. For proper operation, TSPS units require a precisely timed set of handshaking signals in addition to electrical compatibility with the CCIO bus. To avoid changing the peripheral units themselves, the PSI was designed to interface the TSPS periphery with the 3B20D Processor. In addition, an ACHI is provided with the 3B20D Processor to allow PSI indirect access to the CCIO bus. In essence then, the PSI appears as a main I/O channel to the 3B20D Processor. The overall structure incorporating the 3B20D Processor and the PSI into TSPS is shown in Fig. 4; Fig. 5 is a functional block diagram of the PSI circuit.

### 5.2 Overview

The general flow of information between the 3B20D Processor and the TSPS periphery is as follows:

(*i*) A command word along with data information is formatted by the emulation microcode and sent to the ACHI.

(*ii*) The PSI receives the command, decodes the instruction, and starts the appropriate handshaking sequence on the TSPS peripheral buses.

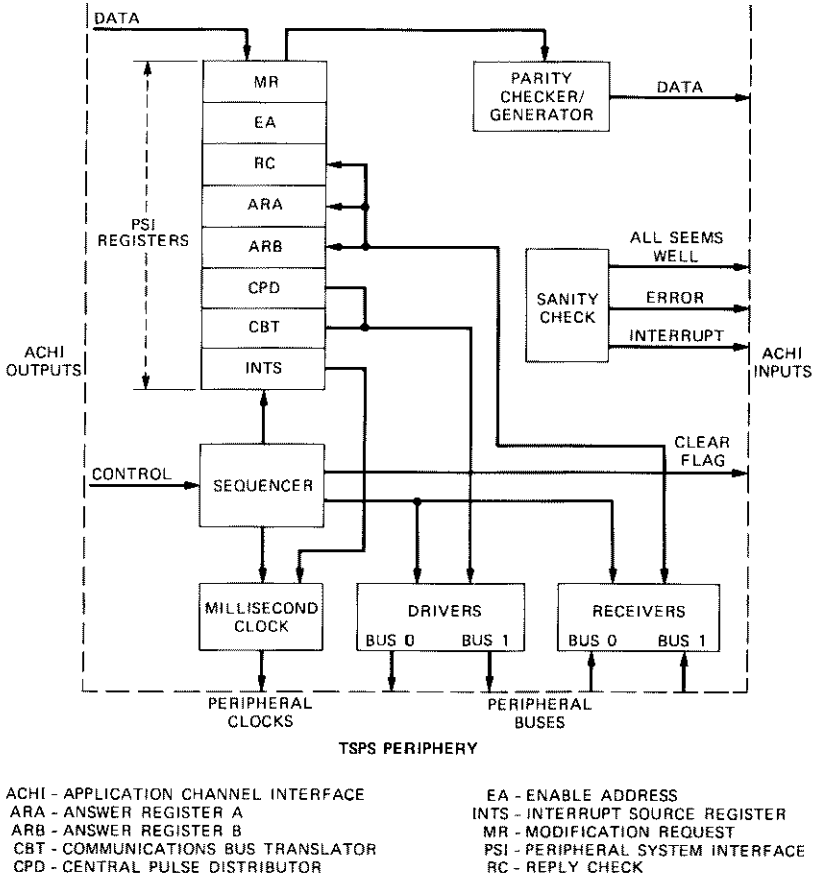
(*iii*) Once the command has begun, the emulation microcode passes a second data word via the ACHI to the PSI. This data is then formatted by the PSI and passed onto the peripheral circuits at the appropriate moment.

(*iv*) Any data sent back from the TSPS periphery along with error check signals are verified and then combined with internal PSI fault-detection information and passed back to the ACHI.

(*v*) The CU receives the data and emulation microcode performs further post analysis and error checking on the data before the conclusion of the peripheral order.

### 5.3 Application channel interface

The ACHI is a 3B20D Processor I/O channel that provides a high-speed, parallel, 32-bit interface between the 3B20D and an application device. Control signals allow the information transfers to be marked as




ACHI - APPLICATION CHANNEL INTERFACE
ARA - ANSWER REGISTER A
ARB - ANSWER REGISTER B
CBT - COMMUNICATIONS BUS TRANSLATOR
CPD - CENTRAL PULSE DISTRIBUTOR
EA - ENABLE ADDRESS
INTS - INTERRUPT SOURCE REGISTER
MR - MODIFICATION REQUEST
PSI - PERIPHERAL SYSTEM INTERFACE
RC - REPLY CHECK

Fig. 5—Block diagram of the Peripheral System Interface circuit.

either data or command information when transmitted over the CCIO bus. When the application passes data to the 3B20D, additional control signals are used to indicate an "all seems well" or error response. Other miscellaneous control signals are also buffered by the ACHI. One such control signal used by the PSI keeps the 3B20D Processor and PSI clocks synchronized, while another is connected to a high-priority processor interrupt to signal when a fatal PSI hardware fault is detected. A third lead is an extension of the CCIO bus I/O inhibit lead. This signal is asserted in the off-line processor to inhibit all channels, thus preventing any off-line channels from interfering with the active processor. The PSI uses the inhibit lead in the same manner; it directly prohibits all PSI peripheral bus drivers from pulsing even though the sequencer may be active.

### 5.4 PSI functional description

The five major functional areas in the PSI are:

(*i*) The ACHI interface
(*ii*) PSI registers
(*iii*) Peripheral bus drivers/receivers
(*iv*) Instruction sequencer and clock circuits
(*v*) Maintenance and fault-recognition circuits.

Each of these areas is described in more detail in this section.

#### 5.4.1 ACHI interface

The ACHI interface uses 5-volt, differential dc signaling on 80 pairs of leads. There are separate input and output buses, with each bus containing 32 data bits, 4 bits of byte parity, and 4 control leads. Data transfers from the ACHI to the PSI are accomplished by making data available on the PSI's input bus and then raising the appropriate command or data flag. This flag starts the PSI sequencer, which gates the data into the appropriate register and resets the control and data flags. The PSI passes data back to the processor by driving its output data bus and then pulsing the normal or maintenance control leads. These leads are used internally in the ACHI to latch the new data and also to indicate that data is present on the next status interrogation by the emulation microcode.

#### 5.4.2 PSI registers

The eight registers in the PSI are used to store data from the ACHI, the periphery, error-check signals accumulated during an order, and to perform maintenance control functions or operations. The CPD and CBT registers are used to store data received from the ACHI for the currently executing peripheral instruction. The outputs of these registers are gated through decoder circuits and then pulsed onto the peripheral buses at the appropriate time. The two answer registers (ARA, ARB) receive data from the duplicated peripheral Scanner Answer Buses. This data is typically matched at the end of the peripheral order sequence and one set is passed to the processor through the ACHI. The Reply Check (RC) register gathers peripheral-error-response data and is used to determine whether to assert the normal or maintenance flag when sending data back to the processor. The Maintenance Register (MR), Enable Address Register (EA), and the Interrupt Source Register (INTS) are all used by diagnostics to provide special maintenance observation and control points during diagnostic exercises.

#### 5.4.3 Peripheral bus drivers/receivers

The PSI must interface to three main TSPS peripheral buses. All peripheral signaling uses 0.5-$\mu$s pulses over AC transformer coupled

buses for maximum isolation and noise immunity. The purpose of each bus is described in Section 2.1. The PSI has access to both halves of each of the duplicated buses.

#### 5.4.4 Instruction sequencer and clock circuits

The PSI sequencer consists of a dual set of timing chains, driven by a 10-MHz oscillator, that can be used to provide pulses from 100 ns to 12 $\mu$s wide. This is accomplished by having the first chain pulse one of its ten output leads every 100 ns, while the other chain pulses one of its twelve output leads every microsecond. These pulses are then logically combined to provide variable-length timing control pulses throughout the execution of the PSI commands.

Four major types of instructions are sequenced by the PSI. The peripheral order is the most common instruction and has three built-in options. These options allow orders to set network points and return response data with or without odd parity, or to set network points only. The pulse order is another instruction that communicates with peripheral equipment and is used to reset network points. Finally, read and write instructions allow direct access to the PSI's registers.

A millisecond clock circuit uses countdown logic, from a 10-MHz source, to produce a 0.5-$\mu$s pulse every 90.9 $\mu$s, 500 $\mu$s, and 5 ms. These pulse chains are used by AMA circuits, TTY buffers, and other miscellaneous peripherals. Through a control lead from the ACHI, the PSI system clocks are aligned with the 3B20D interrupt timers. This feature is necessary to align the AMA data transfers with the clocks to physically write the bytes onto the tape. The alignment process is also used to detect a faulty PSI clock circuit by comparing the states of the 3B/PSI clocks every 5 ms and to generate an interrupt if the PSI clock appears to be fast or slow.

#### 5.4.5 Maintenance and fault-recovery circuits

Since the PSI is considered part of the SPC 1B processor, high reliability and availability plus immediate fault detection is provided. A considerable amount of circuitry is included for special maintenance access for fault resolution and also for immediate error detection during peripheral order sequences. Five separate error sources are combined to generate a processor interrupt if an internal PSI hardware error is detected during the sequencing of an instruction. Command-field parity, data-field parity, multiple commands, data missing, and sequencer sanity checks are made on each order. Sequencer sanity checks are made by comparing a predicted bit stream stored in Programmable Read-Only Memory (PROM) against the actual sequence of pulses. If any error checks fail, an error indication is latched in the PSI INTS register and an interrupt signal is pulsed through the

ACHI and latched into the 3B20D interrupt source register. At the conclusion of the instruction, fault-recovery software is dispatched to take appropriate recovery actions. The clock circuit continuously drives three system clock chains out on the peripheral buses. In addition to possible errors during command sequences, any circuit failures or fast and slow clock errors generate an immediate interrupt, which is passed onto the 3B20D Processor.

Additional maintenance features are included to aid the fault-resolution process and to allow major internal portions of the PSI to be tested before peripheral drivers are enabled out onto the buses. One such feature allows the PSI diagnostic to send data out of the ACHI, through the PSI, and back to the ACHI without any PSI sequencer actions. This allows the ACHI and interface cable faults to be isolated from PSI problems. Another major diagnostic feature allows the preloading of answer and reply check registers before a peripheral order is executed. This allows the diagnostic to check nearly all features of the sequencer and error check hardware before enabling the PSI drivers onto the active peripheral buses. Therefore, this feature helps prevent interference problems between the duplicate processors and allows faults to be distinguished between internal sequencer problems and peripheral interface circuits.

## VI. PHYSICAL DESCRIPTION OF THE SPC 1B

The SPC 1B consists of a number of different frames, units, circuit packs, and interconnections to accommodate the various circuits that comprise the system. These are arranged in configurations that allow for redundancy and growth to achieve a reliable and repairable system.

### *6.1 General*

Most of the wired frames use a newly designed framework developed for the 3B20D Processor.[8] The new framework is 2-ft 2-in. wide, 2-ft deep, and 7-ft high. The greater depth, as compared to existing TSPS frames, which are 1-ft deep, allows for the use of a larger circuit pack and still can accommodate backplane interconnections. Frame uprights are drilled on four surfaces that allow for mounting units of different sizes and dimensions. Some of these holes are used for mounting cable brackets, wire guides, and other hardware for supporting both intraframe and interframe cabling. *BELLPAC** hardware was used in the majority of the unit designs. This consists of apparatus mountings, mounting plates, and other backplane hardware. Units are arranged to accommodate circuit packs of the 8- x 13-inch size. Printed

* Trademark of Western Electric.

wiring backplanes range from two-layer battery and ground to multilayer arrangements for the more complex circuits. One or more units are provided for each frame as required to accommodate the numerous circuits that make up the SPC 1B. The various units on the frame are interconnected with ribbon cable assemblies or similar wiring means.

The circuit packs use dual in-line package devices for most logic functions. Both 200-pin and 300-pin connecters are provided on the circuit packs. Some number of these pins is reserved for battery and ground potentials. Both double-sided-rigid and multilayer printed wiring boards are used. All packs are equipped with face plates and tabs for product-code markings. Circuit packs are mounted on one-inch centers in the apparatus mountings to allow adequate cooling.

The majority of the frames are equipped with their own dc/dc converter units for conversion of central office battery to logic potentials.

The various frames are interconnected with connectorized switchboard cables to form an SPC 1B.

### *6.2 Equipment*

#### *6.2.1 3B20D Processor frames*

The SPC 1B is composed of two PSI frames, two control unit frames, two peripheral control (PC) frames, four Moving Head Disk (MHD) frames, a tape unit (TU) frame, a power distributing frame, a Local Maintenance Position (LMP), a Recent Change/Verify Position (RCVP), and other craft-machine interface hardware. Figure 6 shows the frames for the processor and peripherals line-up except for the local maintenance position, which is illustrated in Fig. 7. A description of the physical design of the 3B20D hardware is given in Ref. 8; additional frames required for an SPC 1B are described below.

#### *6.2.2 Peripheral System Interface frame*

The PSI is a single-bay frame, 2-ft 2-in. wide and 7-ft high, arranged to accommodate the PSI circuitry (see Fig. 8). The frame consists of seven component units. On two of these units up to 20 circuit packs that provide the logic functions and the power control functions can be mounted. Figure 9 shows one such logic unit for the PSI. A third unit at the top of the frame has mountings for transformers, inductors, connecters, and terminations that provide the means for interconnecting the PSI to the existing TSPS peripheral buses. On a fourth unit dc/dc converters can be mounted. The three remaining units are fuse panels. Two of these are for distributing +5V to the logic circuit packs. The remaining fuse panel distributes central office battery to the dc/dc converter input, bus controls, and maintenance functions. A frame filter unit in the frame base filters the central office battery. The logic

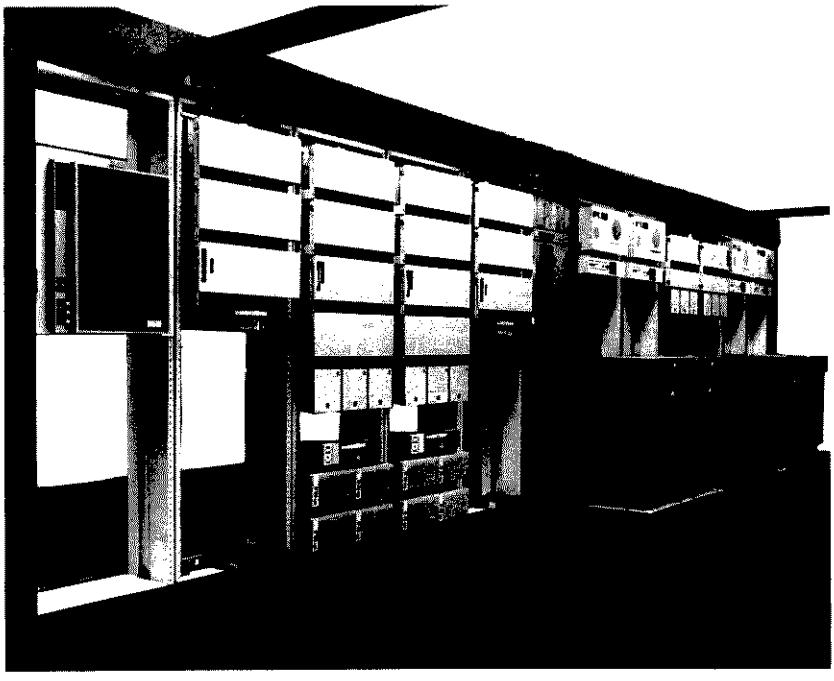


Fig. 6—SPC No. 1B Processor.

units consist of *BELLPAC* mounting hardware. The printed wiring backplane consists of battery and ground layers. Other connections between circuit pack locations are machine wrapped (see Fig. 10). All the circuit packs used in the PSI frame design are of the double-sided, rigid construction equipped with 200-pin connecters (see Fig. 11).

The PSI is duplicated in the SPC 1B. Each frame is associated with and mounted adjacent to a 3B20D Processor control unit frame (see Fig. 6). Connectorized switchboard cables interconnect the PSI frame to the CU frame, the Peripheral Control (PC) frame, and to existing TSPS frames.

#### *6.2.3 Maintenance Control Center*

In the TSPS No. 1B Maintenance Control Center, three bays accommodate the craft-machine interface and related data sets associated with maintenance and administration functions (see Fig. 7).

The Local Maintenance Position (LMP), in two bays, accommodates a shelf-mounted maintenance terminal, a receive-only printer, and data sets for the Switching Control System and the remote field update system[7] interfaces. The Recent Change/Verify position, in one bay, accommodates a shelf-mounted terminal, and data sets associated with

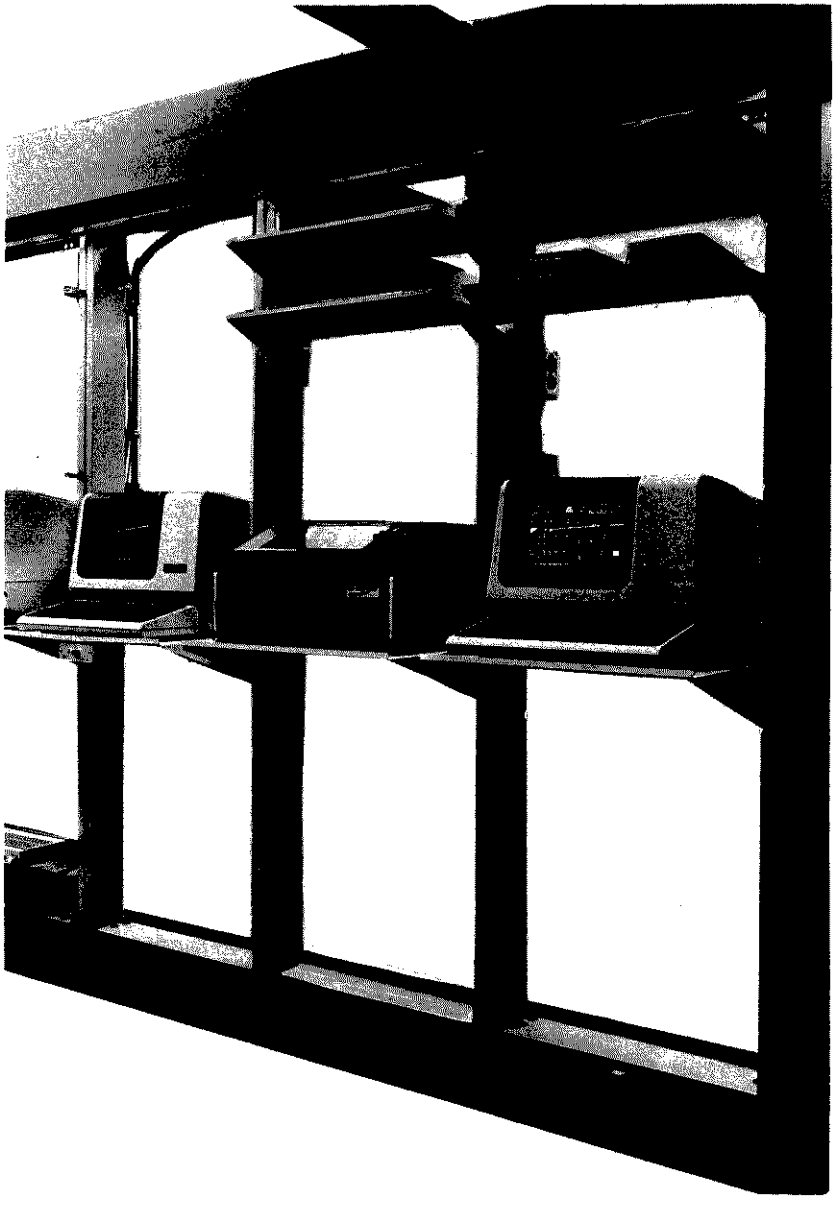


Fig. 7—Local maintenance position.

the remote-maintenance monitor interface and the optional base-unit belt-line maintenance TTY. It also contains a pedestal-mounted auxiliary printer. The LMP bays are mounted adjacent to the existing TSPS No. 1 Control Display and Test frame to facilitate maintenance.

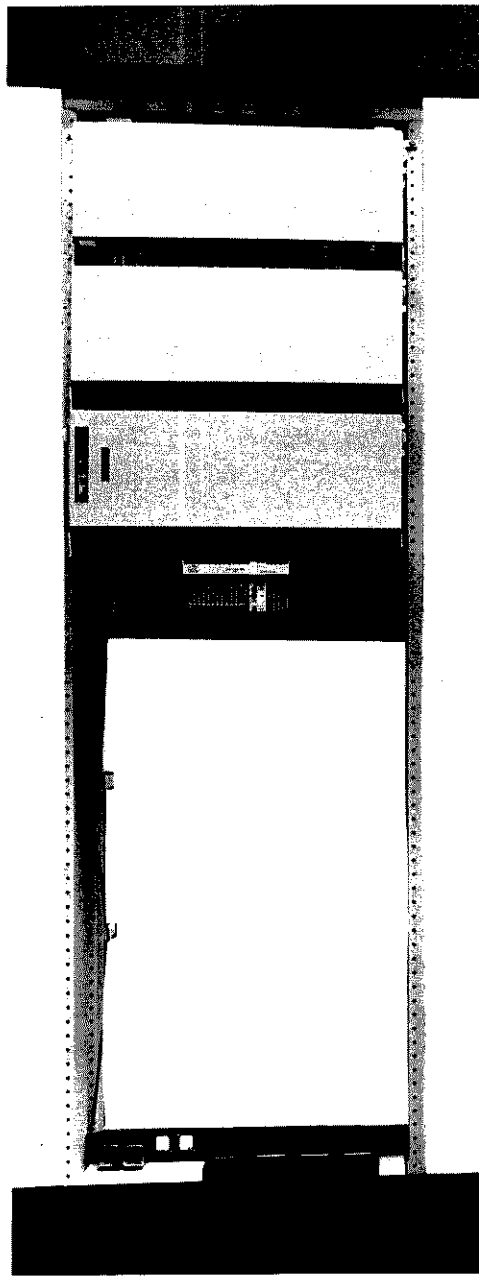


Fig. 8—Peripheral System Interface frame.

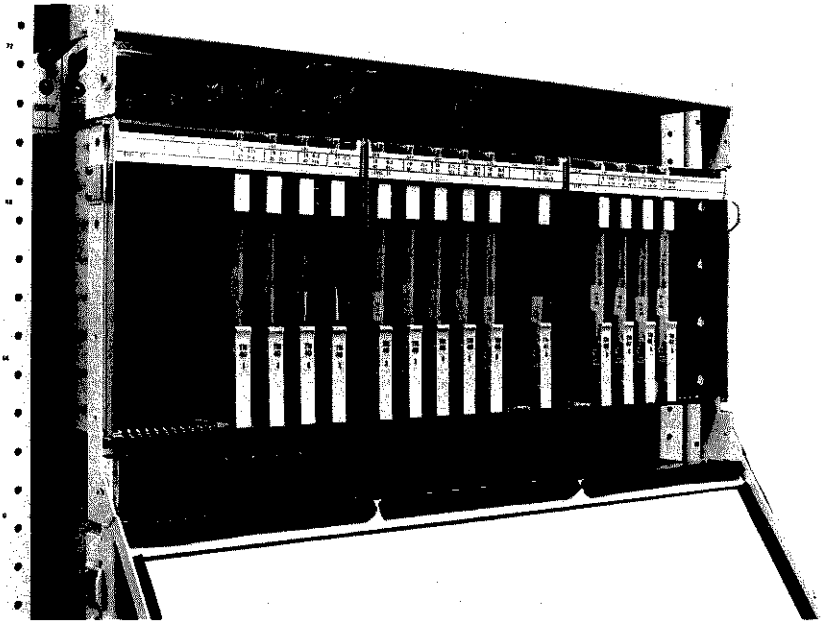


Fig. 9—Peripheral System Interface frame logic unit with circuit packs.

#### 6.2.4 Power

A specially designed power distributing frame distributes central office battery to the various 3B20D Processor frames using one or more pairs of connectorized 10-gauge or 12-gauge feeders. The MHD frame requires a larger pair (4-gauge) for the 30M-byte disk drive. The PSI frames obtain central office battery from the existing TSPS power distributing frames.

The MHD frame also requires 208V single-phase ac for the normal operation of the 300M-byte disk transports. Maintenance terminals, printers, and data sets obtain their protected or essential 110V ac from the existing TSPS ac power distributing unit.

#### 6.2.5 Bus interconnections

The TSPS peripheral buses are duplicated for reliability. Each PSI frame has access to each of the duplicated buses. Connectorized switchboard bus cables are provided between PSI frames and other TSPS frames. As the PSI frame now forms the terminus for these buses, plug-in-terminations are furnished as part of the PSI frame design.

#### 6.2.6 SPC 1B floor plan

The SPC 1B frames are arranged to satisfy the design constraints of the communication links between the individual frames and to comply

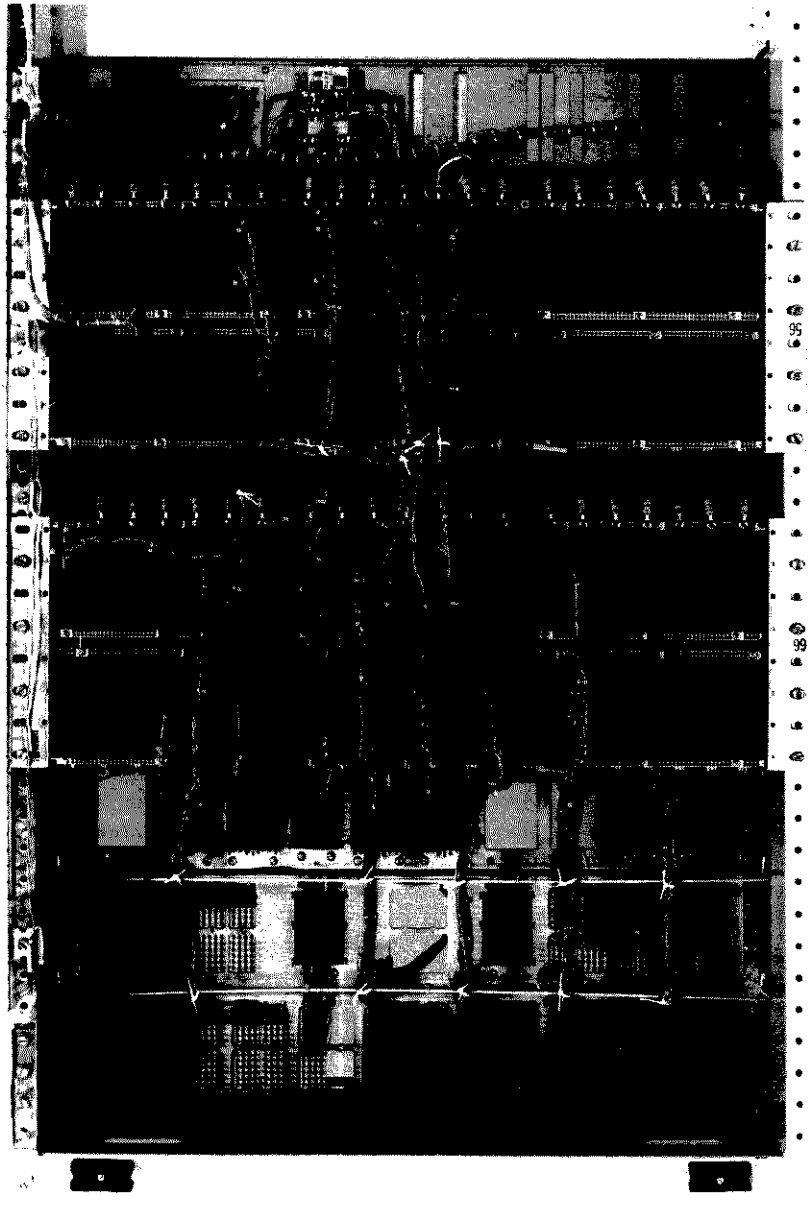


Fig. 10—Peripheral System Interface backplane.

with existing TSPS office equipment arrangements. Figure 12 is an overview of a typical SPC 1B floor plan arrangement designed to meet the constraints described below. The duplicated CU frames must be adjacent to each other and the interconnection from a CU to its

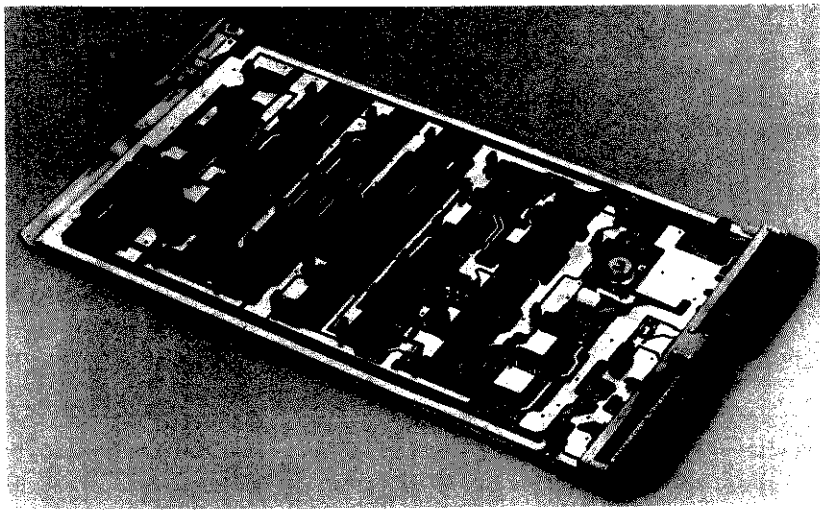


Fig. 11—Typical circuit pack for the Peripheral System Interface frame.

associated PSI frame must be as short as possible. Therefore, the PSIs are located on frames adjacent to the CUs. The MHD frames are located near their associated DFC units, which are mounted on adjacent PC frames. For maintenance of the MHDs, the aisle spacing is increased over the standard TSPS spacing. Location of other units such as the tape drive and IOPs is constrained by cable length between the CU and peripherals driven by the IOP. The 3B power distribution frame is centrally located to minimize length of power feeders to each of the other frames.

## VII. PSI MAINTENANCE SOFTWARE

In addition to the maintenance software provided with the 3B20D Processor and the emulated maintenance software carried over from TSPS No. 1, a maintenance software package was developed for TSPS No. 1B. The major functions of this package are:

(*i*) PSI maintenance

(*ii*) Sanity and integrity of application processes

(*iii*) Interface and coordination of integrity-related activities between DMERT and the TSPS application; i.e., initialization, overload control, and processor switch.

These functions have been implemented in the Application Integrity Monitor (AIM) process, a portion of the native-mode software in the TSPS process, and the PSI diagnostic processes. The following is a description of the PSI fault recovery and PSI diagnostics. Descriptions of items (*ii*) and (*iii*) are provided in Ref. 2.

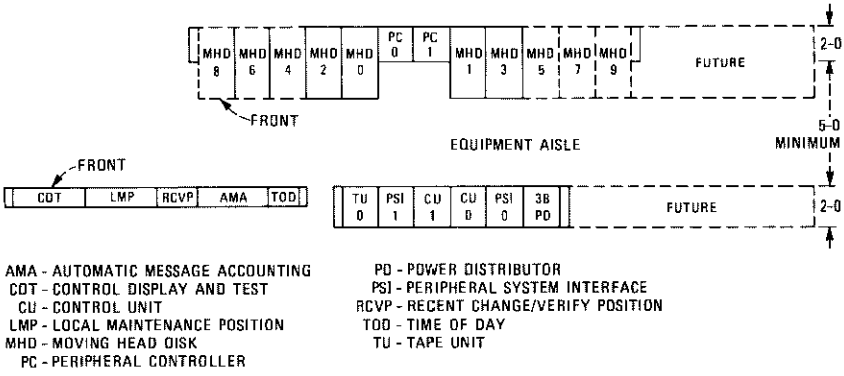



Fig. 12—Typical floor plan for TSPS No. 1B.

The PSI maintenance strategy is based on hardware duplication, fast detection and recovery from service-affecting hardware faults, and the use of deferred diagnostics for the identification of faulty circuit packs. The 3B20D CU and its associated PSI represent a basic switchable entity. This architecture makes the PSI an integral part of the processor. As a result, its maintenance strategy is tightly coupled with the maintenance of the 3B20D Processor.

### *7.1 PSI software data structure integration*

All information about the SPC 1B hardware configuration, status, error counts, and other pertinent data is contained in the Equipment Configuration Database (ECD). The ECD contains a unit control block (UCB) for every equipment entity known or controlled by DMERT. These UCBs are linked data structures that form a hierarchical tree structure representing the physical hardware configuration and the unit interdependencies.[9] This structure links the PSI to the 3B20D hardware architecture and links the PSI to the DMERT maintenance software control structure. The PSI UCB defines the PSI to be a critical CU unit, thus binding the PSI and CU as a single switchable entity. Besides describing the configuration, the UCB also contains error counters and corresponding threshold values for fault recovery, equipment information for service status, and hardware options for diagnostics.

### *7.2 Description of PSI fault detection and recovery*

Detection of PSI faults relies on the 3B20D and PSI self-checking circuits, the TSPS microcode, and the TSPS peripheral-check circuits. Once the fault has been detected, three different fault-recovery programs may be triggered to restore a working system by means of fault resolution and reconfiguration. The fault-recovery program activated

by a fault depends on where the fault is detected. Recovery programs activated may include the TSPS AIM process, the emulated peripheral maintenance program (F-level), or the DMERT Processor Control Process Error-Interrupt Handler (PCPEIH). The following paragraphs will describe the various fault-detection and recovery strategies.

#### 7.2.1 Fault detection

Most PSI faults are detected by the PSI internal-checking circuits. These errors activate an error-interrupt lead to the 3B20D Processor, which is connected to a processor interrupt source register (ISR) bit. The fault-recovery program triggered by this fault depends on what process is active at the time of the error. A fault that occurs during execution of an emulated PSI peripheral order is detected by the TSPS microcode. The microcode generates a maintenance interrupt (F-level) to the TSPS process. The TSPS F-level program, after establishing whether the fault is transient or hard (e.g., after retry of the failing order), informs AIM of the failure by sending a fault event to the AIM process with a fault code indicating whether the fault is transient or hard.

Faults that occur during execution of native-mode PSI instructions and faults that are detected by the 3B/PSI clock-check circuits, cause an ISR bit to be set and result in the dispatching of AIM at the PSI Error-Interrupt Handler (PSIEIH) entry. For these fault types PSIEIH may interrupt and fault the running process to inform it of the failure of the PSI I/O instruction. The decision to fault the running process is made on the basis of its identity. If the running process is TSPS or the PSI Diagnostic Driver (PSIDGDR), it will be faulted. Since no other processes perform I/O with the PSI, there is no need to fault them. Therefore, they will be allowed to continue execution from the point of interrupt after completion of processing by PSIEIH.

Another class of faults are those that are detected by the PSI peripheral order-checking circuits. In most cases this type of fault is a TSPS peripheral-unit fault, but some may be faults in a PSI driver or receiver that are indistinguishable from faults in the TSPS peripheral unit. For these faults, a retry of the failing order through the standby 3B20D CU/PSI is performed. If the off-line retry succeeds, TSPS informs AIM of a PSI fault through a fault code. If the retry fails, TSPS recovers from the peripheral-unit hardware fault by switching TSPS peripheral units.

Finally, certain types of faults in the PSI or the ACHI will result in parity errors on the processor CCIO bus. This type of error appears as a channel error to the CU by activating an ISR. The ISR activates the DMERT fault-recovery process PCPEIH.[10] PCPEIH will then fault the interrupted process. When such a fault occurs while the TSPS

process or the PSIDGDR process is running, it will send a fault to AIM. AIM will reinitialize the PSI and ACHI, and make a reconfiguration decision.

#### 7.2.2 Fault recovery

Once the fault has been detected and resolved to the PSI, it is the responsibility of the AIM process to perform the required recovery action. If the PSI fault is a transient fault, and the PSI/ACHI reinitialization completes successfully, AIM calls DMERT configuration manager functions, which cause error counters in the PSI UCB to be incremented and compared with threshold values. If either counter exceeds its threshold value, removal of the faulty PSI is requested by means of a processor switch. In the case of a hard error, or if the PSI/ACHI reinitialization fails, no error counting is done, and a processor switch is requested immediately. The processor switch is performed by a DMERT service routine.

#### 7.2.3 Off-line PSI fault-detection and recovery

AIM also periodically exercises the off-line PSI. If the exercise fails, the unit is placed out-of-service and diagnostics are scheduled.

### 7.3 PSI initialization

The PSI must be initialized following a PSI power up, as part of a processor switch, and for certain levels of software initialization in TSPS and DMERT. Initialization of the PSI includes the selection of the Scanner Answer Bus, clearing of the PSI maintenance register, and clearing of the Application Channel Interface (ACHI).

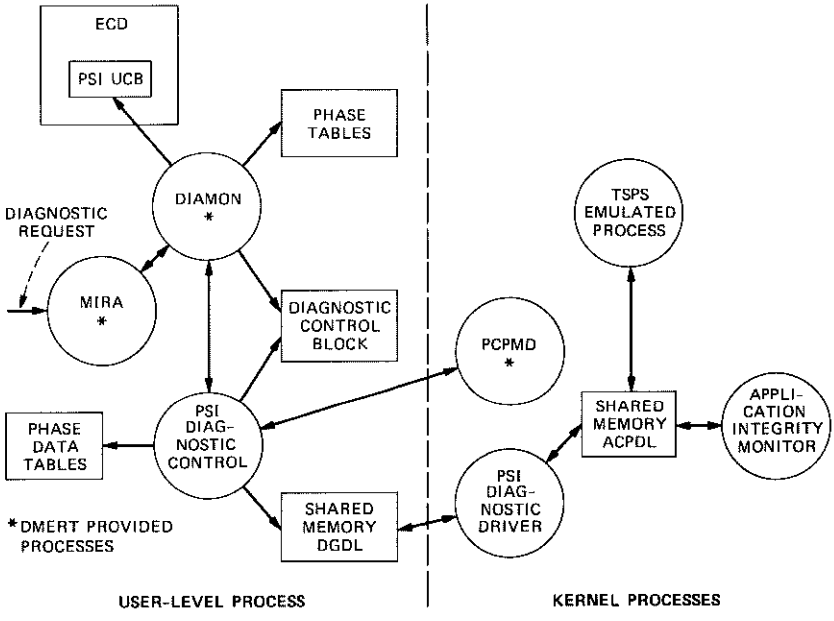
### 7.4 PSI diagnostic

Diagnostics for the 3B20D CU and the TSPS microcode are included in the common DMERT maintenance package. The SPC 1B maintenance package contains the PSI diagnostic.

#### 7.4.1 Operation under DMERT diagnostic control structure

Since the PSI is tightly coupled to the 3B20D Processor and forms a single switchable entity with a 3B20D CU, the PSI diagnostic is designed to operate under the DMERT diagnostic control structure. Diagnostic results are handled by an output message process called the "spooler." A simplified block diagram of this overall software structure is shown in Fig. 13. The PSI diagnostic is initiated and executed under control of the maintenance input request administrator (MIRA) and the diagnostic monitor (DIAMON) process.[11]

Elements of the PSI diagnostic which are unique to the PSI are the Diagnostic Control Process (PSIDC), a Diagnostic Phase Table




ACPDL - APPLICATION CONTROL PROCESS DATA LIBRARY
DGDL - DIAGNOSTIC DATA LIBRARY
DIAMON - DIAGNOSTIC MONITOR
DMERT - DUPLEX MULTI-ENVIRONMENT REAL-TIME
ECD - EQUIPMENT CONFIGURATION DATABASE
MIRA - MAINTENANCE INPUT REQUEST ADMINISTRATOR
PCPMD - PROCESSOR CONTROL PROCESS MAINTENANCE DRIVER
PSI - PERIPHERAL SYSTEM INTERFACE
TSPS - TRAFFIC SERVICE POSITION SYSTEM
UCB - UNIT CONTROL BLOCK

Fig. 13—Maintenance software structure for the Peripheral System Interface.

(DPT), phase data tables (one per phase), task routines, and off-line test routines. Although unique for the PSI application, the implementation of the above diagnostic structure is similar to other 3B20D control unit diagnostics, and a detailed description of the architecture can be found in Ref. 11. Figure 13 provides a high level description of the diagnostic control structure.

### *7.4.2 Off-line execution*

Since the PSI and 3B20D Processor form a single switchable entity, the active processor does not have direct access to the off-line PSI. Therefore, most of the PSI must be diagnosed by test routines which run in the off-line 3B20D Processor.

The off-line tests are divided into two categories of functional tests. Internal PSI functional tests are those which exercise the PSI register logic and sequencer circuits. These tests do not require access to TSPS peripherals and are executed with the I/O inhibit lead activated. The

I/O inhibit lead disables all outpulsing from the PSI to TSPS peripherals. External PSI tests are those that diagnose the drivers and receiver circuits that interface directly with the TSPS periphery. All off-line tests are loaded and executed under the control of the PSI Diagnostic Control (PSIDIAGC) Process with the aid of DMERT library routines that handle communication with the off-line processor. These library routines are provided by the Processor Control Process Maintenance Driver (PCPMD) as described in Refs. 10 and 11.

#### *7.4.3 On-line execution*

Some diagnostic routines must also run on the on-line 3B20D Processor. These routines include the PSI power control tests, the TSPS peripheral bus power tests, and the on-line retries.

The PSI +5V power and the TSPS peripheral bus power are controlled and monitored by a 3B20D common system Scanner and Signal Distributor (SC/SD). The off-line 3B20D Processor cannot communicate with the SC/SD, therefore the on-line processor must perform these tests. The peripheral bus (CPDB and PUAB) power monitoring and control are also connected to the SC/SD. Diagnostic tests of these controls are performed from the on-line processor.

On-line retries of failing off-line tests are performed when it becomes necessary to isolate a fault between the PSI and a TSPS peripheral unit. This situation occurs in the case of a PSI bus driver fault or a fault in a corresponding receiver of a peripheral unit.

#### *7.4.4 PSI diagnostic tests*

The PSI diagnostic test phases are designed to test the hardware in a layered approach such that initial tests start with data bus integrity between the 3B20D and PSI, simple PSI and 3B20D handshaking tests, and then progressing deeper into the PSI hardware until all peripheral driver and receivers have been tested. The test phases of the PSI are divided into internal logic circuit test phases and peripheral driver and receiver test phases. In the following sections the diagnostic test phases are described to show how the layered diagnostic design was implemented.

***7.4.4.1 Internal PSI diagnostic phases.*** The internal diagnostic phases are responsible for detecting faults in the PSI sequencer, registers, and check circuits. These tests are executed with the peripheral bus drivers and receivers disabled to avoid interference with TSPS peripherals. The internal phases can be divided into three sets of tests. The first set of tests relies on maintenance control circuitry to test the basic data paths and ACHI/PSI communication protocols. This phase of testing requires minimal sequencer activity by the PSI, thus providing a high degree of circuit pack resolution by limiting the number of

circuits under test. The second set of internal tests are the tests of the circuit operational functions. All PSI commands are executed to verify proper operation. During these tests error checks are ignored and the commands are tested for proper operation by checking data results. Finally, the remaining test phases of the internal tests are designed to test all internal error detection and matching circuits. Completion of the internal tests provides coverage over most of the PSI logic circuits.

*7.4.4.2 External PSI diagnostic phases.* After the PSI internal circuitry has been tested, the last layer of circuitry, the peripheral bus transmitters and receivers, must be tested.

During these tests the PSI communicates with TSPS peripherals in order to verify operation of drivers and receivers. Therefore, the PSI diagnostic must have access to the equipage and status tables of the TSPS peripherals to check on availability of particular peripheral units. The PSI diagnostic control process provides this access by sharing the memory space of the TSPS emulated process to directly access the office data and status tables for the peripheral units.

Another capability required of the external bus testing is the ability to resolve faults to the PSI or to the peripheral unit used in performing the test. This capability is provided by a special-purpose, on-line diagnostic driver called the PSI Diagnostic Driver (PSIDGDR), which executes the same tests on the on-line processor that were executed in the off-line processor. If the off-line results of the PSI indicated some-test-failed (STF), the on-line test is executed to determine if the fault truly implicated the off-line PSI. The PSI diagnostic control then matches the off-line and on-line results. If the results match, the TSPS peripheral unit is at fault; if the results do not match, the off-line PSI is assumed to be faulty.

### *7.4.5 Isolation from TSPS process*

The structure of the TSPS No. 1B software architecture creates a time-shared environment in which diagnostic execution is performed in time segments that are multiplexed by the DMERT operating system with the TSPS kernel process and other *UNIX** processes. This creates a potential for interference between segments and also creates the possibility of simultaneous execution of an off-line diagnostic routine and on-line TSPS process peripheral orders. Potential interference in these areas requires that the diagnostic control be designed to eliminate interaction. Therefore, a number of capabilities have been developed to provide isolation between the PSI diagnostic and the TSPS emulated process.

* Trademark of Bell Laboratories.

*7.4.5.1 I/O inhibit control.* The 3B20D Processor provides an input/output inhibit control lead on the CCIO bus. This lead is active on the off-line processor and is used by CU I/O channels to block transmission of control signals to the peripheral units. In the case of the ACHI/PSI, the I/O inhibit lead is used to inhibit outpulsing by the PSI drivers. The purpose of the lead is to protect the TSPS peripherals from a faulty PSI sequencer and to allow the diagnostic to execute peripheral orders in the PSI without interfering with the TSPS peripherals.

*7.4.5.2 PSI diagnostic-TSPS process synchronization.* The external PSI tests must utilize the TSPS periphery to test the PSI interface drivers and receivers. Therefore, to ensure that the diagnostic peripheral tests do not interfere with on-line TSPS activity, the PSI diagnostic will be synchronized with the TSPS process. No off-line test routines run while the TSPS emulated process is running. This capability is provided by DMERT library routines, which execute programs in the off-line processor. The library routine checks the TSPS J-level timer (which is generated by the 3B20D clock) and will only initiate execution in the off-line processor when a minimum of 2.5 ms remains prior to the next J-level interrupt. In addition, the routine will retain control of the 3B20D Processor and inhibit all I/O interrupts, timed interrupts, and DMA activity. The 3B20D Processor will not return to TSPS base level during this interval.

## VIII. SUMMARY

A special hardware unit, the Peripheral System Interface, was developed to interface the existing TSPS periphery with the 3B20D Processor. The addition of the PSI required a corresponding development of new maintenance software to integrate the 3B20D maintenance strategy with the TSPS No. 1B maintenance strategy. This article has described the constraints imposed by the differences in technology, and the hardware and corresponding maintenance software required to integrate the 3B20D processor into the TSPS system.

## IX. ACKNOWLEDGMENTS

Many individuals have contributed in a significant way to the design and implementation of the PSI and its maintenance software. In particular, the authors wish to acknowledge the contributions of D. A. Peterson and D. J. Kloc for their part in developing the PSI hardware, and W. J. Proetta for his contributions to the PSI physical design. The following people contributed to the development of the PSI maintenance software: D. L. Brown, K. R. Kulhanek, and J. J. Labut for the PSI diagnostic, E. S. Sachs and M. D. Soneru for the PSI maintenance software.

## REFERENCES


1. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.
2. R. J. Gill, G. J. Kujawinski, and E. H. Stredde, "Traffic Service Position System No. 1B: Real-Time Architecture Utilizing the DMERT Operating System," B.S.T.J., this issue.
3. J. C. Dalby, D. Van Haften, and L. A. Weber, "Traffic Service Position System No. 1B: Retrofitting to TSPS No. 1B," B.S.T.J., this issue.
4. G. R. Durney, et al, "Stored Program Control No. 1A," B.S.T.J., *49*, No. 10 (December 1970), pp. 2445–2507.
5. M. W. Rolund, J. T. Beckett, and D. A. Harms, "The 3B20D Processor & DMERT Operating System: 3B20D Central Processing Unit," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 191–206.
6. J. J. Bodnar, J. R. Daino, and K. A. VanderMeulen, "Traffic Service Position System No. 1B: Switching Control Center System Interface," B.S.T.J., this issue.
7. T. G. Hack, T. Huang, and L. C. Stecher, "Traffic Service Position System No. 1B: Software Development System," B.S.T.J., this issue.
8. S. H. Kulpa, J. M. Brown, and A. W. Fulton, "The 3B20D Processor & DMERT Operating System: 3B20D Packaging and Technology," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 221–34.
9. R. H. Yacobellis, J. H. Miller, B. G. Niedfeldt, and S. S. Weber, "The 3B20D Processor & DMERT Operating System: Field Administration Subsystems" B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 323–39.
10. R. C. Hansen, R. W. Peterson, and N. O. Whittington, "The 3B20D Processor & DMERT Operating System: Fault Detection and Recovery," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 349–65.
11. J. L. Quinn, R. L. Engram and F. M. Goetz, "The 3B20D Processor & DMERT Operating System: Diagnostic Tests and Control Software," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 367–81.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Software Development System

By T. G. HACK, T. HUANG, and L. C. STECHER

(Manuscript received June 30, 1982)

*This article describes the Software Development System for the Traffic Service Position System No. 1B (TSPS No. 1B). It discusses the modern, multicomputer software generation and test facilities that were provided to concurrently support both C-language and emulated, assembly-level software development. The computing environment, software generation and test tools, and standard development process that were developed for the TSPS No. 1B provide a rich, robust programming environment for future network operator services.*

## I. INTRODUCTION

The development and testing of the software for the Traffic Service Position System No. 1B (TSPS No. 1B) was a complex undertaking. In addition to emulating the existing TSPS No. 1 assembly-level program, 3B20 Duplex Processor (3B20D) native-mode (C-language) software was developed to interface to the Duplex Multi-Environment Real Time (DMERT) operating system and to provide the necessary system integrity functions for the TSPS No. 1B.[1,2]

To support software development and testing in this mixed emulation/native mode, it was recognized early in the project that a robust Software Development System (SDS) and a rigorous set of standard software development procedures (or methodology) were necessary. Thus, basic requirements for tools, documentation standards, and control procedures were established for the software development environment for the TSPS No. 1B. These requirements specified that:

(*i*) A simple, interactive user interface to the SDS must be developed. Where possible, commands and procedures for emulation or

native-mode software development should be the same to minimize the complexity of the programming task.

(*ii*) The SDS for emulation-mode software development must be upwardly compatible from the existing TSPS No. 1 SDS. In parallel with the development of TSPS No. 1B, new generic features were being developed and deployed for TSPS No. 1 (such as Automated Calling Card Service).[3] To offer universal service, these features would also have to be concurrently emulated on the TSPS No. 1B.

(*iii*) Complete software change procedures and tools must be established to support the parallel development of TSPS No. 1 and TSPS No. 1B. These procedures and tools would have to support the early development phase of the project when programmers were initially developing code, as well as later phases of the project when system testing is converging to a certified, production software release.

(*iv*) Finally, a set of test facilities was needed to support the integration and system testing of the software. Special tools would be required to test the emulated, assembly programs, as well as the high-level, native-mode software executing under the DMERT operating system.

The following sections of this article describe how the TSPS No. 1B Software Development System was implemented to meet these requirements.

## II. COMPUTING ENVIRONMENT FOR TSPS SOFTWARE DEVELOPMENT

Before discussing the specific software generation tools and test facilities of the TSPS No. 1B SDS, this section describes the computing environment for TSPS development and gives an overview of the development system.

### *2.1 Overview*

TSPS No. 1B software development is supported by a multiprocessor computing system. The system consists of five computers: one remote IBM 3033 processor located at Bell Laboratories in Columbus, Ohio; and four Digital Equipment Corporation (DEC) PDP-11/70 minicomputers co-located with the TSPS development organization at Bell Laboratories in Naperville, Illinois. A variety of data links support interprocessor communications. Figure 1* illustrates the configuration.

The IBM processor and the two PDP-11/70 computers (labeled PSS1 and PSS2) constitute the Programmer Support System (PSS). It is on these systems that source modification, load building, and

* Acronyms and abbreviations used in the figures and text are defined at the back of this Journal.

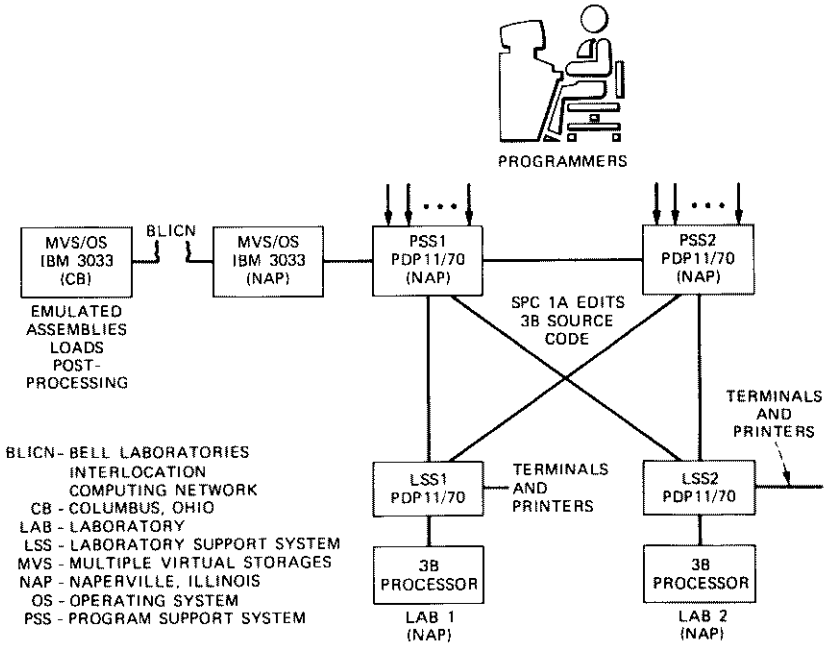



Fig. 1—Software Development System for TSPS No. 1B.

software administration are done. The remaining two PDP-11/70 computers reside in the TSPS system laboratories and are part of the Laboratory Support System (LSS). Each is paired with a 3B20D Processor and is used to support developer, integration, and system testing for 3B20D-based generic programs.

### 2.2 Local interactive development environment

TSPS programmers use the PSS1 and PSS2 computers to modify source files and create executable versions of software. These computers run the *UNIX** time-sharing operating system, which provides a general-purpose, multi-user, interactive development environment.[4] The *UNIX* operating system has a number of characteristics that make it attractive for software development. First, it is extremely easy to learn and use. The command-line interpreter (the "shell"), the interactive editor, and the hierarchical file system are particularly simple in nature. Second, the shell and C-language programming environment provide powerful and flexible facilities to create and combine software tools to support the generation and administration of software. Many of the tools used in TSPS have been created in this

* Trademark of Bell Laboratories.

environment. Finally, the Programmer's Workbench facility,[5] available with the *UNIX* operating system, offers:

(*i*) A sophisticated document preparation system to prepare documentation supporting the development process

(*ii*) A Remote Job Entry (RJE) subsystem that transmits jobs to other computing systems and returns output to appropriate users

(*iii*) A complete Source Code Control System (SCCS) for controlling and maintaining multiple versions of source code.

The software development is partitioned so that C-language developers use the PSS1 system and emulated, assembly-language developers use the PSS2 system.* System load building makes use of both PSS1 and PSS2. C-language development is solely supported on the PSS1 system. Emulated development, on the other hand, is accomplished through the interaction of the PSS2 system and the remote IBM 3033 processor.

### *2.3 Remote batch environment for emulated development*

TSPS emulated software is developed in a part interactive, part batch-oriented computing environment. Much of the SDS for emulated code has been upgraded and carried over from the previously existing batch-oriented SDS for the TSPS No. 1.[6] The PSS2 system was added to the previously existing SDS for the TSPS No. 1 and provides the front-end user interface to the SDS for emulated programs. When the PSS2 system was added, the programmers gained a modern set of interactive commands that allow them to initiate the various software generation jobs for emulated code. Developers log into PSS2 and modify "edit files" that contain edit statements to be applied against the emulated source files, which are stored on the remote IBM system. The application of edits and all software-generation steps run under the Multiple Virtual Storages (MVS) operating system in a batch mode on the IBM processor.

For example, once a programmer has finished modifying the edit file on the PSS2 system, a single swap command can be issued to construct a Job Control Language (JCL) script designed to invoke, through the *UNIX* RJE facility, a remote emulated assembly on the IBM processor. The job file is sent via the Bell Laboratories Interlocation Computing Network and queued to be executed on the IBM/MVS system at Columbus. When the assembly is complete, the listing file is printed locally. Commands similar to swap are provided for link editing and other functions. These commands are described in more detail in Section III.

* Subsequent references to "emulated, assembly" language will be shortened to "emulated" language.

From the programmer's point of view, all activity takes place on the PSS2 system. The commands that create the remote batch jobs look very much like local commands. Since required output data (tapes, files, or listings) are returned to the local users, the use of the IBM/MVS system is transparent to the user.

### *2.4 Local interactive environment for C-Language development*

The PSS1 system provides the computing environment for C-language program development. It can support as many as 36 simultaneous interactive users, although prime-time usage is typically about 25 users. For the most part, terminals are connected through a dial-up arrangement. There are, however, a few "hard-wired" terminal connections into the TSPS System Laboratories to guarantee access to programmers using laboratory test facilities.

Software tools that support C-language development on the PSS1 system include: the text editor, the 3B20D Software Generation System (3BSGS), the Change Management System (CMS), and the Source Code Control System. The text editor is used to modify C-language source files. The 3BSGS is a cross-compilation and link-editing system that generates executable files for the 3B20D Processor from source code. CMS and SCCS control and track the development process and provide the mechanisms to maintain multiple versions of source and object files.

From the programmer's perspective, the development scenario is quite simple. A source file is retrieved from SCCS, modified with the editor, compiled to produce an executable file using the 3BSGS, and data-linked to the Laboratory Support System computers for testing in the TSPS System Laboratories. With the exception of testing, all work takes place interactively on the PSS1 system.

### *2.5 Laboratory support processor*

The support processor used in the TSPS Laboratory Support System (LSS) is a PDP-11/70 computer running under the *UNIX* operating system. The flexible environment of the *UNIX* operating system can support multiple testers simultaneously. For TSPS development, there are two system laboratories that are used to create the laboratory environment for field support and to enable realistic system testing of TSPS software. Section V describes the LSS in more detail.

### *2.6 Networking facilities*

As we can see in Fig. 1, the processors within the TSPS computing environment are interconnected by a variety of data links that create a reliable, secure computer network. Each PSS computer is connected by two separate links to the other PSS computer and the two LSS

computers. One of the connections is a 9.6-kb link designed to be used with the cu command. (The cu command gives a user on one system the appearance of being logged into another.) Commands can be executed on the remote system and files can be transferred in both directions. The second type of connection is a 56-kb link that uses DEC DMC-11 hardware as the primary intercomputer file transfer mechanism. It can support effective file transfer rates of up to 2000 bytes/second. The remaining data link is a component of the Bell Laboratories Interlocation Computing Network connecting the Naperville and Columbus locations. The primary connection is established through 56-kb private lines between two IBM 3033 processors running the MVS operating system, the remote IBM system at Columbus, and a local interface system. The two TSPS PSS computers are connected to the local IBM/MVS system with special-purpose 9.6-kb hardware. Using the RJE subsystem, batch jobs can be sent through the local IBM/MVS system to the IBM/MVS system at Columbus. The capability also exists to return files to the PSS system or generate local output tapes or listings.

## III. SOFTWARE GENERATION TOOLS

### *3.1 Overview*

The primary function of a software generation system is to transform symbolic source language statements into a format executable by the target processor. For TSPS No. 1B, the target processor is the 3B20D Processor. A software generation system is usually a collection of tools that perform this transformation in phases. The compilation (or assembly) phase translates the source module containing symbolic instruction and data statements into machine-readable form, called a "relocatable object file." This file also contains symbol definitions and relocation information to be used in the linking-loading phase. The linking-loading phase enables separately compiled modules (relocatable object files) to be combined into a single executable unit, called a "load file." A final process-loading phase is used to transform the load file to a process-file format.

### *3.2 Emulated software generation*

During the development of the TSPS No. 1B, special-purpose microcode was written for the 3B20D processor to emulate the Stored Program Control 1A (SPC 1A) assembly language used in TSPS No. 1. This allowed a large portion of the existing TSPS software to be transported to the 3B20D Processor without modification. In this approach, the emulated source statements were kept identical to the SPC 1A assembly-language statements. However, the object form (the machine-equivalent form) was modified to take advantage of the

powerful micro-instruction set of the 3B20D. This was done to maximize the real-time processing capability of the TSPS No. 1B system.

To ensure commonality between TSPS No. 1 and TSPS No. 1B assembly-level programs, the existing SPC 1A assembler and loader were retained. The new 3B20D object format for the TSPS No. 1B was produced by two new software generation tools—a "load post-processor" and a "listing post-processor." The load post-processor transforms an SPC 1A load file into a 3B20D load file, and the listing post-processor reformats the assembly listing to reflect the new, emulated instruction formats. In addition, many of the existing SPC 1A SDS tools were retained and significantly upgraded by the addition of the PSS2 system. Figure 2 shows the steps involved in emulated software generation.

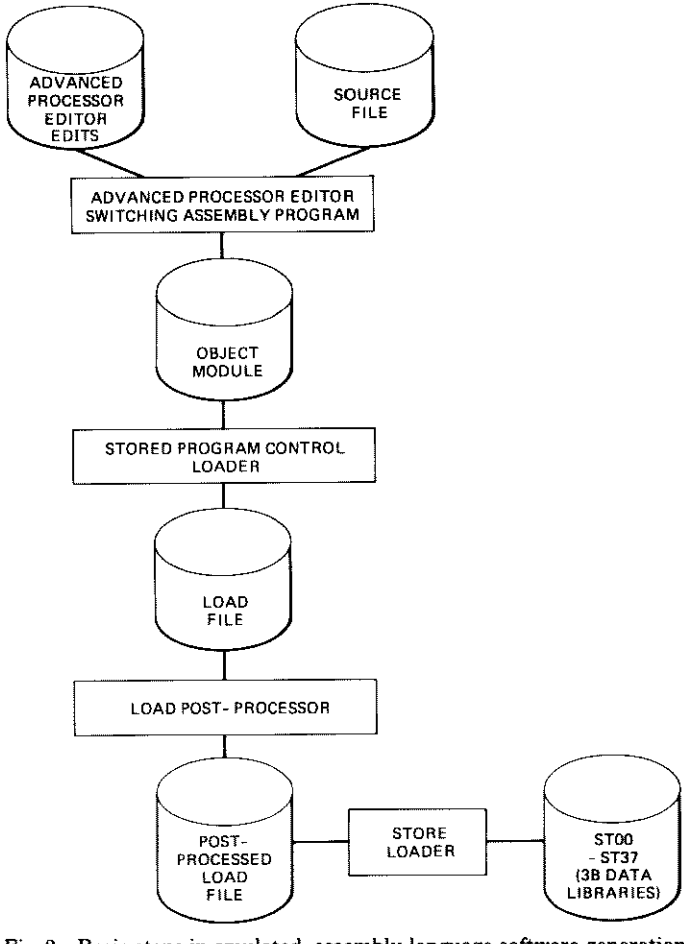



Fig. 2—Basic steps in emulated, assembly-language software generation.

### 3.2.1 The TSPS assembler and loader

The assembly process for TSPS emulated code combines both an editing and an assembly operation. Editor statements are prepared on the PSS2 system and transmitted to the remote IBM/MVS system. These editor statements are then processed and applied to the emulated source file by the Advanced Processor Editor (APE) running under MVS. APE is a simple, line-oriented editor with the basic "insert," "replace," and "delete" functions. It is specifically designed to work with the assembler, passing the edited source as input to the assembly process.

The assembly operation is performed by the powerful SPC-SWAP (Switching Assembly Program) assembler,[7,8] which also runs under MVS. Besides performing the standard source-to-object conversion, it has a sophisticated macro capability and a variety of useful pseudo-operations for controlling listing format and establishing symbol definitions. In addition, there is a mechanism for creating and maintaining special-purpose "library" files. Library files of symbol or macro definitions can be created in one SPC-SWAP run, then later accessed by subsequent source module assemblies for the purpose of symbol or macro resolution. This is not only a convenient mechanism for sharing global symbol definitions between source modules, but also allows a single source file to be assembled in different environments, resulting in different object modules. The latter technique is used in TSPS to develop code for multiple generics from a single source file and is discussed further in Section 4.1.

To perform an assembly, TSPS developers invoke the swap command on the PSS2 system. This command creates a Job Control Language script, which is executed on the remote IBM/MVS system. The primary outputs of SPC-SWAP are an object module, which is retained on the IBM file system, and an assembly listing, which is printed locally.

After the assembly process, the next step in producing an executable file for the 3B20D Processor is to combine the SPC-SWAP object modules for a given generic using the SPC loader. This loader uses a special set of control statements that specify what areas of emulated address space are available for loading object modules, which modules are to be loaded, and, optionally, what their load addresses are. The output of the SPC loader is a file or magnetic tape containing the relocated, fully bound, generic load file. At this point, however, the load file is suitable only for loading into an SPC 1A used in TSPS No. 1. Further post-processing (described in Section 3.2.2) is then done to produce a 3B20D-compatible format.

In addition to a full generic load, the SPC loader also has the capability to produce what is called a "partial-load" file. During the

generation of a full generic load, the SPC loader is directed to create a special file on the IBM/MVS system called a HISTORY. This HISTORY file contains the information necessary to completely characterize the generated load file. This includes the load image itself; the names, starting addresses, and sizes of the object modules; free space; and all external symbol definitions and references. Once a HISTORY file has been created, any TSPS developer on the PSS2 system can reassemble a selected subset of generic source modules and issue the ppload command to build a partial load on the IBM/MVS system. The items input to the ppload command identify the HISTORY file and the reassembled object modules. The partial-load process constructs a new load image from these input files, then compares it to the image contained in the HISTORY file. The final output is a magnetic tape containing the changed instruction or data words. This tape, after post-processing, can be taken into the TSPS system laboratory and overlaid on the full generic load file. The partial-load capability is used extensively in the early stages of TSPS generic development. It is a very flexible and convenient means of creating a developer's private software version for testing.

#### *3.2.2 Post-processing— format conversion from SPC 1A to 3B20D format*

To convert from SPC 1A format to 3B20D format, a load post-processor was developed to transform the output of the existing SPC loader into the new 3B20D encoding. In addition, a listing post-processor was developed to convert program listings to account for the changes in the order encoding and also to provide address and data fields in hexadecimal, which is the native number base for the 3B20D.

The load post-processor accepts as input a HISTORY file produced by the SPC loader. The HISTORY file contains not only the actual load image but also specifies whether an SPC word corresponds to a program or data instruction. Each 40-bit (double-word) SPC program instruction is translated into a 64-bit, 3B20D encoding (two 32-bits words) as required for the emulation.[2] The reformatting that is done depends on the SPC operation code and can completely rearrange the fields in an instruction. The translation algorithm is encoded into a set of common routines that are shared by the load and listing post-processor. This ensures that listing and load translations remain synchronized. Both the load and listing post-processors are run as batch jobs on the IBM/MVS system and are initiated remotely from the PSS2 system.

#### *3.2.3 Creating a DMERT process from an emulated load file*

The TSPS process is a key component of the TSPS No. 1B system since it has primary responsibility for call-processing functions.[2] It is

a DMERT kernel process made up of both C-language and emulated software. The C-language portion is built in the standard fashion using the 3B Software Generation System (see Section 3.3.1). The emulated portion is stored in DMERT data libraries and appears as pure data to the C-language software generation process. The data libraries are stored as separate files on the 3B20D disk and are linked into the TSPS process address space at process creation time. There are 32 data libraries for emulated software, each being one segment (128K bytes) in size.

#### *3.2.4 Overwrite generation*

In the later stages of the development of a generic, rigorous change procedures are introduced to tighten control over changes to the software. This is known as the "frozen" mode of development and begins normally with the start of system testing.[9] In frozen mode, changes in emulated code are applied in overwrite form rather than by full reassembly and linking of source modules. An overwrite consists of just those program and data instructions that are being modified in a program change. Thus, an overwrite tends to be small in size and the impact of the change is local rather than global. Overwrites are usually inserted into a stable program version and tested one at a time in a cumulative fashion. In this way, the generic program evolves in a controlled way with each change being tested before the next is applied.

### *3.3 C-Language software generation*

During the lifetime of the TSPS No. 1 system, all software development was done in assembly language. However, for the introduction of the TSPS No. 1B, as well as for the development of future operator services, significant portions of the new software will be developed in the C programming language. Thus, a new set of software tools was implemented to support the native-mode development process for the TSPS No. 1B.

#### *3.3.1 The 3B Software Generation System*

The 3B Software Generation System (3BSGS) is a collection of software tools available with the 3B20D Processor.[10] These tools transform C-language source code into object files that can be loaded into the 3B20D disk and executed under the DMERT operating system. The main constituents of the 3BSGS are the C compiler (3bcc), the 3B20D assembler (3bas), the 3B20D link editor (3bld), and the 3B20D process loader (3bldp). These programs are designed to execute in the environment of the *UNIX* operating system.

The C compiler accepts C-language source files as input and trans-

forms them into relocatable object files. The 3B link editor, 3bld, can combine several object files into one by relocating program and data instructions and resolving externally defined symbols. It also provides the mechanism to resolve references to library routines. (Libraries are collections of precompiled modules that typically contain commonly used routines that are shared among many modules.) For TSPS No. 1B, the output of 3bld is passed to the 3B20D process loader for final processing.

The 3B20D process loader, 3bldp, takes as input a collection of relocatable object files produced by 3bcc or 3bld. By using a special-purpose specification file, it constructs an output, called a process file, which has a format appropriate for execution under the DMERT operating system.

#### *3.3.2 The Make Program*

For TSPS No. 1B, a utility program, Make, is used extensively in conjunction with the 3BSGS during the software generation process. The purpose of Make is to automate the construction of a process file when one or more of the source files on which it depends have been modified. The generation process is controlled by a special-purpose description file, frequently referred to as a "makefile."

The advantages of the Make program are that it:

(*i*) Ensures that a process file is constructed correctly in that no commands are accidentally forgotten or specified incorrectly

(*ii*) Regenerates only those files that have been affected by a change

(*iii*) Minimizes the programmer effort in building executable processes.

For these reasons, Make is used extensively in TSPS No. 1B development. For the initial TSPS No. 1B generic, there are approximately 50 makefiles used in building over 300 3B20D processes.

#### *3.3.3 Dependence on DMERT header files and libraries*

The TSPS C-language software runs under the DMERT operating system. The two chief software mechanisms that support communication between the TSPS application and the operating system are header files and libraries. Header files are typically used to define data structures that are shared between processes or shared between source files within a single process. For example, there is a header file that contains data declarations specifying the layout of a DMERT Process Control Block (PCB) for a supervisor or user process. Any source file that references the PCB will include this header file through the "#include" mechanism provided by 3bcc, by which the PCB data declarations are made available to the compilation process. Libraries,

on the other hand, typically contain commonly used functions or routines. References to libraries are resolved by 3bld during the software generation process. Libraries are usually organized on a functional basis. For example, the craft interface library contains functions and routines oriented towards applications involved with the handling of craft input messages or the generation of output messages.

The DMERT operating system contains a large number of header files and libraries to support TSPS development. With each DMERT release, applications receive not only the latest version of DMERT software, but also updated versions of the header files, library files, and the 3BSGS. These are installed on the PSS1 system to support TSPS development.

#### *3.3.4 Linking C and emulated software*

As previously mentioned, the TSPS process contains both C-language and emulated programs. The emulated code is allocated one megaword of the available two-megaword-process address space and contains the bulk of the TSPS call-processing software. Naturally, there is a need to transfer control between emulated and C-language programs. Since these programs require different versions of microcode, a processor "mode switch" is required. Two special assembly-language instructions, cale (call emulated) and smt (switch mode and transfer), were developed to perform this function. Cale is a native-mode instruction that effects a mode switch and a transfer to emulated address space. Smt is an emulated instruction that performs a similar function but in the reverse sense.

From a software-generation perspective, the interface between emulated and native mode is quite simple. Routines called through cale or smt are accessed through transfer vectors. An emulated program, called MADEP, contains a table of transfer vectors, one for each emulated routine called from native mode. To link from a native to an emulated routine, the developer defines a symbol, say etv, equal to the address of the transfer vector in MADEP. (These definitions are kept in a header file dedicated for this purpose.) The emulation routine can then be accessed by an "spcxfr(etv)" function call in a C program. Spcxfr is an IS25 assembly-language routine that executes a cale to effect the transfer.

To transfer from emulated to native mode, the developer defines a global symbol, say ntv, equal to the address of the transfer vector for the desired function. These transfer vectors are forced to specific locations by special control statements in the loader specification file of the TSPS process. Thus, their location is known to the programmer. The transfer is accomplished through the use of the smt instruction.

Although this emulated-native transfer mechanism requires some

manual coordination between emulated and C-language files, this effort is only required when a new routine (to be accessed from the opposite mode) is added. These routines can be modified at will without requiring the referencing programs to be regenerated. Since these special-purpose routines are added infrequently, emulated and C-language development can proceed, for the most part, quite independently.

### *3.4 Overwrite considerations for the C-Language environment*

For C-language software generation, there is no direct analogue to an overwrite for emulated software. C-language frozen development involves full recompilation and link editing of TSPS processes. The effect of this on TSPS development is discussed in Section 4.3.3.

### *3.5 Building a 3B20D disk image*

Thus far, the mechanisms for constructing DMERT processes files have been described. To complete the software generation process, it is necessary to build a 3B20D disk image containing the total collection of process files, special-purpose boot files, and equipment configuration information. To accomplish this, the DMERT operating system utilizes two major data bases: the Equipment Configuration Database (ECD) and the System Generation (SG) database. The ECD contains records describing the characteristics and status of all processor and peripheral hardware. The contents of the ECD specify the current hardware configuration of the system. This hardware status information is placed in a central database, the ECD, to eliminate redundant device information and to provide a unified approach to handling and accessing hardware configuration data. The SG database contains information specifying the structure of the 3B20D disk image as designed by the application. This includes operating system parameters, disk partitioning information, file system names and sizes, names and types of individual files, and the make-up of the operating system boot image.

A DMERT tool, 3bmkdsk, provides the capability to construct a disk image from the ECD and SG databases, DMERT process files, and TSPS process files located on the PSS. This program generates three 1600-bit-per-inch (bpi) magnetic tapes containing 32M bytes of data. The image on tape is used to create a 3B20D disk.

## IV. THE SOFTWARE DEVELOPMENT PROCESS

The development of the TSPS No. 1B with many software tools and literally thousands of files requires a rigorous set of procedures (or methodology) to allow the development of software to proceed in an

orderly and controlled fashion. This section describes the methodology employed for TSPS No. 1B development.

### *4.1 General support environment of TSPS*

At the source-language level, the initial TSPS No. 1B generic program comprises approximately 350 emulated source files and 380 C-language source and header files. In executable form, the emulated software amounts to about 2.3M bytes of program. This excludes scratch data areas and office-dependent data (describing the particular line and trunk arrangements of an office), which also reside in the emulation address space. The C-language source is transformed through software generation tools into over 300 3B20D process and data files.

To control development on this large collection of software modules, official and test versions of both emulated edit files and C-language source files are maintained under the Source Code Control System. SCCS is a collection of programs that can store and retrieve multiple versions of a file in a space-efficient manner. This not only maintains a history of changes but allows the retrieval of various versions of source code that, for example, might represent a recent release, the current official software, software under system test, etc.

In addition to using SCCS for source control, TSPS also makes use of a "featuring" concept. Featuring allows developers to maintain a *single* source file regardless of the number of program generics under development. In featuring, lines added, deleted, or replaced in a source file are bracketed by feature-control directives which, during the assembly of the source file, direct the assembler either to assemble or ignore bracketed source code. The feature-control directives used in TSPS emulated development are:

(*i*) INFOR feature-expression
(*ii*) OUTFOR feature-expression
(*iii*) ENDFOR feature-expression.

Feature-expression is a Boolean combination of feature names. In an assembly, each feature name, and consequently each feature-expression, evaluates to a true or false value. If the feature-expression associated with an INFOR directive is true, the source lines between the INFOR and corresponding ENDFOR are assembled. For OUTFOR, if a feature expression is true, it causes the bracketed source lines not to be assembled. A set of feature expressions is associated with each generic. In this way, a single source file can generate multiple distinct object files, each associated with a particular generic program under development. The featuring syntax used for C software is different, but has the same basic capabilities.

There are two modes of development used in TSPS. The "non-

frozen" mode is the first stage in the development process when most of the new feature software for a generic is written. The intent is to give the programmers as much freedom as possible in writing and testing their code. When most of the new feature development is complete, the "frozen mode" is entered. In this mode, tight control is exercised over changes that are applied to the generic. Each change is documented and tested individually to assure a controlled evolution of the generic software. The sections that follow describe in some detail how development works in these two distinct environments.

### *4.2 Non-frozen-mode methodology*

#### *4.2.1 Emulated development*

***4.2.1.1 Creation and modification of source modules.*** In the non-frozen mode, programmers have a large degree of freedom in modifying existing source files or creating new ones. To create a new source file for emulated program development, the developer simply uses the interactive editor to input emulated statements. The new source file can be assembled, put into executable form by the partial load capability, and tested in the TSPS system laboratories. At the time of the next "base load" (see 4.2.1.3), the source file will be installed on the remote IBM/MVS system where all emulated source files are maintained.

Most non-frozen-mode activity involves modifying APE line edit files under the control and administration of the TSPS Subsystem for Non-Frozen Development (TSPNF).

TSPNF uses a strategy that allows multiple developer teams to work in parallel in a non-interfering way. The hierarchical nature of the file system is used to logically group edit files based on generic name, base-load name, and team name. Thus, several teams may have edit files affecting the same source file. Individual team members can create edit files within the team directories via the TSPNF commands, nfget and nfput. The nfget command is used to retrieve from the TSPNF directory structure, for the purpose of editing, the edit file for a source module corresponding to a particular generic, base load, and team combination. If the TSPNF structure has no edit file for the particular team, nfget retrieves the latest official version of the edit file. The edit file is placed in the user's current directory (work area). Once the file is retrieved, the programmer can modify it as described by interactively adding, deleting, or replacing source lines or feature control directives. Once the edit file has been changed to the programmer's satisfaction, it can be returned to TSPNF for safekeeping by the nfput command.

***4.2.1.2 Generation of private versions for testing.*** The PSS2 interface for emulated program development contains a number of commands

to be used for the generation of IBM Job Control Language (JCL) scripts and for the submission of these scripts to the remote IBM/MVS system for execution. Two of these commands, swap and ppload, play a key role in generating software for testing in the non-frozen mode of development. The swap command is used to generate and submit the JCL for a SPC-SWAP assembly.

The swap command allows the programmer to assemble a new source file that has been created on PSS2 or, in the case of a program whose source is already resident on the IBM/MVS system, to specify the name of an APE edit file to be applied to the program source prior to assembly. Options are provided to retrieve the edit file from the user's current directory, the TSPNF team directory structure, or the official set of line edit files.

Regardless of the source of the input, a *generic-name* parameter is used to establish the proper environment for the assembly. This includes setting up references to the correct SWAP symbol and macro libraries, selecting appropriate feature names, and specifying generic-dependent listing-format options. Typical output files from the swap command are the assembly listing file and an object module. These both reside on the remote IBM file system with the object module placed in a special team partitioned data set (a single data set containing a collection of object modules for a specific development team). The listing file can be printed locally in original or post-processed format.

The ppload command provides the programmer interface to the partial load capability described in Section 3.2.1. This command will generate a partial load run using the HISTORY file from the latest base load (see Section 4.2.1.3) and all of the object modules in a specified set of team data sets. The result is a file on the IBM/MVS system in post-processed format that contains just the program differences introduced by the team object modules. Through the use of an optional parameter, this partial load file can be written to tape. This then provides a convenient mechanism to create a private, team version of software that can be taken into the system laboratory, overlayed on the current official version from the last base load, and tested.

The non-frozen development scenario from the programmer's point of view is quite simple. An edit file for the source file to be modified is retrieved with the nfget command, and changed with the interactive editor. Private developer or team test loads can be built by use of the swap and ppload commands. Once the edits have been tested, the edit file can be returned to the team TSPNF structure with the nfput command. All programmer activity with the exception of the testing itself occurs on the PSS2 system.

*4.2.1.3 Submission and integration of software changes.* Periodically, during the process of developing a new generic, newly developed software is integrated with the latest approved version of the generic program and tested. This process is called the "base load" process because it results in a new, independently certified program load that serves as the base for future development. For a base load, each team leader is responsible for submitting a complete description of the team input. This information includes the list of team edit files to be incorporated into the base load, the features that are completed with this submission, and other appropriate documentation as needed. Using this input, a new software version is constructed and turned over to the TSPS Integration Group for certification in the system laboratory. When certification is complete, the software is made available to the developer community.

### 4.2.2 C-Language development

Thus far we have described the non-frozen development environment for emulated programs. This section addresses the C-language development environment. The Change Management System (CMS) plays a key role in both frozen and non-frozen C-language development in TSPS. To understand how C-language development proceeds, it is necessary to grasp the basic concepts employed in CMS.

*4.2.2.1 The Change Management System.* CMS is a collection of *UNIX* programs aimed at controlling the activity of programmers, test-teams, and administrators engaged in C-language software development. In CMS, all development activity is tied to the notion of a modification request (MR).

An instance of CMS has three main components:

(*i*) A set of project source files maintained under the Source Code Control System (SCCS), described earlier. This allows multiple versions of a source file to be kept (for example, the current official version, a version undergoing system test, and a developer's private version).

(*ii*) A relational database that maintains the status of MRs and relates MRs to SCCS (version) identifiers. This not only provides the status of MRs but also permits the retrieval of source file versions that correspond to particular features or code changes.

(*iii*) A set of directory structures called *nodes*, each of which is identical in makeup (i.e., reflects the generic program's directory structure) and is capable of holding all source, object, process, and data files that constitute a generic program.

The use of CMS in TSPS C-language development will be the topic of the next several sections.

*4.2.2.2 Creation and modification of source modules.* CMS maintains

an official source repository that consists of one SCCS file for each source file used in a generic development. For TSPS, this includes C-language source files, header files, and makefiles. These SCCS files are stored in a project directory structure, a subtree of the hierarchical file system, organized so that files making up a single process typically are found in a single directory, while the files associated with related processes are grouped in directory structures that are subtrees of the overall project directory structure.

As previously mentioned, the SCCS files can contain multiple versions of the corresponding source file. To modify a source file, the programmer first extracts the latest version of the file to be modified from the SCCS file in the official repository. The programmer then edits the file with the text editor, builds an executable version of the affected process file, and tests the resulting product. When satisfied with the changes to the source file, the programmer returns the new version of the file to SCCS.

All source file change activity is associated with an MR number. Basically, the MR is a name under which a set of software changes are grouped. The programmer is quite free to use MRs as necessary during the non-frozen mode of development. Typically, the programmer will define an MR to represent a feature or subfeature being developed.

***4.2.2.3 Generation of a private version of software for testing.*** The generation of executable process files from C-language source uses the 3BSGS and makefiles, as descibed in Sections 3.3.1 and 3.3.2. However, rather than using the make command, an enhancement of make, called build is employed by TSPS developers. Build is a software tool provided as part of CMS. It provides a very convenient mechanism to construct a private version of a process or set of processes. The programmer executes a single command and only has to deal with those files that have been modified. Other required files are automatically shared from the official directory structure. This sharing not only saves file space, but also eliminates the need for time-consuming, error-prone copying of files.

In TSPS, each programmer maintains a private node for development. All programmers share the official node. The developer scenario is as follows:

(*i*) A command is issued to retrieve the source files to be modified.

(*ii*) The retrieved source files are edited.

(*iii*) Executable versions of the software are constructed using build.

(*iv*) The executable files are transported to the system laboratory for testing.

(*v*) When changes are complete, the source files are returned to SCCS.

There is little chance for error during software construction since build

automatically generates the commands to reprocess any and all files that have been affected by the editing. This eliminates the manual effort to reprocess files, the potential errors caused by forgetting to reprocess files, and the overhead of unnecessarily reprocessing files.

***4.2.2.4 Submission and integration of software changes.*** In the non-frozen mode, as developers complete testing of new software associated with an MR, the code is submitted by executing the CMS submit command. This changes the status of the MR in the CMS database from "active" to "submitted." In addition, documentation specifying the feature(s) covered by this submission and the source and process files affected is turned over to the TSPS Integration Group.

As with emulated software, (and usually at the same time), C-language software undergoes a "base load" process. After the load is tested and certified, the official node is updated with all files in the test node and the MRs are marked "approved" and "integrated" in the CMS database. Non-frozen development can then proceed with the newly tested and approved official node.

### *4.3 Frozen-mode methodology*

#### *4.3.1 Overview*

In the non-frozen mode of development, the emphasis is on giving the programmer freedom to make large-scale changes to existing software to facilitate new feature development. As system testing begins, TSPS software is placed under rigorous change control procedures. Problems are documented in the form of trouble reports, which are then carefully monitored. The developers submit correction reports (CRs), which include a description of the fix, as well as the software change itself. Each CR is tested individually against the existing approved software version and must be approved by a Change Review Committee with representatives from all involved project teams before it is officially incorporated in the generic.

The intent of this method is to evolve the software in a rigorously controlled manner. Thus, when operational problems arise, they can be localized more easily to a particular area of software. With this approach, a high degree of confidence in the evolving software product is realized.

#### *4.3.2 Emulated development*

From a programmer's point of view, the scenario for creating and testing a correction in emulation code in frozen mode is similar to developing code in the non-frozen mode; however, the unit of change is substantially different. In non-frozen mode, the unit of change was the partial load. In frozen mode, it is the overwrite described in Section 3.2.4.

Special "patch" directives recognized by SPC-SWAP are placed by programmers in the overwrite source code to ensure that a change does not cause the entire object file to be relocated in memory. Replacement of program or data words is permitted, but only by instruction sequences of equal size. If a change is larger than the instruction sequence being replaced, a transfer to "patch" area is made. (Patch area is spare memory space reserved for this purpose.) In this way, object modules remain the same size and the number of memory locations whose contents change is minimized.

#### *4.3.3 C-Language development*

For C-language development, there is little difference between the frozen and non-frozen modes. The same tools are used and the same basic development scenario is followed. The main difference is that each MR represents an existing system problem or minor enhancement rather than a new feature. Therefore, the amount of software associated with an MR tends to be of a smaller quantity. In addition, as with emulated software, software change reports are generated and submitted to document the change and must be approved through the same process.

## V. LABORATORY SUPPORT SYSTEM

The Laboratory Support System is a software testing system that enables users to test their programs in a TSPS system laboratory. It is generally concerned with the laboratory execution environment of TSPS programs.

### *5.1 System laboratory configuration*

To provide sufficient test capabilities for TSPS development, there are two independent TSPS system laboratories. Generally, both system laboratories support parallel execution environments for TSPS programs. The configuration for a TSPS system laboratory is shown in Fig. 3.

#### *5.1.1 System laboratory hardware configuration*

The TSPS No. 1B is controlled by the SPC 1B, which consists of a 3B20D Processor and a Peripheral System Interface (PSI). The complete description of the SPC 1B is detailed in Ref. 11. In addition to the SPC 1B, a TSPS system laboratory also contains a set of TSPS peripheral units, operator consoles, and a Laboratory Support System.

#### *5.1.2 Laboratory Support System*

The Laboratory Support System is primarily used to support debugging on the SPC 1B, for controlling the operations of the SPC 1B

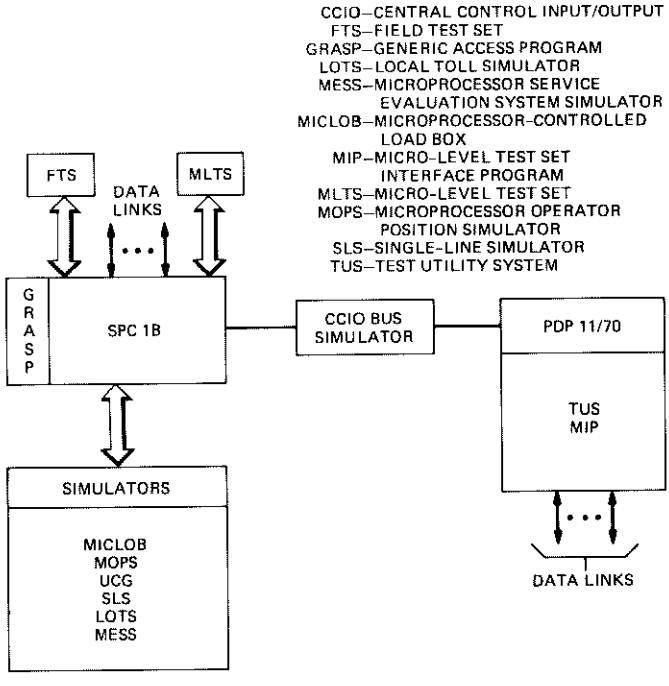



Fig. 3—Laboratory configuration for TSPS No. 1B.

in the test environment, and for loading the SPC 1B memory. The LSS consists of a PDP-11/70 support computer, special laboratory and generic utility systems, and TSPS call simulators, which support program testing. The PDP-11/70 is connected to the SPC 1B by a Central Control Input/Output (CCIO) bus simulator unit.[11]

There are four major debugging tools associated with a TSPS LSS. They are the Test Utility System, the Micro-Level Test Set (MLTS), the Field Test Set (FTS), and the DMERT Generic Access Program (GRASP). Each provides a distinct set of debugging capabilities.

***5.1.2.1 Test Utility System.*** The Test Utility System (TUS) is the principal, high-level, debugging tool for the software developed for the SPC 1B. TUS provides symbolic access to data and a "C"-like utility language. It resides partially on the LSS and partially on the SPC 1B. The TUS support processor subsystem on the LSS performs TUS input and output processing and TUS system control. Users log into TUS on the support processor subsystem; this subsystem converts the user's utility commands into executable command groups that are then sent to the TUS test processor subsystem on the SPC 1B. The test processor subsystem executes under the DMERT operating system. Symbolic references are resolved on the LSS using symbol tables constructed on the PSS as part of the base load process.

As the TUS test processor subsystem on the SPC 1B receives command groups to be executed, requested break points or matchers are set up, and associated utility commands are processed. Any raw data that result from the utility commands are collected and sent back to the TUS support processor subsystem to be processed. Output to the user includes symbolic references and processed data. Commands are available to: read or write any memory location, display or send a message to a process or port, enable or disable a program trace, copy a file from the test processor to the LSS, start or stop a test process, and send an event or fault to a test process with associated data input.

Because of its symbolic access to data and its "C"-like utility language, TUS is a powerful debugging tool. In addition, TUS is an integral part of the system overwrite facility. TUS supports a data link that is capable of transferring files at a rate of 56K baud. Using TUS, overwrites are quickly transferred from the LSS to the SPC 1B when requested by the programmer or tester. More details about TUS can be found in Ref. 10.

***5.1.2.2 Micro-level test set.*** The micro-level test set connects to the microbus of the SPC 1B and is used for direct access to 3B20D registers and memory. From a terminal connected to the LSS, simple commands can be sent to the MLTS to insert breakpoints. After a breakpoint fires, the SPC 1B is halted so that the tester can display or change the contents of registers and memory locations. Once this is done, the processor can be restarted. MLTS (unlike TUS) is a stand-alone tool that is completely independent of the operating system on the SPC 1B and can be used to debug at the kernel level. However, it does not provide symbolic testing capabilities and is used only for low-level program debugging.

***5.1.2.3 Field test set.*** The field test set is a portable debugging tool capable of tracing the execution of processes without interfering with normal processor operations (Fig. 4). As such, it provides an effective debugging capability both in the system laboratory and operational field sites. It interfaces directly with the dual-access utility circuit in the 3B20D Processor to record information about the execution of processes in the SPC 1B. The tracing capabilities of the FTS are controlled by simple commands that set up matchers, trigger functions, and trace memory. The FTS can be used to perform a process trace, transfer trace, data history trace, simultaneous data history and transfer trace, function trace, and simultaneous data history and function trace. Like the MLTS, the FTS is completely independent of DMERT.

***5.1.2.4 Generic Access Program.*** The final debugging tool used in the development of the TSPS No. 1B is the Generic Access Program (GRASP), which is a standard DMERT utility system resident on the 3B20D. It provides basic, non-symbolic capabilities to dump registers

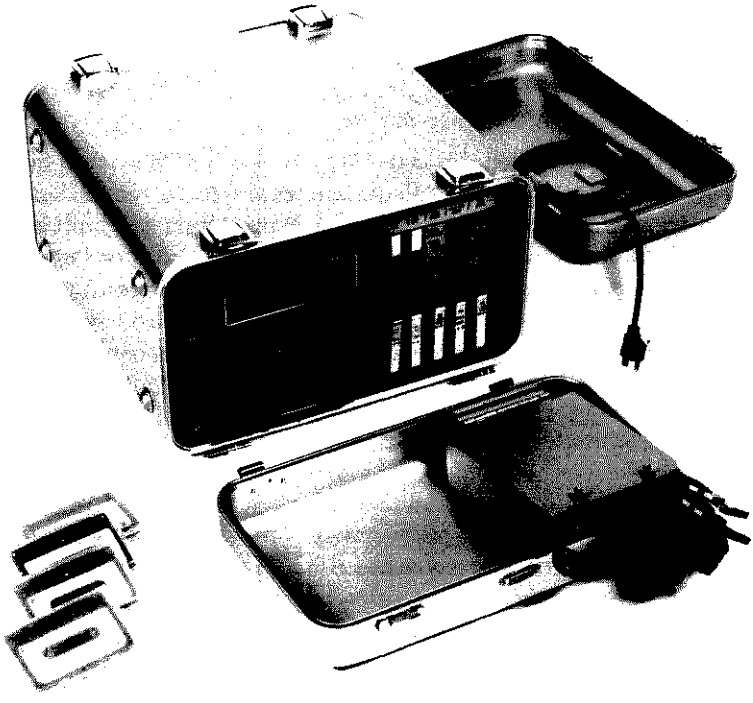


Fig. 4—The field test set.

and memory, and to set up breakpoints to stop program execution at desired points. Using the 3B20D dual-access utility circuit, GRASP also can trace the flow of execution of instructions of a process in a non-interfering fashion. A typical trace might record the "from address" and "to address" of every branch instruction executed. Additional information, such as the contents of the registers, may also be obtained. Because it is running on the same processor as the test processes and requires the support of the DMERT operating system, GRASP will generally interfere (like TUS) with the normal operations of a process and is not effective for kernel debugging.

***5.1.2.5 Choosing the right debugging tool.*** As evident in the previous discussions, the four debugging tools used in the TSPS No. 1B development were designed for specific applications and have overlapping utility capabilities. TUS is a high-level, symbolic debugging tool. The MLTS is a low-level debugging tool. The FTS is a non-interfering debugging tool in the sense that normal instruction sequences and timing are not altered when using it. GRASP is a generic tool available on an operational 3B20D. Within a particular test session, a combi-

nation of these tools may be used. Whenever possible, TUS is used because of its powerful debugging capabilities. The MLTS, FTS and GRASP are used only when needed for a particular problem.

### *5.2 Laboratory software change facility*

When a new TSPS generic is developed, the TSPS system laboratories are used to provide a realistic environment to test the new or changed programs. These new or changed programs are built on the PSS and then down loaded via the LSS to the SPC 1B for laboratory testing. Changes are often required to correct program problems uncovered during the lab session. These changes are in the form of overwrites consisting of program and data instructions being modified.

#### *5.2.1 Emulated program change facility*

To modify emulated programs in the system laboratory, the OVGEN program on the PSS is used to generate an overwrite file. This file is then transferred to the LSS and used as input to an overwrite assembler on the LSS to produce an overwrite object module. Using TUS, this module is then transferred to the SPC 1B disk and a special loader program on the 3B20D is run to overwrite the SPC 1B emulated address space. The overwrite assembler and loader used in this process are described in this section.

*5.2.1.1 The overwrite assembler for emulated code.* NOVA is a utility program on the LSS used to assemble overwrites for SPC 1B emulated code. The input to NOVA is the swap overwrite file produced by OVGEN, and the output is an absolute or relocatable object module and a relocation dictionary for the overwrite. The information contained in the relocation dictionary consists of pointers into the object file and addresses to be assigned by the loader. NOVA also produces overwrite and cross-reference listings to document the change in the laboratory.

*5.2.1.2 The overwrite loader for emulated code.* The loader program (NULOAD) is a relocatable loader for emulation code on the 3B20D. It provides a means by which NOVA-assembled overwrites are loaded into the TSPS process address space in memory. During each lab session, NULOAD automatically monitors the allocation of patch space and assigns patch space as needed by the overwrites. It allows the user to combine various overwrites assembled at different times and by different people without having one patch in one overwrite loaded on top of a patch from another overwrite. NULOAD also is used to load a partial load created on the PSS system.

#### *5.2.2 Native-mode program change facility*

To modify C-language programs in the system laboratory, an entire new process file (pfile) must be built on the PSS. Once this file has

been transferred via the LSS and TUS to the SPC 1B disk, it is installed on the 3B20D disk replacing the previous version of the process file. The next time the process is loaded into memory, the new pfile is used.

### *5.3 Laboratory simulators*

Program testing in the TSPS system laboratory often requires the ability to either place a single call or a substantial traffic load on the TSPS and to automatically handle calls that require operator assistance in a live site. A number of microprocessor-controlled simulators including the Single-Line Simulator (SLS), Local Toll Simulator (LOTS), the Microprocessor-Controlled Load Box (MICLOB), and the Microprocessor Operator Position Simulator (MOPS) have been developed as part of the LSS to provide these capabilities for the TSPS No. 1 system and were also used to test the TSPS No. 1B. These call simulators are described in detail in Ref. 6.

## VI. SUMMARY

The TSPS No. 1B Software Development System provides a complete set of facilities for TSPS generic development on the 3B20D Processor. Software generation tools can compile and link both emulated, assembly-language, and C-language programs. Administrative software and a rigorous development methodology control the evolution of feature development and keep track of the multiple versions of software that exist during the development process. Special-purpose software and hardware in the TSPS system laboratories provide a modern test environment for debugging and certification of the software product. These facilities are an effective and productive software development environment for future network operator services.

## VII. ACKNOWLEDGMENTS

Many individuals have contributed in a significant way to the design and implementation of the TSPS No. 1B Software Development System. In particular, the authors wish to acknowledge the contributions of K. M. Conness, G. F. Gieraltowski, B. E. Holmes, M. S. Koehler, B. T. Rovegno, R. J. Welsch, and S. P. Winker, who were responsible for major components of the Software Development System. In addition, we would like to extend our thanks to D. L. Atkins, B. T. Rovegno, and E. S. Sachs for their many constructive suggestions during the preparation of this article.

## REFERENCES

1. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.

2. R. J. Gill, G. J. Kujawinski, and E. H. Stredde, "Traffic Service Position System No. 1B: Real-Time Architecture Utilizing the DMERT Operating System," B.S.T.J., this issue.
3. B.S.T.J., *61*, No. 7, Part 3 (September 1982), special issue on the Stored Program Controlled Network.
4. D. M. Ritchie and K. Thompson, "*UNIX*™ Time-Sharing System: The *UNIX* Time-Sharing System," B.S.T.J., *57*, No. 6, Part 2 (July-August 1978), pp. 1905–30.
5. T. A. Dolotta, R. C. Haight, and J. R. Mashey, "*UNIX*™ Time-Sharing System: The Programmer's Workbench," B.S.T.J., *57*, No. 6, Part 2 (July-August 1978), pp. 2177–2200.
6. J. J. Stanaway, Jr., J. J. Victor, and R. J. Welsch, "Software Development Tools," B.S.T.J., *58*, No. 6, Part 1 (July-August 1979), pp. 1307–34.
7. M. E. Barton, N. M. Haller, and G. W. Ricker, "Service Programs," B.S.T.J., *48*, No. 8 (October 1969), pp. 2866–80.
8. M. E. Barton, "The Macro Assembler, SWAP—A General Purpose Interpretive Processor," Fall Joint Computer Conference, 1970.
9. J. C. Lund, Jr., M. R. Ordun, and R. J. Wojcik, "Implementation of the Calling Card Service Capability-Application of a Software Methodology," Int. Commun. Conf., Denver, Colorado, June 1981.
10. B.S.T.J., *62*, No. 1, Part 2 (January 1983), special issue on the 3B20D Processor & DMERT Operating System.
11. G. T. Clark, H. A. Hilsinger, J. H. Tendick, and R. A. Weber, "Integration of the 3B20D Processor into TSPS," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Integration and System Testing

By R. AHMARI, R. S. DiPIETRO, S. C. REED, and J. R. WILLIAMS

(Manuscript received June 30, 1982)

*The integration testing and system testing strategies used in the development of the Traffic Service Position System No. 1B (TSPS No. 1B) are described in this article. It discusses novel methods developed for the integration and testing of the first embedded application of the 3B20D Processor and DMERT operating system. The new testing techniques and project management aides, which were vital to the achievement of system quality for the Bell System service, are described. The first TSPS No. 1B went into service on November 20, 1981, in Fresno, California.*

## I. INTRODUCTION

The first Traffic Service Position System No. 1B (TSPS No. 1B) was placed in service in Fresno, California, on November 20, 1981. This newly installed TSPS brought modern Stored Program Control (SPC) operator services capabilities to telephone customers in Fresno and other San Joaquin Valley communities, and permitted Pacific Telephone to close the largest remaining Bell System Cordboard installation in the United States. Modern operator services[1,2] are bringing convenience to telephone customers and increased efficiency and savings to Pacific Telephone.

Placing the Fresno TSPS No. 1B in service marked the conclusion of a major testing program that was vital to the delivery of a high-quality system, capable of meeting all objectives.[1] This was especially critical because of the planned, rapid conversion of existing TSPS No. 1 sites to TSPS No. 1B. Furthermore, since TSPS No. 1B is the first electronic Stored Program Control system to utilize an embedded

3B20 Duplex (3B20D) Processor and its Duplex Multi-Environment Real-Time (DMERT) operating system,[3] confirmation of its capabilities was essential to the success of the TSPS No. 1B program, as well as to other systems utilizing the 3B20D Processor.

Following the successful introduction of TSPS No. 1B in Fresno, a second new start TSPS No. 1B was placed in service in San Antonio (Southwestern Bell Telephone) on January 30, 1982. Both the San Antonio and Fresno sites were instrumental in the overall test program. The first conversion of an existing in-service TSPS No. 1 office to TSPS No. 1B occurred at Redwood City, California, on March 13, 1982.

The balance of this paper will focus on the overall test program for TSPS No. 1B, specifically on integration and system testing. These systemwide test activities are major components of the methodology used for development of TSPS No. 1B (see Fig. 1*). Integration testing specifically includes: the introduction of various system components (hardware and software) into the development laboratories; the integration of these components to form a stable operating environment; and testing to ensure that a sufficient set of capabilities exists to warrant full, comprehensive, and broad testing. System testing includes the later testing and is done to verify the system, as a complete functional product, before its release to the field.

The integration and system testing of TSPS No. 1B and its components—the 3B20D Processor, the TSPS Peripheral System Interface (PSI), the DMERT operating system, and TSPS application software—was a complex effort. Because TSPS was the first user of the 3B20D as an embedded processor, a substantial cooperative effort with the 3B20D development team was scheduled to identify and solve processor and operating system problems that were encountered during the design, integration, and testing of TSPS No. 1B. A tightly scheduled and controlled series of integration tests were performed by a team at the Bell Laboratories Indian Hill Laboratory in Naperville, Illinois, as new hardware and software capabilities were delivered to the system laboratories. After these new capability packages were integrated and certified as ready for system test, they were delivered to the system test teams at Indian Hill, Fresno, and San Antonio. The use of the two field sites provided settings similar to live TSPS offices for early identification of site-dependent problems and for verification of future in-service retrofit procedures with Western Electric engineers.

The system test plan consisted of over 16,000 unique tests of three general types:

* Acronyms and abbreviations used in this paper are defined at the back of this Journal.

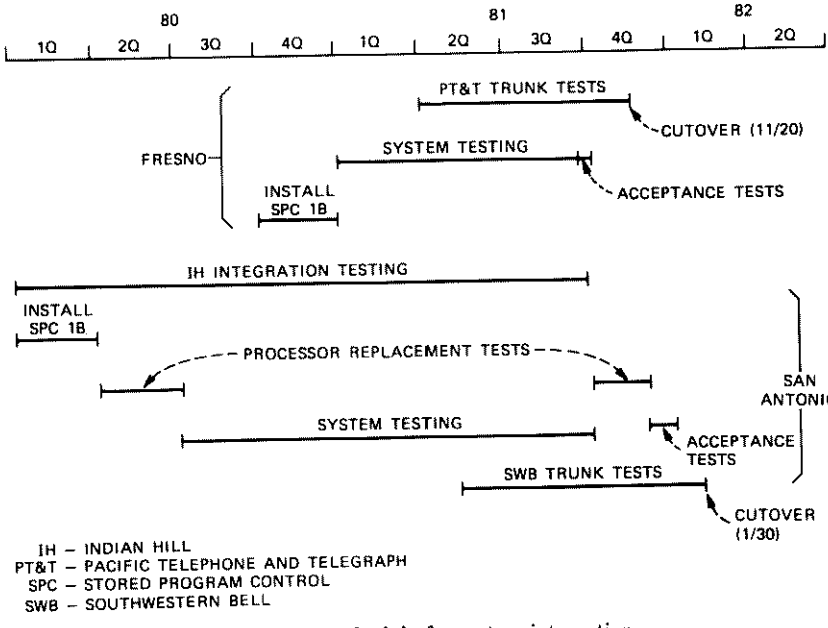



Fig. 1—Test schedule for system integration.

(*i*) Functional tests—to verify that design requirements were met by new features.

(*ii*) Regression tests—to verify continued proper operation of previously delivered features.

(*iii*) Stress tests—to parametrically determine and evaluate system limits, response characteristics, and overall performance under traffic overload conditions and fault conditions.

As depicted in Fig. 2, system test responsibilities were allocated to teams at the three locations with some intended overlap. For example, most of the DMERT feature system testing was scheduled for Fresno, to take advantage of the higher simulated-traffic-load capability at that site. Of these tests, some had been previously run at Indian Hill and at San Antonio, where exposure to different hardware and system configurations was critical. San Antonio was selected as the site to perform exhaustive tests of the existing TSPS periphery and to test external interfaces, such as to the Switching Control Center System (SCCS). In addition, a core of tests was periodically selected for all three system test sites to evaluate key software releases and to regularly assess the performance of a broad spectrum of features. In general, these "Run-for-Record" tests were a subset of the tests included in the system test plan.

Both integration and system testing were pursued for a period of over one year to completely verify the operation of the TSPS No. 1B.

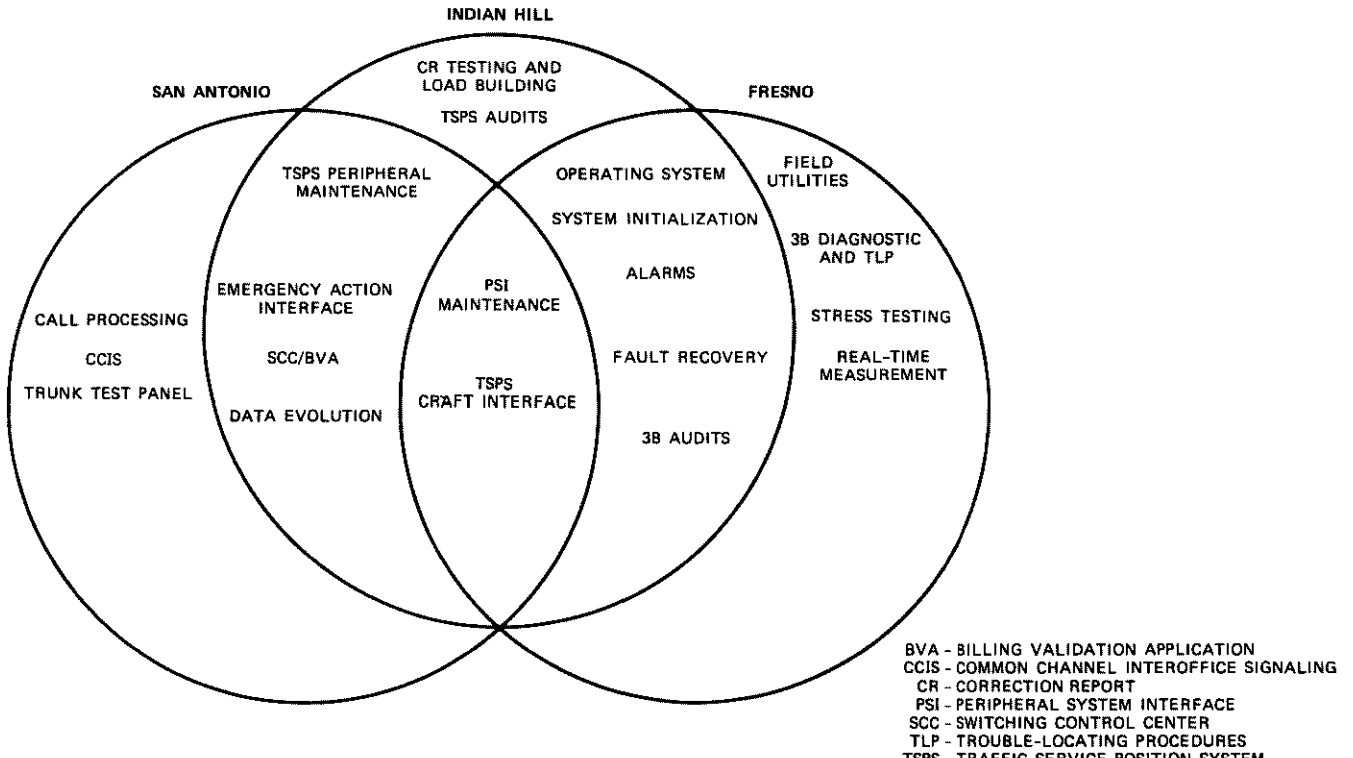



Fig. 2—Test plan overview.

Deliverables were phased to provide a smooth work flow. For this rigorous and demanding test program, tests were conducted ranging from microscopic tests, aimed at verifying operational status of specific elements of system capabilities, to global capacity and performance measurements and evaluations. During the test period, all problems found were tracked and pursued to their resolution. Performance criteria were set, performance was measured to ensure convergence to these criteria, and strong project management techniques were employed to ensure a timely introduction of a high-quality TSPS No. 1B.

The following sections of this article describe how the laboratory integration and site system testing were conducted. The excellent performance of the TSPS No. 1B system at cutover resulted not only from good planning and design but from timely integration and comprehensive testing supported by appropriate documentation, development tools, and change control procedures.

## II. INTEGRATION TESTING

### *2.1 Overview*

A crucial stage of the verification and evaluation process for new 3B20D/DMERT and TSPS capabilities is integration testing. During the early stages of the process, the development environment was considered to be "non-frozen," and designers could freely make large-scale changes to the existing capabilities or introduce new ones. The design and development of various features were accomplished by partitioning them into a series of functional units, which were then tested by the designers to make sure that each unit met the design requirements. Upon the completion of unit testing by the designers, a set of integration tests were performed by an independent team to ensure that each functional unit performed reliably in the total system environment. Upon successful integration of functional units, they were considered "frozen," and changes were made only through formal procedures. The frozen functional units were then ready for system testing and evaluation, the last stage of the verification process.[4]

Certain unique characteristics of the TSPS No. 1B development environment required special attention during the integration process. These characteristics can be summarized as follows:

(*i*) TSPS No. 1B was developed in a multilanguage environment. DMERT software was primarily developed in the high-level C language, while TSPS software was developed in both assembly language (TSPS emulated software) and the high-level C language (TSPS native software).

(*ii*) DMERT software changes were concurrently developed by the common system organization and had to be integrated with the TSPS

software changes developed by the application organization. In some cases a DMERT change required a coordinated change from TSPS or vice versa.

(*iii*) 3B20D firmware/hardware changes had to be similarly integrated with the TSPS firmware, hardware, and software changes, such as those associated with the PSI.

(*iv*) A new generation of TSPS No. 1B support tools was developed, resulting in new load generation, installation, and integration procedures.

The above characteristics were taken into consideration to establish an efficient integration and test strategy, which ensured the quality of the functional units integrated into the system.

### 2.2 Integration testing methodology

The integration testing philosophy adopted for TSPS No. 1B established a rigorous methodology with systematic checks to independently evaluate all functional units supplied by designers in a single environment. To meet this objective, it was decided that the integration group should be independent of the development groups. Independence provided an unbiased and fresh viewpoint in assessing the functional units and interpreting the test results.

The integration test plan started early in the development cycle, subsequent to the availability of feature requirements. The plan took into consideration the availability sequence of various features and functional units within each feature. A strategy was established that guaranteed efficient handling of all units ready for integration testing without interrupting the other development activities. Special efforts were required to coordinate and synchronize the integration testing of TSPS functional units with 3B20D/DMERT features and capabilities. This was especially crucial in interface areas where direct interaction between the two environments existed. Another major consideration in the plan was that the software, firmware, and hardware environments evolve in a manner compatible with the system test plan, thus allowing the system testing activities to proceed without any interruption.

The TSPS No. 1B integration tests were generated independently, following a thorough analysis and review process, which is summarized as follows:

(*i*) Feature requirement and design specification documents were reviewed to identify functional units within each feature.

(*ii*) Functions performed by each unit, along with its input/output characteristics, were identified.

(*iii*) Software interfaces with firmware and hardware were examined and reviewed.

(*iv*) 3B20D/DMERT interfaces with the TSPS application were thoroughly analyzed, and expected input/output responses of the interface units were identified.

(*v*) The control flow among various units and their time sequence description was carefully inspected.

(*vi*) A general layout and description of data used by each unit were identified.

Based on these examinations, a set of integration tests were developed to exercise each functional unit within the total system environment and to stress 3B20D/TSPS interfaces. These tests were later augmented by the information supplied by the designer(s), upon completion of unit testing, to start the integration testing process. During this process it was first verified that each unit correctly performed all intended functions, including its interaction with other units in a single environment. The second, more subtle, objective was to ensure that the unit did not perform any function that, either singly or in combination with others, would degrade the performance of the system. Third, each unit was examined to make certain that it met the standards set for design structure, documentation, and coding.

Problems identified during the process of integration testing were classified according to their impact on the overall system. Problems were prioritized and fed back to the designer(s) for corrective actions. Problems related to the TSPS software were primarily tracked by Trouble Reports (TRs), while Modification Requests (MRs) were used to track 3B20D/DMERT problems. Specific corrective action by a designer was required to close a TR or an MR.[5] The list of all open TRs and MRs was carefully monitored to evaluate the total impact and to establish a plan for closing each individual TR or MR. The primary objective of this plan was to ensure that the evolution of the TSPS No. 1B software, firmware, and hardware could proceed on schedule without any interruption.

#### *2.2.1 DMERT integration testing*

The development of DMERT software was done in parallel with the development of TSPS software. At well-defined points of the development, a full DMERT release was generated by the common system organization and delivered to the various applications after being tested on a stand-alone basis.[6]

Upon delivery, each DMERT release was first installed in the Program Support System (PSS). Subsequently, steps were taken to incorporate the new DMERT release into the TSPS No. 1B environment (see Fig. 3). These steps can be summarized as follows:

(*i*) The Equipment Configuration Database and System Generation Database (ECD/SG) were updated to reflect the latest DMERT

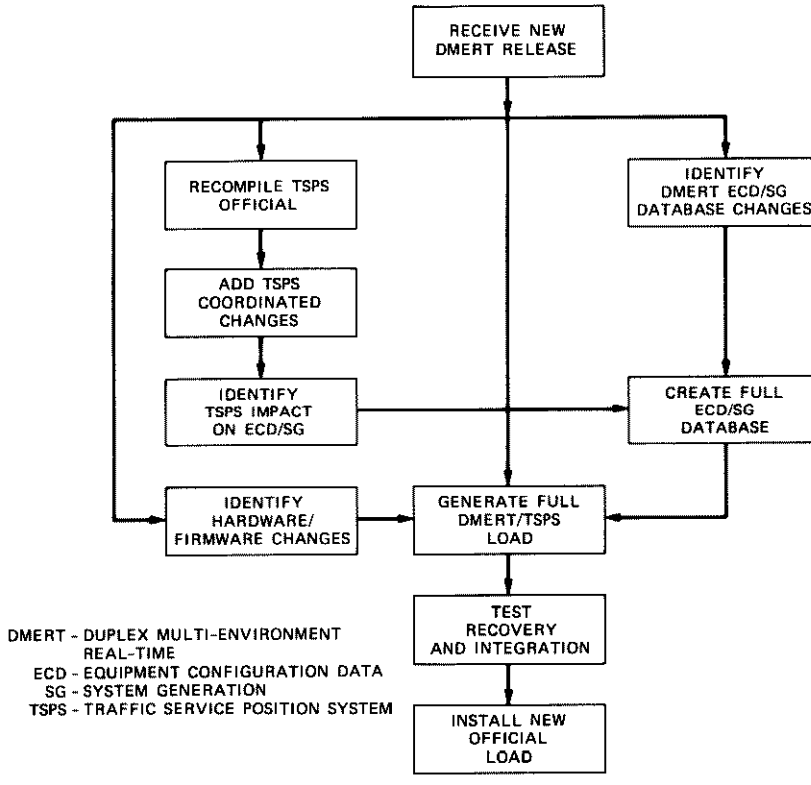



Fig. 3—DMERT integration testing.

changes. These databases contain data structures necessary to generate and run the TSPS No. 1B software system.

(*ii*) Those areas of TSPS software that were directly affected by the new DMERT release were identified. The corresponding TSPS software changes were then developed on the PSS under the control of the Change Management System (CMS). More information about the software generation process may be found in Ref. 5.

(*iii*) The TSPS official native-mode software was recompiled using the new DMERT release to update the TSPS programs that were affected by the DMERT release (such as a library that is shared by both DMERT and TSPS software and is changed by the new DMERT release). All TSPS programs that were changed by the recompilation process were audited for potential impact on the overall system. If necessary, appropriate TSPS software changes were generated [see Step (*ii*)] to compensate for the impact.

(*iv*) Those areas of DMERT that required changes unique to the

TSPS application were identified. These changes were also reflected in the PSS.

Subsequent to these steps, a full TSPS No. 1B load was generated on the PSS and was installed in the system laboratory for load recovery and integration testing. This load contained the latest DMERT release, along with changes unique to the TSPS application; a recompiled version of the TSPS official load; ECD/SG database changes; and the TSPS software changes required by the new DMERT release.

Any problems in the new load were quickly identified using the debugging tools associated with the TSPS No. 1B,[5] and immediate steps were taken to obtain the necessary changes from the appropriate designers (DMERT or TSPS). Once the load was cycling, a set of integration tests were conducted to ensure that each functional unit performed as expected in the total system environment. These tests were based upon the ones independently generated by the TSPS application organization, augmented by those supplied by the 3B20D common system organization. Special emphasis was given to DMERT/TSPS interface modules, TSPS areas directly impacted by the new DMERT release, DMERT changes unique to the TSPS application, and ECD/SG database changes. Problems identified during the process of integration testing were tracked using the strategy described in Section 2.2.

#### 2.2.2 *TSPS integration testing*

Modifications to the TSPS software were introduced into the TSPS No. 1B environment via base loads.[5] Introduction of a new base load began with identification of all source file changes that had occurred since the prior base load. Then a new test load was created, containing additional software changes and associated ECD/SG database changes. New software changes were usually grouped in one or more test packages. The packaging strategy for the TSPS changes took into consideration many factors including:

(*i*) Keeping a high degree of resolution in testing of various changes that have an impact on the same programs.

(*ii*) Optimizing the amount of time used in the system laboratory.

(*iii*) Keeping the PSS effort for compiling various changes to a reasonable amount.

(*iv*) Optimizing the system capacity for processing software changes.

Keeping these objectives in mind, all software changes were first mapped into a series of test packages. Program dependencies played a key role in determining the number and content of these test packages. The load of the test packages, along with coordinated hardware and firmware changes, were then installed in the system

laboratory separate from the TSPS official load before the integration testing could proceed.

Upon installation of the load, a predefined set of integration tests were used to examine each test package. These tests were aimed at quickly identifying problems and attributing them to individual changes. Problems encountered during the testing were classified by their impact on the normal operation of the system and tracked as described in Section 2.2. This process was then repeated for all test packages until the necessary resolution was obtained, and it was ensured that all changes could coexist and perform as expected in a single environment. All changes that successfully passed the integration testing were then installed in the system laboratory as the new official base load (see Fig. 4).

Upon completion of the integration testing, functional units were considered "frozen" and ready for system testing. In the frozen environment a set of stringent change control procedures were used so that the TSPS software evolved in a rigorously controlled manner, leading to a high-quality production release. All problems were documented and tracked by appropriate trouble reports, which were carefully monitored. To close a trouble report, a designer was required to submit a Correction Report (CR), which contained the functional description of the change, all necessary information relevant to the software change, and a description of the unit tests used by the designer. Each change was considered an independent entity and was individually tested in the total system environment before its approval. All changes had to be approved by the Change Review Committee (CRC) before they were incorporated in the official load. The CRC comprised representatives from each hardware/software design and test organization on the project.

### *2.3 Testing of 3B20D and TSPS firmware and hardware changes*

Firmware and hardware changes associated with the 3B20D were also tested in the system laboratory environment by the TSPS integration group to uncover problems unique to the TSPS application. These changes, along with coordinated software modifications, were first installed in the system laboratory environment. A series of integration tests were then performed to stress and exercise the firmware-hardware-software interfaces. Problems encountered during this stage of testing were tracked in a fashion similar to that described in Section 2.2.

In cases where 3B20D microcode changes had an impact on the writable portion of the microstore,[7] no firmware change was necessary, and the microcode change was released in a fashion similar to a software change described in earlier sections.

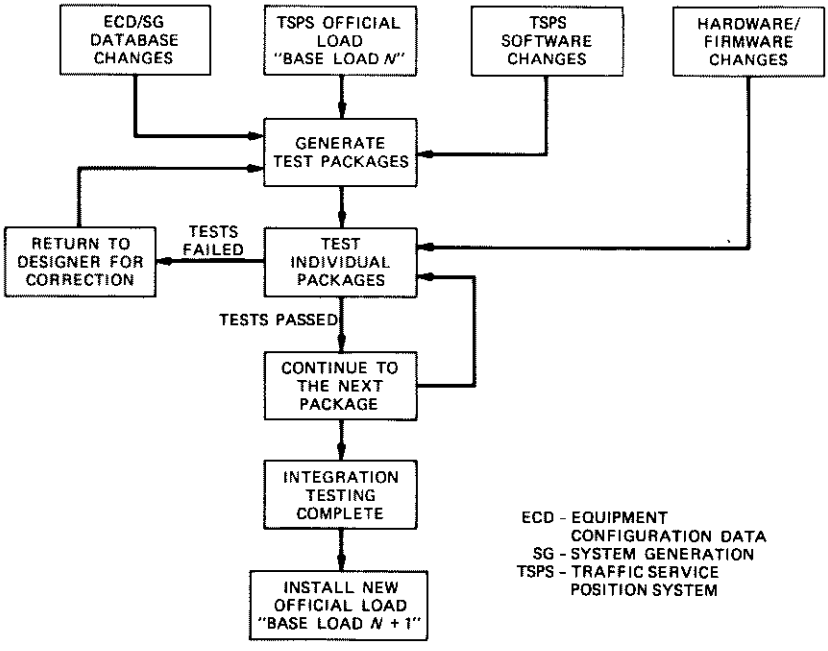



Fig. 4—TSPS integration testing.

Changes to the TSPS microcode were administered under control of CMS in a similar fashion to the TSPS software changes. A change was first developed on the PSS and was then submitted for integration testing. The change was transferred to the system laboratory in one or more files and was installed in the writable microstore unique to the system labs. A series of integration tests were then performed to verify its integrity in the total system environment. Similarly, all hardware changes associated with TSPS were tested in the system laboratory environment using prototype circuits prior to their official availability. Once it was verified that the hardware/firmware changes operated as expected, they were transmitted to Western Electric along with manufacturing and factory test information, and were subsequently installed at the field sites.

### *2.4 Distribution of individual software changes*

Once a stable TSPS No. 1B release was installed in a field site, the subsequent individual software changes were supplied as individual packages and were installed using the field update commands.[5] A package typically consisted of the software change, additional files containing field update commands, and information files required for

installation, testing, soaking, and distribution of the change to the field.

For a change in the 3B20D/DMERT software, the 3B20D common system organization generated a software package that was distributed to each application organization. Before the release of the package to the various applications, it was first tested and certified in the common system environment by the common system organization. Once the change was received by TSPS, it was examined for compatibility with the TSPS environment. In some cases a coordinated TSPS software change was required before actual installation and testing could proceed. A series of tests were then conducted to ensure that the change performed as expected within the total system environment. These tests included those supplied by the common system organization, as well as an independent set generated by TSPS personnel.

Once a software package was successfully tested and soaked in the TSPS system laboratory environment, necessary changes were made to its information files prior to field delivery. These changes were made to delete the test information unique to the system laboratory environment. This information was used by application organizations to verify the modification and was not applicable to the field environment.

The TSPS application software changes required for distribution to the field sites were individually tested and approved using the "frozen environment" methodology described in Section 2.2.2. These changes were then packaged using the standard format described earlier in this section. Necessary considerations were given in the packaging strategy to balancing and optimizing various factors such as the field installation time and the number of system initializations required to install the changes. Also, if necessary, these changes were retested to ensure that no problems were introduced in the final packaging.

## III. SYSTEM TESTING AND EVALUATION

### *3.1 Overview*

System testing also represented a crucial and essential part of the overall verification process. During this stage a complete set of functional tests were written and executed to independently evaluate all TSPS and DMERT operational and maintenance capabilities from a system viewpoint. These tests were systematically performed with the objective of ensuring that the requirements for each feature were met. Special emphasis was given to evaluating the interaction between the new features and the existing features that were emulated from the TSPS No. 1 environment. Furthermore, regression tests were performed to ensure that previous tested capabilities were not being adversely affected by the introduction of new ones. All problems

identified during the system testing interval were documented by appropriate trouble reports.

The system testing effort was divided into several areas with different objectives. They are listed below with the primary objective for each area and are further discussed in Sections 3.2 to 3.5.

(*i*) Call-processing objective—Verification of all emulated call-processing features.

(*ii*) TSPS maintenance software objective—Verification of fault recognition, routine exercises, and diagnostics.

(*iii*) External interfaces objective—Verification of external interfaces such as the Service Evaluation System (SES), Switching Control Center System (SCCS), and Billing Validation Application (BVA), etc. (see Refs. 1, 8, 9, and 10).

(*iv*) DMERT and 3B20D system testing objective—Verification of DMERT and 3B20D features, DMERT and 3B20D/TSPS interfaces, and sample faulting of 3B20D and its periphery in the TSPS environment.

(*v*) Regression testing objective—Determination of impact of new TSPS and DMERT/3B20D releases on previously tested capabilities.

(*vi*) System evaluation runs objective—Analysis of overall system stability and quality at various loads.

System tests were performed both in the system laboratory and the field sites. The field test plan consisted of functional tests, environmental tests, and an overall acceptance test, and was used as a final check that the system met its functional objectives at specified environmental limits. Over 16,000 individual functional system tests were written and executed during the system test interval at the San Antonio and Fresno test sites. The first execution of all system tests was scheduled and accomplished by early in the third quarter of 1981 (see Fig. 5). This timely execution enabled identification of problems, management, resolution, and closure by turnover (see Fig. 6). All significant system tests passed prior to the turnover of the Fresno site to the operating telephone company. The environmental tests verified system performance at the limits of high temperature and low voltage. The functional-site testing effort validated system performance in a fully equipped TSPS office that has been engineered by an operating telephone company with hardware supplied and installed by Western Electric.

### *3.2 Call-processing testing*

The call-processing software in TSPS consists of a set of programs that provide the logic and control for processing telephone calls. These programs supervise the state of the call, transmit and receive signals

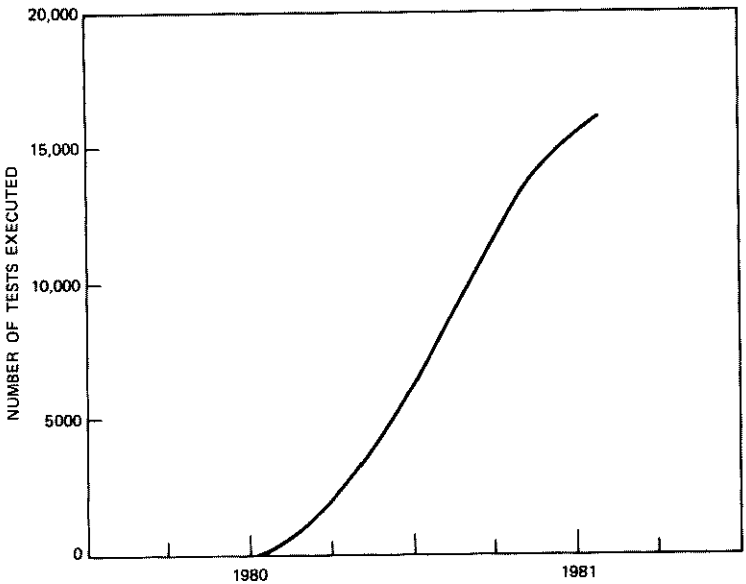



Fig. 5—TSPS No. 1B test schedule.

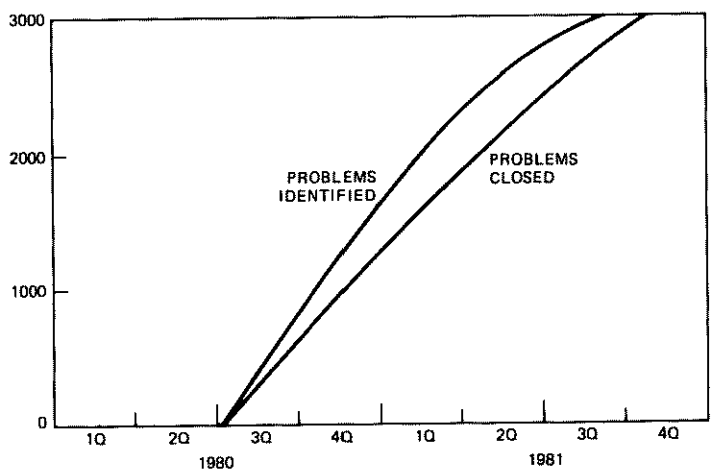



Fig. 6—TSPS No. 1B problem closure.

to and from other switching systems, send information to and receive signals from operators, and record billing details on the calls.

In most cases the call-processing programs were directly emulated from the TSPS No. 1 environment without any structural or code changes. Because the call-processing software was preserved in the TSPS No. 1B, the strategy for testing was mainly to verify the proper

emulation of the existing capabilities. This objective was accomplished by using system test facilities to generate simulated call inputs to exercise various call-processing capabilities, including those involving customers and operators. These capabilities were tested both on a single-call basis and under call loads simulating the normal traffic environment of the system.

### 3.3 TSPS peripheral fault recognition, diagnostics, and exercise

TSPS No. 1 peripheral maintenance programs such as fault recovery, diagnostic, and exercises were also preserved in the TSPS No. 1B environment. These maintenance programs were emulated from the TSPS No. 1 environment, after necessary code and structural changes were made to provide interfacing compatibility with 3B20D hardware, firmware, and maintenance software capabilities. The function of fault recognition is to analyze failure conditions and quickly reconfigure the system to a working state with a minimum interruption to operational call processing. Testing in this area verified that faults were successfully detected and appropriate actions were taken.

Diagnostics are used in TSPS to test the various equipment units in response to fault-recognition requests or manual requests. These capabilities resolve hardware malfunctions by providing trouble-locating messages for use by maintenance craft to repair the faulty hardware. Testing in this area verified that the diagnostics properly detected the presence or absence of troubles and produced messages consistent with expected results without adversely affecting system operation.

Specific routine exercises are used to periodically examine the correct operation of the hardware and to detect various failures that may not show up in normal system operation. In general, exercises utilize the diagnostic capabilities described above to verify the condition of the equipment units on a periodic basis. Testing in this area demonstrated that the exercise routines operate in the expected manner.

#### 3.3.1 Test strategies

Sample physical fault insertion, power faulting, and data faulting, combined with manual actions initiated at the Local Maintenance Position, were the major means of verifying the proper operation of system maintenance functions. These techniques were used to develop specific tailored test strategies for each peripheral unit. Initially, tests were conducted in the absence of call load to verify the proper response. Subsequently, maintenance capabilities were tested in the presence of varying call loads to ensure proper interaction of call handling and maintenance functions.

Physical fault insertion was one of the primary methods of testing the operation of maintenance software and hardware. Physical fault

insertion verified proper cooperation of fault recognition, system reconfiguration, and diagnostics. It further provided a test of the Trouble Locating Manual (TLM) used by the craft to identify faulty circuit packs. It was impractical to do exhaustive fault insertion, such as that utilized in TLM generation. However, it was necessary to develop sample faults to verify that the maintenance software performed within the existing requirements.

Power faulting was the second major means of introducing hardware faults. Power faults were introduced selectively by blowing fuses or intentionally removing power to verify the ability of the system to detect the fault, take the appropriate equipment out of service, and produce the proper alarm and output messages to enable craft personnel to restore the system.

Data faulting was the third faulting technique used in testing maintenance actions. Data faulting is the application of erroneous data to the system. It measures the sensitivity of the system, the adequacy of defensive program checks, and the efficiency of data consistency audits. This type of faulting was useful, for example, in testing response to transmission irregularities between the base and remote subsystems, such as the Remote Trunk Arrangement and Position Subsystem No. 2.[11]

Various manual actions at the Local Maintenance Position were also used to verify the proper response of the system in areas such as reconfiguration, diagnostics, measurements, control and display capabilities, and exercises. These types of actions are usually taken by the craftspeople both on a routine basis and in emergency-action conditions.

#### 3.3.2 *TSPS peripheral test sequence*

Communication between the SPC 1A and peripheral equipment is accomplished by the Central Pulse Distributor (CPD), Communications Bus Translator (CBT), Master Scanner (MS), Universal Trunk Scanner (UTSC), and Universal Trunk Signal Distributor (UTSD). Maintenance testing verified that the emulated maintenance programs detect malfunctions in the above units and take appropriate recovery and diagnostic actions.

The TSPS periphery consists of a number of functional entities or subsystems (see Table I) that are administered by duplicated controllers that interface with the SPC bus system. Extensive tests were performed to verify the operation and maintenance of these functional units under control of the TSPS No. 1B software.

In TSPS, trunks and service circuits provide the interface between the TSPS and external systems. For maintenance purposes, access to these circuits is required to evaluate performance and localize or

Table I—TSPS peripheral subsystems

| |
|---|
| Position and Trunk Link Network |
| Position Subsystem No. 1 |
| Peripheral Control Link |
| —Position Subsystem No. 2 |
| —Remote Trunk Arrangement |
| Station Signaling and Announcement Subsystem |
| Common Channel Interoffice Signaling Links |

isolate a trouble to a particular faulty unit. Comprehensive tests were required to ensure proper operation of these trunk and service circuit routines. Table II lists the TSPS trunks and service circuits for which maintenance routines were tested.

### *3.4 TSPS external interface tests*

A number of systems external to TSPS provide operator administrative functions, or interact with TSPS for services and maintenance. In general, these are self-contained systems, but rely on communications with TSPS over data links and/or channels, thus requiring coordinated testing between the two systems. The communications generally rely on specific protocols or message types. Testing of external interfaces verified that the necessary protocols, message handling, and maintenance states were operational in the TSPS No. 1B environment. Some of the external interfaces include:

(*i*) External administrative centers
(*ii*) The Hotel Billing Information System (HOBIS)[8]
(*iii*) The Service Evaluation System (SES)[9]
(*iv*) The Billing Validation Application (BVA)[1]
(*v*) The Switching Control Center System (SCCS).[10]

The interfaces in items (*i*) through (*iv*) above required only verification that TSPS No. 1B had not introduced operational or maintenance problems. The SCCS interface [item (*v*) above] required substantive testing because of new interface hardware and software (see Ref. 12). This interface was functionally tested between the system laboratories of the SCCS and TSPS development organizations. Field-site testing verified these functional capabilities and stressed load-related features that could not be tested earlier.

### *3.5 3B20D/DMERT system testing*

3B20D/DMERT testing evaluated 3B20D/DMERT hardware and software in the application environment. The 3B20D/DMERT tests concentrated on interfaces, resource utilization, and system response characteristics in the TSPS application environment. As such, overload response, resource audits, and software fault tolerance were extensively tested. The combined craft/machine interface and

Table II—Tests of TSPS trunk, service circuit, and maintenance circuits

| |
|---|
| Service Circuits |
| —Dial-Pulse receivers |
| —Multifrequency receivers |
| —Multifrequency outpulsers |
| —High-Impedance, Multifrequency receivers |
| —Coin Control and Ringback circuits |
| —Reorder Tone and Announcement circuits |
| —Audible Tone trunks |
| —Coin Detection and Announcement circuits—Type 1 |
| —Coin Detection and Announcement circuits—Type 2 |
| —Dual Tone Multifrequency Detection and Announcement circuits |
| —Multifrequency Announcement and Detection circuits |
| Trunk Test Panel |
| AMA maintenance |
| Recorded Announcement equipment |
| Tone and Interrupter circuit |
| Time of Day circuit |

DMERT administrative functions (field utilities and field update, for example) were also extensively exercised.

#### *3.5.1 Fault recovery, initialization, diagnostics, and overload*

The TSPS No. 1B maintenance strategy is based on the 3B20D/DMERT maintenance design augmented by additional capabilities unique to the TSPS application. These additions are mainly in areas where TSPS software directly interacts with the DMERT software, or when specific hardware interfaces are required for the TSPS application.

Specific fault-recovery strategies were developed by TSPS for the PSI, the interface between the 3B20D and the TSPS periphery.[12] In addition, the TSPS Application Integrity Monitor (AIM) was developed to directly interact with the DMERT system integrity monitor (SIM) to ensure the system integrity of TSPS No. 1B. DMERT system initialization and overload features were augmented by TSPS to establish an overall system initialization and overload strategy for the TSPS No. 1B.[13] These areas were tested extensively to determine the proper system response.

The 3B20D and PSI diagnostics, Trouble-Locating Procedures (TLP), and Routine Exerciser (REX) were tested by sample faulting techniques. Manual requests for removal/restoral were also used for verification of capabilities.

The TSPS No. 1 maintenance craft interface was replaced in TSPS No. 1B with a combined 3B20D Processor and TSPS maintenance craft interface. Testing in this area concentrated on combined craft response, software and hardware alarm interfaces, and an assessment of process priorities to provide sufficient terminal response time under load.

### 3.5.2 Administrative functions

Testing in this area concentrated in the areas of 3B20D recent change/verify (for equipment and system configuration database changes), system update (for installation of a "point" issue of a current generic or installation of a subsequent generic), and field update (for installation of one or more broadcast warning messages).

## 3.6 System evaluation runs

### 3.6.1 Purpose and definition

One goal of system testing was to monitor a set of quantified performance criteria to measure overall system quality and to ensure that system design goals were met. Of particular importance for TSPS No. 1B was the evaluation of system behavior at full-load call handling (maximum capacity). This was particularly important since a substantial number of TSPS No. 1 to TSPS No. 1B live in-service retrofits were scheduled for 1982 for offices at or approaching real-time overload of the SPC 1A. Since the Fresno site had a large amount of peripheral equipment, sufficient microprocessor-controlled call simulators and operator simulators were installed on a temporary basis to permit stressing the new TSPS No. 1B above the capacity design objective. This additional performance margin simulated variation among sites with respect to call mix and peakedness in the busy hour(s).

In the first quarter of 1981, a series of weekly System Evaluation Runs (SERs) were begun at both Fresno and San Antonio to measure system performance in an environment approximating post-cutover conditions. Typically, these SERs were executed on weekends and were 24 to 48 hours long. During these SERs the following conditions were met:

(*i*) Operating telephone company maintenance craft were assigned to monitor and maintain the office, performing all TSPS peripheral maintenance and trunk cutover testing simulating an in-service office.

(*ii*) Bell Laboratories and Western Electric site staff had instructed the maintenance craft in the use of normal 3B20D Processor and SPC 1B peripheral maintenance procedures. When craftspeople were troubleshooting equipment faults or potential design problems, normal field repair tools and techniques were used.

(*iii*) A simulated call load was applied to the system. At Fresno, the first quarter 1981 SER was conducted at a load equivalent to 25 percent of the system call capacity. This was gradually increased so that by mid-1981, SERs were conducted above the call-capacity design objective of the TSPS No. 1B.

### 3.6.2 Measurements and objectives

A team of designers, testers, and system analysts defined a set of eighteen measurements that were tracked as part of the SER each

Table III—Fresno Service Acceptance Test Performance* (9/27/81–10/5/81)

| | System Performance | First-Site Turnover Objective |
|---|---|---|
| Call Load (Percent of SPC 1A) | 175% | 160% |
| Mishandled Calls | 0.04% | <0.05% |
| Plug-in Replacements | 0 | <1/10 days |
| System Initializations | 0 | 0.02/wk |

* Includes normal maintenance and PT + T trunk testing.

week to summarize system stability and maintainability. For each measurement, a turnover objective was specified. If several results deviated widely from the objective, a high priority was placed on resolution of these problems. When the measurements in those areas approached the turnover objective, resources were redirected to improving other areas of system performance. Statistics were kept on the frequency and cause of initializations, interrupts, audits, and overloads. The duplex performance of the processor and its disk subsystem was tracked and the degree of automatic fault recovery and identification was evaluated. Hardware failure rates were closely monitored and compared with reliability models. All initial service objectives were achieved prior to cutover. Table III shows Fresno Service acceptance test performance versus key first-site turnover objectives.

## IV. SUMMARY

The strength of the integration and system test plan for the successful field introduction of the TSPS No. 1B involved the use of complementary techniques to set objectives and monitor progress. The comprehensive functional and regression testing, close tracking of correction and modification requests, and prompt integration and delivery of a diverse set of software changes were essential to the steady progress summarized by the periodically executed System Evaluation Runs.

An immediate benefit of this approach was the customer satisfaction expressed by Pacific Telephone and Southwestern Bell Telephone at turnover, cutover, and subsequent operation of their new TSPS No. 1B systems. Furthermore, the implementation of the comprehensive test program described herein is responsible for the excellent performance of the 37 offices now in service (35 of which were live in-service retrofits in high-traffic sites). Current Bell Laboratories and Western Electric efforts involve the continued coordination and support of the large number of TSPS No. 1B live in-service retrofits planned for the next few years.

## V. ACKNOWLEDGMENTS

The authors would like to acknowledge the dedicated efforts and innovative technical contributions of their staff and the test coordination provided by J. C. Dalby and his staff. Special recognition is also given to the 3B20D Development team, particularly J. H. Miller and her staff, for their valuable participation in the TSPS No. 1B testing program.

## REFERENCES

1. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.
2. E. M. Prell, V. L. Ransom, and R. E. Staehler, "The Changing Role of the Operator," Int. Switching Symp., Paris, France, May 1979.
3. Special issue entitled "3B20 Processor & DMERT Operating System," B.S.T.J., *62*, No. 1, Part 2 (January 1983).
4. J. P. Delatore, D. Van Haften, and L. A. Weber, "System Verification and Evaluation Procedures," B.S.T.J., *58*, No. 6, Part 1 (July–August 1979), pp. 1335–46.
5. T. G. Hack, T. Huang, and L. C. Stecher, "Traffic Service Position System No. 1B: Software Development System," B.S.T.J., this issue.
6. W. F. Klinksiek and H. L. Mitchell, "3B20D Processor and DMERT Operating System: System Integration and Test," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 399–410.
7. M. W. Rolund, J. T. Beckett, and D. A. Harms, "3B20D Processor & DMERT Operating System: 3B20D Central Processing Unit," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 191–206.
8. S. Michael and J. Vizcarrondo, "HOBIS: New Designs on Hotel Billing," Bell Lab. Rec. (January 1980), pp. 11–18.
9. T. R. Lehnert, "A Better Way to Measure the Quality of Telephone Service," Bell Lab. Rec., *59*, No. 6 (July–August 1981), pp. 186–9.
10. J. J. Bodnar, J. R. Daino, and K. A. Vandermeulen, "Traffic Service Position System No. 1B: Switching Control Center System Interface," B.S.T.J., this issue.
11. S. M. Bauman, R. S. DiPietro, and R. J. Jaeger, Jr., "Remote Trunk Arrangement: Overall Description and Operational Characteristics," B.S.T.J., *58*, No. 6, Part 1 (July–August 1979), pp. 1109–18.
12. G. T. Clark, H. A. Hilsinger, J. H. Tendick, and R. A. Weber, "Traffic Service Position System No. 1B: Hardware Configuration," B.S.T.J., this issue.
13. R. J. Gill, G. J. Kujawinski, and E. H. Stredde, "Traffic Service Position System No. 1B: Real-Time Architecture Utilizing the DMERT Operating System," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Retrofitting the Processor

By J. C. DALBY, JR., D. VAN HAFTEN, and L. A. WEBER

(Manuscript received June 30, 1982)

*At the end of 1981, over 150 Traffic Service Position System No. 1 (TSPS No. 1) offices were in service, equipped with Stored Program Control No. 1A (SPC 1A) processors. Some of these sites had reached the system capacity with respect to real time or memory. The new SPC 1B, which contains a 3B20 Duplex (3B20D) Processor and a Peripheral System Interface (PSI), provides the TSPS No. 1B with additional processor capabilities for additional capacity and future features. This article discusses the techniques used for achieving a smooth retrofit from the TSPS No. 1 to TSPS No. 1B with virtually no interruption of call processing. Special procedures and tools were developed to introduce the SPC 1B onto existing buses and to verify the interfaces with existing peripherals by means of a cycle-stealing mechanism, while the SPC 1A continues to handle call processing. These procedures were used successfully at the first such retrofit in Redwood City, California, on March 13, 1982. During 1982, 34 additional sites will be retrofitted by Western Electric to accomplish the initial phase of the planned retrofits to TSPS No. 1B.*

## I. INTRODUCTION

Over 150 Traffic Service Position System No. 1 (TSPS No. 1) offices[1] are in service in the United States. These systems give fast, efficient toll operator services to over 95 percent of the Bell System main stations. The TSPS No. 1 consists of a Stored Program Control No. 1A (SPC 1A) and numerous peripheral units. The SPC 1A has performed well since its initial introduction into service in Morristown, New Jersey, in 1969.[2]

The development of the TSPS No. 1B introduced a modern 3B20

Duplex (3B20D) Processor with its peripherals and a Peripheral System Interface (PSI) into TSPS.[3] The 3B20D Processor and the PSI together are called the SPC 1B.[4] The PSI connects the 3B20D to the existing TSPS peripherals in such a manner that the interface to the SPC 1B is the same as the interface to the SPC 1A.

Since many TSPS No. 1 sites are approaching real time and memory exhaust and the new processor capabilities are necessary for new features, a procedure has been developed to allow in-service TSPS sites to retrofit to the SPC 1B. This procedure requires advanced planning for floor space, connectorization of existing buses, and some hardware modifications. Special procedures and tools are used to ensure that the SPC 1B can properly interface with the existing periphery before the new processor is in control of call processing. After cutover and following sufficient soak time on the SPC 1B during which assurance tests are run, the SPC 1A can be removed from the office.

Unlike commercial processor upgrades, where the system is taken off-line, the replacement of the SPC 1A with the SPC 1B must be performed with virtually no interruption of call processing. For this reason, special procedures and tools were developed to prepare the site and check the new equipment and interfaces while the existing system continues to handle telephone traffic. These procedures had to be thorough and be able to completely verify the hardware configuration, since upon cutover the new system must be able to handle a large volume of traffic. Furthermore, since the existing peripherals are maintained, and the new processor does not control the peripherals until cutover, the final cutover procedures must be straightforward and fast. During the retrofit, the operator traffic is interrupted for only approximately the length of a system initialization on the new system.

This article discusses the techniques used to achieve a smooth retrofit from the TSPS No. 1 to TSPS No. 1B. These procedures were successfully used at the first such retrofit in Redwood City, California, on March 13, 1982. During 1982 34 additional sites will be retrofitted by Western Electric to accomplish the initial phase of the planned retrofits to TSPS No. 1B.

## II. SITE PREPARATION

The first step in retrofitting an in-service site is to make the site compatible with the SPC 1B. Site preparation can be planned by the operating telephone company for any time prior to the actual retrofit. The site preparation, both bus modifications and auxiliary unit relocation, is performed according to Western Electric procedures. These procedures do not require special-purpose, processor-replacement software in TSPS No. 1.

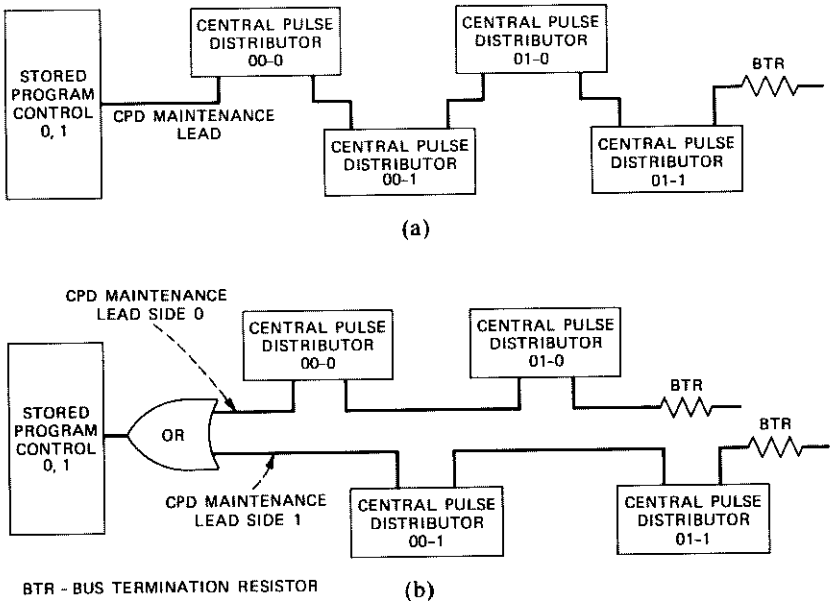



Fig. 1—(a) Simplex configuration for CPD maintenance lead. (b) Duplicated CPD maintenance lead.

### *2.1 Modifications of peripheral buses*

The TSPS No. 1 peripheral buses are modified to be compatible with the SPC 1B and to simplify introducing the SPC 1B. During this bus modification, call processing continues uninterrupted.

The Central Pulse Distributor (CPD) maintenance leads must be split to allow each of the 3B20D/PSI complexes access to the entire set of leads. At present the CPDs are daisy-chained along these complex CPD maintenance leads (see Fig. 1a*). These leads are duplicated by separating them and then feeding the EXCLUSIVE-OR of the two sets back to the processor (see Fig. 1b).

A set of miscellaneous leads are bundled into miscellaneous buses and are rerouted to the Communication Bus Translator (CBT). Each miscellaneous bus has a network reset lead, an auxiliary reset lead, a maintenance scanner lead, and three clock leads. The rerouting enables the clock leads to be disconnected from the SPC 1A when it is removed from the office.

Most TSPS No. 1 peripheral buses must be connectorized to allow insertion of the SPC 1B and the removal of the SPC 1A (see Fig. 2).

* Acronyms and abbreviations used in the figures and text of this paper are defined at the back of this Journal.

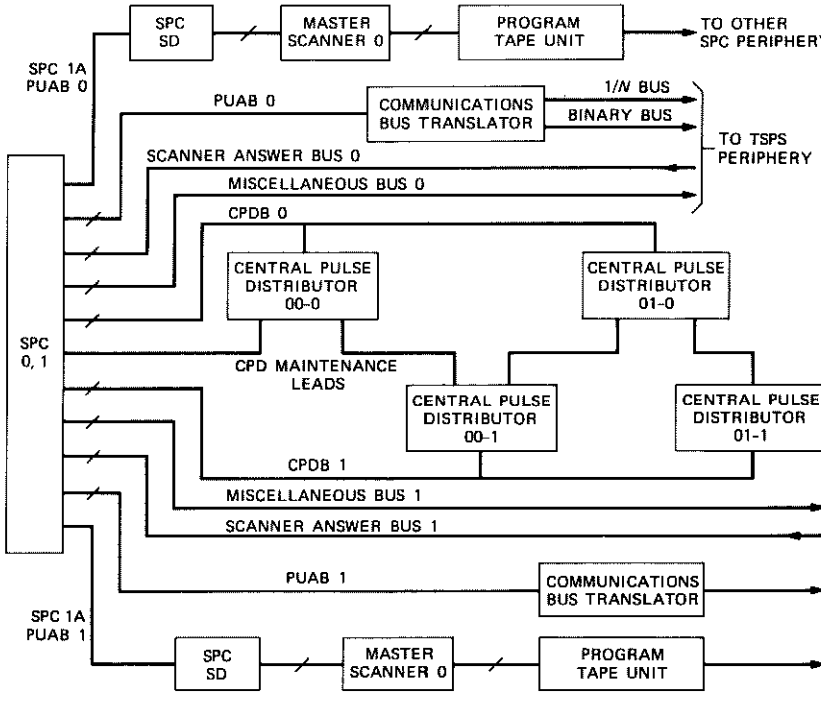



Fig. 2—TSPS No. 1 peripheral buses.

This is the most time-consuming activity of the processor replacement activities since 346 pairs of wires are involved. Connectorization is transparent to the SPC 1A operation. It requires a transition panel unit to be installed on site. The transition panel provides bus continuity for each pair of connectors that result. The panel is normally mounted on top of a cable rack near the location of the PSI frames or in an empty frame adjacent to the PSIs. The connectorization procedure is first run on one bus and then repeated for the mate bus.

In addition to the buses for the 3B20D Processor being connectorized, the SPC 1A Peripheral-Unit Address Bus (PUAB) and the one-out-of-$N$ (1/$N$) bus must be connectorized to accommodate the relocation of Master Scanner 0 (MS0).

### *2.2 Relocation of auxiliary units*

The SPC 1A Signal Distributor (SD), the Program Tape Unit (PTU), and MS0 reside on the SPC 1A PUAB. This bus is not utilized by the SPC 1B because the SD and PTU are replaced by 3B20D units.

However, MS0 must remain since it monitors TSPS units. This is resolved by moving the appearance of MS0 from the SPC 1A PUAB to the $1/N$ bus and by indicating the move in office data.

## III. RETROFIT PROCEDURES

After the site-preparation activities are completed, the actual retrofit procedures can begin. Two special software packages for the SPC 1A and the SPC 1B, plus special-purpose circuitry, are used to assist the retrofit.

### *3.1 Loading the SPC 1A retrofit programs*

Special retrofit software is loaded into the SPC 1A. It provides the capabilities for the initialization of TSPS periphery for 3B20D access tests, the bimodal feature of Station Signaling and Announcement Subsystem (SSAS) (see Section 3.2), and the ability to time-share buses with the 3B20D. The SPC 1A retrofit load overlays recent change programs and can be backed out whenever retrofit testing is at one of many safe stop points. Thus, the operating telephone company can have access to recent change capabilities when processor replacement programs are not in use.

### *3.2 Modification of bus access to SSAS announcement stores*

Processor replacement also requires the modification of the bus access to the SSAS announcement stores.[5] The Announcement Stores (AST) reside on the SPC 1A store bus (see Fig. 3). Since the store bus is not retained for the SPC 1B, the only access to the ASTs for loading announcements and for maintenance is through the SSAS controller after the SPC 1B retrofit. However, during the retrofit interval, the store bus access to the ASTs is left intact but not used. This allows emergency loading of ASTs if mate failures occur. The modification of bus access to the SSAS announcement stores is accomplished one SSAS side at a time. Therefore, for a short period, one SSAS side's AST is accessed via the SPC 1A store bus while the other SSAS side's AST is accessed through the SSAS controller. To handle this unique access situation, special-purpose bimodal software included in the SPC 1A retrofit program must remain loaded. This software ensures SSAS integrity during the bus modifications. As part of the modification the firmware in the SSAS controller is upgraded. This new firmware is designed to interface with both the SPC 1A and SPC 1B.

After the SSAS bus access modification is complete, the system is ready to have the new SPC 1B introduced on the buses. Figure 4 shows the site configuration.

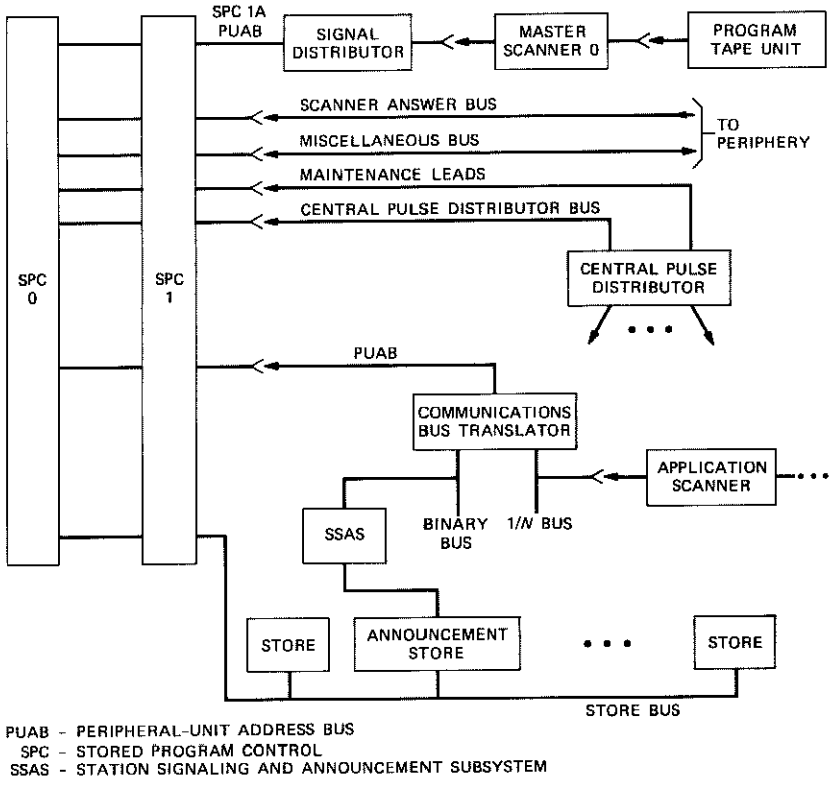



Fig. 3—Connectorized TSPS buses.

### *3.3 SPC 1B installation and testing*

In parallel with or subsequent to site preparation, the SPC 1B is installed. First, the 3B20D Processor is brought up in a stand-alone configuration by Western Electric according to standard procedures using the Duplex Multi-Environment Real Time (DMERT) operating system[6] and 3B20D diagnostics. Next, the PSIs are installed and connected to both halves of the 3B20D. Standard installation procedures are used to check out the connection. These procedures use the PSI diagnostic phases that test all internal PSI circuitry up to the bus drivers.

### *3.4 Insertion of the SPC 1B*

The SPC 1B is now ready to be inserted onto the TSPS peripheral buses. This involves removing the bus connectors on top of the transition panel and then inserting them into the inductor/transformer area of the PSI frame. As a result, the PSI not only has access to the TSPS peripheral buses, but also gives bus continuity when idled or

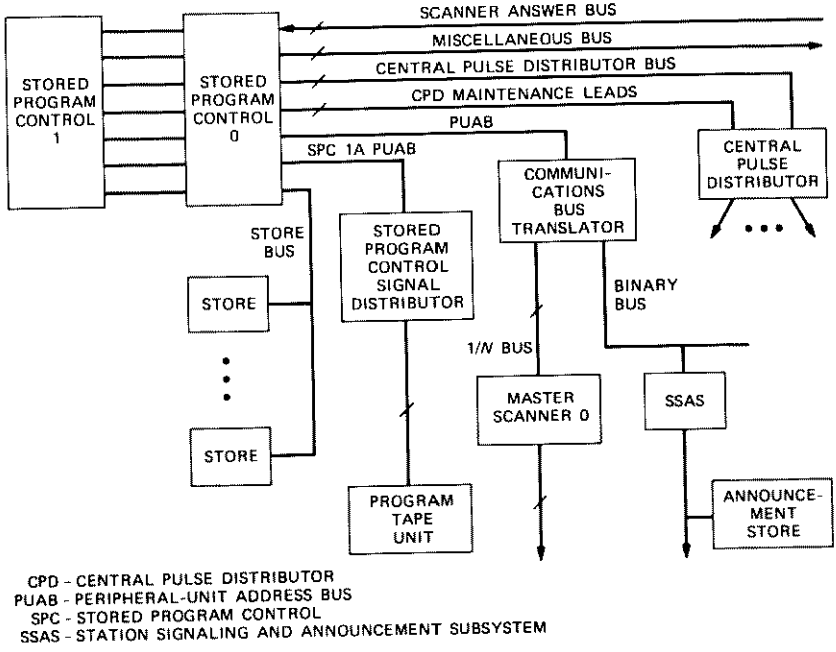



Fig. 4—Site preparation.

powered down. The retrofit software ensures that the SPC 1B complex never accesses the buses at the same time that SPC 1A does. The SPC 1A and SPC 1B operate in a time-sharing mode.

The insertion procedure connects one bus at a time utilizing existing TSPS diagnostics to verify correct movement. During the insertion the PSI access to the miscellaneous bus (see Section 2.1) is disabled to ensure that there is only one processor sending out signals over the clock and reset leads at any one time. Not until cutover does the SPC 1B have control of these leads.

### *3.5 SPC 1A/3B20D interface hardware and software*

Some additional special retrofit hardware and software is required to provide synchronization between the SPC 1A and the 3B20D for bus time-sharing. Every 25 ms the SPC 1A allows the 3B20D complete access to the TSPS peripheral buses for a period of 2 ms. It does this by sending the 3B20D a "start" (interrupt) signal. During the 2-ms interval, the SPC 1A keeps itself cycling. At the end of 2 ms, the SPC 1A sends the 3B20D a "stop" signal by disabling the PSI and then regains control of the buses. The SPC 1A sends out the "start" and "stop" signals via the unused CPD Execute leads in the office (see Fig. 5).

The SPC 1A retrofit software controls the bus time-sharing through

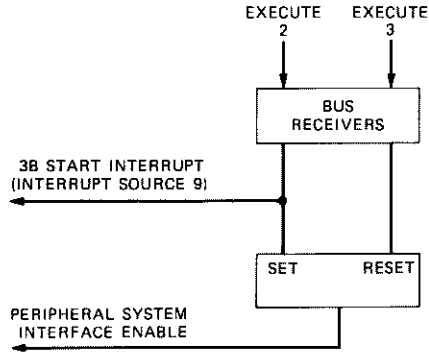



Fig. 5—Start and stop signals from SPC 1A.

the use of a base-level teletypewriter (TTY) message-handler program and a J-level program with a 25-ms entry rate. For a particular retrofit test, the SPC 1A software checks for the proper system configuration. If the system is not configured correctly, the retrofit test request is rejected; otherwise, the SPC 1A starts sending the proper start and stop pulses to the SPC 1B. Every 25 ms the SPC 1A sends an interrupt to the 3B20D ("go-ahead" signal) and, at the same time, an enable to the appropriate PSI frame. After the SPC 1A has cycled for 2 ms, it sends a disable to the PSI frame; the SPC 1A now has access to the peripheral buses for approximately 23 ms, and the cycle then repeats itself. Hence, the 3B20D must run its retrofit tests in 2-ms intervals so that it does not exceed its allotted time on the buses.

If for any reason the SPC 1A encounters a hardware interrupt during retrofit testing, it automatically stops sending interrupts to the 3B20D and initializes the SPC 1A retrofit software before exiting. The system must be restored to normal and the problem cleared before retrofit testing can continue.

The SPC 1B retrofit software has three major capabilities: an input message handler, an output message handler, and the retrofit test and control software. The input message handler, which is a user process, interprets the input message typed on the maintenance terminal. It then sends a message to the retrofit test and control software at the kernel level, which executes the requested test. Only one retrofit test can be run at a time. Most tests run in their entirety during the 2-ms interval. The emulated diagnostics, however, must run a section (<2 ms) at a time until completion. The retrofit control software ensures that this takes place. Finally, the output message handler (a user process) is used to print out any raw data words generated by the emulated diagnostics. This printing occurs on the maintenance terminal after the kernel process sends a message to the output message handler.

### 3.6 Testing of the 3B20Ds access to TSPS periphery

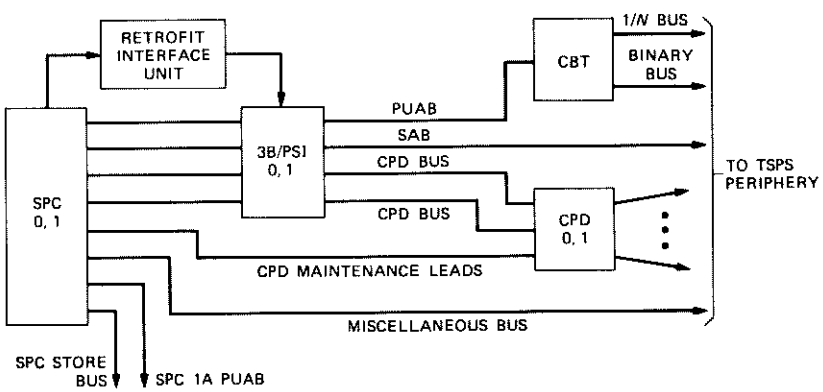
A special set of retrofit tests and procedures have been developed to verify that the processor has been replaced correctly. These procedures and tests verify the ability of the SPC 1B to drive the TSPS peripheral buses, and check that correct connections are made between the SPC 1B and TSPS peripherals. The system configuration during this testing is shown in Fig. 6. The retrofit tests include:

(*i*) SPC 1A/3B20D Interface Test—The interface test verifies numerous capabilities needed for subsequent tests. The SPC 1A must be sending start and stop signals to the SPC 1B complex, the time-share cable containing the CPD execute leads must be wired properly, and the hardware interface must be properly converting the signals from the SPC into a 2-ms enable pulse for the PSI.

(*ii*) CPD Scope Test—The CPD scope test verifies the proper bus connection and the system's ability to drive each lead. It involves having the SPC 1B send specific patterns of pulses equal to the lead number over the CPD leads and verifying them.

(*iii*) Communications Bus Translator (CBT) Scope Test—The CBT scope test verifies the proper connectorization and driving of the CBT leads. It involves having the SPC 1B send specific patterns of pulses equal to the lead number over the PUAB to the CBT and verifying them.

(*iv*) Central Pulse Distributor (CPD) Diagnostic—The CPD diagnostic is divided into two sections. One section runs with the CPDs inhibited. This section uses diagnostic circuitry in the CPDs to verify CPD bus input and appropriate responses. The other section runs with




CBT - COMMUNICATIONS BUS TRANSLATOR
CPD - CENTRAL PULSE DISTRIBUTOR
PSI - PERIPHERAL SYSTEM INTERFACE
PUAB - PERIPHERAL-UNIT ADDRESS BUS
SAB - SCANNER ANSWER BUS
SPC - STORED PROGRAM CONTROL

Fig. 6—Retrofit testing configuration.

the CPDs uninhibited. It checks the CPD operation by sending actual orders. This test marks the first time during retrofit testing that an actual CPD enable is sent over the buses. If either section fails, it prints out raw data words identical to the printout of the CPD diagnostic with the TSPS No. 1 generic.

(*v*) Station Signaling and Announcement Subsystem (SSAS) Loop-Around Test—The SSAS loop-around test consists of selected phases of the SSAS diagnostic. The Station Signaling and Announcement Subsystem (SSAS) receives a predetermined pattern via the CBT and returns the pattern on the Scan Answer Bus (SAB). The SPC 1B then checks to see if the loop-around was successful. This test verifies the integrity of CBT and SAB, and the integrity of all CPD and CBT bus routes to SSAS.

(*vi*) Miscellaneous Tests—The miscellaneous tests are required to check the remaining peripheral bus leads. These tests are run in two modes: one with the SPC 1A sending out pulses over these leads, the other with the 3B20D sending pulses. The SPC 1A-driven test checks the correct connections for the miscellaneous leads to the SPC 1B complex, while the 3B20D-driven test verifies its ability to send out proper pulses over these leads.

### *3.7 Mate SPC 1B complex*

Retrofit testing must be successful on both halves of the SPC 1B's (side 0 and side 1) before cutover. Therefore, all of the retrofit procedures that are run from one side must be executed from the mate side. This requires installing the SPC 1A/3B20D hardware interface on the mate SPC 1B complex. Once the hardware interface has been installed, retrofit testing can be performed from the mate SPC 1B.

## IV. OFFICE DATA TRANSFER AND RETROFIT

After the ability of the SPC 1B to interface with the buses is verified, the office data from the SPC 1A must be transferred to the SPC 1B. An on-line method has been developed to do this. Special wiring is installed between the SPC 1B and the teletypewriter buffer of the SPC 1A. This wiring allows the 3B20D to be viewed as a teletypewriter from the SPC 1A. After this connection is made, the 3B20D sends memory-read commands to the SPC 1A. The results of the reads are checked for correct parity and are reread when errors are found. The rereading ensures correctness. These reads continue until the entire office data spectrum is transferred.

After the office data have been transferred to the 3B20D, a special office data retrofit program is run. This program performs the required changes in the data structures and contents to make the data compat-

ible with TSPS No. 1B. It does not introduce new office-dependent data or modify any assignments specified in office-dependent data.

## V. CUTOVER PROCEDURES

Once all retrofit tests are successfully completed and the office data is transferred, the retrofit hardware interface is removed and the office data are merged with the official TSPS No. 1B generic software. To begin the actual cutover, the two miscellaneous bus connectors, which include the system clock leads, are connected to the inactive SPC 1B. The clock in the SPC 1A is stopped and the SPC 1B is initialized. Call processing is interrupted only from the time the SPC 1A clock is stopped until the resulting SPC 1B initialization is complete, usually less than 2 minutes. At this point, the 3B20D processors should be handling the TSPS call load. If any difficulties are encountered at this time, the SPC 1A configuration can still be brought up by powering down the PSI frames, restarting the SPC 1A clocks, and initializing the SPC 1A.

After cutover, a series of system diagnostics and verification tests are performed. In addition, a recommended soak interval of a few days is allowed to ensure system operation. Once the operating telephone company has determined that the TSPS No. 1B is performing satisfactorily, unneeded equipment can be removed by Western Electric. This removal requires no additional software or hardware capabilities.

## VI. SUMMARY

The first live retrofit was performed on March 13, 1982, in Redwood City, California. No major difficulties were encountered in this retrofit. During the remainder of 1982, 34 additional retrofits occurred.

## VII. ACKNOWLEDGMENTS

A number of people made contributions to the development and design of these retrofit procedures. Significant contributions were made by Carl Amodio, Ken Handy, Frank Maesky, and D. A. Wood of Western Electric and by Mark Koehler, Kevin Kulhanek, Ron Michelsen, J. D. Peterson, Dennis Shank, R. A. Tengelsen, Neil Walgenbach, Stan Windes, and several others of Bell Laboratories.

## REFERENCES

1. R. J. Jaeger, Jr., and A. E. Joel, Jr., "TSPS No. 1: System Organization and Objectives," B.S.T.J., *49*, No. 3 (December 1970), pp. 2417–43.
2. G. R. Durney, H. W. Kettler, E. M. Prell, G. Riddell, and W. B. Rohn, "TSPS No. 1: Stored Program Control No. 1A," B.S.T.J., *49*, No. 3 (December 1970), pp. 2445–508.
3. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.

4. G. T. Clark, H. A. Hilsinger, J. H. Tendick, and R. A. Weber, "Traffic Service Position System No. 1B: Hardware Configuration," B.S.T.J., this issue.
5. G. T. Clark, K. Streisand, and D. H. Larson, "TSPS No. 1: Station Signaling and Announcement Subsystem: Hardware for Automated Coin Toll Service," B.S.T.J., *58*, No. 6 (July-August 1979), pp. 1225–49.
6. R. J. Gill, G. J. Kujawinski, and E. H. Stredde, "Traffic Service Position System No. 1B: Real-Time Architecture Utilizing the DMERT Operating System," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Capacity and Reliability Evaluation

By B. A. CRANE and D. S. SUK

(Manuscript received June 30, 1982)

*This paper describes the prediction and evaluation of the call-processing capacity and reliability resulting from use of the 3B20 Duplex (3B20D) Processor in the Traffic Service Position System No. 1B (TSPS No. 1B). The call-processing capacity was predicted using a processor real-time model whose parameter values were determined by laboratory and test-site measurements. The system reliability was predicted using Markov modeling techniques. Performing an evaluation during TSPS No. 1B development provided a means for monitoring progress toward meeting the capacity and reliability objectives.*

## I. INTRODUCTION

One of the development objectives of the Traffic Service Position System No. 1B (TSPS No. 1B) was to improve the call-handling capacity of the TSPS No. 1 by replacing the Stored Program Control No. 1A (SPC 1A) with the Stored Program Control No. 1B (SPC 1B).[1] The SPC 1B consists of the 3B20 Duplex (3B20D) Processor together with the Peripheral System Interface (PSI) unit, which adapts the 3B20D to existing TSPS peripherals. The 3B20D is microprogrammed to execute the SPC 1A instructions, thus allowing TSPS call-processing software developed for the SPC 1A to be ported to the SPC 1B with minimal changes. This emulated TSPS software executes as a kernel process under the DMERT operating system. References 2 and 3 contain further details on the SPC 1B architecture.

### *1.1 Call-processing capacity analysis*

The increased speed of the SPC 1B in executing the emulated SPC 1A instructions provides the increase in call-processing capacity. The

initial objective for capacity increase established for TSPS No. 1B was that the SPC 1B call-processing capacity should be at least 160 percent that of the SPC 1A.

Early in TSPS No. 1B development, a capacity prediction and evaluation plan was established for monitoring the progress in meeting the capacity improvement objective. This plan involved formulating a mathematical model of the SPC 1B real-time usage, where the parameters of this model represent the various call-processing and overhead activities performed by the processor. Laboratory and test-site measurements of these parameters during development provided, through use of the real-time model, estimates of the call-processing capacity. In this way, any problem areas having an adverse effect on call-processing capacity could be identified as candidates for improvement during continued development. This same real-time model, at the completion of development, has been incorporated into the TSPSCAP program[4] used by the operating telephone companies to determine the call-processing capacity of specific TSPS No. 1B sites. The formulation of this real-time model, the laboratory and test-site measurement techniques, and the resulting capacity performance data are described in subsequent sections of this paper.

### *1.2 System reliability analysis*

An important step in the development of highly reliable switching systems is the prediction of their reliability. To provide uninterrupted service, TSPS No. 1B has the same reliability objectives as other Bell System electronic switching systems (ESSs), namely: an average downtime of less than 3.0 minutes per year.[1] In TSPS No. 1B, most of the TSPS peripherals are retained and their maintenance strategy remains virtually unchanged from the TSPS No. 1. Thus, the reliability objectives of the TSPS peripherals will not change in TSPS No. 1B from 1.0 minute per year average downtime and, consequently, the SPC 1B reliability must achieve the objective of less than 2.0 minutes per year average downtime.

To predict the reliability of SPC 1B hardware, continuous-time, finite-state Markov models were used. The Markov model approach for the reliability calculation of repairable systems is described in Ref. 5. Throughout the development period of TSPS No. 1B, the reliability model was updated to accurately reflect architectural modifications or design changes in the subsystems. The reliability estimates of various configurations were compared to monitor the system reliability, to identify limiting subsystems, and to determine if modifications would improve the overall reliability. This will be described in subsequent sections of this paper.

## II. CAPACITY EVALUATION

### *2.1 Approach taken*

The approach taken to model the real-time usage of the SPC 1B was to modify the real-time model of the SPC 1A.[6] Emulation of the SPC 1A code by the SPC 1B made this approach possible. The modified model contains parameters that represent the speedup in SPC 1B instruction execution relative to the SPC 1A and, also, the effects of the DMERT operating system.

These modifications were characterized by making measurements of real-time usage at various call loads ranging from idle to over 160 percent of the SPC 1A capacity. Call loads were applied through use of electronic, programmable call generators attached to TSPS trunks. The response of TSPS operators to these calls was simulated by other electronic, programmable units. The measurements of real-time usage were made by non-interfering monitoring equipment, which sampled and recorded the system execution state every 10 microseconds. Other measurements consisted of various TSPS traffic counts periodically printed out on the standard output devices.

### *2.2 TSPS No. 1 capacity analysis*

Because the SPC 1A real-time model forms the basis for the SPC 1B real-time model, it is briefly described here. References 6, 7, and 8 should be consulted for greater detail.

#### *2.2.1 SPC 1A software architecture*

During normal operation, most of the real-time usage of the SPC 1A occurs at two priority levels, called J-level and base level. J-level has the higher priority of the two, and is entered every 5 ms through a hardware interrupt to perform necessary input/output operations involved in communicating with the TSPS peripherals. Although a higher-priority H-level is also involved in these operations, H-level and J-level will hereafter be jointly referred to as J-level except where distinction is necessary. In the SPC 1A, base-level work has the lowest system priority and is performed whenever there are no higher-priority interrupts. Most of the call-processing work is performed in base level.

Each base-level program is assigned to one of five classes of work: A, B, C, D, or E. Each class is periodically visited by a control program to determine whether there is any work to do and to perform the work if present. The control program endlessly repeats the following fixed visitation sequence:

$$\cdots \text{ABACABADABACABABACABADABACABAE} \cdots .$$

We can see that from one class-E visitation to the next, termed an E-E cycle, the five classes are visited according to the ratio

$$A:B:C:D:E = 15:8:4:2:1.$$

Base-level programs are assigned to these classes in accordance with the acceptable delays in their execution; class A contains those programs requiring fastest response.

The time duration of an E-E cycle increases with the call load because, as the call load increases, there is more work to be done during each class visitation. However, a fixed amount of base-level work must be performed no matter what the call load is (e.g., determining if there is any work to do) and this work is referred to as the E-E cycle overhead.

#### 2.2.2 SPC 1A real-time model

The real-time model developed for the SPC 1A consists of the equation

$$900 = t_N N + T_{CR} + t_E E. \tag{1}$$

This equation expresses how a quarter-hour (900 seconds) of processor real time is shared by three different kinds of processor work: trunk-seizure work, represented by $t_N N$; constant-rate work, represented by $T_{CR}$; and E-E cycle overhead work, represented by $t_E E$. Each of these three terms is expressed in seconds per quarter-hour.

A TSPS call begins as a seizure (request for service) of a special TSPS trunk from a local office to a toll office. Most trunk seizures result in completed TSPS calls, but a few become uncompleted attempts because of customer abandonments, busy circuits, etc. Although these uncompleted attempts do not require as much processor real time as completed calls, they must be included as part of the processor real-time load. In the real-time equation, $N$ represents the number of trunk seizures per quarter-hour; and $t_N$ represents the average amount of processor real time (in seconds) required per trunk seizure. The value of $t_N$ depends on the mix of various types of completed TSPS calls and uncompleted attempts. About two-thirds of $t_N$ occurs in TSPS base level, and the other third in J-level. The TSPS call-processing capacity is expressed in terms of trunk seizures per quarter hour.

Constant-rate work is the processor work that is performed at fixed time intervals and is independent of trunk-seizure rate. For example, one type of TSPS trunk is scanned every 100 ms to determine whether a trunk seizure has occurred. The value of $T_{CR}$, in seconds per quarter hour, depends on the number of TSPS peripherals in use, and most of this time is spent in J-level.

The E-E cycle overhead work uses all processor real time not used by trunk seizures or constant-rate work. $E$ represents the number of E-E cycles that are executed per quarter hour, and $t_E$ represents the

average processor real time (in seconds) spent per E-E cycle in doing E-E cycle overhead work, which is independent of trunk-seizure load. By definition, all of this time occurs in base level.

Equation (1) is linear in terms of $E$ and $N$. Figure 1 plots $E$ as a function of $N$ for a typical SPC 1A site. Such a plot is referred to as a load line, which describes how the E-E cycle rate, $E$, varies with respect to the trunk-seizure rate, $N$. The slope of this load line is $-t_N/t_E$ and the intercept, corresponding to an idle system (i.e., when $N = 0$), is $(900 - T_{CR})/t_E$.

Also shown in Fig. 1 is a value of $E$, called $E_{MIN}$, which is the lowest E-E rate that can be sustained while still providing adequate system response. At rates below $E_{MIN}$ the visitation rate to the previously described base-level classes of work becomes too low and delays in serving requests become too long to meet service criteria. The trunk-seizure rate corresponding to $E_{MIN}$ is defined as the quarter-hour trunk-seizure capacity, $N_{CAP}$.

### 2.2.3 TSPSCAP program

TSPSCAP is an interactive, time-shared program used by the operating companies to determine the trunk-seizure capacity of specific TSPS sites. The user inputs the call mix and hardware configuration of a site, and TSPSCAP calculates the values of $t_N$, $T_{CR}$, and $E_{MIN}$ corresponding to these input values for use with the above real-time equation. TSPSCAP then outputs the value of $N_{CAP}$ for that site, together with auxiliary information such as an equation for the $E$

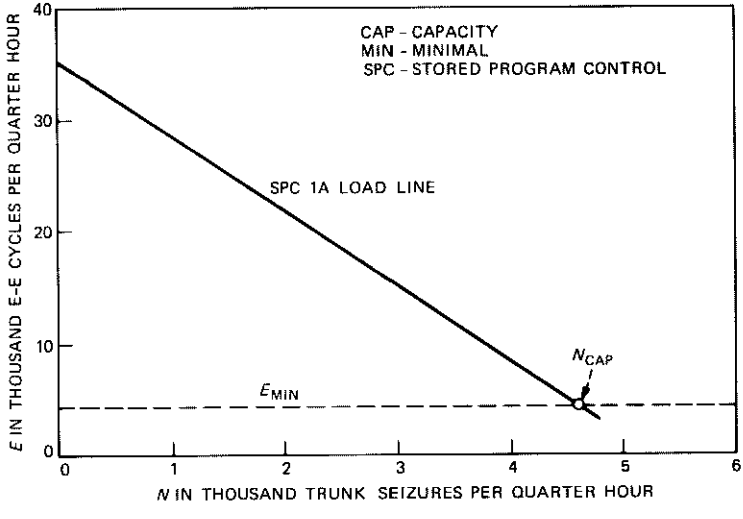



Fig. 1—Typical SPC 1A load line.

versus $N$ load line. This information is used by the operating telephone companies in growth planning to determine how close a TSPS No. 1 site is to its capacity limit.

### 2.3 TSPS No. 1B capacity analysis

As we mentioned earlier, emulation of the TSPS No. 1 software allows construction of the SPC 1B real-time model by modifying the SPC 1A real-time model. The modifications represent the speedup in instruction execution and the effects of the DMERT operating system. To understand these modifications the reader should know how the emulated code executes in the SPC 1B environment. This is briefly described below; Ref. 3 should be consulted for a more complete description.

#### 2.3.1 SPC 1B software architecture

**2.3.1.1 System execution levels.** The DMERT operating system has sixteen execution levels (ELs), numbered 0 through 15, that determine the relative priorities for process execution; EL 15 has the highest priority. Kernel processes can use ELs 15 through 2, and supervisor/user processes are restricted to ELs 1 and 0. The emulated TSPS call-processing software executes as a kernel process.

Table I shows the ELs for those processes that influence the SPC 1B real-time usage. H-level and high-priority J-level of the emulated TSPS process execute at EL 12, and low-priority J-level executes at EL 11. Base level executes at EL 5. The DMERT timer, at EL 15, provides a timing function for other processes by notifying a requesting process after a specified time period has elapsed. Processes involved with I/O, file management, and memory management execute at ELs 10, 7, and 2, respectively. The scheduler at EL 2 schedules the supervisor/user processes at ELs 1 and 0. Diagnostics for the 3B20D and PSI execute at EL 0, whereas diagnostics for the TSPS peripherals remain as part of class-E work in emulated TSPS base level at EL 5. The new TSPS craft interface software, which uses DMERT facilities to provide maintenance input-output message capability and system

Table I—DMERT execution levels

| Execution Level | Process |
|---|---|
| 15 | Timer |
| 12, 11 | TSPS J-Level |
| 10 | Disk Driver |
| 7 | File Manager |
| 5 | TSPS Base-Level |
| 2 | Memory Manager, Scheduler |
| 1, 0 | SPC 1B Diagnostics, TSPS Craft Interface, Other Supervisor/User Processes |

status display, also executes at EL 0. Software for the other TSPS output messages (e.g., for periodic traffic counts) remains as part of the emulated J-level and base-level code.

A 3B20/DMERT timer hardware interrupt occurs every 10 ms to service any timing requests. As with the SPC 1A, a J-level interrupt occurs every 5 ms. These two interrupts are synchronized such that the J-level interrupts lead the timer interrupts by 1 ms. As with the SPC 1A, the emulated base level executes whenever nothing at a higher EL is executing, with the exception that base level periodically relinquishes control (goes to sleep) so as to allow processes at lower ELs to execute.

Just before going to sleep, base level requests that the DMERT timer wake it after a specified period has elapsed. The low-level processes at ELs 4 through 0 can then execute, subject to interrupts by processes at higher ELs. However, if all the low-level processes complete their work before the timer awakens base level, then base level is prematurely awakened by a software interrupt and the pending timer request is deactivated. Thus, any real time not needed by the low-level processes is given back to base level, which uses this real time to execute additional E-E cycles.

***2.3.1.2 Speedup factors.*** The increased speed of the SPC 1B causes a net speedup in the execution of the emulated TSPS process relative to the SPC 1A. Not all portions of the emulated code experience the same degree of speedup, however, because of dependence on dynamic instruction mix, cache hit ratio, and ATB hit ratio.

The dependence on dynamic instruction mix occurs because some SPC 1A instructions could be emulated more efficiently than others. Also, in the SPC 1B, the execution time of some emulated SPC 1A instructions depends on what instruction options (e.g., rotating and masking) are exercised, whereas no such dependency exists in the SPC 1A.

To reduce memory access time, the SPC 1B employs a cache memory to contain the most recently accessed words of main memory. The cache is searched prior to each memory access and, if the word is in the cache (i.e., a cache hit), less real time is used because main memory need not be accessed. The cache is shared in common by all processes in the system.

The SPC 1B also employs eight Address Translation Buffers (ATBs), which speed up the task of translating from virtual memory address to physical memory address. Each ATB is essentially a cache memory that contains the physical addresses of the most recently accessed pages of virtual memory assigned to that ATB (a page is a 512-word block of main memory). If the page address is not in the ATB (i.e., an ATB miss), extra time is used in translation, which can

increase an instruction's execution time. To reduce ATB misses, J-level is exclusively assigned to one ATB and base level is exclusively assigned to another.

Parameters called "speedup factors" have been introduced to characterize the increased speed of the SPC 1B in executing the emulated TSPS process. Because of the effects of cache and ATB hits, speedup factors apply to execution of portions of code rather than to individual instructions. Thus, the speedup factor for a given portion of emulated code depends on the mix of executed instructions, and on the cache and ATB hit ratios experienced by those instructions.

*2.3.1.3 DMERT operating system.* For TSPS No. 1B, some real-time requirements of the DMERT operating system are application independent and others are application dependent. The application-independent requirements are for those functions that are necessary for maintaining a stable system environment. For example, the real time allocated to diagnose the 3B20D Processor would fall into this category. The application-dependent requirements are for those TSPS functions that make use of DMERT-supplied facilities. Two examples are: the real time required by the new TSPS craft interface, and the real time required to interface DMERT to TSPS J-level.

Parameters have been introduced that represent the combined TSPS-independent and TSPS-dependent DMERT real-time requirements for TSPS No. 1B. One parameter represents the combined high-level requirements (at ELs 15 through 5), and a second represents the combined low-level requirements (at ELs 4 through 0). Other parameters represent the real time used in handling TSPS J-level interrupts and base-level sleep requests.

### 2.3.2 SPC 1B real-time model

The SPC 1B real-time model is formed by adding speedup and operating system parameters to eq. (1) so as to obtain the new equation:

$$900 = t'_{\mathrm{N}} N' + T'_{\mathrm{CR}} + t'_{\mathrm{E}} E' + T'_{\mathrm{H}}, \tag{2}$$

where

$N'$ = trunk-seizures per quarter hour serviced by the SPC 1B
$E'$ = E-E cycles per quarter hour executed by the SPC 1B
$T'_{\mathrm{H}}$ = seconds per quarter hour used by high-level processes (at ELs 15 through 5) associated with TSPS-independent and TSPS-dependent DMERT work

and where $t'_{\mathrm{N}}$, $T'_{\mathrm{CR}}$, and $t'_{\mathrm{E}}$ are as defined in the following paragraphs.

The value of $t'_{\mathrm{N}}$, the average processor seconds per trunk seizure, is defined as

$$t'_N = \frac{t_{NB}}{K_{NB}} + \frac{t_{NJ}}{K_{NJ}}, \tag{3}$$

where

$t_{NB}$ = average processor seconds used per trunk seizure by the SPC 1A in base level
$K_{NB}$ = SPC 1B speedup factor for $t_{NB}$
$t_{NJ}$ = average processor seconds used per trunk seizure by the SPC 1A in J-level
$K_{NJ}$ = SPC 1B speedup factor for $t_{NJ}$.

Separate base-level and J-level speedup factors are defined because base-level and J-level each has its own dynamic instruction mix and, also, its own ATB and associated ATB hit ratio.

The value of $T'_{CR}$, the processor seconds per quarter hour of constant-rate work, is defined as

$$T'_{CR} = \frac{T_{CRB}}{K_{CRB}} + \frac{T_{CRJ}}{K_{CRJ}} + 18 \times 10^4\, t_{DJ}, \tag{4}$$

where

$T_{CRB}$ = processor seconds per quarter hour used in constant-rate work by the SPC 1A in base level
$K_{CRB}$ = SPC 1B speedup factor for $T_{CRB}$
$T_{CRJ}$ = processor seconds per quarter hour used in constant-rate work by the SPC 1A in J-level
$K_{CRJ}$ = SPC 1B speedup factor for $T_{CRJ}$
$t_{DJ}$ = processor seconds used by the SPC 1B in handling each J-level interrupt.

Separate base-level and J-level speedup factors are defined for the same reasons stated above.

The value of $t'_E$, the processor seconds per E-E cycle to perform E-E cycle overhead work in base level, is defined as

$$t'_E = \frac{t_E}{K_E} + b(t_{DB} + \bar{s}A), \tag{5}$$

where

$t_E$ = processor seconds used per E-E cycle by the SPC 1A in performing E-E cycle overhead work in base level
$K_E$ = SPC 1B speedup factor for $t_E$
$b$ = number of base-level sleep periods executed per E-E cycle by the SPC 1B
$t_{DB}$ = processor seconds used by the SPC 1B in handling each base-level sleep-period request
$\bar{s}$ = average duration (in seconds) of each base-level sleep period

$A$ = average fraction of each base-level sleep period that is available to low-level processes (at ELs 4 through 0).

In this formulation it can be seen that the real time used by the low-level processes is treated as part of the SPC 1B E-E cycle overhead. The value of $A$ decreases as high-level interrupts increase and, therefore, $A$ decreases as call load increases.

The values of $b$ and $\bar{s}$ must satisfy the constraint

$$b\bar{s}AE^* = T'_L, \tag{6}$$

where

$T'_L$ = seconds per quarter hour to be allocated to low-level processes (at ELs 4 through 0) associated with TSPS-independent and TSPS-dependent DMERT work

$E^*$ = lowest SPC 1B E-E cycle rate at which $T'_L$ is to be allocated by base-level sleep periods.

At E-E cycle rates less than $E^*$, insufficient base-level sleep periods will occur to satisfy eq. (6). At E-E cycle rates higher than $E^*$, more than $T'_L$ can be used by low-level work if necessary.

The value of $b$ is a software parameter, and the value of $\bar{s}$ is determined by the value of $s$, which is another software parameter. When base level goes to sleep, it requests that it be awakened after $s$ milliseconds have elapsed. Because this request can be made at any time relative to the 10-ms DMERT timer interrupt, $\bar{s}$ is around 5 ms longer than $s$.

Equations (2) through (6) constitute the SPC 1B real-time model.

#### 2.3.3 Determination of real-time model parameters

The newly introduced SPC 1B real-time parameters have been characterized through measurements made in the TSPS system laboratories and at the test site in Fresno, California, prior to cutover. The basic measurement technique involved measuring the percentage of processor real time used at each execution level under a number of different loads applied to the system. Other auxiliary measurements were also made.

**2.3.3.1 Real-time measurement techniques.** Processor real-time usage at the sixteen execution levels was measured through use of *Dynaprobe** monitoring equipment manufactured by the NCR COMTEN Corporation. The *Dynaprobe*, through means of high-impedance probes attached to the SPC 1B backplane, was used to sample the execution-level bits of the Program Status Word (PSW) every 10 microseconds to determine the relative frequencies of execution-level

* Registered trademark of NCR COMTEN Corporation.

occupancies. Other signals were also sampled at the same time to: (*i*) distinguish between emulated and native-mode code; (*ii*) count the number of times that each emulated SPC 1A instruction was executed during the measurement period; and (*iii*) measure the hit ratios experienced by the cache and by the base-level and J-level ATBs. The raw counts for all these data were written onto magnetic tape for subsequent off-line analysis.

During parameter measurement, simulated calls were generated by means of MICLOB (Microprocessor Controlled Load Box) units attached to the TSPS trunks. The response of TSPS operators, for those simulated calls that required operator assistance, was simulated by MOPS (Microprocessor Operator Position Simulator) units. The MICLOB and MOPS units are described in Ref. 9. Complete parameter characterization required taking measurements under various system conditions. Call loads were varied from zero to the maximum applicable simulated load. Different degrees of low-level activity were obtained by running processor and memory diagnostics and by causing different rates of output messages to be generated by the craft interface.

***2.3.3.2 Measurement of speedup factors.*** Values for each of the defined speedup factors were calculated from measurements taken at the Fresno test site. Figure 2 shows the calculated values for each of the speedup factors plotted with respect to $E'$, the SPC 1B quarter-hour E-E cycle rate. The value of $E'$ is inversely related to call load; $E' = 58{,}000$ corresponds to an idle system and $E' = 10{,}000$ corresponds to the maximum applied call load.

Figure 2 shows that the two J-level speedup factors, $K_{NJ}$ and $K_{CRJ}$,

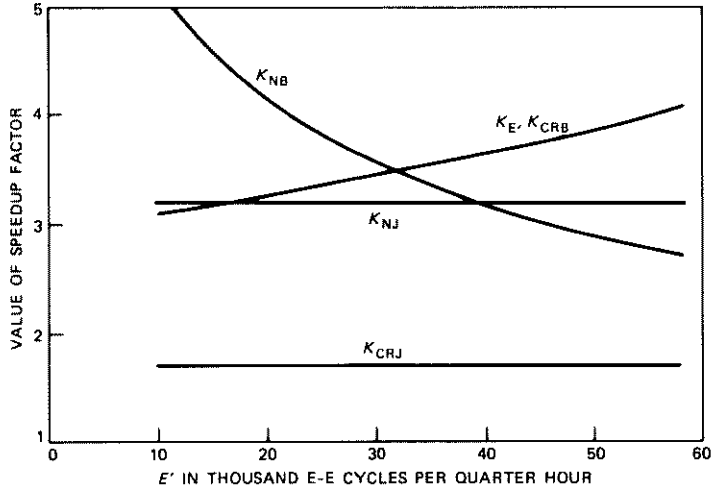



Fig. 2—Speedup factors.

are constant with respect to $E'$. The value of $K_{CRJ}$ is less than $K_{NJ}$ because J-level constant-rate work makes heavier use of the instructions that have relatively low-emulation efficiencies. Although the cache and ATB hit ratios associated with both of these speedup factors were observed to slightly increase with decreasing $E'$ (increasing call load), the effect of these changes was compensated for by a slight change in the dynamic instruction mix for $K_{NJ}$ and, for $K_{CRJ}$, a higher percentage of conditional transfers taken.

For base level, the three speedup factors, $K_E$, $K_{CRB}$, and $K_{NB}$ are seen to change with $E'$. The value of $K_E$, the speedup factor for E-E cycle overhead work, decreases with decreasing $E'$ (increasing call load) because of a marked decrease in the base-level cache and ATB hit ratios as $E'$ decreases. At zero call load, a relatively small portion of emulated code (the E-E cycle overhead work) is executed for a relatively high percentage of the time, causing the cache and ATB hit ratios to be at their highest values. The value of $K_{CRB}$, the speedup factor for base-level constant-rate work, was not measured directly but is set equal to $K_E$ because this type of work is quite similar to E-E cycle overhead work and because only a small percentage of real time (less than 2 percent) is involved.

The value of $K_{NB}$, the speedup factor for base-level trunk-seizure work, is seen to increase with decreasing $E'$ (increasing call load) even though the cache and ATB hit ratios are decreasing. This increase is caused by a decrease in the number of base-level instructions (excluding constant-rate and E-E overhead instructions) executed per trunk seizure as $E'$ decreases. Figure 3 shows this effect. Measured values of $I_{NB}$, the number of base-level instructions executed per trunk seizure, are plotted versus $E'$. The dependence of $I_{NB}$ is seen to be approximately linear with respect to $E'$ over a wide range of values.

Investigation has indicated that this effect is at least partly caused by queueing for busy facilities (e.g., digit receivers). During each E-E cycle, if a queue exists, an attempt is made to remove all entries from the queue. Those entries that cannot be removed remain for the next E-E cycle, thereby causing extra instructions to be executed. As the call load increases, the probability of queue formation also increases. The E-E cycle rate decreases, however, thereby producing a net decrease in the number of base-level instructions executed per trunk seizure. This effect also occurs with the SPC 1A, but to a lesser degree because, as will be seen, the E-E cycle rate of the SPC 1A is lower than that of the SPC 1B when both are operating at the same trunk-seizure rate.

Curves were fitted to the calculated values of the speedup factors shown in Fig. 2 to obtain expressions for the parameters used in the SPC 1B real-time model. These expressions are:

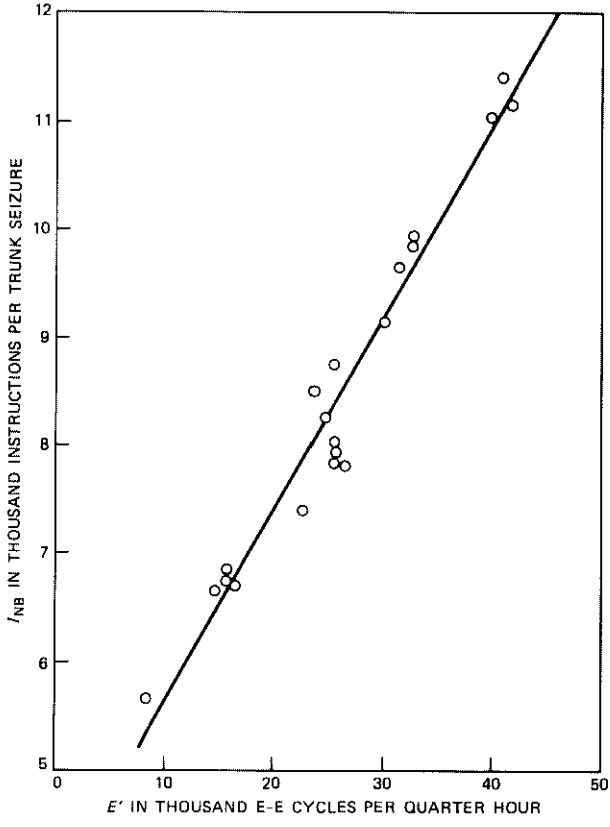



Fig. 3—Base-level instructions for each trunk seizure versus $E'$.

$$K_{CRJ} = 1.70;\ K_{NJ} = 3.20$$

$$K_{CRB} = K_E = \frac{8.2762}{(2.7929 - 13.104 \times 10^{-6}E')}$$

$$K_{NB} = \frac{20.698}{(1 + 45.884 \times 10^{-6}E')(2.8649 - 14.179 \times 10^{-6}E')},$$

where, as previously defined, $E'$ is the quarter-hour E-E cycle rate.

***2.3.3.3 Measurement of DMERT real-time requirements for TSPS No. 1B.*** Parameters representing the real-time requirements of DMERT for TSPS No. 1B combine both TSPS-independent and TSPS-dependent work. TSPS-independent DMERT work includes DMERT functional work (e.g., audits, timer, etc.) and maintenance work associated with the SPC 1B (e.g., 3B20D diagnostics). TSPS-dependent DMERT work includes the TSPS craft interface work and work associated with handling the TSPS J-level interrupts and base-level sleep-period requests. These parameters were characterized by

*Dynaprobe* measurements of the real time used at each execution level and under various system operating conditions.

The TSPS craft interface real time is primarily used in producing output messages for maintenance purposes, and is a function of message rate, message length, and the number of output devices in use. The message rate is, in turn, a function of call load. Characterization involved measuring the real-time cost on a per-character basis and analyzing output messages generated by SPC 1A sites to determine representative message rates and lengths.

Measurements of $T'_H$, the parameter combining TSPS-independent and TSPS-dependent real-time requirements for high-level DMERT work, yielded

$$T'_H = 45.2 + 10^{-3}\, N' \text{ s/QH},$$

where $N'$ is the TSPS No. 1B quarter-hour trunk-seizure rate. Measurements of $T'_L$, the parameter combining TSPS-independent and TSPS-dependent real-time requirements for low-level DMERT work, yielded

$$T'_L = 101.0 + 7.5 \times 10^{-3}\, N' \text{ s/QH}.$$

The value of $T'_L$ is the amount of real time that should be allocated to achieve satisfactory execution of low-level activities under worst-case conditions (e.g., high maintenance activity during call overload). Under normal conditions, the actual value of $T'_L$ is considerably less than this allocated value so that more real time is available to call processing.

To satisfy eq. (6), the values chosen for $b$, the number of base-level sleep periods per E-E cycle, and $s$, the requested duration of each sleep period, are

$$b = 5;\ s = 10 \text{ ms}.$$

*Dynaprobe* measurements also yielded

$$t_{DJ} = 84 \text{ microseconds}$$

for each J-level interrupt [see eq. (4)] and

$$t_{DB} = 1.5 \text{ ms}$$

for each base-level sleep request [see eq. (5)].

#### 2.3.4 Model evaluation

Figure 4 shows measured and predicted values of the quarter-hour E-E cycle rate, $E'$, plotted versus the quarter-hour trunk-seizure rate, $N'$, for the Fresno TSPS site. The SPC 1B real-time model was used to predict three different $E$ versus $N$ load lines, each corresponding to a different low-level activity rate. The separate load lines occur be-

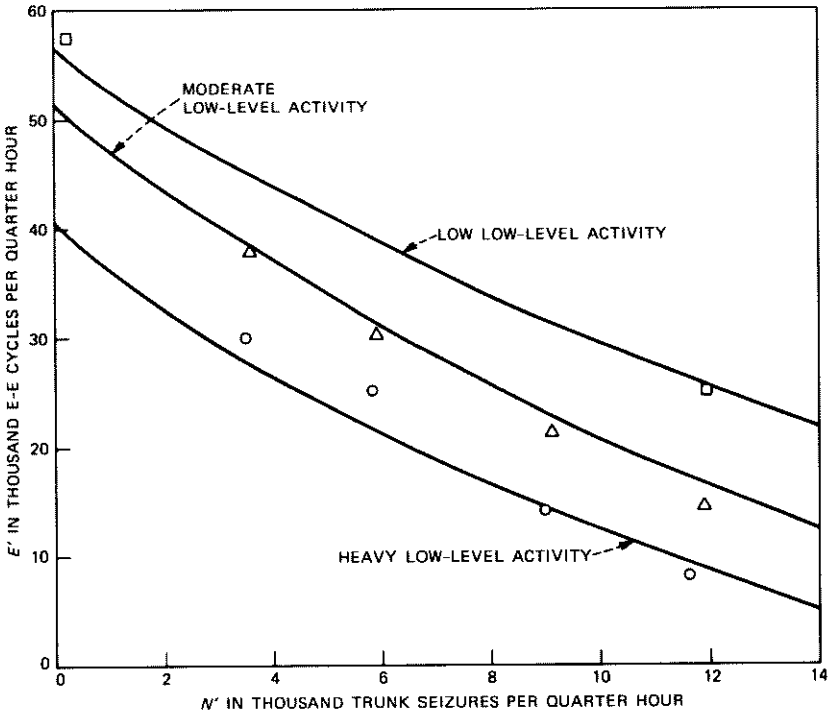



Fig. 4—SPC 1B load lines.

cause, as previously described, any real time not used by low-level processes is given back to base level, which uses this real time to execute additional E-E cycles.

The upper SPC 1B load line shows $E$ versus $N$ behavior when the low-level activity rate is low. Quarter-hour measurements were taken under these conditions at zero and the maximum applied load, and good agreement is seen between measured and predicted values. The middle SPC 1B load line corresponds to moderate low-level activity, and the lower SPC 1B load line corresponds to the condition when the low-level activity is heavy. Again, agreement between measured and predicted values is quite good.

### *2.3.5 SPC 1B capacity increase*

Figure 5 shows two $E$ versus $N$ load lines that indicate the increase in call-processing capacity provided by the SPC 1B. The upper load line depicts the $E$ versus $N$ behavior for the Fresno TSPS site as predicted by use of the SPC 1B real-time model for a typical low-level activity rate. The lower load line shows the $E$ versus $N$ behavior of the

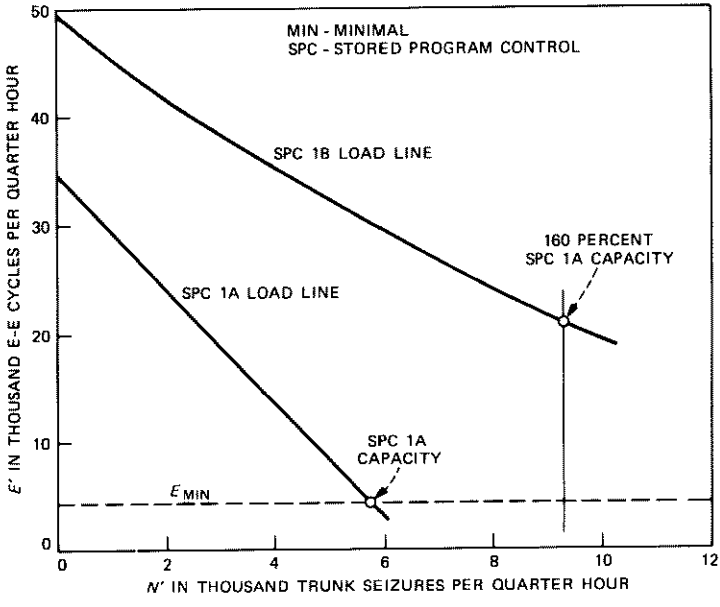



Fig. 5—Comparison between SPC 1A and SPC 1B load lines.

Fresno TSPS site as predicted using the SPC 1A real-time model, indicating how the site would perform if it were to use the SPC 1A. Both load lines assume the same call mix.

The SPC 1A load line shows that the SPC 1A would reach its capacity at about 5800 trunk seizures per quarter hour, since it is at that trunk-seizure rate that the E-E cycle rate equals 4180 E-Es per quarter hour, the SPC 1A value of $E_{MIN}$. Analysis and experiments conducted at Fresno indicate that $E'_{MIN}$ for the SPC 1B should be less than $E_{MIN}$. Therefore, since SPC 1B measurements were conducted at Fresno at around 9300 trunk seizures per quarter hour with good system performance (see Fig. 4), it can be concluded that the capacity of the SPC 1B is at least 160 percent of the SPC 1A capacity. Furthermore, because the SPC 1B E-E cycle rate at 9300 trunk seizures per quarter hour is high with respect to the indicated value of $E_{MIN}$, it appears that the SPC 1B capacity is comfortably greater than 160 percent of the SPC 1A. This additional capacity serves as a margin to accommodate variation among sites with respect to call mix and peakedness in busy-hour load.

#### *2.3.6 TSPSCAP program for TSPS No. 1B*

A TSPSCAP program was developed for the TSPS No. 1B incorporating the SPC 1B real-time model. As has been seen, a load-line

equation for the SPC 1B is considerably more complex than for the SPC 1A, and depends to a large extent on the amount of SPC 1B diagnostic and craft interface activity. Therefore, instead of providing a load-line equation, the TSPSCAP program provides calculated values of $E'$ and $N'$ which can be used to plot two load lines for the site in question. These two load lines, similar to the upper and lower load lines shown in Fig. 4, define what can be termed as a load-zone of normal system behavior. That is, the quarter-hour E-E cycle and trunk-seizure measurements for a site that is experiencing normal operation should fall within this load zone.

## III. RELIABILITY EVALUATION

### *3.1 Reliability requirements*

The SPC 1B reliability requirements are similar to those of a traditional ESS-type processor, having four fault categories: hardware faults, recovery deficiencies, procedural errors, and software deficiencies.[10]

Hardware failures are allocated 0.4 minute of downtime per year. The SPC 1B is divided into three subsystems, each of which is duplicated to achieve high reliability. Thus, one failure in one side of a subsystem will not cause a system outage. Hardware faults can cause a system outage only when both sides of a subsystem are experiencing failures (i.e., before the first failure is repaired, another failure occurs on the other side of the subsystem). When this occurs, the system is unable to establish a working configuration until one side of the failed subsystem is repaired and system integrity is reestablished. The hardware reliability is a function of the failure rates of the subsystems, the system architecture, and the repair rates of the subsystems.

Recovery deficiencies are allocated 0.7 minute of downtime per year. When a hardware failure condition is detected, an automatic fault-recovery action occurs to establish a working configuration. Unsuccessful recovery actions are classified as recovery deficiencies. These are due to either design errors or limitations in fault-recovery programs.

Procedural errors are allocated 0.6 minute of downtime per year. An improper maintenance procedure can cause a system outage. Providing easy-to-follow documentation and reducing the number of manual steps help to minimize procedural errors.

Errors in operational programs and data are allocated 0.3 minute of downtime per year. The amount of bootstrap time required to recover the system from software deficiencies is considered to be a part of system downtime under this category. To minimize this source of downtime, overall software execution is monitored continually, data

integrity is checked using extensive auditing procedures, and thorough system integration tests are performed after program changes are introduced.

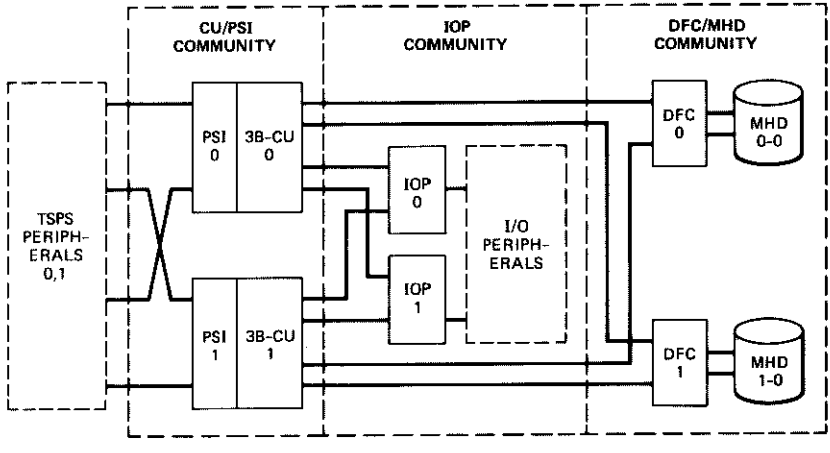
All four potential causes of system outage are closely interrelated. For example, improper procedures combined with certain hardware faults may prevent system recovery. In this paper, only SPC 1B outages induced by hardware faults are considered.

### *3.2 Reliability estimates*

#### *3.2.1 Reliability model*

To provide a basis for the relationship between the reliability model and the system architecture, a brief review of the SPC 1B architecture is presented. A complete description of the SPC 1B architecture can be found in Ref. 11, and a more detailed description of 3B20D architecture can be obtained from Ref. 12.

As shown in Fig. 6, the SPC 1B consists of three subsystems or communities: a duplicated 3B20D Control Unit and PSI (CU/PSI), a duplicated Input/Output Processor (IOP), and a duplicated Disk File Controller with Movable Head Disk (DFC/MHD). Either half of the duplicated CU/PSI community can access either side of the duplicated TSPS peripheral bus system. The IOP community has duplicated




CU - CONTROL UNIT
DFC - DISK FILE CONTROLLER
I/O - INPUT/OUTPUT
IOP - INPUT/OUTPUT PROCESSOR
MHD - MOVABLE HEAD DISK
PSI - PERIPHERAL SYSTEM INTERFACE
TSPS - TRAFFIC SERVICE POSITION SYSTEM

Fig. 6—SPC 1B architecture.

IOPs, each controlling a number of Peripheral Controllers (PCs). Each type of PC is designed to control a specific peripheral device, such as tape drives, teletypewriters, etc. The DFC/MHD community consists of a pair of DFCs, each capable of controlling one or more associated MHDs. One MHD on each half of the DFC/MHD community contains software required to bootstrap the SPC 1B.

Certain reliability measures are required to predict the probability of successful operation of the system. In expressing reliability of a switching system, "availability" is a more widely used term. Availability is defined as the fraction of time, on the average, that a system is expected to be in an operating condition. The availability of a switching system is a function of the system architecture and of subsystem failure rates and repair rates. Estimating availability of a system requires a mathematical model that can reflect the system architecture appropriately. The continuous-time, finite-state Markov model was used for the SPC 1B availability calculation, where an exponential probability distribution was assumed for the failure rates and repair rates. Detailed descriptions of how to use the Markov model to calculate system availability can be found in Refs. 5 and 13.

The reliability model for each of the three communities contains three states: a "duplex state," a "simplex state," and a "down state." The duplex state is a state where both halves of the community are fault-free and operational. Upon detecting a fault in either half of the community, a transition to the simplex state occurs. The rate of the transition is determined by the failure rate of the community. While a community is in the simplex state, one of two transitions is possible. A transition to the duplex state could occur if the faulty half of the community is repaired before a failure occurs in the other half of the community. On the other hand, a transition to the down state may occur if a new fault is detected in the remaining half before the initial fault is successfully repaired. A transition from the down state to the simplex state occurs when one of the faulty halves is repaired and put back to service. The rates of transitions, from the down state to the simplex state and from the simplex state to the duplex state, are determined by the repair rates, which are the reciprocals of corresponding mean time to repairs (MTTRs) for the community.

The probability that a community is in the down state is defined as the unavailability of the community. The SPC 1B is considered out of service when any of the CU/PSI, IOP, or DFC/MHD subsystems of the SPC 1B are in a down state. Hence, the unavailability of the system can be obtained by calculating the sum of unavailabilities of these three communities. The expected downtime per year for the SPC 1B can be estimated directly from the unavailability of the system.

### 3.2.2 *Availability estimates and modifications*

To evaluate the unavailability of each community, the reliability model was converted to a set of simultaneous equations where the unknowns are the probabilities of the states. Programs were written to solve the sets of equations corresponding to various architectural configurations. When these programs are used, sensitivity of the system downtime to the architectural variations as well as to the parametric values such as repair rates and failure rates of each community could be investigated. Coefficients of the equations were determined by the failure rates and repair rates of each community. Failure rates of the three communities were estimated from their component failure rates.

The repair rate of each community is estimated from the MTTR of mechanical failures, the MTTR of electrical failures, and a craft dispatch time. The MTTR of mechanical failures is considered separately from the MTTR of electrical failures because, for an MHD, the MTTR of mechanical failures is an order of magnitude longer than that of electrical failures. To minimize the MTTR, extensive diagnostic programs are included in the TSPS No. 1B, which can locate a fault within the resolution of three circuit packs. Detailed descriptions of diagnostic programs can be found in Refs. 11 and 14. A craft dispatch time is added to the MTTR when determining the repair rates of each community because the SPC 1B can be maintained by craft personnel located at a remote site. The dispatch time depends on the average travel time from the remote site and on the ratio between the average staffed hours and unstaffed hours per day of the TSPS No. 1B office.

The failure rates of the CU/PSI, DFC, and IOP are principally due to electrical failures. On the other hand, the failure rate of the MHD is due to roughly half mechanical and half electrical failures. Consequently, the MTTR of an MHD is much longer than the MTTRs of the other units. For the TSPS No. 1B application, the MTTR of an MHD has been improved through use of a spare MHD for each system.

Evaluation of the reliability model using current parameters shows that the reliability objectives for the TSPS No. 1B have been met.

## IV. CONCLUSION

This paper has described the prediction and evaluation of the call-processing capacity and system reliability of the SPC 1B. The call-processing capacity has been estimated through means of a processor real-time model whose parameter values have been determined by laboratory and test-site measurements. The system reliability has been predicted through use of Markov modeling techniques. Performing this evaluation during TSPS No. 1B development to monitor progress was instrumental in meeting the capacity and reliability objectives.

## V. ACKNOWLEDGMENTS

G. J. Kujawinski and the authors worked together as a team in evaluating the SPC 1B capacity; his contributions were essential to the success of the effort.

## REFERENCES

1. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.
2. N. X. DeLessio and N. A. Martellotto, "Traffic Service Position System No. 1B: System Description," B.S.T.J., this issue.
3. R. J. Gill, G. J. Kujawinski, and E. H. Stredde, "Traffic Service Position System No. 1B: Real-Time Architecture Utilizing the DMERT Operating System," B.S.T.J., this issue.
4. L. B. Brisson and R. L. Potter, "A Time-Shared Program for Predicting TSPS No. 1 Capacity," Proc. Int. Conf. Commun., Seattle, 1973.
5. J. A. Buzacott, "Markov Approach to Finding Failure Times of Repairable Systems," IEEE Trans. Reliability, *R-19* (November 1970), pp. 128–34.
6. R. J. Jaeger, Jr. and R. L. Potter, "Analysis of Processor Usage in Stored Program Controlled Telephone Switching Systems," Proc. Int. Conf. Commun., Philadelphia, PA, 1972.
7. Special issue entitled "Traffic Service Position System No. 1," B.S.T.J., *49*, No. 10 (December 1970).
8. N. Farber, "A Model for Estimating the Real-Time Capacity of Certain Classes of Central Processors," Paper 426, Proc. Sixth Int. Teletraffic Congress.
9. J. J. Stanaway, Jr., J. J. Victor, and R. J. Welsch, "Software Development Tools," B.S.T.J., *58*, No. 6, Part 1 (July 1979), pp. 1307–34.
10. P. W. Bowman, M. R. Dubman, F. M. Goetz, R. F. Kranzmann, E. H. Stredde, and R. J. Watters, "Maintenance Software," B.S.T.J., *56*, No. 2 (February 1977), pp. 255–87.
11. G. T. Clark; H. A. Hilsinger, J. H. Tendick, and R. A. Weber, "Traffic Service Position System No. 1B: Hardware Configuration," B.S.T.J., this issue.
12. M. W. Rolund, J. T. Beckett, and D. A. Harms, "3B20D Processor & DMERT Operating System: 3B20D Central Processing Unit," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 191–206.
13. M. D. Soneriu and D. S. Suk, "Markov Model for Estimating the Reliability of Duplicated and Repairable Computing Systems," Nineteenth Annual Tech. Symp., Pathways to System Integrity, Washington, D. C. (June 1980), pp. 87–95.
14. J. L. Quinn, F. M. Goetz, and R. L. Engram, "3B20D Processor & DMERT Operating System: Diagnostic Tests and Control Software," B.S.T.J., *62*, No. 1, Part 2 (January 1983), pp. 367–81.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Switching Control Center System Interface

By J. J. BODNAR, J. R. DAINO and K. A. VANDERMEULEN

(Manuscript received July 30, 1982)

*Over the years, the centralization of the maintenance and operation of Stored Program Control Systems (SPCS) has proved to be an economically attractive and effective methodology. At the core of the centralized maintenance plan is the Switching Control Center (SCC), which has responsibility for the surveillance and control of a number of SPCS. This center is supported by the Switching Control Center System (SCCS), which is a minicomputer-based system that provides automation of, or mechanized support for, the functions of the SCC. Since SCCS support for the Traffic Service Position System No. 1 (TSPS No. 1) was already available, these capabilities needed to be carried forward to support the operation and maintenance of TSPS No. 1B. In addition, the use of the new 3B20D Processor with a new craftperson interface allowed for a number of improvements and extensions. The SCCS interface to TSPS No. 1B has made use of new technology and techniques by incorporating microprocessors, video terminal interfaces, and BX.25 protocol in the design. The result is a flexible interface with software-driven craft displays.*

## I. CENTRALIZED MAINTENANCE—OVERVIEW

Since the early 1970's the Bell Operating Companies (BOCs) have been taking increasing advantage of the concept of centralized maintenance of switching systems. To economically operate, administer, and maintain electronic switching systems, operations centers were formed where technical experts equipped with computer-based support systems can apply their skills to many switching systems. In the case of the Traffic Service Position System (TSPS), the daily maintenance and operations functions may be provided by the Switching Control Center (SCC).

If a TSPS is maintained and operated from an SCC, the SCC manager has overall responsibility for the quality of service provided by the TSPS and the cost of maintaining the system. To provide high-quality service at a reasonable cost, a number of functions must be performed efficiently at the SCC. Since in this environment the TSPS is unattended, except when field personnel are assigned to specific tasks in the office, the SCC personnel monitor the real-time status of the TSPS on a continuous basis. In the event of failure of equipment in the TSPS complex, the SCC exercises appropriate controls to ensure service protection. The cause of the failure is then analyzed and isolated. Finally, when the trouble is identified, field personnel are dispatched to perform any necessary on-site repair.

To accomplish these functions, a number of work positions have been established in the SCC and individual craft have been assigned particular responsibilities.

The Office Controller is responsible for real-time surveillance of several switching systems and as such has the most frequent interface to the TSPS. (Most BOCs will maintain several types of electronic switching systems from each SCC.) The controller responds to system alarms and takes necessary service-protection action. The Analyzer is responsible for those more complex problems that require sectionalization and trouble identification. The Analyzer interacts with the TSPS to collect additional data and uses previously reported data to isolate system troubles. The craftperson at the trunk-maintenance position is responsible for identifying faulty trunks and directing their repair. In addition, dispatch and administration functions exist in the SCC to identify work for field forces, facilitate proper information flow, and keep records.

## II. TRADITIONAL INTERFACE

The No. 2 Switching Control Center System (No. 2 SCCS) began deployment in the mid-1970's as a computer-based support system to aid the SCC personnel in performing their jobs. It has evolved over time by providing new features and interfacing to Stored Program Controlled Systems (SPCS) such as TSPS No. 1.[1,2] With the development of TSPS No. 1B, a parallel SCCS development was done to integrate the new system into the existing operations environment while taking advantage of opportunities afforded by the new 3B20D Processor.

### *2.1 The TSPS No. 1—SCCS interface*

The traditional interface provided by No. 2 SCCS consists of a dedicated channel from a TSPS No. 1 that parallels the Maintenance

Teletypewriter (MTTY), plus a dedicated channel that terminates on an E2A telemetry unit at the TSPS No. 1 Master Control Center (MCC). At the SCC, the TTY channel is connected to a minicomputer system, known as the Computer Subsystem (CSS) and the telemetry channel terminates on a telemetry unit that collects data from many switching offices. This basic architecture is shown in Fig. 1. The TTY channel is a 110-baud asynchronous serial communication line. The data output by the TSPS No. 1 on this channel reports detailed information in line-oriented output messages suitable for hard-copy teleprinters. In addition, line-oriented input commands may be sent to the TSPS No. 1 on this channel. The TTY channel data are collected and stored in the CSS. The data are available for real-time alerting and long-term analysis.

The telemetry system serves the purpose of reporting real-time equipment status and allows controls to be activated remotely. At the TSPS No. 1 MCC an E2A remote unit is provided to collect detailed

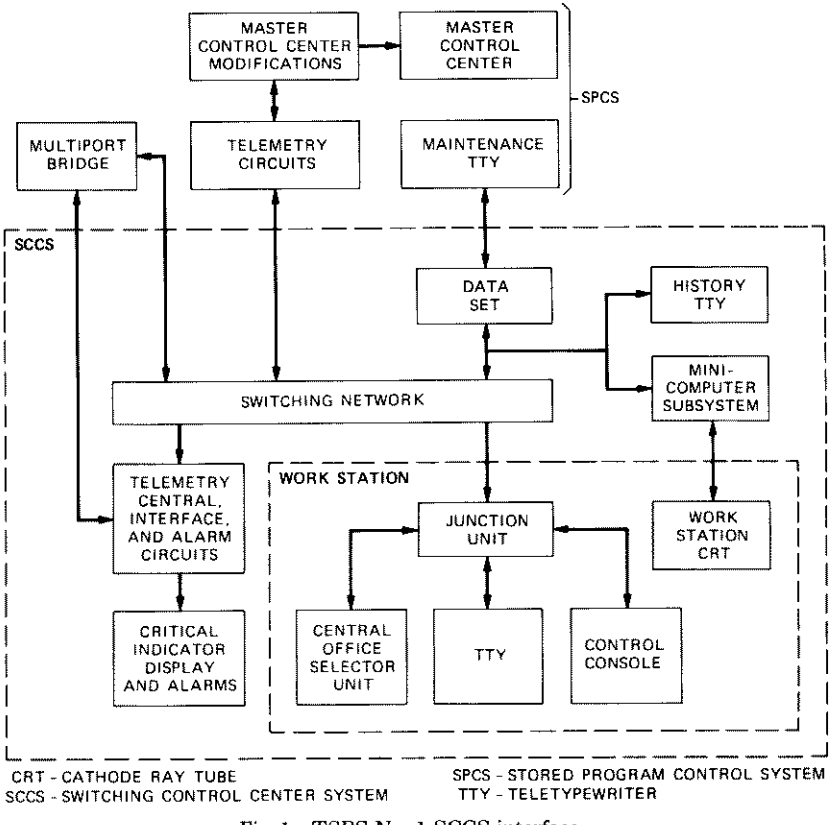



Fig. 1—TSPS No. 1-SCCS interface.

status of TSPS No. 1 hardware states. This is accomplished by connecting discrete scan points in the E2A telemetry unit to individual lamp displays on the MCC. Control capability is provided by individually connecting relay contact closures in the E2A telemetry unit in parallel with the pushbutton keys on the MCC. The interface is thus hardwired in place.

An E2A central unit in the SCCS polls up to 16 remote E2A units on a continuous basis. The polling process takes place over a data network that has a multiport bridge at its hub. The central unit transmits a request for data along with a unique address of one of the remote units. The remote units on the network respond only when their address is contained in the polling request. The message format used consists of start and stop bits, 16 data bits, and a 7-bit cyclic redundancy check sum. This data word contains a summary of the status of critical hardware units in each office. The data are displayed in real-time on a wall-mounted status panel called the Critical Indicator Panel (CIP) in the SCC.

To gain access to more detailed status-and-control capability, a control console at the SCC can be connected to the E2A remote unit at the TSPS No. 1. This is accomplished by the use of switching equipment in the SCCS to disconnect the E2A from the multiport network and connect it on a point-to-point basis with a control console at a work station in the SCC. Two types of consoles exist for interconnection with a TSPS No. 1. The original type of console provided was a wired logic console with a fixed display and keyboard. The display consisted of status lamps with labels that corresponded to labels printed on the MCC. Pushbutton keys on a keyboard were labeled with the same designations as used on the keys at the MCC.

Housed inside the console was an E2A central unit which was configured to collect all the data available from the remote unit. Its memory was hardwired to the lamps on the display panel. Such a console was then limited to functioning only with a TSPS No. 1. Any changes or additions to the MCC required corresponding changes to the consoles at the SCC. A second type of console is now provided, which consists of a CRT and microprocessor. In this type, the microprocessor communicates with the E2A remote unit to collect status information. The interface to the craft at the SCC consists of a CRT display. On this display, the lamps and keys on the MCC are represented by their labels, as shown in Fig. 2. Active status or controls are shown by displaying the label in reverse video (black characters on a white field). Keys are distinguished by preceding the label with a plus sign. Keys are activated by cursor positioning. This software-driven display allowed for the use of the same physical equipment with other types of switching systems.

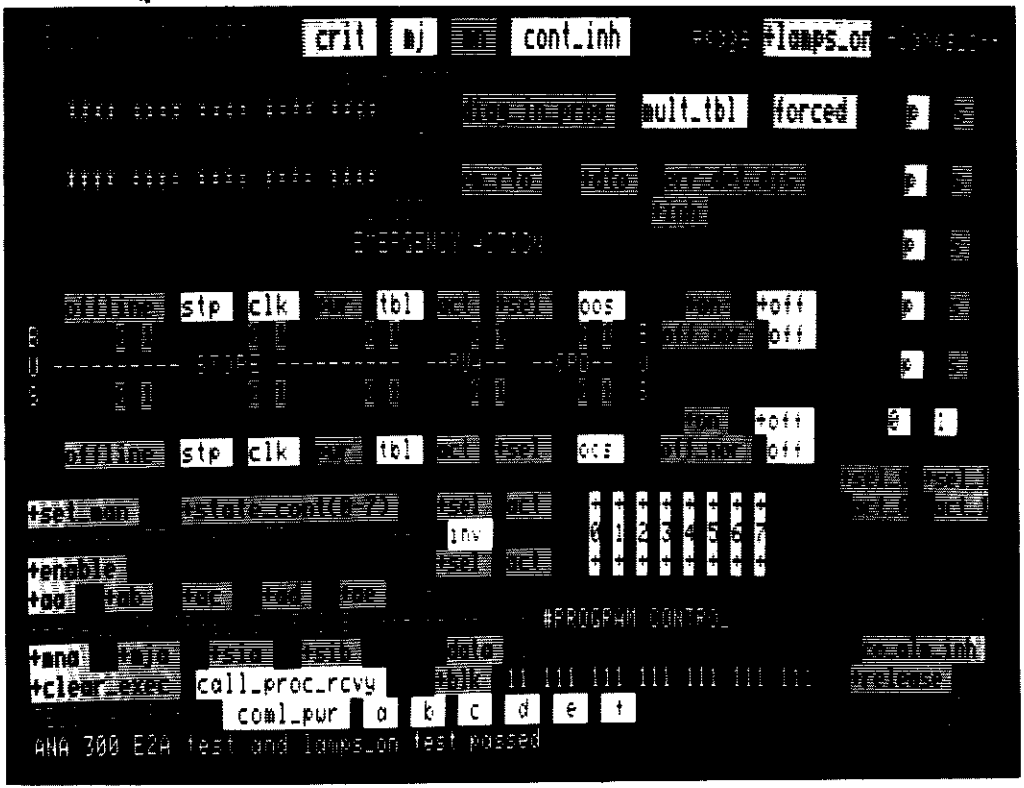



Fig. 2—TSPS No. 1 MCC display page.

### 2.2 Human interface

A number of human interfaces are provided to alert SCC craft to trouble conditions and to allow for further data collection, analysis, and control. As data are collected via the telemetry network in real time, alarms and status of critical equipment units are displayed on a highly visible CIP. This provides alarm visibility and long-term equipment status information to the SCC managers and craft. TTY messages are examined in real time as they are received and stored by the CSS. Messages reporting alarmed events are used to trigger a video display on an alarm monitor, which is a large CRT device. The identity of the switching system reporting the trouble and the alarm level are displayed along with a one-line summary of the condition being reported.

The office controller is responsible for responding to the real-time events reported on the CIP and the alarm monitor. The controller's work station is equipped with two CRTs that serve as additional interfaces to the system. One CRT, known as the CSS work station, is connected to the CSS. Via this device the controller can examine previously reported data that have been stored by the CSS. In addition, the CSS work station can be placed in the monitor mode where it is connected to the TSPS No. 1 via the CSS and becomes the functional equivalent of the maintenance TTY at the TSPS No. 1. That is, in this mode the work station receives each character output by the TSPS No. 1 as it reaches the CSS and any characters typed by the user are sent directly to the TSPS No. 1.

The second CRT is the microprocessor-driven console described above, which interfaces to the telemetry network. This equipment constitutes the control console and is the functional equivalent of the local MCC lamp and key panel. Thus the office controller has the same capabilities at the SCC work station that a craft at the TSPS No. 1 has via the maintenance TTY and MCC panel. The control console and CSS work station functions are provided on separate terminals and via separate processors and links in order to enhance the availability of the interface in the face of a simplex outage.

The analyzer work station is also equipped with these devices. The analyzer, however, makes more use of SCCS programs in the CSS which analyze large volumes of data and reduce them to summarized outputs and reports. For example, filtering and patterning capabilities allow for the identification of repetitive events and aid in the trouble-sectionalization process. While the office controller uses the control console for service-protection purposes, the analyzer uses it for more detailed data collection and observation of system performance under specific conditions.

It is this operations environment into which the new TSPS No. 1B

was integrated. The new interface serves as the foundation for remote maintenance of TSPS No. 1B as well as other 3B20D Processor applications.

## III. THE TSPS NO. 1B—SCCS INTERFACE

### *3.1 Elements of the new local maintenance position*

With the use of the 3B20D Processor and the DMERT operating system came the opportunity for new flexibility at the local maintenance position.[3] The MCC and the MTTY of the TSPS No. 1 have been replaced by the Maintenance CRT (MCRT) and a Receive-Only Printer (ROP).[4] The ROP provides a paper history function by recording each maintenance event or alarm condition in the form of line-oriented output messages.

The MCRT has several modes of operation. The first, the Emergency Action Interface (EAI), can be thought of as the replacement of the emergency recovery functions of the MCC. Actually, the EAI provides a hardware control function that is independent of the system software. It provides recovery capabilities for the TSPS No. 1B even in the absence of software sanity. The MCRT also has a test input/output mode. This allows the user to input line-oriented commands and to receive responses to these commands as well as spontaneously generated output. This is equivalent to the old MTTY input and output mode. In addition, the MCRT has a Control and Display mode (C&D), which provides the user with page-oriented displays of system status and with a menu of commands used to affect the status or configuration of the system. The C&D mode provides a very effective human interface for the daily functions of trouble isolation and correction. Finally, the MCRT provides continuous status information on the health of the system via a critical indicator display.

These different modes of operation are provided by dividing the screen into several sections and providing keys for toggling between modes (see Figs. 3 and 4). The top of the CRT is a header line. The second and third lines of the MCRT contain critical indicator information. The fourth line is reserved for command input. Lines five through twenty-two are used alternatively to display menus and pages in the C&D mode or to display the EAI page. Finally, any lines that are unused by the page that is currently being displayed are used for text input and output. At least two lines are always available for text input and output. However, if the currently displayed page does not use the entire page region, the remaining area may also be used for output.

The MCRT and the ROP interface to the 3B20D Processor through a firmware controlled unit called the Maintenance Teletype Controller (MTTYC). This unit provides the EAI mode and interfaces between

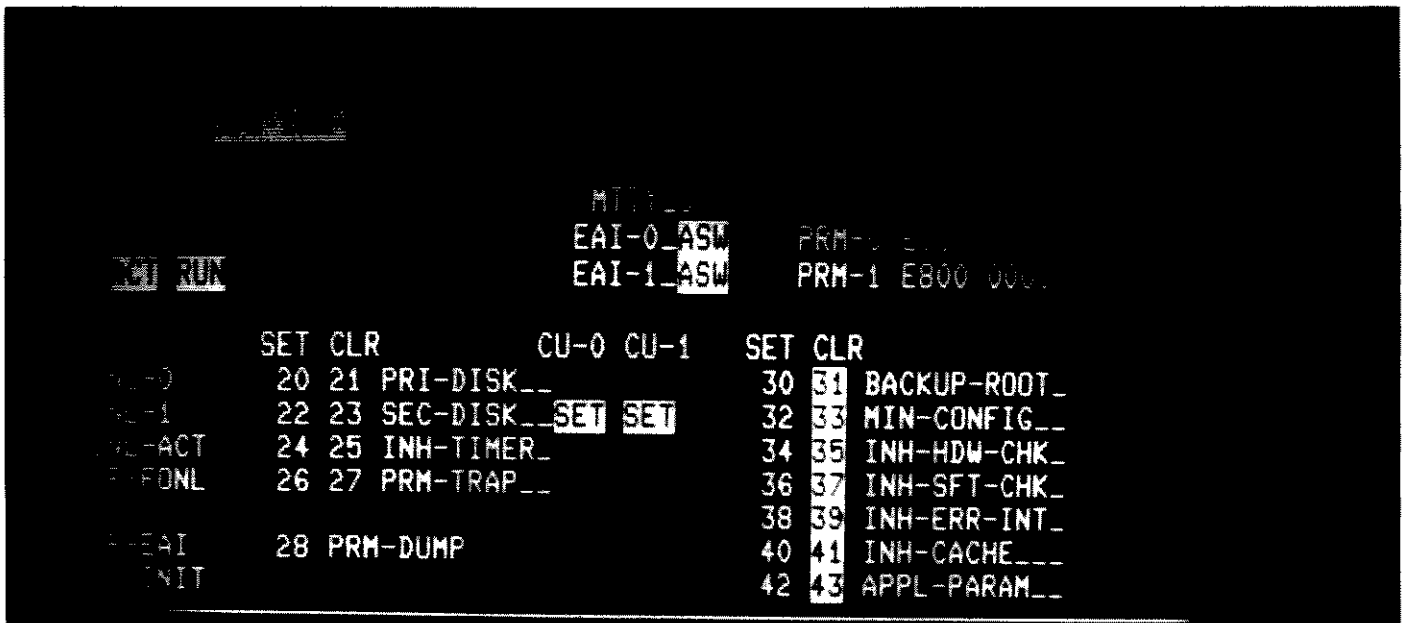



Fig. 3—TSPS No. 1B EAI page.

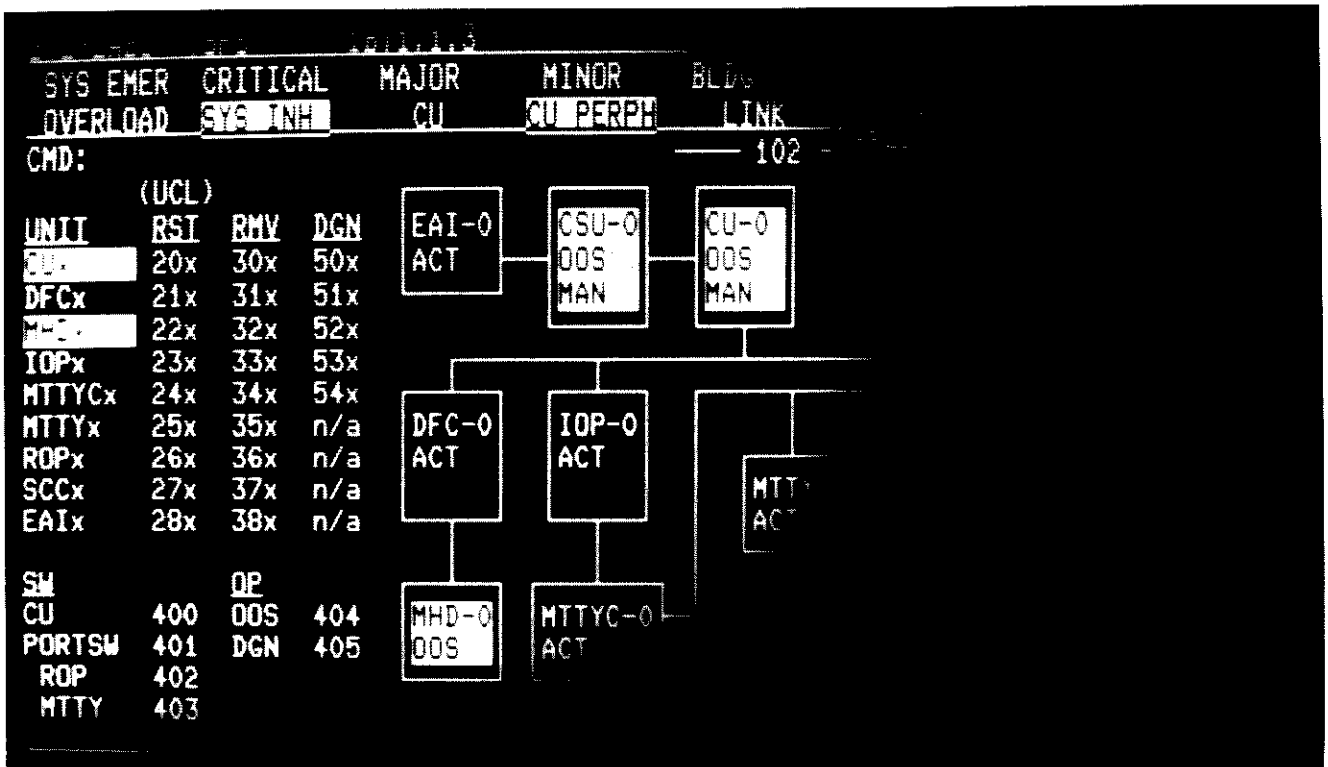



Fig. 4—TSPS No. 1B control and display page.

the terminals and the 3B20D Processor in other modes. The firmware in the MTTYC is responsible for controlling the MCRT in the EAI mode to allow access to the system in the absence of 3B20D Processor sanity. The MTTYC also controls the interface to the SCCS. All other modes of operation and information on the MCRT are controlled by software in the 3B20D Processor.

### *3.2 Elements of the remote maintenance interface*

The allocation of information and modes of operation at the SCC is consistent with both the local interface at the TSPS No. 1B and the existing SCC functions. The critical indicator information is displayed on the CIP as was done for the TSPS No. 1. In addition, the text input/output and the data on the ROP are logged in the CSS. This provides the work station with the Monitor mode and allows the CSS to drive the alarm monitor. This will also allow the existing work station features for the controller and the analyzer to operate as before. Also, the EAI interface is available to the Control Console (CC) just as the MCC functions of the TSPS No. 1 were provided there. This allows the controller to recover the TSPS No. 1B when necessary. Providing this function on the CC gives added assurance that it will be available in the event that the CSS is unavailable.

Furthermore, to provide consistent operation with the local interface, a CRT display mode essentially identical to that at the MCRT is provided on both the CC and the CSS work station. The only exception is that the CC is not provided with the text input/output functions at the bottom of the screen. This means that both the CSS work station and the CC have access to the EAI mode and the C&D mode, and the CSS work station has full TTY input/output capabilities. Also, all the mechanized features provided by the CSS to support the controller, analyzer, and trunk-maintenance positions remain available.

To communicate between the TSPS No. 1B and SCCS in the C&D mode, a language was defined with which the TSPS No. 1B tells the SCCS what should be displayed on the screen. The SCCS then sends each user command to the TSPS No. 1B, which processes the command and sends back a description of the display, complete with all the up-to-date status information. The SCCS translates that data stream into one that is understood by the SCC terminal. The language that is used to communicate this information is known as Virtual Terminal Protocol (VTP). By this method, the SCCS is not required to have complete up-to-date information describing each of the many C&D pages provided by TSPS No. 1B. This helps keep the system relatively independent with respect to changes and enhancements.

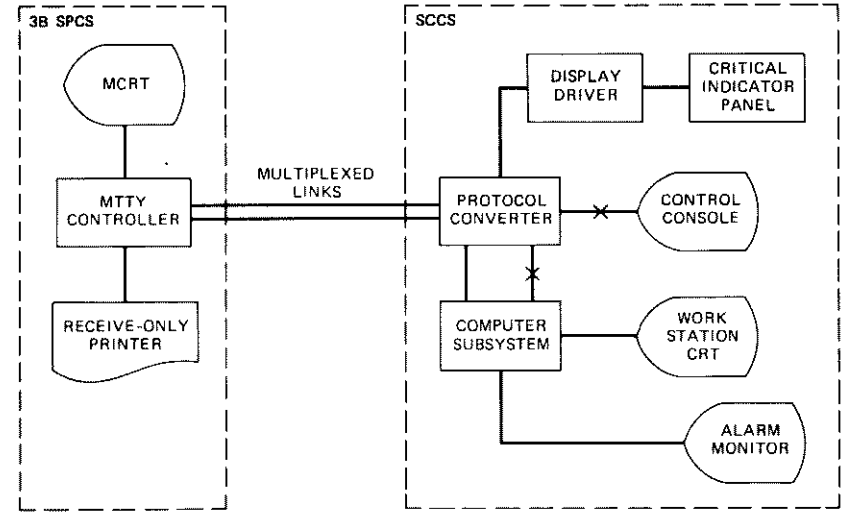
With respect to the EAI mode and the critical indicators, however, much less information is involved and it is less subject to change. The

EAI consists of a single page, which is controlled by the firmware in the MTTYC. There are up to twelve critical indicators, the definition of which is very stable for each SPCS. Therefore, the SCCS is capable of generating a critical indicator display as well as an EAI display, and the TSPS No. 1B simply sends the data required to describe the status of each of the indicators or display components.

### *3.3 Basic architecture of the interface*

Since the existing TTY and E2A interfaces do not have the capacity to support the amount and form of information that must be exchanged between the TSPS No. 1B and SCCS, a new interface was designed. In particular, the information necessary to drive the EAI and C&D modes requires highly reliable transmission of a significant amount of binary information. The new interface that was designed for communication between the TSPS No. 1B and the SCC, Fig. 5, consists of a pair of 2400-baud synchronous links between the MTTYC and a new device at the SCCS called the Protocol Converter (PC). One link is normally active while the other is used as a backup and is therefore normally in the standby state. The link carries seven logical streams of data under the BX.25 protocol using the virtual channel assignment capabilities of that protocol. The virtual channels (VC) are assigned as follows:

VC1—Emergency action interface
VC2—Critical indicator information




CRT - CATHODE RAY TUBE
MCRT - MAINTENANCE CRT
MTTY - MAINTENANCE TELETYPEWRITER

SCCS - SWITCHING CONTROL CENTER SYSTEM
SPCS - STORED PROGRAM CONTROL SYSTEM

Fig. 5—TSPS No. 1B—SCCS interface.

VC3—C&D interface
VC4—TTY input and acknowledgment
VC5—TTY output
VC6—Spare
VC7—Spare

The two links provide simple backup for each other. If either end detects a link outage at level 2 of the protocol, it will switch to the other link and establish communication on that link. The function of the PC is to handle the BX.25 protocol and to divide the virtual channels into separate physical connections bound for the different subsystems of the SCCS.

The PC is a firmware-controlled microprocessor system. Each microprocessor can support up to three 3B20D-based systems. The PC is housed in a newly designed cabinet, known as the T-cabinet, which can hold up to three PCs. In addition, this cabinet contains one spare PC, which can be manually activated to replace any one of the three active PCs, should they fail.

The PC sends the critical indicator information contained on VC2 to another firmware-controlled microprocessor called the display driver, which interfaces to the CIP. The display driver is also housed in the T-cabinet and can support up to 18 systems.

The PC sends and receives the TTY information contained on VC4 and VC5 to and from the CSS. The interface is a 1200-baud asynchronous serial communication line similar to the MTTY interface in the previous architecture. Since this is a character-oriented interface and the BX.25 link to the TSPS No. 1B is a message-oriented interface, the PC provides for the translation between the two. The fact that the TSPS No. 1B emulates the TSPS No. 1 has allowed the software in the CSS—used to assist the SCC in the maintenance and operation of TSPS—to be carried forward to support TSPS No. 1B with only minor modification. That is, most information received from a TSPS No. 1B, except that which deals directly with the processor itself, is the same in form and content as in TSPS No. 1. Thus, all the mechanized tools provided by the SCCS are still applicable with very little modification. This fact significantly enhances the ease with which the TSPS No. 1B can be integrated into the SCC environment as it gradually replaces the existing TSPS No. 1's.

To provide the MCRT function, the PC multiplexes the data contained on VC1, VC2, and VC3 onto a single physical link which can be connected to either the control console or the CSS, depending on whether the control console or CSS work station is currently accessing the office. This interface is also a 1200-baud asynchronous communication line, and a simplified version of levels 2 and 3 of the BX.25 protocol has been designed for this link. This allows for the three logical data streams to coexist on this link and also allows for some

error checking to increase the reliability of the interface. In addition, this link allows the user to input C&D commands or EAI commands into the TSPS No. 1B through the PC.

In summary then, the interface described in this section permits the distribution of functions among the subsystems of the SCCS, which is consistent with existing operations and allows the SCC user to access the TSPS No. 1B in the same manner as the user at the local maintenance position. The control console has access to the EAI, critical indicator, and C&D information when it is attached to the office. In addition, the CSS work station can "display" the office information and receive the same view as the MCRT at the TSPS No. 1B, including the text input/output functions at the bottom of the CRT. Finally, all the status and alarm information output at the ROP and all the input/output activity on the TTY channel—both from the SCC and the local position—are logged in the history files in the CSS. This information then becomes the basis for the alarms displayed on the alarm monitor and for the sophisticated tools provided by the SCCS to support the controller, analyzer, and trunk-maintenance positions at the SCCS.

## IV. OPERATIONAL IMPACT

The TSPS No. 1B-SCCS interface introduces several new features to the SCC craft. The use of software and microprocessor technology at the SCC and the TSPS No. 1B have enabled various design objectives to be met. The available technology allows the use of the BX.25 protocol providing a multiplexing arrangement on the TSPS No. 1B-SCCS link. Intelligent control equipment administers the BX.25 protocol that provides error detection and correction on the link. Software-controlled video displays replace hardwired control panels provided on the TSPS No. 1.

The 3B20D Processor will be used in other Western Electric switching systems, which will permit the development of a common SCC interface for all such systems, These, and other new features and capabilities described below, take advantage of the available new technology. This section discusses how the features and capabilities of the architecture described in Section III affect the SCC operations.

### *4.1 Consistent operation with local TSPS No. 1B maintenance terminal*

The CSS work station operation with the TSPS No. 1B is essentially identical with the on-site TSPS No. 1B maintenance terminal. In addition to all capabilities provided at the on-site TSPS No. 1B terminal, the CSS work station has access to the features provided by the SCCS computer subsystem. In the EAI and C&D modes, the control console also has identical operation as the on-site TSPS No. 1B maintenance terminal. Both the SCC work station and the control

console have full access to the EAI page to monitor and initiate TSPS No. 1B status and recovery functions. By reducing the differences between the local and remote interfaces, training requirements are reduced and craft efficiency is enhanced.

### *4.2 Consistent operation with existing SCC*

The design provides the TSPS No. 1B with support which is operationally similar to other SPCSs interfacing to the SCCS.

The TSPS No. 1B may be monitored and controlled from the CSS work station with the CSS providing the support functions of logging, sorting, and browsing. All the tools provided by the CSS for the support of TSPS No. 1 are extended to TSPS No. 1B. This means that operational procedures that are in place with respect to TSPS No. 1 are still applicable. The alarm monitor display of alarms is also fully functional.

In the event of a CSS failure, the control console provides craft access to the TSPS No. 1B as it does in other SPCS-SCCS interfaces. Alarms are displayed on the critical indicator panel in a standard line-up arrangement as with other systems. All CSS features operate in the same manner as with previous systems.

While enhancements have been made to the human interface, the basic SCC functions (described in Section II) have been maintained with the TSPS No. 1B.

### *4.3 Improvements to the human interface*

Since the TSPS No. 1B introduces many new features and capabilities in the maintenance terminal, these features are also available at the CSS work station and control console as a result of the new TSPS No. 1B-SCCS interface.

For example, in addition to the text input/output message capability, used in the TSPS No. 1, the TSPS No. 1B-SCCS interface includes the C&D mode. The C&D mode provides a series of displays that give current status of various sections of the system. These displays may be requested by the craftperson using a menu selection process. The menu lists several possible display pages with a three-digit-selection menu number. In addition, the user can initiate many of the typical maintenance and recovery actions required to operate a TSPS No. 1B using menu numbers provided on these displays. This provides a significant improvement in speed and accuracy for frequently used functions over the text input and output capabilities of the TSPS No. 1 and SCCS interface.

The EAI page, Fig. 3, follows the same format as other display pages. From this page the craftperson is able to initiate various recovery functions using a two-digit menu selection. Also available on this page

is progressive system status during recovery sequences. This is a significant improvement over the previous MCC interface of the TSPS No. 1 in several ways. In particular, the EAI provides more information—because of the video display—and more consistency between the emergency interface and other kinds of access to the system. Consistency between the local interface and the remote interface at the SCC is also enhanced. In the TSPS No. 1 the local interface was a hardware panel of lamps and keys and the remote interface used a video display and keyboard to emulate the local interface. In the new design the local and remote interface are identical. The system also provides four special function keys at the video terminal. These keys are designed to allow the craft fast and convenient use of the capabilities of the human interface. The keys provide one-button operation to perform the following functions:

(*i*) Request the EAI page (EA DISP)

(*ii*) Exit the EAI page (NORM DISP)

(*iii*) Change between the text input and menu input mode (CMD/MSG)

(*iv*) Retire alarms (ALM RLS).

To illustrate the advantages of the control and display operation available with the interface between the TSPS No. 1B and the SCCS, consider the example of the alerting and analysis of a major alarm associated with a moving head disk in the TSPS No. 1B. As with other systems, a TSPS No. 1B major alarm results in a visual indication at the critical indicator panel, a one-line alarm summary message is displayed at the alarm monitor, and, if activated, an audible alarm sounds. If the controller then connects the CSS work station to the office in the display mode, the display will indicate the alarm with the "Major" indication and flashing "CU PERPH" (Control Unit Peripheral problem) in reverse video in the critical indicators section of the display (second line on the screen).

The controller can retire the alarms by depressing the alarm-release special function key. This will return the major indication to normal on the screen and result in "CU PERPH" in solid reverse video. The controller can then select the CU PERPH page from the page index by entering the three-digit menu number associated with that page. The CU PERPH page will display the status of the 3B20D Processor and peripherals. On the display, the moving head disk will be indicated as out-of-service. Once called in by the controller, the analyzer may then request diagnostics be run on the disk using either a menu-selection number (from the menu of commands displayed on the left part of this display page) or, by using the special function key to change to the text input mode and typing a text diagnostic message. The result of this diagnostic should lead the analyzer to further action, (e.g. dispatch to repair or replace the faulty hardware).

Video displays and menu selection provide an effective human interface, which results in a low probability for human errors occurring and a high level of job satisfaction.

### *4.4 High reliability*

To meet the high reliability required of the TSPS No. 1B-SCCS interface for unattended operation of TSPS, the interface is designed with a highly redundant architecture. The 2400-baud data link between the two systems is fully duplicated and the BX.25 protocol provides error detection and correction. The TSPS No. 1B components that control and administer the SCCS interface (e.g., the MTTYC) are fully duplicated and provide full access both to the local and remote users in the event of single-unit failures.

The SCCS components are also backed up. In particular, the T-cabinet provides for a sparing arrangement in the event of a PC failure, and the duplication of capabilities at both the control console and CSS work station allows these terminals to back each other up in the event of failures in either subsystem. Finally, the CIP provides alarm and status information even in the event of a CSS failure that disables the alarm monitor.

In addition, this design provides for some improvement in the capabilities provided to the SCC when single-link failures occur. In previous SPCS-SCCS interfaces, the text message data and the E2A telemetry data containing critical indicator information and emergency action information were transmitted over separate facilities. Therefore, if one facility failed, the SCC lost partial capability. The TSPS No. 1B-SCCS interfaces, using two fully duplicated links, provide the ability to transmit all the necessary data with no loss of capabilities at the SCC in the event of a single transmission-channel failure.

## V. CONCLUSION

The SCCS interface to TSPS No. 1B has made use of new technology and techniques by incorporating microprocessors, video terminal interfaces, and BX.25 protocol in the design. The result is a flexible interface with software-driven craft displays. At the SCC, the craft interface devices are the same physical devices used to interface with older technology SPCSs, yet they provide new video displays and menu selection inputs. Thus, the new technology of TSPS No. 1B can be maintained from the SCC as it is introduced into the network and can also be used to enhance SCCS craft interfaces while remaining consistent with overall SCC operations.

## VI. ACKNOWLEDGMENTS

The authors would like to recognize the efforts of a large number of

people over a several-year period in the development and testing of the TSPS-SCC interface.

## REFERENCES


1. R. J. Jaeger, Jr. and A. E. Joel, Jr., "Traffic Service Position System No. 1," B.S.T.J., *49*, No. 10 (December 1970), pp. 2417–31.
2. R. E. Staehler and W. S. Hayward, Jr., "Traffic Service Position System No. 1, Recent Developments," B.S.T.J., *58*, No. 6, Part 1 (July–Aug. 1979), pp. 1109–1367.
3. G. T. Clark, H. A. Hilsinger, J. H. Tendick, and R. A. Weber, "Traffic Service Position System No. 1B: Hardware Configuration," B.S.T.J., this issue.
4. T. G. Hack, T. Huang, and L. C. Stecher, "Traffic Service Position System No. 1B: Software Development System," B.S.T.J., this issue.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
*Printed in U.S.A.*

## *Traffic Service Position System No. 1B:*

# Long-Range Planning Tools

By P. L. BASTIEN and B. R. WYCHERLEY

(Manuscript received August 9, 1982)

*This article describes the issues involved in operator services planning and the structure of the computer tools developed to address these issues. The approach to long-range planning of operator services networks is very flexible, and the availability of the Traffic Service Position System No. 1B further enhances this flexibility. This planning effort is a complex process that has been automated by computer tools that are both accurate and user-friendly.*

### I. INTRODUCTION

The planning of Traffic Service Position System (TSPS) installations and growth is rather different from that usually encountered in engineering central office equipment.[1] One of the reasons for this is the great flexibility that the engineer has in selecting the placement of operator position subsystems and how traffic will be routed to the various TSPSs that serve a particular area. The availability of the TSPS No. 1B offers new choices: formerly, when a TSPS exhausted its call processing capacity the alternatives were to divert the traffic overload to another TSPS that had surplus capacity or to purchase a new TSPS and divert load to it. Under the new system a TSPS No. 1 can be retrofitted to a TSPS No. 1B with much greater call capacity.[2] The resulting efficiencies, including avoidance of trunk rearrangements, trunk splintering, and opening of new operator groups, provide significant economic benefits to the operating telephone company. The TSPS No. 1B also provides the engineer more opportunities to consolidate, that is, to retrofit a subset of a group of TSPS No. 1's to TSPS No. 1B's and retire some or all of the remaining TSPS No. 1's in the group.

The Operator Services Traffic Network Planning System (OSTNPS) is a tool designed to aid telephone companies in making long-range plans for TSPS and related network evolution in order to guide short-range planning activities.[3] OSTNPS has been available to telephone companies since November 1981. Using interactive programs and a *UNIX** operating system environment, OSTNPS lets the user make the critical decisions while taking over the detailed calculations that can often severely limit the number of alternatives that a planner considers. OSTNPS combines maximum user flexibility with fast turnaround: the user can generate and analyze an alternative satisfying all the necessary constraints in about two hours of terminal time (CPU time is negligible). Since a typical alternative may involve 45 nodes and an associated network configuration over a 20-year period, design considerations and human factors engineering were of the utmost importance in developing OSTNPS.

Many of the factors to be considered existed for the TSPS No. 1 before the development of the TSPS No. 1B; the general approach taken with the program design made addition of the new degree of freedom relatively simple. This article describes the problems involved in planning TSPSs, and the software programs developed to guide the user in this planning.

## II. PLANNING ISSUES

### *2.1 Long-range planning*

The purpose of long-range planning is to evaluate the results of the most likely trends in traffic, services, and technology in order to optimize the long-term results of short-range decisions. The usual long-range plan covers a period of 20 years. The procedure consists of generating a set of alternatives and comparing the economic worth of each. The alternatives usually include a continue-as-is base plan with which the other alternatives are compared.

Because 20-year forecasts involve many potential contingencies, the long-range planner must apply judgment and experience to select the most reasonable alternatives and to perform appropriate sensitivity analysis on these alternatives.

### *2.2 Operator services long-range planning issues*

The choices specific to operator services include the following.

#### *2.2.1 Purchase of new nodes*

Usually the purchase of a new TSPS, Remote Trunk Arrangement

* Trademark of Bell Laboratories.

(RTA), or Position Subsystem (PSS) is made to relieve the exhausts of existing nodes.[4] The planner must decide where to place the new node and where its traffic is to come from. For example, a new TSPS may be loaded in such a way as to delay its exhaust and that of neighboring systems for as long as possible, or it may be placed in a way such that its resultant traffic is routed over facilities that are as short as possible. Usually a new PSS is placed where there is a large operator labor market. The issues involved in an RTA purchase will be discussed in Section 2.2.4.

#### *2.2.2 Retrofit to TSPS No. 1B*

A TSPS No. 1B has greater real-time and memory capabilities than a TSPS No. 1. If the TSPS No. 1 exhausts real time or memory, then the savings in rearrangements and/or new node purchases must be weighed against the costs of retrofit and, in a growth environment, the eventual purchase of a new node.

#### *2.2.3 Load balancing*

Load balancing consists of moving trunks, and hence traffic, from one node to another in such a way that all exhausts are relieved. Load balancing eliminates the capital expenditure of a new operator node in the given year, but the purchase is only deferred and load balancing can be quite expensive in itself because of trunk rearrangements.

#### *2.2.4 RTA issues*

An RTA concentrates trunks and homes them in on a nearby TSPS via base-remote (BR) trunks.[4] The planning issues specific to RTAs are as follows.

- Direct versus RTA trunking. If a set of local offices is located a long distance (say 100 miles) from the TSPS on which the offices are to be homed, one alternative is to trunk them directly to the TSPS and another is to concentrate these trunks via an RTA. In the former case, trunk and facility costs dominate; in the latter, purchase of an RTA is necessary. In general, purchasing an RTA is desirable if many trunks are involved and the distance to the TSPS is large.
- Colocated RTA. If a TSPS reaches a condition of trunk exhaust, an RTA may be purchased and colocated with the TSPS: the RTA concentrates some of the TSPS incoming trunk calls and relieves the exhaust. Further, a colocated RTA-TSPS pair gives more flexibility in load balancing if the TSPS later becomes exhausted for any other reason: the RTA could then be rehomed on a nearby TSPS with spare capacity. An RTA rehome is defined as a move of its base-remote trunks; if the RTA were not there, a

move of about eight times as many TSPS incoming trunks would be necessary.

#### 2.2.5 PSS rehoming

TSPS operators are assigned to PSSs, consisting of groups of operator consoles and the associated intelligence.[4] A move involving TSPS/RTA traffic usually means that position requirements at that TSPS will change. If a sufficient amount of traffic is moved from one TSPS to another, one or more PSSs may be rehomed between these TSPSs. The planner generally has great flexibility in these rehomings except for mileage constraints.

#### 2.2.6 Impact on toll trunking

Sometimes a move of TSPS/RTA traffic impacts on toll trunking as well. Since TSPS and RTA incoming trunks are bridged between the local office and toll switch, a move of these trunks implies a corresponding connect/disconnect of trunks at the associated toll switches. Moreover, such a move will also cause a change in intertoll trunking. Every local area generates operator traffic that returns to itself via the toll switch. When TSPS/RTA incoming trunks are moved from one toll switch to another, the traffic must have a path from the second toll switch back to the first via Direct Distance Dialing (DDD) trunks. This path would not have been necessary had the move not taken place. On the other hand, an RTA rehome does not change toll trunking since only BR trunks are involved. Detailed examples of these issues will be considered in Section IV.

### 2.3 New considerations with TSPS No. 1B

A TSPS No. 1 that exhausts real time or memory can be relieved by retrofitting a 3B20D Processor in place of the present Stored Program Control No. 1A—a procedure that converts TSPS No. 1 to TSPS No. 1B.[2] The demand for the 3B20D Processor for TSPSs existed because real time or memory is the limiting factor on most TSPSs today. From a planning point of view, this method of relief is easiest because no traffic rearrangements are needed.

#### 2.3.1 Selective conversion to TSPS No. 1B

Because of capital constraints, a planner may convert some but not all exhausted TSPS No. 1's to TSPS No. 1B in a given year. This strategy introduces new effects on planning: a combination of conversion to TSPS No. 1B, TSPS/RTA purchases, and load balancing may take place. Consider the following example. Suppose that both TSPSs A and B (both TSPS No. 1's) exhaust real time in some study year and that a single conversion to TSPS No. 1B is to be made in that

year. The planner retrofits A, giving it extra real-time capacity. To relieve B, the planner takes the following steps: two RTAs are purchased, colocated with B, loaded with incoming trunks from B, and homed on A. B is thus relieved and survives real-time exhaust two more years, at which time B is retrofit and the RTAs are rehomed back to B.

#### *2.3.2 TSPS consolidation*

Planning issues are different in low-growth regions: the added real-time capacity of TSPS No. 1B increases the possibility of TSPS consolidation. First a consolidation candidate must be selected from a group of nearby TSPSs. It may be the oldest, most centrally located, or most lightly loaded TSPS of the group. The TSPS is retired by moving all of its traffic to the other TSPSs in the group. After consolidation, the remaining TSPSs may have to be converted to TSPS No. 1B earlier than if consolidation had not taken place, since they will exhaust sooner.

## III. USER-CONTROLLED PLANNING TOOL

### *3.1 Design considerations*

OSTNPS was developed so as to combine extreme flexibility with very fast turnaround. Extreme flexibility means that the user can generate any alternative desired—the programs check that no constraints are violated along the way. Very fast turnaround means that the input database and modeling must be both representative and simple, so that generation and analysis of an alternative can be done in an hour or two. A typical operator services alternative consists of 5 TSPSs, 8 RTAs, 20 PSSs, and a dozen toll switches all evolving as a function of time with consolidation, purchase of new nodes, or load balancing. The usual study period is 20 years so that, with a base year, there may be up to 21 toll-connect and intertoll sets of trunk requirements ("trunk fields") associated with the alternative. A typical number of such solutions is ten.

OSTNPS does not optimize but rather lets users generate their own alternatives. For a given study area, it is not known how close a given plan is to the optimum because the optimum is not known to begin with. However, analysis of all possible alternatives for a given simple study area suggests that the economic worth of a typical "intelligent" user plan will be within 5 percent of the optimal plan.

### *3.2 Modeling considerations*

Modeling considerations are as important as design considerations, because inputs must be at once representative and simple if the objective of fast turnaround is to be met.

### 3.2.1 Input database

For mechanized long-range planning, it is desirable to obtain reasonable results using data that are easy for users to collect. OSTNPS requires comparatively little user input data, and all data come from standard sources. Implied in the small size of the input data base are several assumptions—summarized below—used to make forecasts.

The input database contains only three types of call volumes: total trunk seizures on TSPS (whether or not an operator is accessed), trunk seizures that reach an operator for any purpose, and trunk seizures that reach an operator only for Operator Number Identification (ONI) of the calling party. These three volumes, for every TSPS and study year, are the primary inputs into all of the forecasting algorithms.

Some of the algorithms are linear regressions based on these volumes. One such regression forecasts required operator consoles (positions), the independent variable in this case being the product of the number of attempts reaching an operator and the average operator work time per call (both input database items). In reality, the relationship is not quite linear because large operator teams are more efficient than small operator teams. However, this lack of linearity occurs primarily in lightly loaded TSPSs; for moderately to heavily loaded TSPSs, a linear relation is sufficient.

Simplifying assumptions are also made in modeling the required TSPS trunks. First, minor trunk types are considered as a single category, a simplification that cuts down required input data considerably. Second, for each trunk type considered by the program, it is assumed that groups are of equal size, and that each group handles the same number of attempts with the same holding time per attempt. The average trunk group size is an input item, and the holding time is derived from input items. These approximations are quite suitable for long-range planning, and because of them, the input database is made far easier for the user to create.

### 3.2.2 Network

Even though OSTNPS is fundamentally an operator node planner, there must be a way to estimate the cost of trunk movements resulting from shifts in loads. Trunk movements occur in the toll-connect and intertoll fields. In the toll-connect field, it is adequate to assume that when a deload occurs, the appropriate number of trunks are disconnected from the deloaded operator node and the same number reconnected to the other (if the two operator nodes are colocated, the costs are different than if they are not, and this is properly taken into account). This information, together with percentages of 2-wire and 4-wire bridges, facilities, and average mileages between the local offices and toll switches before and after, is sufficient to get a good estimate

of the toll-connect movement costs. Intertoll movement also occurs after a deload since the load on all toll switches is affected, depending on community of interest. It would be unnecessarily complex to reengineer the toll network after each deload or purchase. Instead, the change in offered load between every pair of toll switches is estimated and converted to trunks. Because intertoll trunk groups are generally larger and thus more efficient than toll-connect trunk groups, it is assumed that an intertoll trunk carries twice the load as a toll-connect trunk.

## IV. STRUCTURE OF PROGRAMS

### *4.1 Introduction*

The job of generating and evaluating a long-range operator services alternative can be split into four separate tasks, each programmed as a separate OSTNPS module. Figure 1 shows a general flowchart of the process, and Fig. 2 illustrates the scope of each of these programs in the overall operator services network.

(*i*) Node exhaust. The Node Exhaust program calculates requirements for each TSPS/RTA node in each year, informs the user about possible exhausts, and lets the user relieve the exhausts by purchasing new nodes, moving traffic, or converting to TSPS No. 1B. This program does not involve itself with the network connecting the nodes—this task is performed later.

(*ii*) PSS. The PSS program, given overall position requirements from the Node Exhaust program, assigns specific PSSs for these requirements and builds the network joining those PSSs to the TSPSs.

(*iii*) Remainder of network. The Network program assigns specific trunk groups between local areas, TSPS/RTA's, and toll switches. As discussed in Section 4.4, the detailed network can be reconstructed given the global requirements output by the Node Exhaust program for each TSPS/RTA.

(*iv*) Translation for Economic Analysis. The Translator program combines the output of the preceding two programs and transforms it into a "network file" for the economic evaluator program TASP (Toll Alternatives Studies Program).[5] TASP, a very detailed economic evaluator based on Capital Utilization Criteria (CUCRIT), requires a user-supplied "cost file" as well as the above network file.

Following is a detailed description of each of these programs.

### *4.2 Node Exhaust program*

#### *4.2.1 Summary*

The Node Exhaust program is the first module of OSTNPS. Starting with a set of user-supplied input data describing the operator nodes

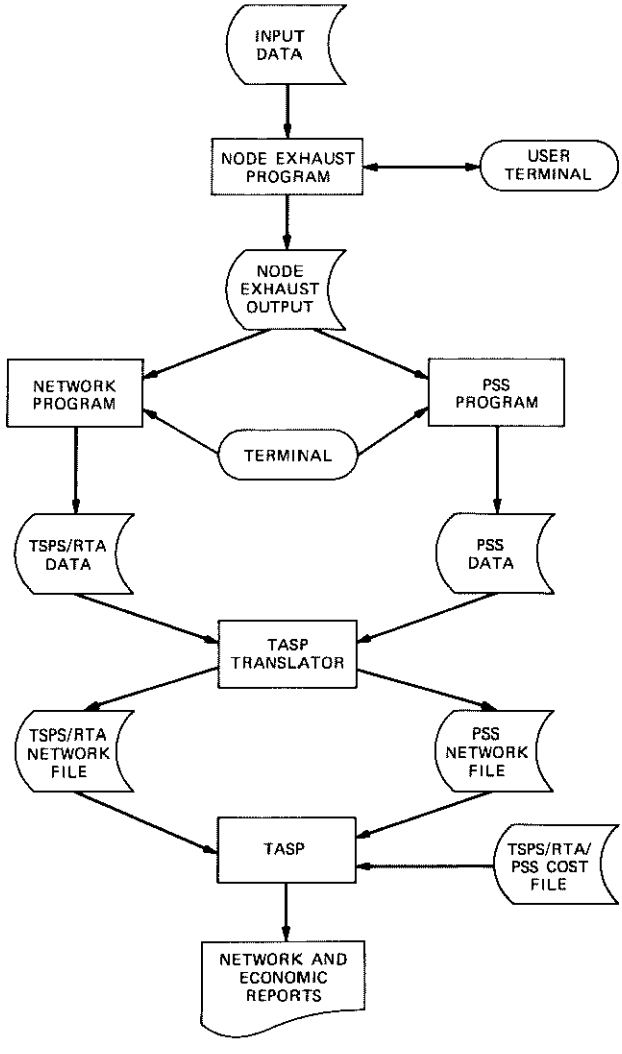



Fig. 1—Flowchart of OSTNPS.

(TSPSs and RTAs) in the study area, the program generates part of an alternative interactively. The alternative is refined further in succeeding OSTNPS modules. Figure 3 shows a general flowchart of the Node Exhaust program.

For each study year, the program forecasts requirements for relevant TSPS/RTA items and compares these numbers with their respective

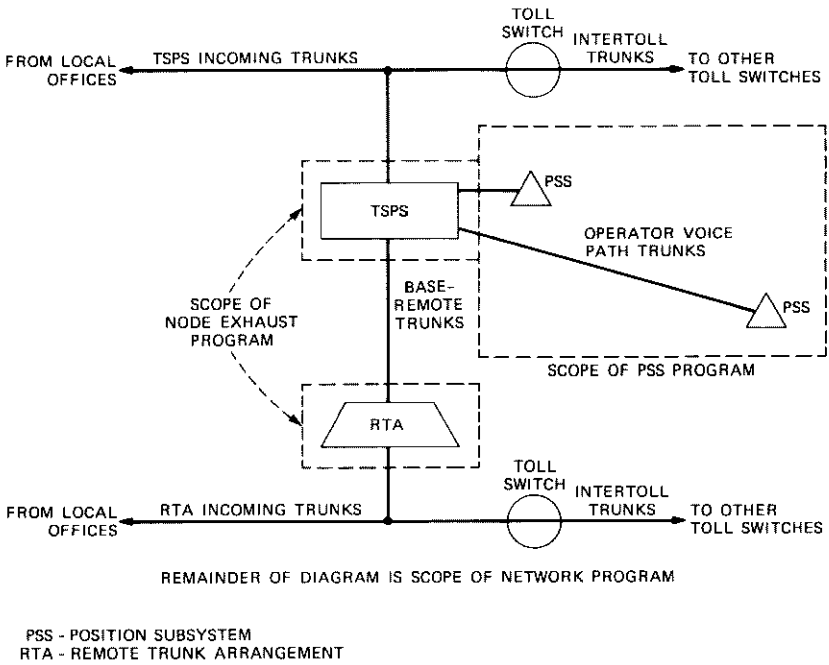



Fig. 2—Scope of OSTNPS program.

capacities. If an exhaust occurs, the program prints out the operator node that is exhausted, along with the cause(s) of exhaust, and a complete status report on all other operator nodes. The user is given an opportunity to purchase new nodes, convert to TSPS No. 1B, or to move traffic from one node to another. The program will proceed to the next study year only if there are no unrelieved exhausts. After completion of the last study year, the alternative is stored in a data base to be used by subsequent modules of OSTNPS.

It should be emphasized that the user interacts with the program on a year-by-year basis while the program is running. It is not necessary for the user to know exactly what traffic moves and node purchases to make before the study begins.

#### *4.2.2 Description of exhaust causes*

The Node Exhaust program considers TSPS/RTA exhaust causes of several types. The most common type of exhaust for a TSPS No. 1 is real time; with the development of TSPS No. 1B and its associated increase in real-time capacity, a TSPS may exhaust owing to a number of other causes. Following is a description of other exhaust causes.

***4.2.2.1 TLN terminations and individual trunk types.*** Various types of

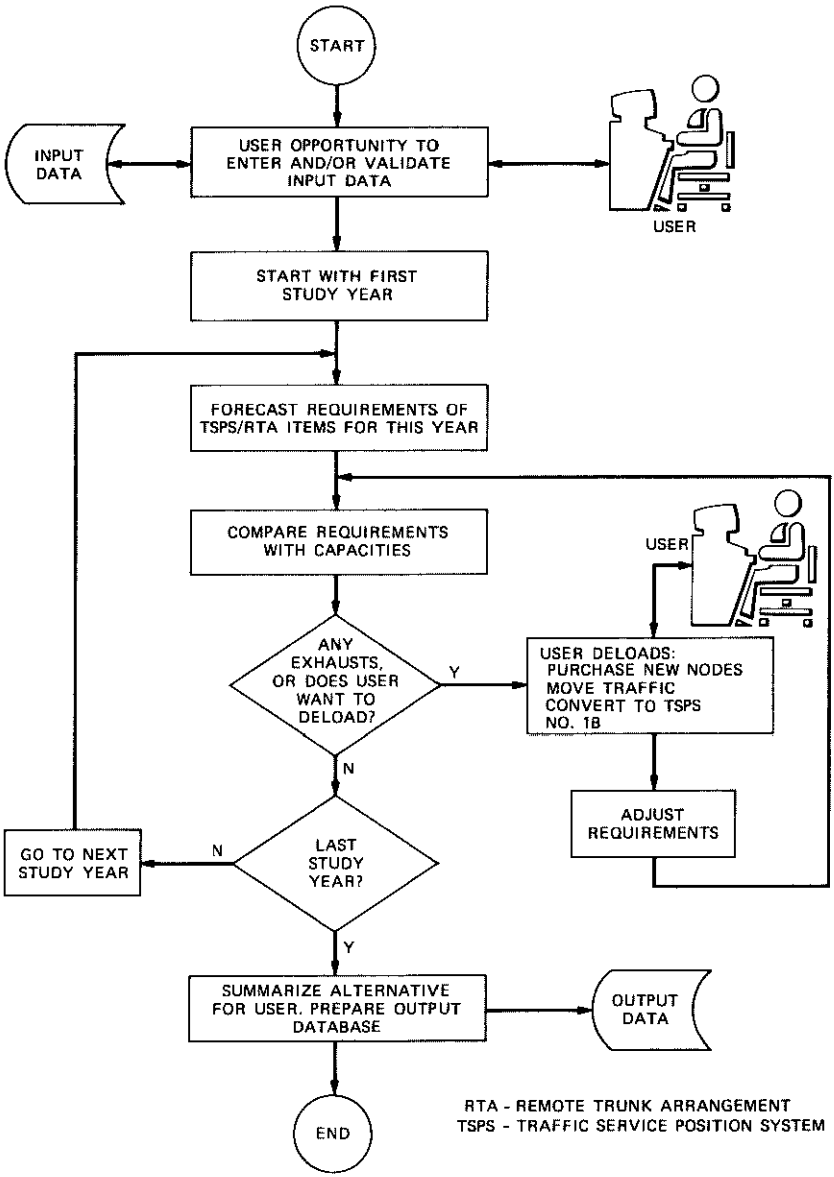



Fig. 3—Flowchart of Node Exhaust program.

TSPS trunks terminate on Universal Trunk Frames (UTFs) which in turn attach to the Trunk Link Networks (TLNs). These trunk types include: incoming ("universal") trunks, Transfer Centralized Automatic Message Accounting (XCAMA) trunks, Base-remote (BR) trunks, and all other TSPS trunks considered as a single category.[6]

A TSPS becomes TLN-exhausted if, after engineering each TLN trunk type and fitting these trunks on UTFs, the required number of UTFs (including an administrative margin) exceeds the hardware limitation. In addition, a TSPS can exhaust if one of the individual trunk types exceeds its software, design, or hardware limitation.

*4.2.2.2 PLN appearances and positions.* The Position Link Networks (PLNs) contain terminations by which operator positions and various service circuits (receivers, outpulsers, announcement circuits, etc.) connect to the TSPS. As in the TLN case, a TSPS becomes PLN-exhausted if, after engineering positions and service circuits, the total number of PLN appearances exceeds hardware limitations.

*4.2.2.3 TSPS network.* TSPS traffic travels between trunks requiring service and circuits/operators providing service via the TSPS network. A given trunk can connect to a given position or service circuit via one of eight paths composed of "A," "B," and "C" links. The TSPS network is limiting if its offered Erlang load is sufficiently great to result in a blocking probability greater than 0.001.

*4.2.2.4 TSPS office data memory.* TSPS memory is subdivided into two types: the generic program and office data. Office data memory contains information specific to the design and traffic characteristics of a given TSPS site. Conversion of a TSPS No. 1 to a TSPS No. 1B increases the total amount of addressable memory as well as real time.

### *4.2.3 Input data*

The input data to the Node Exhaust program include, for each TSPS/RTA in the study area, base year requirements for various trunk types as well as current and forecast call volumes of various types.

The following data are required:

- Call volumes: trunk seizures, position seizures, and XCAMA trunk seizures
- Trunks: base and RTA incoming trunks, XCAMA trunks, and all other trunks except BR
- Real-time capacity in trunk seizures
- Operator Average Work Time (AWT).

### *4.2.4 Forecast of TSPS/RTA items*

For a given study year, the Node Exhaust program uses these input data to forecast TSPS/RTA items that determine exhausts. The program uses simple formulas or linear regressions in place of detailed calculations.

Figure 4a shows a sample printout of these items from an actual study. At user request, this printout will appear in any study year while the Node Exhaust program is running.

| | TSP1 | TSP2 | TSP3 |
|---|---|---|---|
| BASE INC TSZ | 5226 | 11730 | 8241 |
| XCAMA TSZ | 800 | 1300 | 1200 |
| RTA RTA1 - TSZ | 2264 | 0 | 0 |
| RTA RTA2 - TSZ | 1144 | 0 | 0 |
| RTA RTA3 - TSZ | 1116 | 0 | 0 |
| RTA RTA4 - TSZ | 1400 | 0 | 0 |
| RTA RTA5 - TSZ | 0 | 0 | 984 |
| RTA RTA6 - TSZ | 0 | 0 | 1230 |
| RTA RTA7 - TSZ | 0 | 0 | 1038 |
| RTA RTA8 - TSZ | 0 | 0 | 1146 |
| TOTAL NON-XC TSZ | 11150 | 11730 | 12639 |
| TOTAL TSZ | 11950 | 13030 | 13839 |
| R-T CAPY (TSZ) | 16007 | 16587 | 17621 |
| REALTIME - % RTC | 74 | 78 | 78 |
| BASE INC TKS | 555 | 1786 | 959 |
| XCAMA TKS | 24 | 36 | 24 |
| RTA RTA1 - INC TKS | 283 | 0 | 0 |
| RTA RTA2 - INC TKS | 143 | 0 | 0 |
| RTA RTA3 - INC TKS | 186 | 0 | 0 |
| RTA RTA4 - INC TKS | 280 | 0 | 0 |
| RTA RTA5 - INC TKS | 0 | 0 | 246 |
| RTA RTA6 - INC TKS | 0 | 0 | 246 |
| RTA RTA7 - INC TKS | 0 | 0 | 173 |
| RTA RTA8 - INC TKS | 0 | 0 | 191 |
| OTH BASE TKS, EX BR | 73 | 213 | 68 |
| TLN - TOTAL UTF | 5 | 8 | 6 |
| POSITIONS (TRAFFIC) | 71 | 80 | 53 |
| PLN - TOTAL APPR | 276 | 300 | 251 |
| NETWORK (ERLANGS) | 75 | 92 | 105 |
| MEMORY (OFC NMCDS) | 4 | 5 | 5 |
| POSITION SEIZURES | 7205 | 8082 | 5552 |

Fig. 4(a)—Node Exhaust printout before user move—1984.

The following forecasts are made:

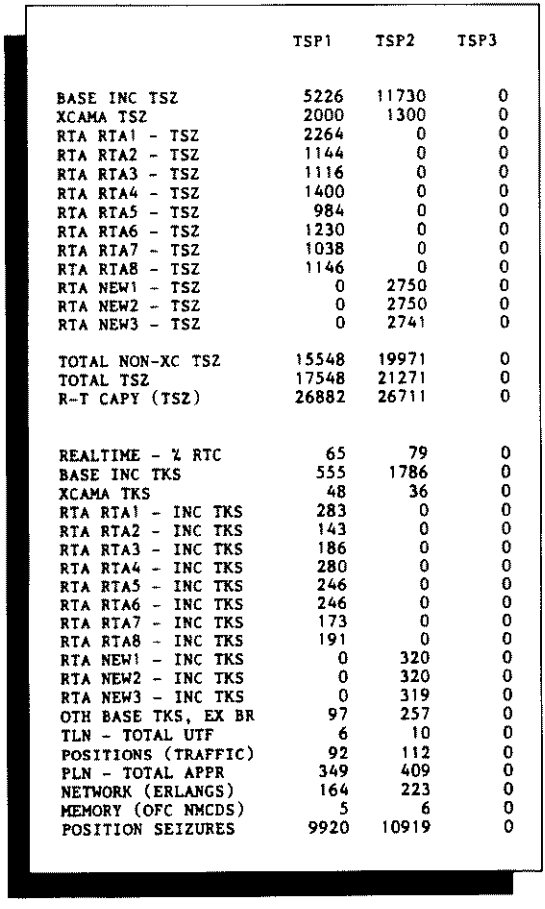
- Real-time usage
- Trunks of various types
- Memory
- Operator positions
- Service circuits
- TSPS network load.

#### 4.2.5 Exhaust prediction

Once the above items are forecast in a given study year for every TSPS/RTA in the study, the program compares these numbers with their respective upper limits. The program thus determines for each TSPS/RTA whether it is exhausted and for what reasons.

#### 4.2.6 User-specified changes to network

Input commands to the Node Exhaust program allow the user to purchase new operator nodes or move traffic from one node to another



| | TSP1 | TSP2 | TSP3 |
|---|---|---|---|
| BASE INC TSZ | 5226 | 11730 | 0 |
| XCAMA TSZ | 2000 | 1300 | 0 |
| RTA RTA1 - TSZ | 2264 | 0 | 0 |
| RTA RTA2 - TSZ | 1144 | 0 | 0 |
| RTA RTA3 - TSZ | 1116 | 0 | 0 |
| RTA RTA4 - TSZ | 1400 | 0 | 0 |
| RTA RTA5 - TSZ | 984 | 0 | 0 |
| RTA RTA6 - TSZ | 1230 | 0 | 0 |
| RTA RTA7 - TSZ | 1038 | 0 | 0 |
| RTA RTA8 - TSZ | 1146 | 0 | 0 |
| RTA NEW1 - TSZ | 0 | 2750 | 0 |
| RTA NEW2 - TSZ | 0 | 2750 | 0 |
| RTA NEW3 - TSZ | 0 | 2741 | 0 |
| TOTAL NON-XC TSZ | 15548 | 19971 | 0 |
| TOTAL TSZ | 17548 | 21271 | 0 |
| R-T CAPY (TSZ) | 26882 | 26711 | 0 |
| REALTIME - % RTC | 65 | 79 | 0 |
| BASE INC TKS | 555 | 1786 | 0 |
| XCAMA TKS | 48 | 36 | 0 |
| RTA RTA1 - INC TKS | 283 | 0 | 0 |
| RTA RTA2 - INC TKS | 143 | 0 | 0 |
| RTA RTA3 - INC TKS | 186 | 0 | 0 |
| RTA RTA4 - INC TKS | 280 | 0 | 0 |
| RTA RTA5 - INC TKS | 246 | 0 | 0 |
| RTA RTA6 - INC TKS | 246 | 0 | 0 |
| RTA RTA7 - INC TKS | 173 | 0 | 0 |
| RTA RTA8 - INC TKS | 191 | 0 | 0 |
| RTA NEW1 - INC TKS | 0 | 320 | 0 |
| RTA NEW2 - INC TKS | 0 | 320 | 0 |
| RTA NEW3 - INC TKS | 0 | 319 | 0 |
| OTH BASE TKS, EX BR | 97 | 257 | 0 |
| TLN - TOTAL UTF | 6 | 10 | 0 |
| POSITIONS (TRAFFIC) | 92 | 112 | 0 |
| PLN - TOTAL APPR | 349 | 409 | 0 |
| NETWORK (ERLANGS) | 164 | 223 | 0 |
| MEMORY (OFC NMCDS) | 5 | 6 | 0 |
| POSITION SEIZURES | 9920 | 10919 | 0 |

Fig. 4(b)—Node Exhaust printout after user move—1984.

at any time. These steps are mandatory if there is an unrelieved exhaust; otherwise they are optional (as in the case of TSPS consolidation). The possible options are as follows:

- The user may convert a TSPS No. 1 to TSPS No. 1B.
- The user may purchase a new TSPS or RTA.
- The user may move TSPS or RTA incoming trunk seizures from one node to another.
- The user may rehome an RTA from one TSPS to another.
- The user may move XCAMA trunk seizures from one TSPS to another.
- The user may retire a TSPS or RTA.

If exhausts are still not relieved after such a series of steps, the

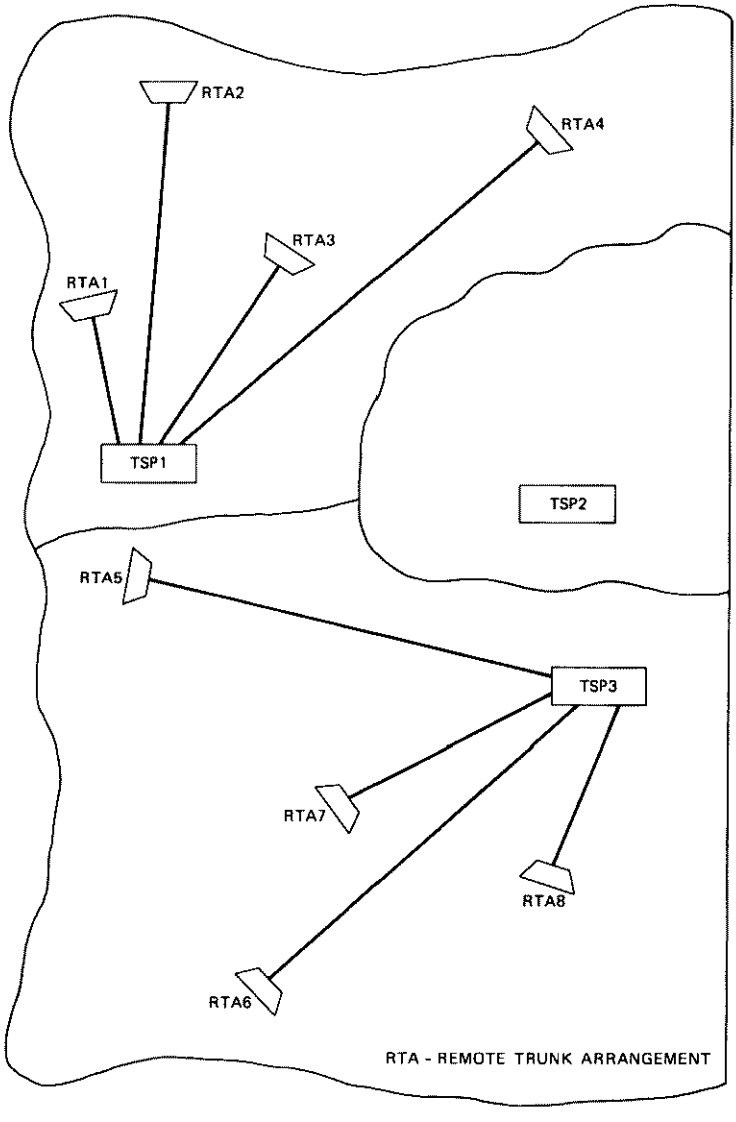



Fig. 5(a)—TSPS/RTA configuration before user move.

program will not proceed to the next study year. The user must start over with the given study year and try another strategy.

#### *4.2.7 Example*

Figures 4 and 5 illustrate the results of a set of moves made during a run of the Node Exhaust program. Figure 4 lists the relevant forecasts

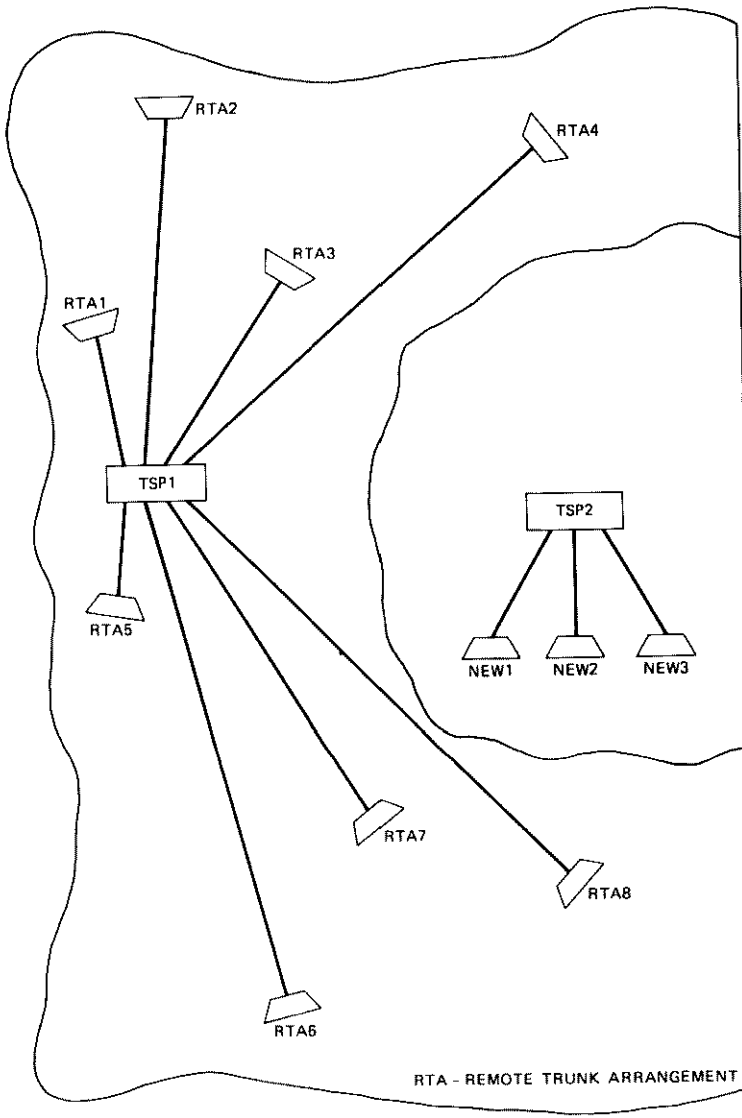



Fig. 5(b)—TSPS/RTA configuration after user move.

before and after the moves, and Fig. 5 shows a pictorial representation of this situation.

The study area consists of three TSPSs: TSP1, TSP2, and TSP3, all of which are TSPS No. 1's. These are the columns in Figures 4a and 4b. Each row represents a given forecast item for the study year, in this case, 1984. The tables are split into two sections: "User-Controlled

Items" (trunk seizures which the user may directly move from one operator node to another) and "Other Relevant Items" (exhaust-determining items which indirectly change as a result of such a move).

To evolve from Fig. 4a to 4b, the user retires TSP3 in the following way. First, RTAs RTA5, RTA6, RTA7, and RTA8, presently homed on TSP3, are rehomed to TSP1. Second, RTAs NEW1, NEW2, and NEW3 are purchased, homed on TSP2, and loaded with all of TSP3's base incoming traffic. Finally, all of TSP3's XCAMA traffic is moved to TSP1, leaving TSP3 with no traffic and hence retired. The remaining TSPSs TSP1 and TSP2 exhaust real time as a result of the move, and they are converted to TSPS No 1B's.

#### *4.2.8 Completion of run*

Once all years in the study period have been covered, the relevant information for the alternative is printed on the user's terminal. The information is also organized into an output database for use by the PSS and Network programs described below.

### *4.3 PSS program*

#### *4.3.1 Summary*

The main function of the PSS program is to establish a schedule for Position Subsystems (PSSs). This task can be separated from all others because its only input is the total required number of operator positions on each TSPS for each study year. Numbers of positions are established in the Node Exhaust program depending on incoming traffic to the TSPSs. Once the schedule is established, an additional task is to obtain facilities between each PSS and its home TSPS.

*4.3.1.1 PSS schedule.* The schedule is determined by the following considerations:

- Total number of positions required on each TSPS in each study year.
- Equipment actually in the field in the base year.
- The absolute and recommended maximum number of positions per PSS.
- The type of PSS (PSS1 or PSS2).*
- Minimization of the number of PSSs in the study, subject to administrative constraints and operator availability.

The solution consists in presenting to the user a schedule that is compatible with all the constraints, as in Fig. 6a, and letting the user modify it as desired by means of interactive commands, as in Fig. 6b.

Following is a description of Fig. 6a. First, equipment in the field

* PSS1 positions are no longer being manufactured, and PSS1s remain capped at their present position levels.

| | 1980 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TSPS1 | | | | | | | | | | | | | | | | | | | | | |
| PS11* | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 |
| PS12 | 30 | 18 | 23 | 27 | 33 | 34 | 36 | 38 | 40 | 42 | 45 | 48 | 50 | 50 | 28 | 30 | 33 | 35 | 38 | 40 | 43 |
| PS13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TSPS2 | | | | | | | | | | | | | | | | | | | | | |
| PS21* | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 |
| PS22* | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 |
| PS23 | 30 | 0 | 0 | 0 | 14 | 14 | 17 | 20 | 23 | 25 | 28 | 32 | 35 | 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TSPS3 | | | | | | | | | | | | | | | | | | | | | |
| PS31* | 55 | 55 | 55 | 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PS32* | 28 | 28 | 28 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PS33 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TSPS4 | | | | | | | | | | | | | | | | | | | | | |
| PS41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 | 50 | 50 | 50 | 50 | 50 | 50 |
| PS42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 29 | 32 | 34 | 37 | 40 | 43 |

Fig. 6(a)—A first-order PSS field.

| | 1980 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TSPS1 | | | | | | | | | | | | | | | | | | | | | |
| PS11* | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 |
| PS12 | 30 | 18 | 23 | 27 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 28 | 30 | 33 | 35 | 38 | 40 | 43 |
| PS13 | 0 | 0 | 0 | 0 | 3 | 4 | 6 | 8 | 10 | 12 | 15 | 18 | 20 | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TSPS2 | | | | | | | | | | | | | | | | | | | | | |
| PS21* | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 |
| PS22* | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 | 47 |
| PS23 | 30 | 0 | 0 | 0 | 14 | 14 | 17 | 20 | 23 | 25 | 28 | 32 | 35 | 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TSPS3 | | | | | | | | | | | | | | | | | | | | | |
| PS31* | 55 | 55 | 55 | 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PS32* | 28 | 28 | 28 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PS33 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TSPS4 | | | | | | | | | | | | | | | | | | | | | |
| PS41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 | 50 | 50 | 50 | 50 | 50 | 50 |
| PS42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 29 | 32 | 34 | 37 | 40 | 43 |

Fig. 6(b)—PSS field after the first move.

(base year) is entered by the user (1980 in this case). PSS1's, for which positions are capped, are marked with a "*". For the study years (1981–2000), the scheduler bases its calculations on position requirements and purchases new PSSs where needed.

Given this PSS schedule, the user can modify it with the following interactive commands:

- For a given TSPS, the user may redistribute the number of PSS2 positions.
- The user may rehome a PSS2 from one TSPS to another.

- The user may or may not release unneeded PSS2 positions for reuse.

Consider Fig. 6b. The first move consists in all excess positions beyond 30 in the base year being moved from PS12 to PS13 for all study years up to 1994. A possible second move consists of making PS33, PS13, and PS42 the same PSS by rehoming PS33 (originally homed on TSPS3) to TSPS1 in 1984, then to TSPS4 in 1994.

*4.3.1.2 PSS facilities.* Once the schedule is established, the program asks the user for facility data between TSPSs and the PSSs. This, together with the schedule, completely determines the input to the economic analysis program.

#### *4.4 TSPS/RTA Network program*

Besides the network joining TSPSs to PSSs, there is the network joining local offices and toll switches to TSPS/RTA's. This network can be divided into three distinct portions:

- Toll connect trunks (local office to TSPS/RTA to toll switch).
- Base remote trunks (RTA to TSPS).
- Intertoll trunks.

Figure 7b shows an example of such a network.

The Node Exhaust program allocated total trunk seizures amongst the TSPS/RTA's without making any assumptions about the network. The separation of exhaust planning and network planning allows simplified program design, and allows the user to concentrate on each issue separately. The method used to model the network allows this separation to work; the network program guides the user through the steps of establishing the network corresponding to the alternative formulated by the Node-Exhaust program.

At the beginning of the study, there is by definition a one-to-one correspondence between "local areas" and TSPS/RTAs (Fig. 7a). User-specified traffic moves made during a node exhaust run may affect the network in the following ways (see Fig. 7b):

- A move of base-remote trunks, also called an RTA rehome (a) in Fig. 7b.
- A move of toll-connect trunks (b) in Fig. 7b.
- Changes in the intertoll network. A certain percentage of operator traffic leaving a local area must go back to the same local area via its toll switch. Corresponding to each toll-connect move (b) is an increase in required intertoll trunks (b'). No such intertoll adjustment is necessary for a base-remote move.

The Network program obtains the following information from the user:

- Toll switch information: names, types, homing TSPS/RTAs.
- Base-remote mileage and facility information.

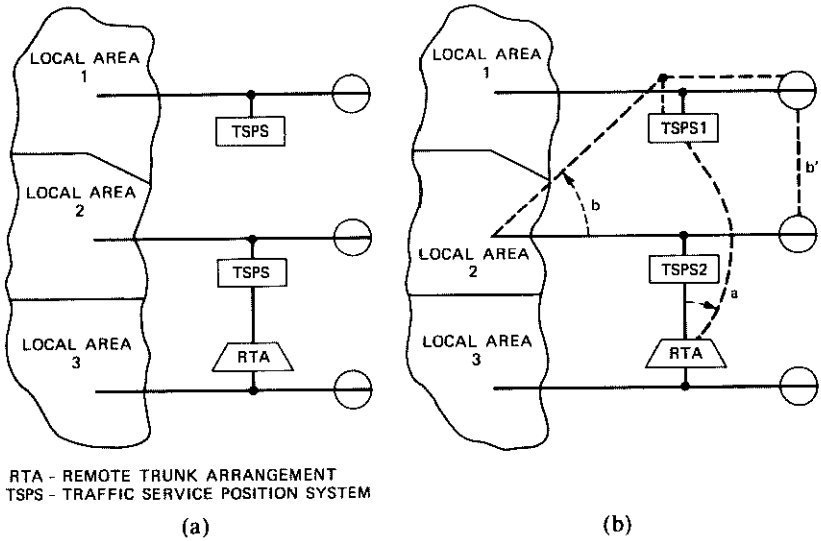



Fig. 7—(a) Relationship of local offices and toll switches to TSPS/RTA's—the correspondence is one to one. (b) Effect on network of user-specified traffic moves during a Node Exhaust run.

- Toll-connect mileage and facility information, including the percentage of 2-wire and 4-wire bridges.
- Community of interest and intertoll mileage/facility information.

### *4.5 Economic evaluation and translator programs*

Economic evaluation of the alternatives is performed by TASP, which has already been mentioned. Telephone company planners are very familiar with this program, which has been in use for years. TASP requires a considerable amount of input, even for studies involving only a few switches. A fourth program, the Translator program, transcribes the output of the three preceding programs into TASP format without user intervention. The PSS and TSPS/RTA Network programs write files in a compact language which is translated into TASP language by the Translator program.

## V. RUNNING THE PROGRAMS

The generation and economic evaluation of a complete alternative with OSTNPS involves the following steps to be taken by the user:

- Run the Node Exhaust program.
- For the above Node Exhaust program alternative, run one or a few appropriate PSS alternatives.
- Run each of these PSS alternative through TASP and choose the best.

- Run the Network program for the above Node Exhaust program alternative.
- Concatenate the best PSS alternative with the Network program alternative, and run this complete alternative through TASP.

This TASP output constitutes the economic evaluation of this complete alternative, to be compared with others.

## VI. CONCLUSION

OSTNPS is an operator services long-range planning tool. It allows the user to analyze in a short time different strategies of operator services network evolution. For high-growth areas, OSTNPS evaluates various types of load-balancing alternatives versus purchase of new equipment. For low-growth areas with underutilized nodes, OSTNPS helps the user determine whether or not to retire those nodes. For both of these cases, OSTNPS allows the user to easily incorporate the conversion to TSPS No. 1B into the planning process. OSTNPS is not an optimizer, but instead lets users generate and evaluate their own alternatives.

## VII. ACKNOWLEDGMENTS

The authors wish to thank R. G. Crafton of AT&T, who contributed in innumerable ways, and the operator services planning organizations of Mountain Bell Telephone and Bell of Indiana, who helped us during program development.

## REFERENCES

1. R. J. Jaeger, Jr. and A. E. Joel, Jr., "TSPS No. 1: System Organization and Objectives," B.S.T.J., *49*, No. 10 (December 1970), pp. 2417–43.
2. R. E. Staehler and J. I. Cochrane, "Traffic Service Position System No. 1B: Overview and Objectives," B.S.T.J., this issue.
3. R. G. Crafton, "Network Planning for Operator Services Systems," Int. Switching Symp., Montreal, 1981, Session 21, C Paper 3, pp. 1–5.
4. S. M. Bauman, R. S. DiPietro, and R. J. Jaeger, Jr., "TSPS No. 1: Remote Trunk Arrangement: Overall Description and Operational Characteristics," B.S.T.J., *58*, No. 6 (July–August 1979), pp. 1119–35.
5. B. H. Fetz and P. M. Moricz, "Tipping the Scales for No. 4 ESS," Bell Lab. Rec., *58*, No. 5 (May 1980), pp. 138–45.
6. W. K. Comella, C. M. Day, Jr., and J. A. Hackett, "TSPS No. 1: Peripheral Circuits," B.S.T.J., *49*, No. 10 (December 1970), pp. 2561–623.

## ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| 3B20D | 3B20 Duplex Processor |
| 3bas | 3BD assembler |
| 3bcc | 3BSGS C compiler |
| 3bld | 3B20D link editor |
| 3bldp | 3B20D process loader |
| 3BPEST | 3B20D inhibits |
| 3BSGS | 3B20D Software Generation System |
| ACHI | application channel interface |
| ACPDL | application control process data library |
| AIM | application integrity monitor |
| ALU | arithmetic/logic unit |
| AMA | automatic message accounting |
| APE | advanced processor editor |
| ARA | answer register A |
| ARB | answer register B |
| AST | announcement stores |
| ASW | all seems well |
| ATB | address translation buffer |
| AWT | average work time |
| BBR | buffer bus to register |
| BPI | bit per inch |
| BR | base remote |
| BTR | bus terminating resistor |
| BVA | billing validation application |
| CALE | call to emulation |
| CAR | channel address register |
| CBB | constant to buffer bus |
| CBT | communications bus translator |
| CC | central control |
| CC | control console |
| CCIO | central control input/output |
| CCIS | common channel interoffice signaling |
| CD | control and display |
| C&D | Control and Display mode |
| CDR | channel data register |
| CDT | control display and test |
| CHK/GEN | check/generator |
| CIP | Critical Indicator panel |
| CLRFLG | clear flag |
| CMON | craft monitor |
| CMS | Change Management System |
| CPDB | central pulse distributor bus |

| | |
|---|---|
| CPD | central pulse distributor |
| CRC | Change Review Committee |
| CR | correction report |
| CRT | cathode ray tube |
| CSOP | controller of output spooler process |
| CSS | computer subsystem |
| CU | control unit |
| CUCRIT | Capital Utilization Criteria |
| CU/PSI | control unit/peripheral system interface |
| CUPERPH | control unit peripheral problem |
| DAP | display administration process |
| DFC | disk file controller |
| DFC/MHD | disk file controller with removable head disk |
| DGDL | diagnostic data library |
| DIAMON | diagnostic monitor |
| DIST | distributes |
| DMA | direct memory access |
| DMERT | Duplex Multi-Environment Real-Time |
| DMU | data manipulation unit |
| DRMO | detect rightmost one |
| DSCH | dual serial channel |
| DUC | dual utility circuit |
| DZRMO | detect and zero rightmost one |
| EA | enable address |
| EAI | emergency action interface |
| ECD | equipment configuration data |
| ECD | Equipment Configuration Database |
| ECD/SG | Equipment Configuration Database/System Generation Database |
| ECIO | executive control for input/output |
| ECMP | executive control for the main program |
| EGBN | execute go back to normal |
| EIH | error-interrupt handler |
| EL | execution level |
| ERD | emergency recovery display |
| ESS | Electronic Switching System |
| EXC | execute |
| FLZ | find low zero |
| FSI | file system interface |
| FTS | field test set |
| GBNHJ | go back to normal H-level or J-level |
| GRASP | Generic Access Program |
| HOBIS | Hotel Billing Information System |
| HSR | hardware status register |

| | |
|---|---|
| I | inhibit interrupt |
| IB | instruction buffer |
| ILAF | interrupt-level activity flags |
| IM | interrupt mask |
| INT | interrupt |
| INTS | interrupt source register |
| I/O | input/output |
| IOP | input/output processor |
| IS | interrupt source |
| ISR | interrupt source register |
| JCL | job control language |
| LMP | local maintenance position |
| LOTS | local toll simulator |
| LSS | Laboratory Support System |
| MAIS | maintenance interrupt sources |
| MCC | maintenance control center |
| MCC | master control center |
| MCCD | maintenance control center data |
| MCCI | maintenance control center interrupts |
| MCH | maintenance channel |
| MCRT | maintenance CRT |
| MESS | microprocessor service evaluation system simulator |
| MHD | moving head disk |
| MICLOB | microprocessor-controlled load box |
| MIP | micro-level test set interface program |
| MIRA | maintenance input request administrator |
| MLTS | micro-level test set |
| MNA | minor audit initialization |
| MOPS | microprocessor operator position simulator |
| MR | maintenance register |
| MR | memory to register |
| MR | modification request |
| MSEC | millisecond clock state |
| MS | master scanner |
| MTTR | mean time to repairs |
| MTTYC | maintenance teletype controller |
| MTTY | maintenance teletypewriter |
| MVS | multiple virtual storages |
| OBB | ones to buffer bus |
| ONI | operator number identification |
| OSTNPS | Operator Services Traffic Network Planning System |
| OST | operating system trap |
| PA | present address |
| PBCP | peripheral bus control process |

| | |
|---|---|
| PC | peripheral control |
| PC | peripheral controller |
| PC | protocol converter |
| PCB | process control block |
| PCPEIH | processor control process error-interrupt handler |
| PCPMD | processor control process maintenance driver |
| PD | peripheral device |
| PD | power distribution |
| PEST | interrupt inhibits |
| PIR | program interrupt request |
| PLN | position link network |
| PPR | pulse point register |
| PROM | programmable read-only memory |
| PSI | peripheral system interface |
| PSIDGDR | peripheral system interface diagnostic driver |
| PSIDIAGC | peripheral system interface diagnostic control |
| PSIEIH | peripheral system interface error-interrupt handler |
| PSS | position subsystem |
| PSS | Program Support System |
| PSW | program status word |
| PTU | program tape unit |
| PUAB | peripheral-unit address bus |
| PUMS | peripheral-unit maintenance summary |
| RAM | random access memory |
| RBB | register to buffer bus |
| RC | reply check |
| RCVP | recent change/verify position |
| RETE | return to emulation |
| REX | routine exerciser |
| RJE | remote job entry |
| RM | register to memory |
| RMU | rotate mask unit |
| ROP | read-only printer |
| ROP | receive-only printer |
| RRM | registers to memory |
| RTA | remote trunk arrangement |
| SAB | scanner answer bus |
| SAR | store address register |
| SCAN | scans |
| SCCS | Source Code Control System |
| SCCS | Switching Control Center System |
| SCC | Switching Control Center |
| SCR | store control register |
| SC/SD | scanner/signal distributor |

| | |
|---|---|
| SDR | store data register |
| SD | signal distributor |
| SDS | Software Development System |
| SER | system evaluation run |
| SES | Service Evaluation System |
| SG | system generation |
| SIA | system initialization A |
| SIB | system initialization B |
| SIM | system integrity monitor |
| SIR | store instruction register |
| SLS | single-line simulator |
| SMT | switch mode and transfer |
| SPC | stored program control |
| SPCS | Stored Program Control System |
| SSAS | station signaling and announcement subsystem |
| STF | some tests failed |
| SWAP | Switching Assembly Program |
| TASP | Toll Alternatives Studies Program |
| TLM | Trouble Locating Manual |
| TLN | trunk line network |
| TLP | trouble-locating procedures |
| TOD | time of day |
| TPCP | TSPS display page control process |
| TR | trouble report |
| TSIP | TSPS spooler input process |
| TSPNF | TSPS non-frozen development |
| TSPS | Traffic Service Position System |
| TSPSCAP | Traffic Service Position System Real-Time Capacity Program |
| TU | tape unit |
| TUS | Test Utility System |
| UCB | unit control block |
| UCG | universal call generator |
| ULARP | user-level automatic restart process |
| UTF | universal trunk frame |
| UTSC | universal trunk scanner |
| UTSD | universal trunk signal distributor |
| VC | virtual channels |
| VTP | virtual terminal protocol |
| WRMI | we really mean it |
| XCAMA | transfer centralized automatic message accounting |

## CONTRIBUTORS TO THIS ISSUE

**R. Ahmari,** B.S.E.E., 1966, University of Tehran; M.S. (E.E.), 1969, and Ph.D. (E.E.), 1972, Illinois Institute of Technology; Assistant Professor, Manhattan College, New York, 1972–1973; Bell Laboratories, 1973—. Mr. Ahmari worked on system planning for private networks, design and development of fault-tolerant systems, and system testing and integration for TSPS. He is currently supervising a group engaged in software design, and development coordination of TSPS No. 1B Generic 1BT2. Mr. Ahmari is a Registered Professional Engineer in the State of Illinois. Member, IEEE.

**Pierre L. Bastien,** B. S. (Physics), University of Montreal; Ph.D. (Physics), University of California at Berkeley; Bell Laboratories, 1973—. Upon joining Bell Laboratories in 1973 Mr. Bastien first studied the effects of No. 4 ESS failure on the toll network and local offices. More recently he has worked on long-range planning tools: The Operator Services Traffic Network Planning System for operator services; and the Switching Studies Tool, for toll planning. Currently, Mr. Bastien is a Member of Technical Staff in the SPC Network Planning Methodology Group working on development of a planning tool for long-range toll switch studies and point-of-presence (interconnection) planning for post-divestiture.

**Joseph J. Bodnar,** B.S.E.E., 1969, Rutgers University, M.S.E.E., 1971, Polytechnic Institute of Brooklyn; M.S. (Advanced Management), 1980, Pace University; Bell Laboratories, 1969–1976; AT&T, 1976–1979; Bell Laboratories, 1979—. Mr. Bodnar worked on the development of the Switching Control Center System at Bell Laboratories. He then joined AT&T Network Operations where he was responsible for TSPS and ESS operations methods and technical support. He returned to Bell Laboratories in systems engineering and is currently Supervisor of the Switching Control Center Systems group. Member, Tau Beta Pi, Eta Kappa Nu.

**G. T. Clark,** B.S.M.E., 1952, Bradley University; Western Electric, 1956–1961; Bell Laboratories, 1961—. Mr. Clark was first engaged in the physical design of step-by-step common control equipment and later worked on the design of 758C PBX equipment. He has coordinated the physical design of TSPS No. 1 equipment, including detail design of network, position subsystem, TTY trunk and buffer, the station signaling and announcement subsystem, Peripheral System

Interface frame, and processor replacement for TSPS No. 1B. He was also engaged in the physical design of AIS equipment, including the file subsystem No. 2. He is currently working on physical design of new features for TSPS No. 1B.

**J. I. Cochrane,** B.S.E.E., 1962, M.S.E.E., 1966, Georgia Institute of Technology; Ph.D., E.E. (Communication Theory), N.Y.U.; Honeywell, Inc., 1963–1965; Bell Laboratories, 1966—. From 1966–1969, Mr. Cochrane worked on the design of prototype signal processors for ballistic missile defense radars. In 1969 he was appointed Supervisor of a group responsible for the system design of advanced missile defense radars. From 1972–1973 he supervised the group responsible for the Perimeter Acquisition Radar software performance requirements. From 1973 to 1976 he supervised the formulation of long-range plans for the Navy Telecommunications System. From 1976 to 1977 he supervised the determination of the requirements for new features and equipment to be used in the operations and administration of voice-grade switched networks for large corporate customers. In 1977 and 1978 he supervised a group responsible for formulating and analyzing concepts for new services for large corporate customers using emerging telecommunications technology. In 1978 he was appointed Head of the Network Operations Planning Department responsible for formulating long-range plans for computer-based support systems to support Bell System network operations. In 1981 Mr. Cochrane was appointed Director of the Network Switching and International Systems Engineering Center; this center is responsible for the Bell Laboratories systems engineering work on Operator Services systems and for much of the planning and systems engineering work in support of AT&T International. Licensed Professional Engineer (NJ); member, Tau Beta Pi, Eta Kappa Nu.

**Bently A. Crane,** B.S. (Physics), 1954, M.S. (Physics), 1955, University of Michigan; M.S.E.E., 1959, New York University; Bell Laboratories, 1957—. Mr. Crane was involved in exploratory development of advanced computers for ballistic missile defense systems until 1972. He then began Bell System work, which has included exploratory development of auxiliary processors and voice message systems. Mr. Crane is currently a Consultant in the Operator Services Project Planning Department, working on TSPS system performance evaluation. He recently received the Bell Laboratories Distinguished Technical Staff Award.

**John R. Daino,** B.S.E.E., 1967, Syracuse University; M.S.E.E.,

1969, Ohio State University; AT&T, 1977–1980; Bell Laboratories, 1967–1977, 1980—. After joining Bell Laboratories in 1967, Mr. Daino worked on switching systems development in No. 5 Crossbar and No. 3 ESS. In 1972, he transferred to systems engineering to work on billing systems for electronic and electromechanical switching systems. Mr. Daino transferred to AT&T Network Design in 1977 where he was responsible for ESS generic planning as well as individual switching projects. In 1980, he returned to Bell Laboratories in his present position as Supervisor of the Suburban and Local Systems Central Office Operations group.

**John C. Dalby,** Jr., B. S. (Applied Mathematics), 1968, M.S.E. (Computer, Information, and Control Engineering), 1969, University of Michigan; Masters of Philosophy (Computer Science), 1977, Rutgers University; Bell Laboratories, 1970—. Since joining Bell Laboratories Mr. Dalby has been involved in TSPS No. 1 development designing call-processing and maintenance software and writing system development requirements for new TSPS No. 1 features. Presently, he supervises a group doing No. 5 ESS Operator Services Position System Planning. Member, Tau Beta Pi.

**Noel X. DeLessio,** B.S.E.E., 1960, M.S.E.E., 1961, Ph.D. (E.E.), 1966, Polytechnic Institute of Brooklyn; Bell Laboratories, 1966—. Mr. DeLessio worked on Safeguard system design and supervised guidance design for the SPRINT missile system. Subsequently, he supervised the Exploratory Development Group of the Operator System Laboratory and is currently Head of the No. 5 ESS OSPS Development Department in the Operator Services and Digital Switching Applications Laboratory.

**Richard S. DiPietro,** B.S. (Engineering Science), 1970, Northwestern University; M.S.E.E., 1972, New York University; Bell Laboratories, 1970—. Mr. DiPietro worked on military system performance evaluation until 1974. He then worked on hardware design and system testing for the remote TSPS trunk arrangement. Since 1977, he has been Supervisor of TSPS field support, system test, and call-programming groups.

**R. J. Gill,** B.S., 1970 (Computer Science), Purdue University; M.S., 1971 (Applied Mathematics), University of Michigan; Bell Laboratories, 1970—. Mr. Gill began his work at Bell Laboratories on the Safeguard ballistic missile defense system in the area of software

architecture, design, and analysis. From 1974 through 1981 he worked on TSPS, designing call-processing software, doing exploratory development, and then supervising the Processor Application Group for TSPS No. 1B. In 1982 he became Supervisor of the OSPS Architecture Group for the No. 5 ESS Operator Services Position System.

**T. G. Hack,** B.S., 1966 (Mathematics), Xavier University; M.S., 1967, Ph.D, 1970 (Applied Mathematics), Purdue University; Bell Laboratories, 1966—. Mr. Hack initially worked on software generation tools for the No. 2 ESS and No. 3 ESS projects and then in the area of change management for C-language-based developments. In 1979, he became Supervisor of a group responsible for the software development environment for the TSPS project. He currently supervises a group responsible for the development of operational software for No. 5 ESS.

**Harry A. Hilsinger,** B.S.E.E., 1954, Newark College of Engineering; M.S.E.E., 1958, New York University; Bell Laboratories, 1954—. Mr. Hilsinger initially worked on airborne guidance systems for the Nike Zeus project. In 1961 he became Supervisor of a group doing physical design for the first electronic PBX system. He has been participating in switching system physical design since that time and is currently responsible for physical design associated with Network Operator Services. Member, Tau Beta Pi, Eta Kappa Nu.

**Teddy Huang,** B.S. (Electrical Engineering), Purdue University; M.S. (Electrical Engineering), MIT; Ph.D (Electrical Engineering), Purdue University; Bell Laboratories, 1972—. Since joining Bell Laboratories, Mr. Huang has been engaged in system simulation, maintenance software, and test utility systems development. He is currently a Supervisor of a group responsible for the overall integrity of Operator Services Position Systems.

**Gary J. Kujawinski,** B.S.E.E., 1975, M.S.E.E., 1977, University of Illinois; Bell Laboratories, 1977—. Mr. Kujawinski has worked on the TSPS No. 1B system in the software emulation, performance measurement, and field support areas. He currently supervises the TSPS Test Facility Group. Member, IEEE.

**N. A. Martellotto,** B.E.E, and B.S. (Applied Mathematics), 1957, Georgia Institute of Technology; M.E.E, 1959, New York University;

M.B.A., 1970, University of Chicago; Bell Laboratories, 1957—. Starting with the Bell System Data Processor project in 1957, where he did logic design and programming, Mr. Martellotto has been involved with computers and software development throughout his career at Bell Laboratories. He worked on EPBX and No. 1 ESS and holds a patent related to the basic notion of ESS generic programs. In 1966, he became Head of the Indian Hill Computation Center (IHCC). In 1976, he resumed design and development of ESS software development support programs and other related work. In late 1979, Mr. Martellotto became DMERT project manager and for the next two years was involved with all aspects of the project, from operating system development to field support. He is now Head of the Software Development Systems Department at Indian Hill. Member, IEEE, Tau Beta Pi, Eta Kappa Nu.

**Sherman C. Reed,** B.S.E.E., 1956, University of Okalahoma; M.S.E.E., 1958, Newark College of Engineering; Bell Laboratories, 1956—. At Bell Laboratories, Mr. Reed had many assignments in ballistic missile defense activities from 1956 to 1974. He began work on TSPS in 1974 and has been involved with call programming and operator actions software design, new feature planning development coordination and system testing. He is currently Supervisor of the TSPS No. 1B 1BT2 System Testing Group. Member, Tau Beta Pi, Eta Kappa Nu, Pi Mu Epsilon.

**Robert E. Staehler,** B.S.E.E., 1947, The College of the City of New York; M.S.E.E., 1948, Polytechnic Institute of Brooklyn; Bell Laboratories, 1948—. Mr. Staehler's early work was on No. 5 Crossbar, toll signaling systems, and trainers for guided missile systems. In 1953, he worked on the development of electronic switching systems, specifically, the processor memory for the experimental central office in Morris, Illinois, and the processor logic and call memory for No. 1 ESS. He was appointed Director of the Electronic Switching Projects Laboratory in 1964 with responsibility for special applications for No. 1 ESS to military and data networks, including No. 1 ESS AUTOVON. In 1968, he became Director of the Electronic Systems Design Laboratory with responsibility for development of the 1A Processor for No. 1 ESS and No. 4 ESS. In 1976, he became Director of the Network Operator Services and Digital Switching Applications Laboratory with responsibility for developing operator services for both domestic and international applications, along with the No. 5 ESS remote switching vehicles. Senior member, IEEE. Member, Eta Kappa Nu, Tau Beta Pi, Sigma Xi.

**L. C. Stecher,** B.S., 1967, Loyola University; M.S., 1968, Northwestern University; Ph.D., 1972 (Applied Mathematics/Computer Science), Northwestern University; Bell Laboratories, 1970—. Mr. Stecher was involved in the design and development of the No. 4 ESS maintenance and call-processing subsystems. In 1975, he became Supervisor of the No. 4 ESS trunk maintenance development effort and later became Supervisor of an exploratory effort to define new network services for the evolving Stored Program Controlled Network. In 1980, he became Head of a department responsible for development of software for the Traffic Services Position System. This software interacts with operators, customers, and the network to handle toll and assistance traffic. In addition, the department is responsible for the Software Development System and Programmer Support Systems used in the Operator Services and Digital Switching Applications Laboratory.

**E. H. Stredde,** B.S.E.E., 1966, M.S.E.E., 1967, University of Illinois; Bell Laboratories, 1967—. Mr. Stredde has developed maintenance and operational software for No. 1 ESS, the 1A Processor, No. 4 ESS, and TSPS. He is currently Supervisor of the Remote Switch Module Switch Maintenance Group. Member, Eta Kappa Nu, Tau Beta Pi.

**D. S. Suk,** B.S.E.E., 1969, Seoul National University; M.S. (E.E.), 1977, and Ph.D. (E.E.), 1978, University of Iowa; Bell Laboratories, 1978—. Mr. Suk has been involved in maintenance software design and the evaluation of TSPS No. 1B performance and reliability. He is currently working on project management and software development methodology. Member, IEEE, Eta Kappa Nu.

**James H. Tendick,** B.S.E.E., 1977, University of Illinois; M.S.E.E., 1978 Stanford University; Bell Laboratories, 1977—. Mr. Tendick was initially involved in the hardware design of the Peripheral Systems Interface for the TSPS No. 1B. He was also involved in system testing at the first live TSPS No. 1B site in Fresno, California. Other work activities included speech synthesis and recognition and field support. Presently, he is supervising a group responsible for system test and integration of the current TSPS Generic. Member, Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, Sigma Xi.

**Kendra A. VanderMeulen,** B.S. (Mathematics), 1973, Marietta College; M.S. (Computer Science), 1977, Ohio State University; Bell Laboratories, 1973—. Ms. VanderMeulen has participated in the de-

sign, development, testing, and maintenance of minicomputer-based operations systems for service evaluation and SPCS maintenance. She is currently Head of the SCCS Applications Development Department responsible for the development of SCCS application software, development of a world-class switching operations system for the international market, and development of an operations maintenance system for minicomputer maintenance and operations centers.

**Daniel Van Haften,** B.S., M.S. (Mathematics), 1970, Michigan State University; Ph.D. (Electrical Engineering), 1977, Stevens Institute of Technology; Bell Laboratories, 1970—. Mr. Van Haften initially worked on the Safeguard project. In 1974 he joined the Network Operator Services Laboratory, where he was involved in TSPS system testing, field support, call-processing development, and processor replacement development. In 1981 he became involved in No. 5 ESS site testing in Seneca, Illinois, and is presently Supervisor of the Factory Test Software Design Group in the Local Digital Switching Software Laboratory. Member, Phi Beta Kappa, Phi Kappa Phi, Pi Mu Epsilon.

**Laurance A. Weber,** B.E.E., 1945, Cornell University; M.E.E., 1955, Polytechnic Institute of Brooklyn; Bell Laboratories, 1946—. Mr. Weber was initially involved in the design of signaling circuits. Later he participated in the design of circuits for crossbar tandem systems. Following this assignment, he was appointed Supervisor in charge of designing data sets for the mechanization of TWX service. He was appointed Head of the 101 ESS Design Department in 1960. He has had subsequent assignments in No. 2, No. 2B, and No. 3 ESS. He is presently Head of the Operator Services and Digital Switching Application Laboratory System Testing and Laboratory Administration Department. Member, IEEE, Tau Beta Pi, Sigma Xi, Eta Kappa Nu.

**R. A. Weber,** B.S.E.E., 1970, Iowa State University; M.S.E.E., 1971, Stanford University; Bell Laboratories, 1970—. Mr. Weber was initially a member of the Ocean Systems Laboratory, where he worked on underwater cable systems. In 1974 he transferred to the Operator Services and Digital Switching Applications Laboratory, where he has done circuit design and diagnostic programming on the Remote Trunk Arrangement and the SPC 1B Processor. He currently is the Supervisor of a group responsible for the TSPS System software, which interfaces with the DMERT operating system. Member, Tau Beta Pi.

**J. R. Williams,** B.S.E.E., 1960, Vanderbilt University; M.S.E.E., 1961, University of Illinois; U. S. Navy Submarine Service, 1961–1964;

Bell Laboratories, 1964—. Mr. Williams has had a variety of assignments in Electronic Switching System development. His early work included assignments in system design, test system development, and operational software design for No. 1 ESS ADF, a store and forward message switching system. In 1969, he became involved in maintenance planning and hardware design for No. 4 ESS. Later assignments included responsibilities in the area of No. 4 ESS maintenance and call-processing software development. In 1977, he became responsible for No. 5 ESS maintenance planning, and in 1979, joined the TSPS project with responsibility for operational software development. In 1980, he assumed responsibility for planning future operator system developments and for system testing new TSPS features. In 1982, Mr. Williams returned to No. 5 ESS development, with responsibility for maintenance software.

**Bruce R. Wycherley,** B.A., 1976, M.A., 1978 (Mathematics), University of Oklahoma; Bell Laboratories, 1978—. At Bell Laboratories, Mr. Wycherley has worked on the development of TSPS market models, examined TSPS limiting components, and developed operator services long-range planning tools for use by telephone companies. He is presently working in the Operator Services Planning group, investigating applications of automatic speech recognition and studying the impact of divestiture on TSPS. Member, Pi Mu Epsilon, Phi Beta Kappa.

THE BELL SYSTEM
TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 April 1983 Number 4, Part 1

Copyright © 1983 American Telephone and Telegraph Company. Printed in U.S.A.

# Analysis of Thermally Induced Loss in Fiber-Optic Ribbons

By G. S. BROCKWAY* and M. R. SANTANA

(Manuscript received October 1, 1982)

*In this paper, added loss during temperature cycling in a given ribboned fiber is shown to be caused by thermally induced axial compressive strain imparted to the fiber. A microbending-sensitivity parameter $\delta$ is introduced which reduces all loss-strain curves corresponding to different fibers to one characteristic master curve. Thermoviscoelasticity theory is used to calculate the time- and temperature-dependent compressive strain imparted to a ribboned fiber during a standard environmental cycle. Combining these analytical results with environmental data, the functional relationship between fiber-compressive strain and the added loss for a fiber of any given $\delta$ in an Adhesive-Sandwich Ribbon (ASR) with Urethane-Acrylate (UA) coated fibers has been determined. Using this analysis, the added loss for a UA ASR can now be predicted for any environmental cycle. The critical material properties that dominate the environmental performance of ASRs are the tape shrinkback at elevated temperatures and the product $\alpha EA$ of the coefficient $\alpha$ of thermal expansion, the time- and temperature-dependent relaxation modulus $E$, and the area $A$ of the coating.*

## I. INTRODUCTION

Unless special precautions are taken, fiber-optic cables installed in the outside plant could experience temperatures ranging from −45°F

* Plastics Engineering Consultants, Inc., Lawrenceville, GA.

to +190°F. Thus, optical transmission loss resulting from this thermal history directly impacts the coating and ribbon choices for a particular system design. An understanding of the relationships among the thermally induced strains on the "ribboned" glass fibers, the resulting added transmission loss, and fiber parameters is crucial to properly evaluating candidate fiber-coating materials, ribbon structures, and/or ribbon-matrix materials. Moreover, it is desirable to be able to predict long-term behavior from short-term testing, thereby simplifying the environmental testing procedure.

The added loss in environmental testing is generally thought to be associated with the microbending of the axis of the fiber. Moreover, it has been shown by other investigators that when intimate contact is forced between a fiber and a microscopically rough surface, losses due to microbending can be substantial and are a function of the fiber geometric and optical parameters as well as the elastic modulus of the rough surface.[1,2] Therefore, thermally induced strains on the ribbon structure can indirectly result in added loss by increasing the contact forces between fibers. In addition, the sensitivity of a particular fiber to microbending loss is said to be associated with irregularities at the core-cladding interface, which may be due to core-diameter variations and/or refractive index variations.[3]

In this paper, a thermoviscoelastic analysis is used to compute the axial and transverse strains that are imparted to a fiber-optic ribbon when it is subjected to any given thermal cycle. This analysis shows that for the Bell System Adhesive-Sandwich-Ribbon[4] (ASR) construction with Urethane-Acrylate (UA) coated fibers, the transverse strain due to lowering the temperature results, contrary to one's expectation, in a reduction in the contact force between fibers. On the other hand, the resulting axial compressive strain evidently either increases this contact force or induces fiber buckling at the critical spatial wavelength, and thus creates added optical loss. Moreover, for a given ribboned fiber we establish herein the existence of a relationship between the environmental added loss and any measure of microbending sensitivity of the fiber. This equivalence enables us to predict the environmental performance of any ribboned fiber over a wide temperature-time span from data gathered over a relatively short time. In particular, the loss data collected from one excursion to −45°F can be combined with thermoviscoelastic data and analysis to predict the loss of any ribbon configuration with any choice of constituent materials, provided the tested fibers have a suitable range of microbending sensitivity.

## II. ENVIRONMENTAL TEST PROCEDURE

The intent of environmental testing is to determine the effect of thermal exposure on the optical performance of fiber-optic ribbons. All

Table I—Environmental test cycle

| Cycle Number | Exposure Temperature, $T$ (°F) | Exposure Time, $t$ (days) |
|---|---|---|
| I | 75 | 2 |
| | −45 | 2 |
| | −15 | 2 |
| | 15 | 2 |
| | 75 | 2 |
| | 170 | 30 |
| II | 75 | 2 |
| | −45 | 2 |
| | −15 | 2 |
| | 15 | 2 |
| | 75 | 2 |
| | 190 | 34 |
| III | 75 | 2 |
| | −45 | 2 |
| | −15 | 2 |
| | 15 | 2 |
| | 75 | 2 |

the ribbons tested were placed in a 23-inch-diameter cardboard cylindrical container in a stem-pack fashion. The containers were then placed in a "walk-in" environmental chamber with both ends available for measuring purposes. Input and output array connectors[5] were fabricated to measure loss by the reference-fiber technique. Namely, the 0.63-$\mu$m loss of a fiber is obtained by taking the ratio of its output power to the average output power of ten reference fibers that are short enough to have negligible loss.

Table I is a summary of the environmental testing cycle used for this investigation. The loss of the ribboned fibers* was obtained at the end of each exposure. The change in loss for a typical ribboned fiber is plotted in Fig. 1 for each temperature in Cycles I, II, and III. The loss of the ribboned fibers increases with decreasing temperatures, the maximum loss occurring at the lowest temperature. As is evident in Fig. 1, the loss also increases with increasing cycle number. We subsequently show that this "pumping effect" is due to polyester-tape shrinkback† and stress relaxation effects during the high-temperature exposure between cycles.

It follows that for a 40-year design life, the worst-case environmental cycle would be a 40-year exposure at 190°F followed by a low-temperature exposure. This worst-case cycle is summarized in Table II. Note

* Fibers used in this study had cladding and core diameters of 110 and 55 $\mu$m, respectively.

† Shrinkback is the recovery of process-induced strains.

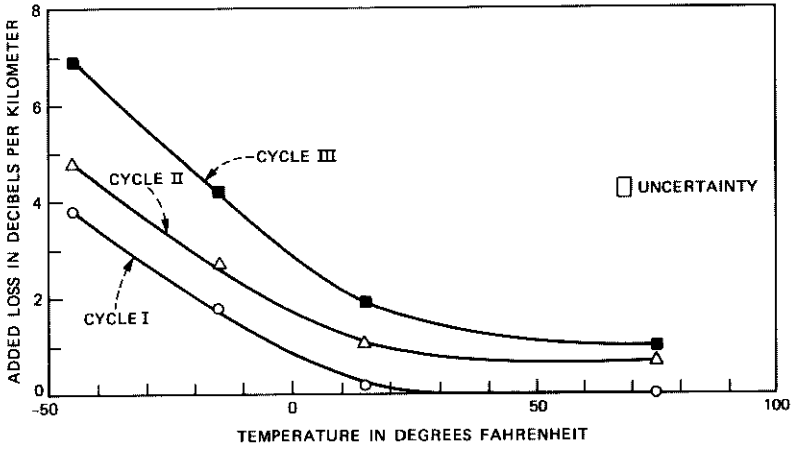



Fig. 1—Environmental added loss at 0.63 μm vs. temperature (UA-coated fiber, ASR construction).

Table II—Worst-case environmental cycle

| Exposure temperature, $T$ (°F) | Exposure Time, $t$ (days) |
|---|---|
| 190 | 14,610 (40 years) |
| 75 | 2 |
| −45 | 2 |
| −15 | 2 |
| 15 | 2 |
| 75 | 2 |

that the cycle of Table II is not suggested for testing purposes but will only be used to predict the most pessimistic estimates of performance.

## III. DUALITY BETWEEN MICROBENDING SENSITIVITY AND STRAIN

A convenient means for measuring the microbending sensitivity of different fibers is to determine the wavelength-independent loss coefficient[6] of the fiber when wound in several layers on a 6-inch-diameter reel under a tension fixed for all fibers. Let $\delta$ denote the wavelength-independent loss coefficient measured under these conditions. This method was contrived to produce artificially high losses, thereby magnifying the loss contribution due to microbending. Of course, the losses for these same fibers will be substantially smaller when measured in a stress-free configuration.

The added-loss response to a given cycle (see Fig. 1) is characteristic of all fibers of a single given $\delta$ (as defined above) in a given structure

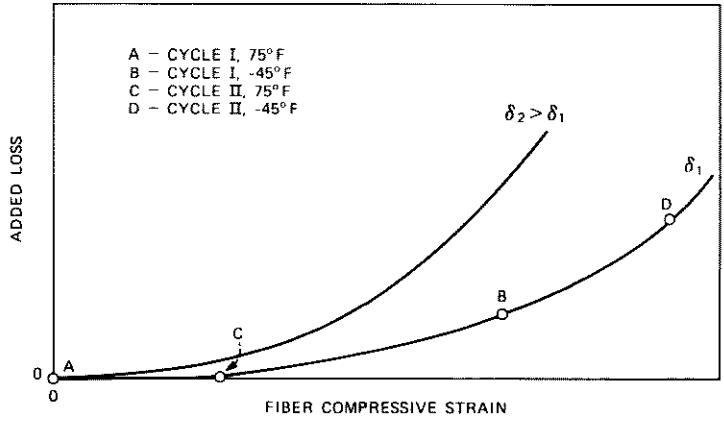



Fig. 2—Environmental added loss vs. fiber compressive strain.

but will change for fibers having different $\delta$'s. Moreover, examination of all the environmental data collected in this study indicates that the change in loss at a given point in any cycle increases with increasing $\delta$.

As is well known, microbending loss occurs when the axis of a fiber is bent into a curve whose spectrum contains a certain critical frequency.[3] Transverse pressure against a fiber on a microscopically rough surface has been shown to introduce loss in this way.[1,2] However, in the case of environmentally induced loss in an ASR ribbon, it is shown in the Appendix that the *transverse* strain due to thermal contraction *reduces* the contact forces between fibers. Thus, the added-loss response of Fig. 1 must be due to another mechanism. The only reasonable possibility is that the compressive axial strain induces microbending loss through fiber buckling. This loss would occur if the fiber buckling were at the critical wavelength or if buckling increases the contact forces between the fiber and its surroundings. It is not our intent to identify which of these means dominates the behavior of the fiber but only to characterize its manifestations. In other words, we simply correlate the added loss with the compressive strain imparted to the fiber.

Both the added loss and compressive strain increase with decreasing temperature; their relationship* is shown schematically in Fig. 2. Here, the losses at 75°F and −45°F in Cycle I are denoted by $A$ and $B$, respectively. After high-temperature exposure, the shrinkback of the polyester tape increases the strain on the fiber, so that $C$ and $D$

* If strain and added loss monotonically increase with decreasing temperature, then it is easy to show that the added loss monotonically increases with increasing strain.

represent 75°F and −45°F in the second cycle. This explains the increased loss with each cycle that is evidenced in Fig. 1. This phenomenon has been observed consistently in all environmental testing and, as previously noted, is called the pumping effect.

Since the loss increases with increasing $\delta$, the added-loss response to strain becomes steeper with increasing $\delta$, as illustrated in Fig. 2. The effect of $\delta$ in this graph is obviously a contraction of the strain scale with increasing $\delta$. As to whether this contraction is uniform or not is readily tested by plotting the loss versus logarithmic strain, where a uniform contraction would then appear as a horizontal, rigid translation of the added-loss curves with $\delta$. As will be demonstrated in Section V, the loss-strain data may indeed be reduced in this fashion. Since the strain-scale contraction factor is a function of $\delta$ alone (the contraction is uniform), it creates a one-to-one correspondence (duality) between $\delta$ and compressive strain, changes in $\delta$ being equivalent to changes in strain. With this in mind, we proceed to a calculation of the axial strain in the environmental cycle.

## IV. THERMOMECHANICAL ANALYSIS

This section is devoted to the summary of the formulas needed to calculate the axial strain history to which a fiber-optic ribbon is subjected during any environmental cycle. All of the plastics used in an ASR structure possess time- and temperature-dependent moduli, two examples of which are shown in Fig. 3. If the plastic is instantaneously strained, the stress required to sustain that strain relaxes with time according to the given curve. Notice that the relaxation that occurs in the UA modulus at 140°F in one hour takes more than 40 years at room temperature. The relaxation at −45°F is slower yet, so that the UA coating is as stiff as the polyester tape over significant time periods at these low temperatures.

On the other hand, the coefficients of thermal expansion of these plastics are constant with time provided the temperature changes do not encompass their glass transition temperatures, $T_g$. Even so, by virtue of the time dependence of the plastic moduli, the ribbon itself exhibits time- and temperature-dependent thermal expansion and contraction during the thermal cycle. This situation is depicted in Fig. 4. Since the expansion coefficients $\alpha_P$ of the plastics are much greater than that ($\alpha_G$) for glass, the fibers restrain the contraction of the plastic, the net contraction $\alpha \Delta T$ of the ribbon being determined by equilibrium (force balance) in the structure. The relatively high short-time modulus of the plastic (see Fig. 3) causes the initial contraction (expansion) to be high. As the modulus of the plastic relaxes, the energy stored in the glass causes the ribbon to recover some of this high initial strain. Of course, the rate of this recovery depends on the temperature at which it occurs.

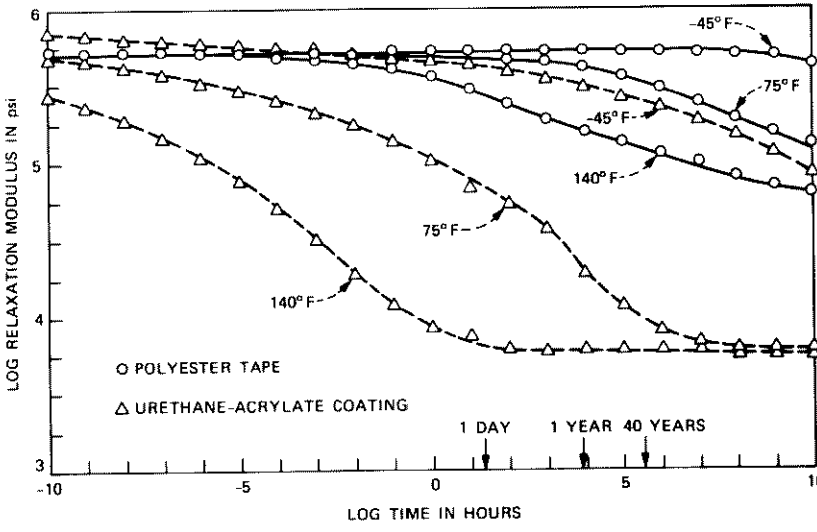



Fig. 3—Time dependence of moduli for ribbon constituents at various temperatures (generated by time-temperature superposition from data collected by R. P. DeFabritis).

The glass also resists shrinkback $\epsilon_S^{TA}$ of the polyester tape during exposure to elevated temperatures. If it were unrestrained, the tape would shrink back according to the results in Fig. 5. As illustrated in Fig. 6, the shrinkback $\epsilon_S$ that the ribbon experiences is much less than that of the free tape.

The compressive strain induced in the ribbon by its thermal contraction upon an excursion to −45°F is thus increased in each subsequent cycle by the shrinkback during the high-temperature exposure. We now proceed to a calculation of these strains.

### 4.1 Axial thermal expansion

If the plastic phases were considered to be elastic, the axial tensile modulus of a fiber-optic ribbon would be approximated well by the *rule of mixtures*

$$E_e = \frac{\sum_i E_i A_i}{\sum_i A_i}, \tag{1}$$

where $E_i$ is the modulus of the $i$th constituent, $A_i$ its area, and the summation is taken over all phases. In reality, (1) is a lower bound for the effective elastic modulus, being exact if all the constituents have the same Poisson's ratio.[7] Furthermore, eq. (1) can be derived in an elementary fashion by considering the equilibrium of the ribbon if one ignores Poisson's effect and assumes that all constituents are equally strained.

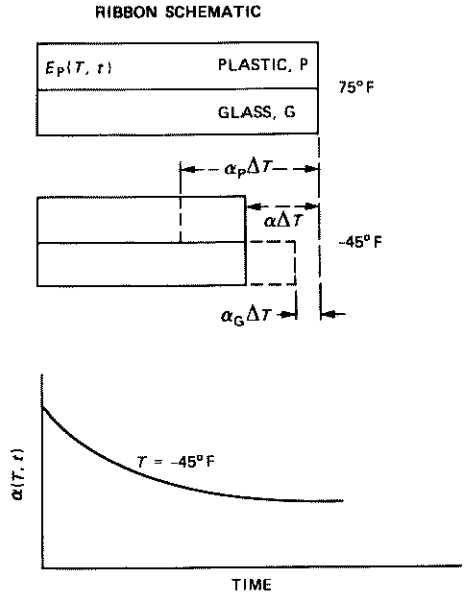



Fig. 4—Mechanics of ribbon thermal contraction.

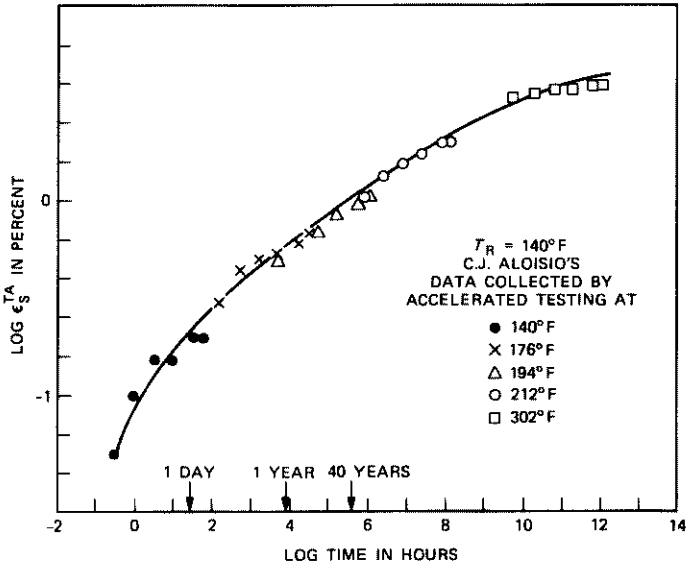



Fig. 5—Shrinkback strain vs. time for 3M No. 5 polyester tape.

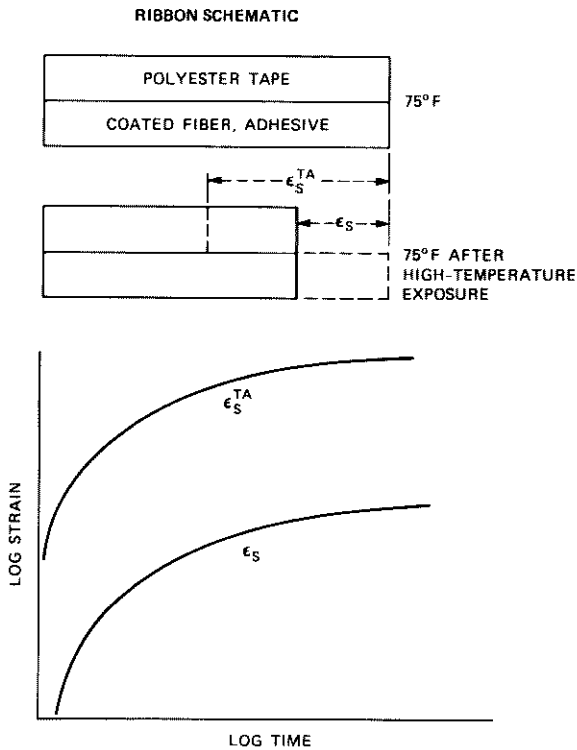



Fig. 6—Mechanics of ribbon shrinkback.

This same kind of elementary argument leads to the formula

$$\alpha_e = \frac{\sum_i \alpha_i E_i A_i}{\sum_i E_i A_i} \tag{2}$$

for the coefficient of thermal expansion of the ribbon, given elastic phases with moduli $E_i$, areas $A_i$ and coefficients of thermal expansion $\alpha_i$. Schapery has shown that (2) is exact if each phase has the same Poisson's ratio and is a very good approximation in any case.[7] These elastic results may be used to generate their viscoelastic counterparts. To see this, assume that all constituents obey constitutive equations of the *form*[8]

$$\sigma(t) = \int_0^t E(T(t), t - \tau)\dot{\epsilon}_o(\tau)d\tau$$

$$E(T, t) = a_G(T)E(T_R, \xi), \qquad \xi = \int_0^t \frac{d\tau}{a_T(T(\tau))} \tag{3}$$

and

$$\epsilon_\sigma(t) = \epsilon(t) - \alpha[T(t) - T_0], \qquad T_0 = T(0), \tag{4}$$

where $\epsilon$, $\sigma$, and $E(T, \cdot)$ are the strain, the stress and the relaxation modulus at temperature $T$, respectively. Here, a superposed dot on a function denotes differentiation with respect to its argument. On a log-log plot of modulus versus time, $\log a_G(T)$ and $\log a_T(T)$ represent, respectively, the vertical and horizontal distances that the curve for a temperature $T$ must be translated to lay over that for the reference temperature $T_R$. All the materials considered here, including glass as a trivial case, conform to this hypothesis.

When (3) and (4) are met, it is readily shown that equation (1) continues to hold in the viscoelastic case, so that

$$E(T, t) = \frac{\sum_i E_i(T, t)A_i}{\sum_i A_i} \tag{5}$$

is exact (Poisson's effect being ignored). Moreover, as long as the curve of the logarithm of the modulus versus logarithmic time has small curvature, a good approximation to $\alpha(T, t)$ is provided by the so-called quasi-elastic approximation[8]

$$\alpha(T, t) = \frac{\sum_i \alpha_i E_i(T, t)A_i}{\sum_i E_i(T, t)A_i}. \tag{6}$$

The strain in the ribbon due to a temperature history $T$ in the absence of stress is characterized by

$$\epsilon(t) = \int_0^t \alpha(T(t), t - \tau)\dot{T}(\tau)d\tau. \tag{7}$$

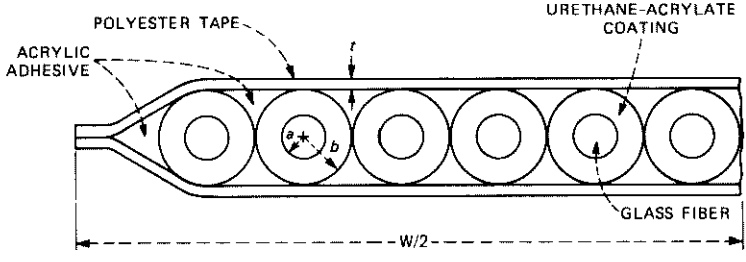
Notice that the ribbon has a time-dependent expansion coefficient even though we have supposed that the constituent plastics do not. This latter assumption is valid as long as the temperature excursion does not encompass the glass transition temperature $T_g$ of any constituent. Equation (6) can easily be shown to continue to hold even if the constituents have time-dependent expansion coefficients. One need only replace each $\alpha_i$ by $\alpha_i(T, t)$. For the purposes of this investigation, it is sufficient to assume each $\alpha_i$ to be constant.

The data needed to calculate the effective modulus $E$ and coefficient $\alpha$ of thermal expansion of the ribbon at −45°F and 75°F for a time of 10 hours is given in Table III. The geometric data of Table III was calculated using the nominal dimensions shown on the cross-sectional view of a typical ribbon in Fig. 7. Notice that although the glass makes far and away the most significant contribution to the stiffness of the

Table III—Geometric and material properties of ribbon constituents

| | $A$ ($10^{-4}$ in.$^2$) | $\alpha$ ($10^{-5}$ °F$^{-1}$) | Ten-Hour Modulus $E$($10^5$ psi) | | $EA$($10^2$ lb) | | $\alpha EA$ ($10^{-3}$lb/°F) | |
|---|---|---|---|---|---|---|---|---|
| | | | −45°F | 75°F | −45°F | 75°F | −45°F | 75°F |
| 3M No. 5 Polyester Tapes | 3.00 | 1.06 | 5.27 | 4.67 | 1.58 | 1.40 | 1.67 | 1.48 |
| 3M No. 5 Acrylic Adhesive | 3.60 | 7.72 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Urethane-Acrylate Coatings | 5.87 | 3.33 | 4.22 | 0.74 | 2.48 | 0.43 | 8.26 | 1.43 |
| Glass Fibers | 1.77 | 0.028 | 107 | 107 | 18.94 | 18.94 | 0.53 | 0.53 |
| Total | 14.24 | | | | 23.00 | 20.77 | 10.46 | 3.44 |

Thus, eq. (6) $\rightarrow \alpha = \frac{\sum \alpha EA}{\sum EA} = \frac{4.55 \times 10^{-6}\ °F^{-1} @ -45°F}{1.66 \times 10^{-6}\ °F^{-1} @ 75°F}$




| DIMENSIONS |
|---|
| $a$ = 2.17 mils |
| $b$ = 9.00 mils |
| $t$ = 1.00 mil |
| w = 150 mils |

Fig. 7—Cross section of a twelve-fiber ribbon.

ribbon, other constituents play a major role in the calculation of $\alpha$. Indeed, at −45°F the $\alpha EA$ of the glass may well be neglected in comparison with that of the coating. At 75°F, on the other hand, the tape, coating and glass make comparable contributions to the effective thermal expansion coefficient.

In Fig. 8, $\alpha$ and $E$ are plotted logarithmically against the logarithm of time at 75°F. It is evident that the change in $\alpha$ with respect to temperature and time is much more significant than the change in ribbon modulus. The coefficients of thermal expansion listed for each of the constituents in Table III are for temperatures below their respective glass transition temperatures, since our interests are in

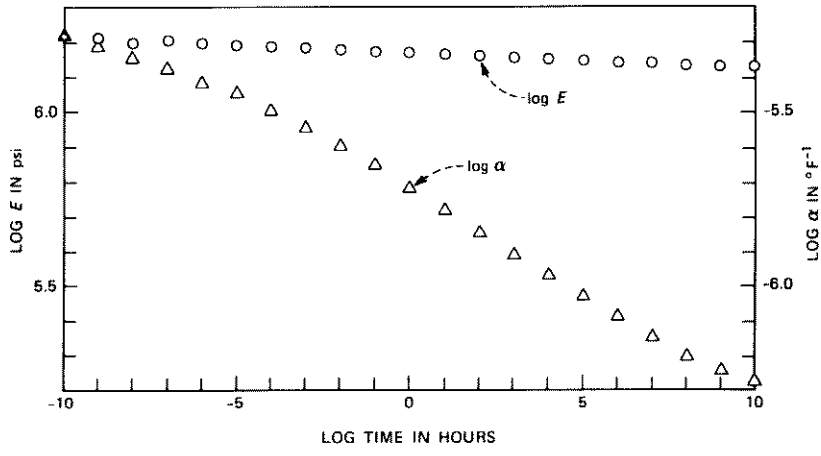



Fig. 8—Time dependence of the modulus and the coefficient of thermal expansion for the ribbon at 75°F (UA-coated fibers, ASR construction).

calculating the compressive strains imparted to the fiber during the low-temperature excursion.

If the ribbon is exposed to the temperature history

$$T(t) = T_0 + \sum_{k=1}^{n} (T_k - T_{k-1})H(t - t_k),\ H(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \tag{8}$$

$0 < t_1 < t_2 < \cdots < t_n < t$, the resulting strain history is

$$\epsilon(t) = \sum_{k=1}^{n} (T_k - T_{k-1})\alpha(T_n, t - t_k), \tag{9}$$

as found from substituting (8) into (7). Observe that eq. (9) contains all relaxation effects associated with the temperature history in (8).

### 4.2 Axial shrinkback

We now set out to calculate the shrinkback induced in the ribbon by the shrinkage of the polyester tape during high-temperature exposure. Denote by $\epsilon_S^{TA}(T_R, t)$ the shrinkback (see Fig. 5) the tape would experience in the time period $[0, t]$ at some reference temperature* $T_R$ if it were unrestrained. At any other temperature $T$, the unrestrained shrinkback in the same time period would be

$$\epsilon_S^{TA}(T, t) = \epsilon_S^{TA}(T_R, t/a_T(T)), \tag{10}$$

where $a_T$ is the temperature-dependent scale-contraction factor. If the tape is subjected to temperature $T_1$ for a time $t_1$ and then to $T_2$ for $t_2$, the total shrinkback would be

* $T_R = 60°$ in Fig. 5.

$$\epsilon_S^{TA}(T_R, t_1/a_T(T_1) + t_2/a_T(T_2)).$$

For the polyester tape used in the ASR construction, shrinkback is observable in the environmental cycle only during the high-temperature exposures, the time scales being much too long at the lower temperatures.

An analysis similar to that outlined in Section 4.1 for the time-dependent modulus and coefficient of thermal expansion of the ribbon can be used to obtain the approximate formula for the shrinkback strain of the ribbon $\epsilon_S$.

$$\epsilon_S(t) \cong S(T, t)\epsilon_S^{TA}(t), \tag{11}$$

where

$$S(T, t) = \frac{E_{TA}(T, t)A_{TA}}{\sum_i E_i(T, t)A_i}. \tag{12}$$

A double logarithmic plot of the ribbon shrinkback function $S$ as calculated from (12) is shown in Fig. 9 for $T = 140°F$. Observe that the shape of $\log S(T, \cdot)$ is very much like that of the logarithm of the polyester modulus in Fig. 3. The ribbon shrinkback during any high-temperature exposure can be calculated by substituting the appropriate mechanical and shrinkback data (e.g., Figs. 3 and 5) into eqs. (11) and (12). The error made in using eq. (11) at 140°F has been shown to decrease monotonically from 10 percent at two hours to 8.8 percent at ten hours. The approximation of eq. (11) increasingly improves as time goes on and as temperature increases. Since our interest is in times greater than forty-eight hours at temperatures over 170°F, (11) is quite acceptable.

### *4.3 Calculation of fiber compressive strains in the environmental test cycle*

As discussed in Section III, a compressive strain applied to a riboned fiber induces added optical loss. This strain $\epsilon_F$ for the temperature history $T$ [see eq. (8)] is given by

$$\epsilon_F(t) = \epsilon_R(t) - \alpha_F[T(t) - T_0], \tag{13}$$

where $\alpha_F$ is the linear coefficient of thermal expansion of the glass fiber and

$$\epsilon_R(t) = \epsilon(t) + \epsilon_S(t) \tag{14}$$

is the ribbon strain. In (14), $\epsilon$ is the strain due to the thermal contraction of the ribbon [eq. (9)], and $\epsilon_S$ is the ribbon shrinkback strain [eq. (11)].

Thermoviscoelastic data on each of the constituent materials were incorporated into computer-programmed versions of eqs. (6), (9), (11),

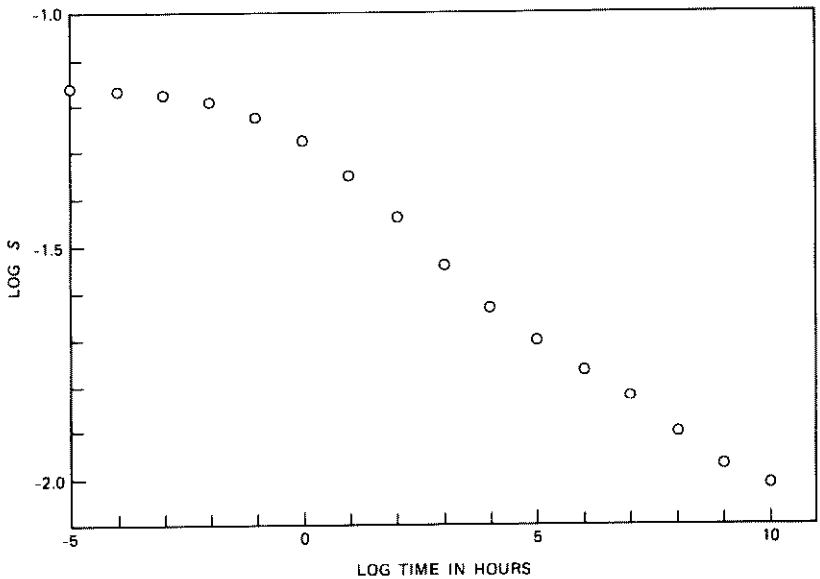



Fig. 9—Time dependence of ribbon-shrinkback function at 140°F (UA-coated fibers, ASR construction).

(12), and (13) to obtain the fiber strains on each excursion to −45°F. The results are plotted in Fig. 10.

## V. THE MASTER CURVE FOR ADDED LOSS VERSUS COMPRESSIVE STRAIN

The purpose of this section is to construct the curve of added loss versus compressive strain (master curve) for a fiber having an arbitrary value of the microbending sensitivity parameter $\delta$. This will be done by demonstrating that the effect of $\delta$ on the loss-strain curve is to uniformly contract the strain scale, which appears as a rigid shift (translation) of the curves with $\delta$ when the loss data are plotted against logarithmic strain. This master curve together with the $\delta$-shift curve can then be used to predict the performance of any fiber ($\delta$ known) in any environmental cycle.

In Fig. 11, the environmental added-loss data for five fibers are plotted versus the logarithm of the axial compressive strain on the glass fiber as calculated in Section IV (Fig. 10). These data include all the measurements for Cycles I through III according to Table I. Neither of the points for 75°F in Cycle I appears in these figures because for the first the strain is zero and for the second it is tensile (negative), so that both logarithms are undefined.

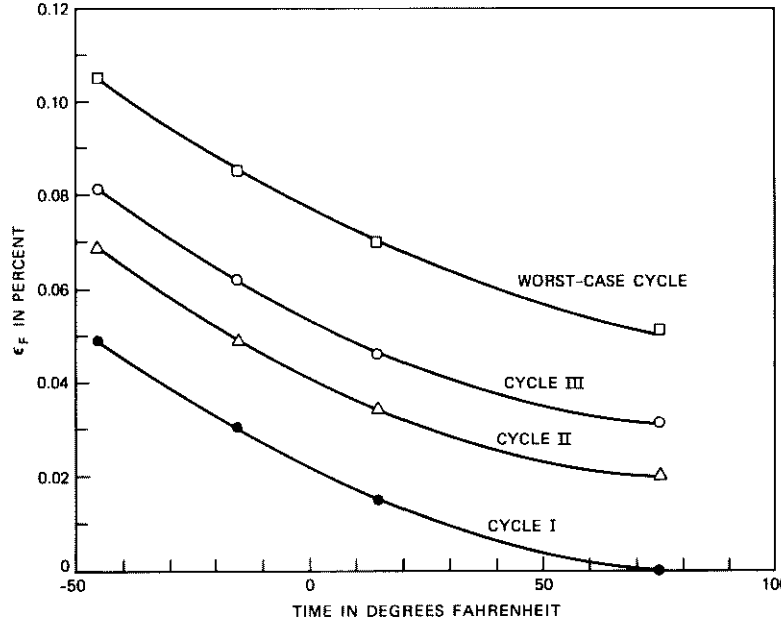



Fig. 10—Calculated fiber compressive strain vs. temperature in the environmental cycle (UA-coated fiber, ASR construction).

### *5.1 Shifting with δ to form the master curve*

The plots in Fig. 11 were laid over one another and shifted horizontally by hand until they formed a single, smooth curve. The amount of shift log $a_\epsilon(\delta)$ required for each of the five values of $\delta$ is shown in Fig. 12 referenced to $\delta = 1.0$ dB/km. The linear, least-squares fit shown in Fig. 12 results in an excellent approximation of the shift data. For simplicity this linear approximation is used in the subsequent construction of the master curve.

The resulting master curve is shown in Fig. 13. The data points for the five different fibers are shown with different symbols on this plot. This master curve together with the $\delta$-shift curve of Fig. 12 can be used to obtain the added-loss-versus-strain profile for a fiber of any given $\delta$. A fourth-order polynomial has been fitted to the master curve data and is included in Fig. 13. Observe that the scatter about the best-fit polynomial is within the loss measurement uncertainty. Notice also that there is significant overlap in the data from fiber to fiber. On account of the functional relationship between $a_\epsilon$ and $\delta$ (illustrated in Fig. 12), we may view the master curve of Fig. 13 as added loss versus strain for a fixed value of $\delta$ or added loss versus $\delta$ for a fixed level of strain. This makes precise the $\delta$-strain duality alluded to in Section III.

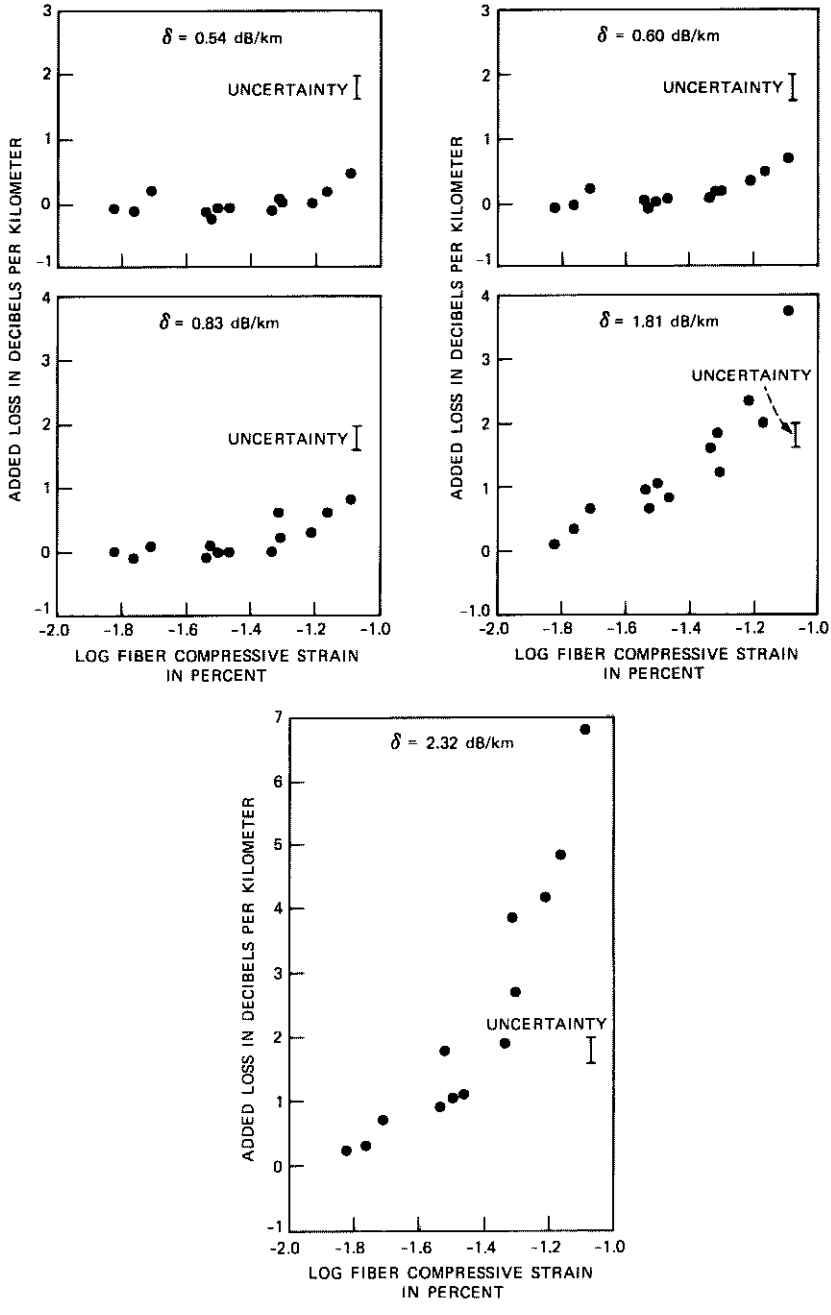



Fig. 11—Environmental added loss at 0.63 μm vs. calculated fiber strain in the environmental test (UA-coated fibers, ASR construction).

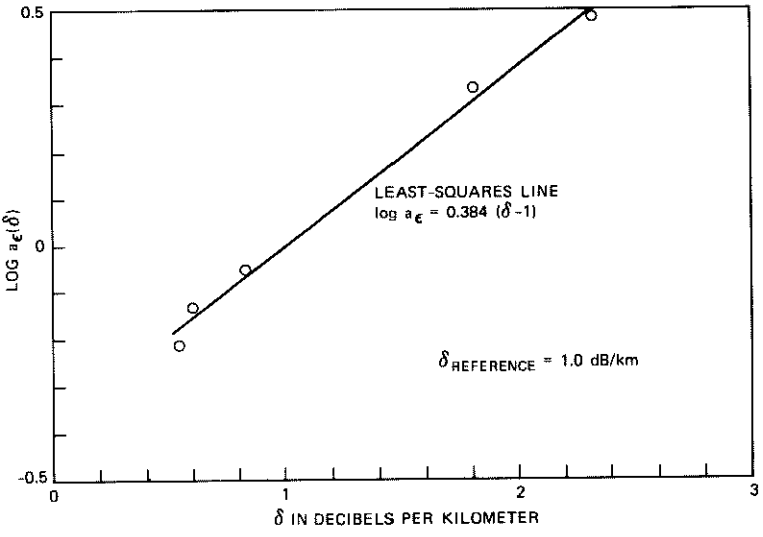



Fig. 12—Strain-scale-contraction factor vs. microbending sensitivity parameter $\delta$ (referenced to $\delta$ = 1.0 dB/km, UA-coated fibers, ASR construction).

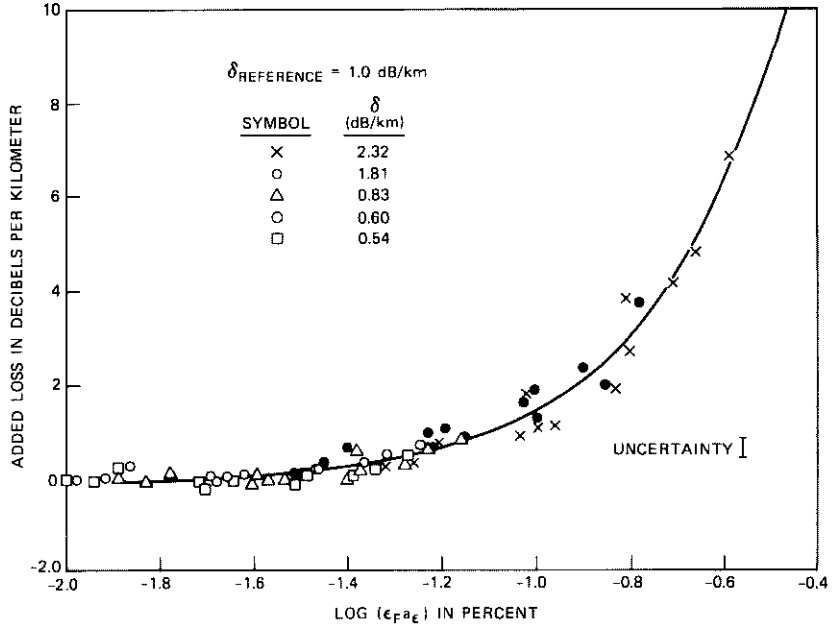



Fig. 13—Environmental added loss vs. $\delta$-reduced fiber strain (referenced to $\delta$ = 1.0 dB/km, UA-coated fibers, ASR construction).

Although the master curve of Fig. 13 has been constructed from data collected in all three cycles, this was not necessary. Indeed, a polynomial fit through Cycle I is virtually indistinguishable from the one shown in Fig. 13. Thus, we are led to conclude that only one low-temperature excursion is necessary to complete an analysis of the type presented here, provided fibers having a suitable range of $\delta$'s are tested. This observation greatly simplifies the environmental-testing procedure for fiber-optic ribbons.

### *5.2 Prediction of loss in the worst-case cycle*

We are now in a position to predict the behavior of a UA ASR in the worst-case environmental cycle. To do this we need only add to the strain associated with each temperature and time (see Fig. 10) the shift factor associated with the appropriate value of $\delta$ (see Fig. 12) for the fiber in question. The loss can then be read off the master curve (Fig. 13). For example, the strain $\epsilon_F$ in the worst-case cycle at −45°F obtained from Fig. 10 is 0.105%. The shift corresponding to a $\delta$ of 2.32 dB/km from the equation in Fig. 12 is $\log a_\epsilon(2.32) = 0.51$. Thus, the log of the $\delta$-reduced strain [$\log \epsilon_F a_\epsilon(2.32)$], −0.47, yields an added loss of 9.8 dB/km from Fig. 13. This procedure was employed to generate the curves in Fig. 14 for the ribbon behavior in the worst-case cycle for different values of $\delta$.

Notice for a $\delta$ of 2.32 dB/km in Fig. 14 that the loss in the worst-case cycle is much worse than that of Cycle III shown in Fig. 1. This

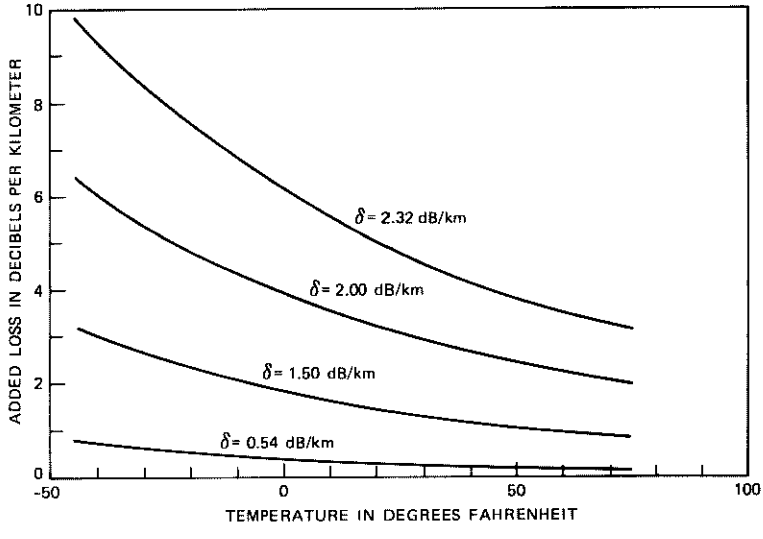



Fig. 14—Environmental added loss at 0.63 μm vs. temperature in the worst-case cycle (UA-coated fibers, ASR construction).

of course is the shrinkback effect previously discussed. Also, observe from comparing Figs. 11 and 14 that a fiber with $\delta = 0.54$ dB/km would meet a 0.5 dB/km added-loss criterion after Cycle III but not after the worst-case cycle. It is therefore clear that data sufficient to allow long-term predictions are essential to any meaningful ribbon evaluation. Finally, we remark that the added loss for a UA ASR can be predicted in the way demonstrated above for *any* environmental cycle provided one calculates the appropriate strain history by means of the formulas in Section IV.

## VI. APPLICATIONS

### 6.1 A Criterion for coating/ribbon-structure comparisons

If one were to choose a ribbon-performance requirement of 1.0 dB/km in the worst-case cycle, then it is clear from Fig. 14 that an ASR made with a population of UA-coated fibers having high $\delta$'s would be unacceptable. The same structure manufactured with low-$\delta$ fibers would satisfy the requirement. Thus, one can easily err out of ignorance of the strong effect arising from the sensitivity of the fiber to microbending (as reflected in $\delta$). Obviously, for a given performance requirement there is a maximum permissible $\delta$-value, $\beta$. That is, if ribbons are manufactured from a fiber selection having $\delta$'s less than $\beta$, the given performance requirement will be met. Clearly then, it is not a question of whether the UA-ASR structure is acceptable or not, but rather how restrictive the value of $\beta$ is. The parameter $\beta$ is therefore an effective criterion for comparing various coating and ribbon-structure combinations.

To calculate this maximum permissible $\delta$-value, $\beta$, for the UA ASR given a 1.0 dB/km added loss in the worst-case cycle, we enter into the master curve (Fig. 13) at a loss of 1.0 dB/km and read off the associated reduced strain

$$\log \epsilon_F a_\epsilon(\beta) = -1.09. \tag{15}$$

Next, introduce the equation for the $\delta$-shift curve (Fig. 12),

$$\log a_\epsilon(\beta) = 0.384(\beta - 1)$$

into (15) to arrive at

$$\beta = 1 - \frac{1.09 + \log \epsilon_F}{0.384}. \tag{16}$$

Now the $-45°$F strain in the worst-case cycle is 0.105 percent, so that (16) yields

$$\beta = 0.7 \text{ dB/km}.$$

Similarly, for a performance requirement of 0.5 dB/km, one finds

$$\beta = 0.2 \text{ dB/km}.$$

Thus, UA-coated fibers selected with $\delta$'s less than $\beta$ could be used in an ASR and satisfy the chosen performance requirement for a 40-year design life.

### 6.2 Critical material and geometric properties

The maximum permissible $\delta$, $\beta$, can also be used to evaluate how material and geometric properties might be altered to improve environmental performance. It is obvious that a reduction in polyester-tape shrinkback would reduce environmental added loss. This is illustrated dramatically in Fig. 15, where the parameter $\beta$ is plotted versus shrinkback normalized with respect to that of the present tape in the worst-case cycle (see Fig. 5). The two curves are for performance standards of 0.5 and 1.0 dB/km in the worst-case cycle. Notice that a 50-percent reduction increases $\beta$ from 0.2 to 0.5 dB/km for the 0.5 dB/km performance requirement, while elimination of the shrinkback

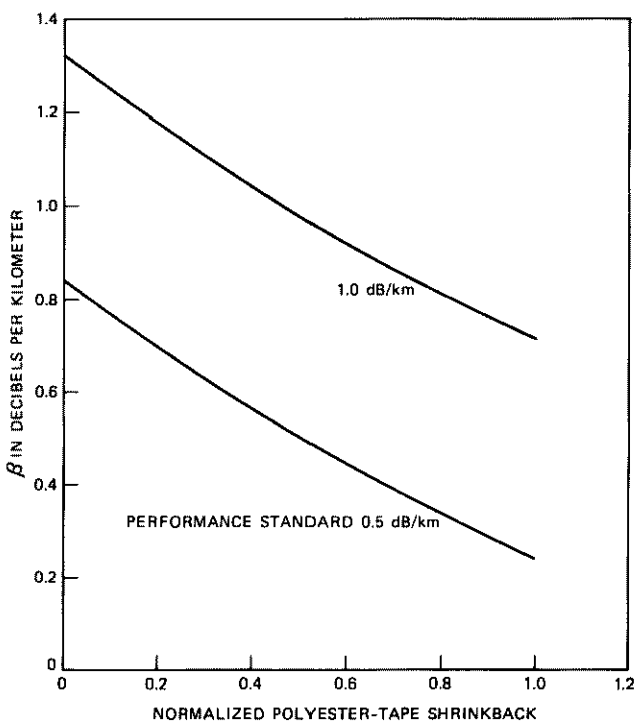



Fig. 15—Effect of shrinkback on the maximum permissible $\delta$-value for the worst-case cycle (UA-coated fibers, ASR construction).

altogether increases $\beta$ to 0.8 dB/km. Observe from eqs. (11) and (12) that the ribbon shrinkback is linear with respect to the area of the tape. Thus, Fig. 15 can also be viewed as a plot of $\beta$ versus reduced area normalized to that of the present tape (see Table III).

A consideration of Table III reveals that reducing the $\alpha EA$ product of the coating will likewise diminish the strain on the fibers, since the coefficient of thermal expansion of the ribbon would then be reduced. The coefficients $\alpha$ of thermal expansion are essentially the same for all plastics though the moduli $E$ can vary substantially from one material to another. Notice further that the area $A$ of the coating is greater than that of any other ribbon constituent. As to whether this reduction in fiber strain is accompanied by an environmental performance improvement depends on the extent to which the added-loss-versus-compressive-strain master curve and its associated $\delta$-shift characteristic are influenced by the changes adopted to reduce the fiber strain. Changing $\alpha$ of the coating has no effect on these curves while changing coating modulus $E$ and/or the ribbon geometry may have a substantial effect. The variation in the shape of these curves with changes in these various parameters can be ascertained only by completing a characterization of the type carried out here for the UA ASR.

Figure 16 shows how strong an effect reducing thermal contraction can be when the change in the added-loss-reduced-strain characteristic is negligible. Here we have plotted $\beta$ against the coated-fiber outer diameter holding the fiber diameter constant. Observe that if the outer diameter of the coated fiber is reduced from 9 to 6 mils the effect on $\beta$ is bigger than the effect of a 50-percent reduction in tape shrinkback. Letting the coating thickness tend to zero results in an increase in $\beta$ from 0.2 to 1.0 dB/km for the 0.5 dB/km performance requirement. If a 50-percent reduction in tape shrinkback is combined with a reduction in coated-fiber outer diameter to 6.5 mils, one can show that $\beta$ increases from 0.2 to 0.9 dB/km. Finally, we remark that these same dramatic effects should occur with reductions in the coating modulus.

Other investigators have established that dual-coated fibers having a soft primary coating are less sensitive to microbending than fibers having a hard, single coating.[1] Such dual-coated fibers in an ASR should also behave well environmentally. To see this, consider a secondary coating of UA over silicone where the area of the UA skin is $1.56 \times 10^{-5}$ in.$^2$ Neglecting the stiffness of the silicone (it is three orders of magnitude smaller than that of UA), we obtained an effective single-coated UA diameter of 6.4 mils. The corresponding increase in $\beta$ (Fig. 16) is from 0.2 to 0.6 dB/km for the 0.5 dB/km performance criterion. As previously mentioned these conclusions are predicated upon the assumption that the basic loss-strain characteristic of the ASR structure is relatively insensitive to changes in coating modulus.

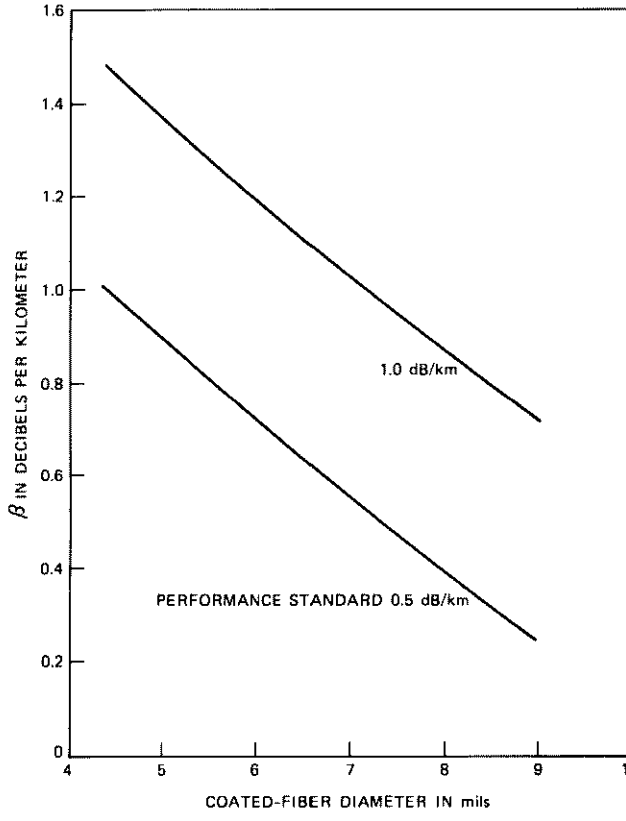



Fig. 16—Effect of coating thickness on the maximum permissible $\delta$-value for the worst-case cycle (UA-coated fibers, ASR construction, fiber diameter = 4.33 mils).

## VII. CONCLUSIONS AND RECOMMENDATIONS

We have shown that:

(*i*) Environmental added loss in an ASR is associated with axial compressive strain (not transverse strain) imparted to the fibers by thermal contraction of the ribbon in the low-temperature excursion and ribbon shrinkback together with relaxation during the high-temperature exposure.

(*ii*) There is a duality between a measure $\delta$ of the microbending sensitivity of a fiber and the fiber-compressive strain; viz., the effect of $\delta$ on the characteristic added-loss response to compressive strain is a uniform contraction of the strain scale.

(*iii*) Points (*i*) and (*ii*) above result in a master curve of environmental added loss versus $\delta$-reduced fiber strain together with a plot of the strain-scale contraction factor versus $\delta$-value. We are thus led to the algorithm depicted schematically in Fig. 17. Therefore, when

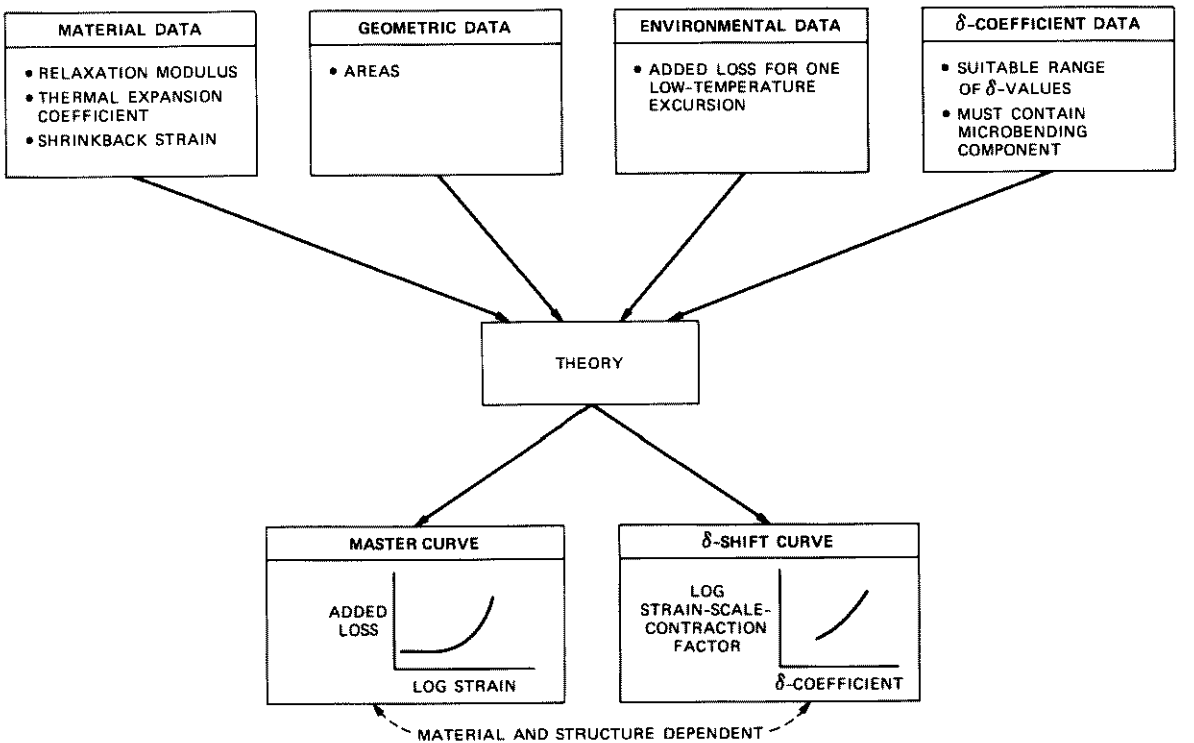



Fig. 17—Schematic of new environmental testing algorithm.

suitable material, geometric, environmental, and optical data are synthesized by way of our theory, we obtain a master curve and a $\delta$-shift curve that are sufficient to predict the environmental added loss for the candidate ribbon configuration for any desired environmental cycle. Notice that this environmental-testing scheme eliminates the necessity for long-term testing, only one low-temperature excursion being required.

(*iv*) There is a maximum value of $\delta$ that a particular ribbon design can accommodate and still meet a given performance requirement. This maximum permissible $\delta$-value is an excellent criterion for making coating/ribbon-structure comparisons.

By application of these techniques, we have found:

(*v*) The critical material and geometric parameters in ASR design are the shrinkback of the polyester tape and the product $\alpha EA$ of the coefficient $\alpha$ of thermal expansion, the time- and temperature-dependent relaxation modulus $E$, and the area $A$ associated with the fiber coating.

(*vi*) Any method of reducing the $\alpha EA$ product of the coating such as introducing a soft, single coating or a dual system with a soft primary coating should improve environmental performance.

## VIII. ACKNOWLEDGMENTS

This work was performed jointly with P. F. Mahr. The authors greatly appreciate the invaluable assistance provided by C. J. Aloisio, M. J. Buckler, R. P. DeFabritis, B. J. Overton, D. N. Ridgway, and C. R. Taylor all of Bell Laboratories in Atlanta, Georgia.

## APPENDIX

### *Transverse Thermal Contraction of an Adhesive-Sandwich Ribbon*

Consider a section of an ASR containing two coated fibers as shown in Fig. 18. If the temperature is instantaneously lowered by an amount

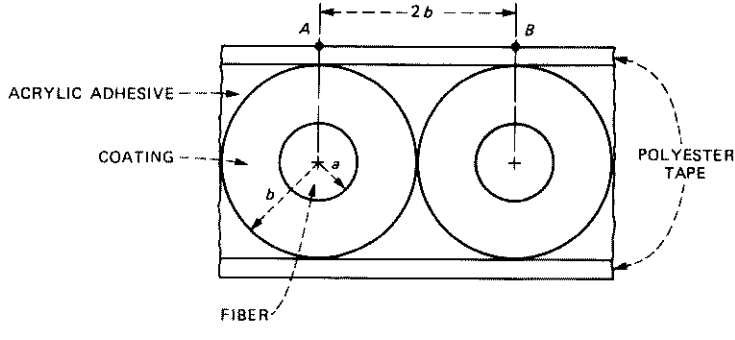



Fig. 18—Two-fiber portion of an ASR cross section prior to thermal contraction.

$|\Delta T|$ ($\Delta T < 0$) from the temperature $T_0$, the section $AB$ of tape would have an unstressed length $l_{TA}$ of

$$l_{TA} = 2b(1 + \alpha_{TA}\Delta T), \tag{17}$$

where $b$ is the outer radius of the coated fiber and $\alpha_{TA}$ is the coefficient of transverse thermal expansion of the polyester tape.

A coated fiber would also shrink in diameter, the final radius $b_f$ conforming to

$$b_f = b(1 + \alpha_*\Delta T), \tag{18}$$

where $\alpha_*$ is a number somewhere between the coefficients of thermal expansion of the glass and the plastic coating. To estimate $\alpha_*$ we appeal to linear thermoelasticity theory.

Denote by $u$ and $\sigma$ the radial displacement and stress fields (assumed axisymmetric) in the coating after the temperature is lowered. The stress field $\sigma$ is to vanish at $r = b$, and it must match the stress in the glass fiber at $r = a$. The displacements must also match at this fiber-coating interface. Since the glass is many times stiffer than the plastic, we approximate the boundary conditions at $r = a$ by supposing the glass-coating interface to shrink with the glass coefficient of thermal expansion. Thus,

$$u(a) = a\alpha_F\Delta T, \sigma(b) = 0, \tag{19}$$

where $\alpha_F$ is the coefficient of thermal expansion of the glass fiber.

When the axisymmetric solution to the requisite field equations[9] is substituted into (19), a pair of simultaneous, linear algebraic equations for two unknown constants result. Solving these equations and doing some algebra, one deduces that

$$u(b) = \alpha_* b\Delta T, \tag{20}$$

where

$$\alpha_* = \frac{4}{3b^2/a^2 + 1}[(b^2/a^2 - 1)\alpha + \alpha_F] \tag{21}$$

when Poisson's ratio of the plastic is taken as 1/3.

For the case at hand, we use the values for $a$ and $b$ from the table in Fig. 7 and select $\alpha$ and $\alpha_F$ for the urethane-acrylate coating and the glass, respectively, from Table III. Substitution of these numbers into (21) gives

$$\alpha_* = 3.18 \times 10^{-5}\ {}^\circ F^{-1}, \tag{22}$$

which is 95 percent of the urethane-acrylate value.

If the final length $l_{TA}$ of the tape segment is less than the final diameter $2b_f$ of the coated fiber, pressure will be created between the

fibers and microbending loss may result. Otherwise, the fibers will separate. Thus, the critical quantity is

$$d = l_{TA} - 2b_f. \tag{23}$$

Incorporating eqs. (17) and (18) into (23) results in

$$d = 2b\Delta T(\alpha_{TA} - \alpha_*). \tag{24}$$

For the case at hand $\alpha_{TA} < \alpha_*$ since the transverse thermal expansion coefficient for the tape is even less than the axial value reported in Table III. We thus conclude from (24) that $d$ is positive (recall that $\Delta T < 0$).

Thus as the temperature is lowered the fibers separate and microbending loss from transverse contraction is therefore impossible. As the temperature is raised, however, the fibers obviously approach one another since at high temperatures $\alpha_{TA}$ is still less than $\alpha_*$, but now $\Delta T > 0$. Moreover, there is transverse shrinkback at high temperatures. There is indeed environmental loss data which indicate increased loss at high temperatures.

## REFERENCES


1. D. Gloge, "Optical-Fiber Packaging and its Influence on Fiber Straightness and Loss," B.S.T.J., *54*, No. 2 (February 1975), pp. 245–262.
2. W. B. Gardner, "Microbending Loss in Optical Fibers," B.S.T.J., *54*, No. 2 (February 1975), pp. 457–465.
3. D. Marcuse, "Losses and Impulse Response of a Parabolic Index Fiber With Random Bends," B.S.T.J., *52*, No. 8 (October 1973), pp. 1423–1437.
4. M. J. Saunders and W. L. Parham, "Adhesive Sandwich Optical Fiber Ribbons," B.S.T.J., *56*, No. 6 (July–August 1977), pp. 1013–1014.
5. C. M. Miller, "Fiber-Optic Array Splicing With Etched Silicon Chips," B.S.T.J., *57*, No. 1 (January, 1978), pp. 75–90.
6. K. Inada, "A New Graphical Method Relating to Optical Fiber Attenuation," Optics Commun., *19*, No. 3 (December 1976), pp. 437–438.
7. R. A. Schapery, "Thermal Expansion Coefficients of Composite Materials Based on Energy Principles," J. Composite Mater., *2* (July 1968), pp. 380–404.
8. R. A. Schapery, "Viscoelastic Behavior and Analysis of Composite Materials," *Composite Materials*, Vol. 2, Mechanics of Composite Materials, Ed. G. P. Sendeckyj, New York: Academic Press, 1974, pp. 85–168.
9. B. A. Boley and J. H. Weiner, *Theory of Thermal Stress*, New York: Wiley, 1960, p. 289.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 4, April 1983
*Printed in U.S.A.*

# Adaptive Linearization of Power Amplifiers in Digital Radio Systems

By A. A. M. SALEH and J. SALZ

(Manuscript received October 5, 1982)

*High-frequency power amplifiers operate most efficiently at saturation, i.e., in the nonlinear range of their input/output characteristics. This phenomenon has traditionally dictated the use of constant envelope modulation methods for data transmission, resulting in circular signal constellations. This approach has inherently limited the admissible data rates in digital radio. In this paper we present a method for solving this problem without sacrificing amplifier power efficiency. We describe and analyze an adaptive linearizer that can automatically compensate for amplifier nonlinearity and thus make it possible to transmit multilevel quadrature amplitude modulated signals without incurring intolerable constellation distortions. The linearizer utilizes a real-time, data-directed, recursive algorithm for predistorting the signal constellation. Our analysis and computer simulations indicate that the algorithm is robust and converges rapidly from a blind start. Furthermore, the signal constellation and the average transmitted power can both be changed through software.*

## I. INTRODUCTION

Progress in high-speed data transmission over radio channels has lagged behind that of the voiceband channel. Inherent difficulties associated with implementing automatic equalizers are partially responsible, but the application of multilevel quadrature amplitude modulation (QAM) also has been inhibited by the amplitude (AM/AM) and phase (AM/PM) nonlinearities present in radio-frequency (RF) power amplifiers.

Recent work[1] has evolved design principles showing a possibility of substantial improvements in QAM performance over linear fading radio channels. A crucial obstacle to achieving these gains is the nonlinear distortion introduced by power amplifiers. Attempts to re-

alize high-speed data transmission over these channels force consideration of methods to cope with this nonlinear distortion.

One approach is to back off from saturation sufficiently so that the signal level is restricted to the linear range of amplification. The required amount of power back-off can be several decibels, resulting in an inefficient operation of the power amplifier. Moreover, the achievement of a given desired level of average transmitted power would require the use of a large, expensive, and high-power-consuming amplifier.

It has been realized[2-7] that some improvements in this regard can be obtained by using fixed signal predistortion circuits prior to amplification. Such circuits, however, cannot compensate for drifts in power amplifier nonlinearities caused by temperature changes, dc power variations, and component aging. These fluctuations can considerably degrade performance of systems employing constellations with large numbers of points, say 64 or higher, and the use of an adaptive technique is necessary in these applications.

Conceptually, the nonlinear distortion introduced by the power amplifier can be minimized at the receiver by an adaptive nonlinear equalizer. Such schemes have been proposed and studied in voiceband data transmission[8] and for filtered PSK signals operating over satellite channels.[9] This approach does not seem to be reasonable or necessary in our application, since the source of the nonlinearity is at the transmitter. Thus, it would appear logical to equalize the nonlinearity at the transmitter, where it occurs and where the transmitted bits are available.

Thus, this paper focuses on the problem of adaptive predistortion linearization. We describe a transmitter-based recursive algorithm for predistorting the signal constellation, thereby rendering a virtually linear transmitter. The algorithm operates in real time and is data directed. The predistortion is accomplished within a digital memory, which is used to generate the desired baseband signal. This maximizes the use of digital technology, and increases the reliability and flexibility of the system. For example, it is possible to change the signal constellation and the average transmitted power through software. Our treatment applies only to single-valued, memoryless nonlinearities.

The idea of adaptive predistortion of signal constellations has been previously suggested.[10] In this reference, the predistortion of each point of the constellation is accomplished by switching the RF or intermediate-frequency (IF) input signal to a separate path containing an adjustable diode attenuator and an adjustable diode phase shifter. The analog hardware required for such an implementation would be quite involved, especially for a large number of points. We have also found in the patent literature[11] a description of a digital adaptive

predistorter that appears to be similar to the one we present here. However, no detailed description of the algorithm or its behavior is provided.

Our basic ideas are described in Section II. Mathematical analysis is provided in Section III, and simulation results of the adaptive linearizer with finite-precision arithmetic are given in Section IV.

## II. GENERAL DESCRIPTION AND REQUIREMENTS OF OPERATION

A block diagram of a QAM transmitter with the proposed predistortion linearizer is shown in Fig. 1. A random access memory (RAM) contains the predistorted values of the in-phase and quadrature voltages of each point on the QAM constellation. A memory-lookup encoder obtains each input data symbol and generates the RAM addresses of the desired signal point. The corresponding stored, predistorted voltage values are converted to analog voltages using a pair of digital-to-analog (D/A) converters. These voltages drive a quadrature modulator, which generates the desired predistorted RF signal for the duration of the input symbol. That signal is then amplified, filtered, and transmitted. This part of the linearizer is similar in operation to a recently proposed memory-based encoder described in Ref. 12.

To update the RAM information, the amplifier output is sampled by a directional coupler and demodulated using the same local oscillator used for the modulator, which eliminates the need for carrier recovery. The output in-phase and quadrature voltages of the demodulator are converted to digital form using a pair of analog-to-digital (A/D) converters. A linearizing processor, which is the heart of the linearizer, receives this information, compares it with the input data, and computes the resulting error. A recursive algorithm, which is discussed in the next section, uses the error to update the voltage values in the RAM corresponding to the particular data point under consideration. Note that each point on the signal constellation is treated separately. Thus, the linearizer can support any desired constellation.

The memory-lookup encoder and the D/A converters have to operate, of course, at the full signaling rate. However, the linearizing processor and the A/D converters can operate at a much reduced rate, since the updating process is only needed to compensate for drifts that occur on a much slower time scale than the data rate.

We now emphasize a crucial point. The proper operation of the linearizer as described above assumes that the amplifier is memoryless, and requires that the signal not be filtered before the power amplifier. Thus, all pulse shaping and filtering must be performed by the combination of the post-amplifier, RF bandpass filter, and the receiver filters. The former filter should be designed just to meet FCC (or

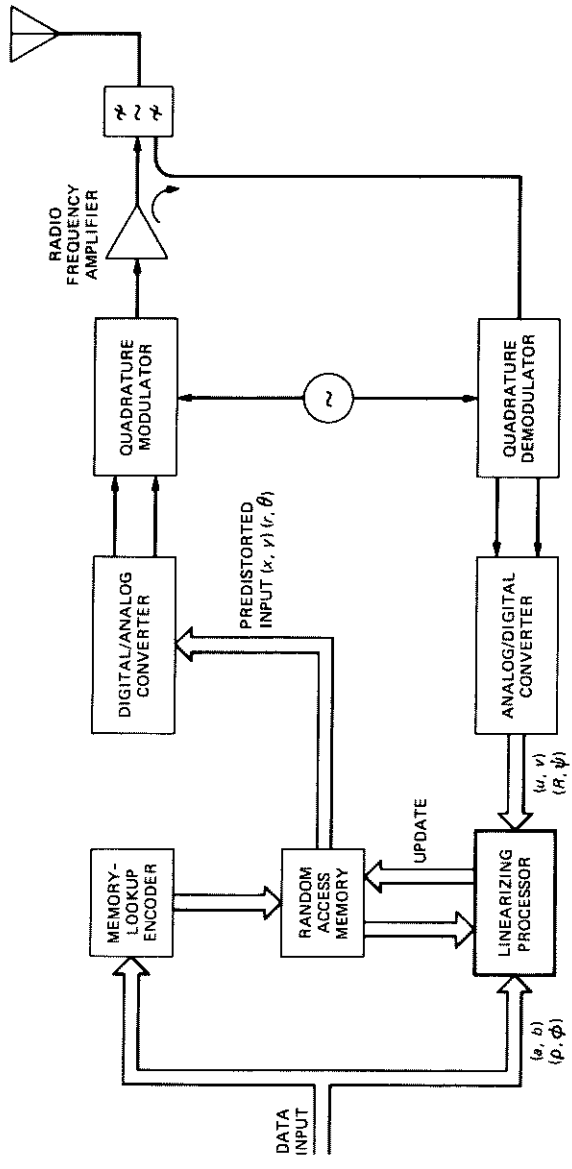



Fig. 1—Schematic representation of the adaptive, digital, predistortion linearizer.

other) emission rules for square input pulses, as described in Ref. 13. Such an implementation may require an automatic equalizer at the receiver to eliminate residual intersymbol interference. It was shown[1] that, even with ideal filtering, adaptive equalization would still be necessary to compensate for multipath fading using QAM signals with large numbers of levels. Thus, the elimination of pre-amplifier pulse shaping appears to be a mild constraint.

Because of the more stringent requirement on the post-amplifier, RF bandpass filter, its loss would of course be increased over that of the conventional case where preamplifier filtering does the spectrum shaping. However, computations[14] based on practical filters[15] operating in the 6-GHz band show that the RF filter loss would only increase from about 0.5 dB to 1.5 dB. However, we will see in Section IV that the linearizer in our system would allow the operation of the power amplifier to approach saturation. This would result in several decibels of power increase, which would more than compensate for the one-decibel increase in the filter loss.

## III. THE RECURSIVE ALGORITHM

As we already mentioned, our approach is applicable to any signal constellation. However, for clarity of exposition, we restrict our analysis to a rectangular constellation. Referring to Figs. 1 and 2, we denote a point on the rectangular QAM constellation by the complex number

$$a + ib = \rho e^{i\phi},$$

where, for an $L^2$-level system, $a$ and $b$ assume values on the lattice $\pm 1 \pm 3 \pm 5 \pm \cdots \pm (L-1)$. We denote the predistorted point by the

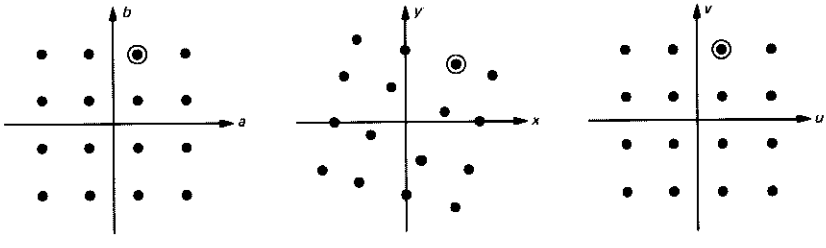



Fig. 2—The signal constellations at various points in the linearizer of Fig. 1. (a) Quadrature amplitude modulation data. (b) Digital/analog data. (c) Analog/digital data.

complex number

$$x + iy = re^{i\theta}.$$

A sequence of these points amplitude and phase modulates a carrier at frequency $f_c$ and the resulting signal,

$$S(t) = \sum_m r^{(m)} p(t - mT) e^{i[2\pi f_c t + \theta^{(m)}]}, \tag{1}$$

is applied to the amplifier. In (1), $p(t)$ is a rectangular pulse, $1/T$ is the signaling rate, and $r^{(m)}$, $\theta^{(m)}$ represent the amplitude and phase of the $m$th data symbol.

The amplifier is customarily represented by a pair of memoryless nonlinear functions.[16,17] The amplitude function, $A(r)$, causes amplitude distortion (AM/AM) and the phase function, $\Phi(r)$, causes amplitude to phase conversion (AM/PM). Thus the signal, (1), after amplification becomes

$$S_o(t) = \sum_m A(r^{(m)}) p(t - mT) e^{i\{2\pi f_c t + \theta^{(m)} + \Phi[r^{(m)}]\}}. \tag{2}$$

Figure 3 shows sketches of typical curves $A(r)$ and $\Phi(r)$ for a traveling-wave tube (TWT) power amplifier.

We remark that if the functions $A(\cdot)$ and $\Phi(\cdot)$ were known exactly we could choose transformations $g(\cdot)$ and $h(\cdot)$ from points $(\rho, \phi)$ to

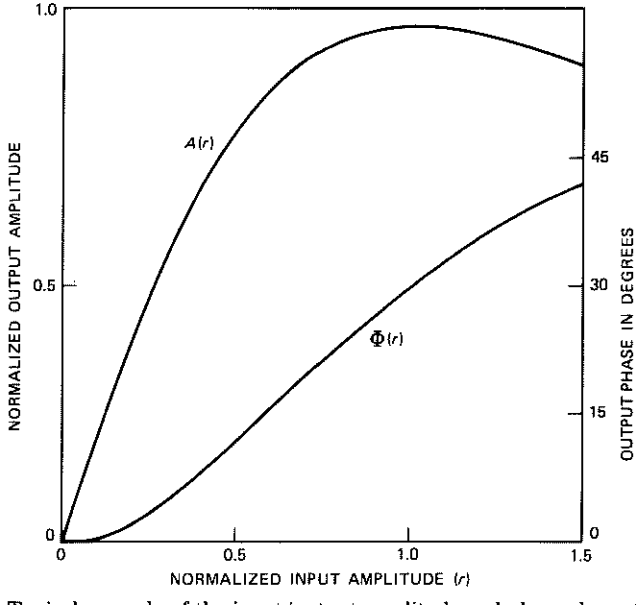



Fig. 3—Typical example of the input/output amplitude and phase characteristics of a TWT amplifier.

$(r, \theta)$ so that

$$A(r) = A[g(\rho)] = G\rho, \tag{3}$$

where

$$r = g(\rho)$$

and

$$\Phi(r) + h(\phi, \rho) = \phi + \zeta, \tag{4}$$

where

$$\theta = h(\phi, \rho).$$

The constant gain, $G$, in (3) is some desired gain, taking into account the linear gain of the power amplifier and the coupling factor of the sampling directional coupler. The fixed phase shift, $\zeta$, in (4) is arbitrary and can be set to zero without loss of generality.

Finally, let the measured data point from the A/D converter be

$$u + iv = Re^{i\psi}.$$

The object of the predistortion is to find a solution

$$r = \hat{r}$$

and

$$\theta = \hat{\theta},$$

such that

$$Re^{i\psi} = G\rho e^{i\phi}. \tag{5}$$

We now describe an iteration procedure that converges to (5).

Since data are usually scrambled, a specific data point $(\rho, \phi)$ will occur at random. Let $(r_n, \theta_n)$ be the predistorted RAM data point and $(R_n, \psi_n)$ be the measured data point at the $n$th time the desired data point $(\rho, \phi)$ occurs. The measured output radius $R_n$ is then

$$R_n = A(r_n) + \nu_n \tag{6a}$$

while the measured angle is

$$\psi_n = \Phi(r_n) + \theta_n + \mu_n, \tag{6b}$$

where $\nu_n$ and $\mu_n$ are zero-mean measurement errors. If the measurements were perfect and noiseless, we would simply solve the following two nonlinear equations

$$f(\hat{r}) = A(\hat{r}) - G\rho = 0 \tag{7}$$

and

$$\Phi(\hat{r}) + \hat{\theta} - \phi = 0 \tag{8}$$

for $\hat{r}$ and $\hat{\theta}$.

A great number of iterative procedures are known for solving the nonlinear equations given in (7) and (8). In the presence of measurement noise (with convex functions $A$ and $\Phi$), stochastic approximation algorithms[18] provide efficient methods. So, if one chooses step sizes $\alpha_n$ and $\beta_n$, which behave as $o(1/n)$, the following recursions are known to converge to the true solutions $\hat{r}$ and $\hat{\theta}$ in a mean-square sense:

$$r_{n+1} = r_n - \alpha_n(R_n - G\rho) \tag{9a}$$

$$\theta_{n+1} = \theta_n - \beta_n(\psi_n - \phi). \tag{9b}$$

To provide a rationale for the above recursions we analyze, in some detail, the behavior of (9a). The behavior of (9b) is similar. Since $f(X)$, eq. (7), is continuous on $(X_0, X_{\max})$ and if, by hypothesis, the derivative $f'(X)$ of $f(X)$ exists on this interval, then there is a $\xi$ such that

$$X_0 \le \xi \le X_{\max}$$

$$f'(\xi) = \frac{f(X_{\max}) - f(X_0)}{X_{\max} - X_0}. \tag{10}$$

Applying this mean-value theorem to $f(r_n)$, eq. (7), we get

$$f(r_n) = A(r_n) - G\rho = f(\hat{r}) + f'(\xi_n)(r_n - \hat{r}), \tag{11}$$

where $\xi_n$ lies between $r_n$ and $\hat{r}$. Substituting this into (9a) we get

$$r_{n+1} - \hat{r} = r_n - \hat{r} - \alpha_n[A'(\xi_n)(r_n - \hat{r}) + \nu_n]. \tag{12}$$

In (12) we made use of the fact that $f(\hat{r}) = 0$ and subtracted $\hat{r}$ from both sides of the equation.

Letting $\alpha_n A'(\xi_n) = \gamma_n$, $r_n - \hat{r} = \delta_n$ and then iterating (12) we obtain

$$\begin{aligned}\delta_{n+1} &= (1 - \gamma_n)\delta_n - \frac{\gamma_n \nu_n}{A'(\xi_n)} \\ &= \delta_1 \prod_{i=1}^{n} (1 - \gamma_i) - \sum_{k=1}^{n} \frac{\nu_k}{A'(\xi_k)} \frac{\gamma_k}{1 - \gamma_k} \prod_{i=k}^{n} (1 - \gamma_i).\end{aligned} \tag{13}$$

We now compute the mathematical expectation of $\delta_n$ and the variance:

$$\begin{aligned}\bar{\delta}_{n+1} = E\{\delta_{n+1}\} &= E\{\delta_1\} \prod_{i=1}^{n} (1 - \gamma_i) \\ &\le E\{\delta_1\} e^{-\sum_{i=1}^{n} \gamma_i}\end{aligned} \tag{14}$$

and

$$\sigma_{n+1}^2 = E\{\delta_{n+1} - \bar{\delta}_{n+1}\}^2 = \sigma_\nu^2 \sum_{k=1}^{n} \frac{\gamma_k^2}{[A'(\xi_k)]^2(1 - \gamma_k)^2} \prod_{i=k}^{n} (1 - \gamma_i)^2, \tag{15}$$

where $\sigma_\nu^2 = E\{\nu_n^2\}$. We see from (14) and (15) that rapid convergence of the algorithm is critically dependent on the step-size sequence, $\gamma_n$, which in turn depends on the derivative of the nonlinearity in the neighborhood of the solution. While the derivative is unknown, in most applications it can be bounded away from zero and this makes it possible to estimate the best step-size sequence. We see from (14) that, even in the absence of noise, the algorithm is guaranteed to converge to the true solution only if the step-size series diverges, i.e.,

$$\sum_{i=1}^{n} \gamma_i = \sum_{i=1}^{n} \alpha_i A'(\xi_i) \to \infty, \qquad n \to \infty. \tag{16}$$

If $\alpha_i$ and $A'(\xi_i)$ are restricted to be positive, (16) is equivalent to requiring $\sum_{i=1}^{n} \alpha_i$ to diverge. So it follows that the structure of the sequence can be of the form

$$\alpha_n = \frac{a}{n^\eta}, \tag{17}$$

where $a$ is a positive constant and $0 \le \eta \le 1$. It can be shown that for this choice of $\alpha_n$, and for $c = aA'(\hat{r})$,

$$\frac{E\{\delta_{n+1}\}}{E\{\delta_1\}} = e^{-\frac{cn^{(1-\eta)}}{1-\eta} + cq(\eta)}, \qquad (0 \le \eta < 1) \tag{18}$$

and

$$[A'(\hat{r})]^2 \frac{\sigma_{n+1}^2}{\sigma_\nu^2} = \frac{c}{2n^\eta}, \qquad (0 < \eta < 1), \tag{19}$$

where

$$q(\eta) = \lim_{n \to \infty} \left\{ \frac{n^{1-\eta}}{1-\eta} - \sum_{k=1}^{n} \frac{1}{k^\eta} \right\};$$

e.g., $q(0) = 0$ and $q(1/2) \approx 1.46$. Table I shows the behavior of the statistics for the special cases $\eta = 0$ and $\eta = 1$, where (18) and (19) are not applicable.

Table I—Behavior of statistics for $\eta = 0$ and $\eta = 1$

| $\eta$ | $c = aA'(\hat{r})$ | $\frac{\bar{\delta}_{n+1}}{\bar{\delta}_1}$ | $\frac{\sigma_{n+1}^2}{\sigma_\nu^2}[A'(\hat{r})]^2$ |
|---|---|---|---|
| 0 | $0 < c < 2$ | $(1-c)^n < e^{-cn}$ | $\frac{c}{2-c}[1-(1-c)^{2n}] \sim \frac{c}{2-c}$ |
| 1 | $c > \frac{1}{2}$ | $e^{-0.577}/n^c$ | $\frac{c^2}{2c-1} \times \frac{1}{n}$ |
| 1 | 1 | $e^{-0.577}/n$ | $\frac{1}{n}$ |
| 1 | 1/2 | $e^{\frac{0.577}{2}}/\sqrt{n}$ | $\frac{ln(n)}{4n}$ |

It is noted from eqs. (17) through (19) and Table I that the choice of a fixed step size ($\eta = 0$) gives the fastest convergence of the mean, $\bar{\delta}_{n+1}$, but results in a finite variance, $\sigma^2_{n+1}$, as $n \to \infty$. On the other hand, a variable, progressively smaller step size ($\eta \neq 0$) gives a slower convergence of the mean, but results in a variance approaching zero as $n \to \infty$. Thus, a fixed step size should be used if the measurement error variance, $\sigma^2_\nu$, is very small, and a variable step size should be used if the variance is large. Actually, when finite-precision arithmetic is employed, the choice of a progressively smaller step size can lead to large errors.[19] Thus, in our particular application where the measurement error can be made small, and where the use of finite-precision arithmetic is necessary, a fixed step size is more suitable.

We note that our algorithm, (9), is based on the polar representation of the data symbols, while the hardware implementation of Fig. 1 is based on the rectangular representation. Thus, the linearizing processor is required to convert back and forth between the two representations. One could avoid this conversion by replacing (9) by the rectangular-based algorithm

$$x_{n+1} + iy_{n+1} = x_n + iy_n - (\alpha_n + i\beta_n)[u_n + iv_n - G(a + ib)].$$

Unfortunately, however, the convergence of this algorithm is not guaranteed, even for some well-behaved amplifier characteristics.

## IV. SIMULATIONS

Here we present the results of computer simulations of the algorithm, (9), with fixed step sizes and finite-precision arithmetic. The precision is limited by the finite number of bits of the D/A and A/D converters (Fig. 1). As an example, we consider a TWT amplifier with normalized amplitude and phase nonlinearities of the form[17]

$$A(r) = \frac{2r}{1 + r^2}, \tag{20a}$$

$$\Phi(r) = 60^\circ \frac{r^2}{1 + r^2}, \tag{20b}$$

which are sketched in Fig. 3. Note that at saturation, $r = 1$, $A(r) = 1$, and $\Phi(r) = 30^\circ$. Figure 4 shows the severe distortion of the output constellation obtained with such an amplifier for a 64-QAM input signal driven with its corner point at saturation.

Figures 5, 6, and 7 show the simulated results of the output signal constellations after the application of (9) for three different cases (explained below). In all cases the amplifier drive is maintained at saturation. The four quadrants in each figure represent different combinations of the numbers of bits of the D/A and A/D converters, as indicated in Table II.

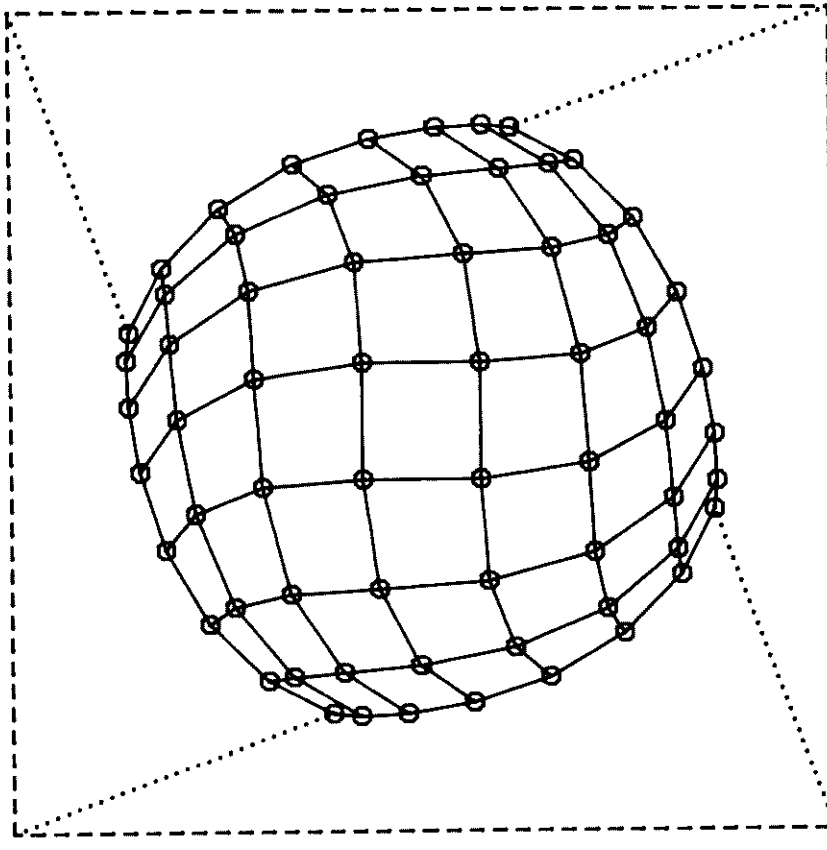


Fig. 4—The distortion of the signal constellation obtained with the amplifier of Fig. 3 for a 64-QAM input signal with its corner points at saturation.

Figure 5 corresponds to the case of no measurement error, i.e., $\nu_n = \mu_n = 0$ in (6). The step sizes used in (9) are $\alpha_n = 0.5$ and $\beta_n = 1.0$. Since the normalized small-signal gain, $A'(r)$, of the amplifier is about 2, the choice of $\alpha_n = 0.5$ results in a value for $c$ of about 1. This results in the fastest convergence of (9a), as can be seen from the first row of Table I. Similarly, the choice of $\beta_n = 1.0$ results in the fastest convergence of (9b) since that equation is linear in $\theta$. The initial guesses of $r$ and $\theta$ for each constellation point in Fig. 5 were chosen at random, i.e., from a blind start. The results indicated in the figure are those after 25 iterations for each constellation point. The point scatter shown is entirely due to the finite resolution of the D/A and A/D converters since no measurement errors were assumed.

It is clear from Fig. 5 that the use of 8 bits for both the D/A and A/D converters (fourth quadrant) gives quite an acceptable performance for 64 QAM. The use of 9 bits for each converter (first quadrant)

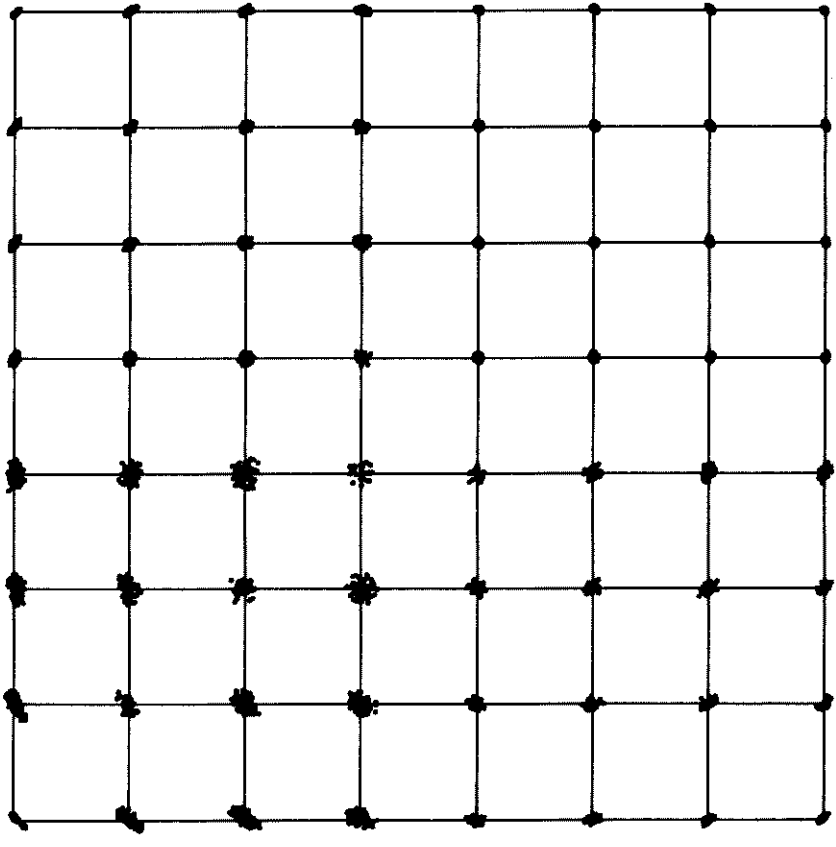


Fig. 5—Simulated results of the output signal constellation after 25 iterations with no measurement errors, and for step sizes $\alpha_n = 0.5$ and $\beta_n = 1.0$. The number of bits of the D/A and A/D converters are different in each quadrant as indicated in Table 2.

results in an almost perfect output constellation. In general, for $L^2$-QAM, where $L = 2^M$, D/A and A/D converters with $(M + 5)$ bits are needed for acceptable performance, and $(M + 6)$ or more bits are needed for almost perfect performance.

In Fig. 6, the step sizes employed are $\alpha_n = 0.5$ and $\beta_n = 1.0$, as in Fig. 5. However, a measurement error was introduced that is equivalent to a 30-dB signal-to-noise ratio at the corner points of the constellation. With the normalization used for $A(r)$ in (20a), this noise corresponds to $\sigma_\nu^2 = A^2(r)\sigma_\mu^2 = 0.0005$, where $\sigma_\nu^2$ and $\sigma_\mu^2$ are the variances of the errors $\nu_n$ and $\mu_n$ defined in (6). It is clear from Fig. 6 that such a level of noise, in combination with the large step sizes employed, gives unacceptable results.

In Fig. 7, the same noise level as that of Fig. 6 is employed. However, the step sizes were reduced by a factor of 10, i.e., $\alpha_n = 0.05$ and $\beta_n =$

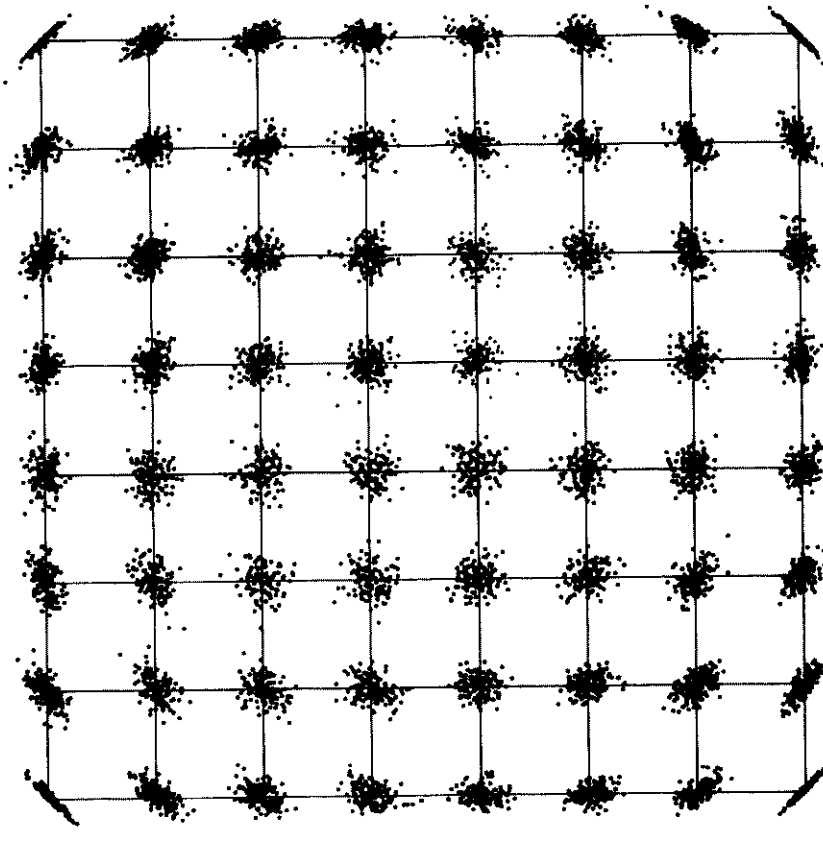


Fig. 6—Same as Fig. 5, but with a measurement error equivalent to 30-dB of signal-to-noise ratio at the corner points of the constellation.

0.1, resulting in a greatly improved performance. About 100 steps of iteration were needed in this case to reach convergence. Note that in spite of the measurement noise, the performance in the third quadrant of Fig. 7 is better than the corresponding performance in Fig. 5, where no noise is present. This is due to the reduced step sizes, and to the fact that the noise tends to smooth out quantization errors.

## V. SUMMARY AND CONCLUSIONS

We have proposed and analyzed a transmitter-based, adaptive linearizer, which automatically compensates for power amplifier nonlinearity in digital radio systems employing multilevel quadrature amplitude modulation. The linearizer utilizes a real-time, data-directed recursive algorithm for predistorting the signal constellation. The algorithm is robust and results in rapid convergence, even from a blind

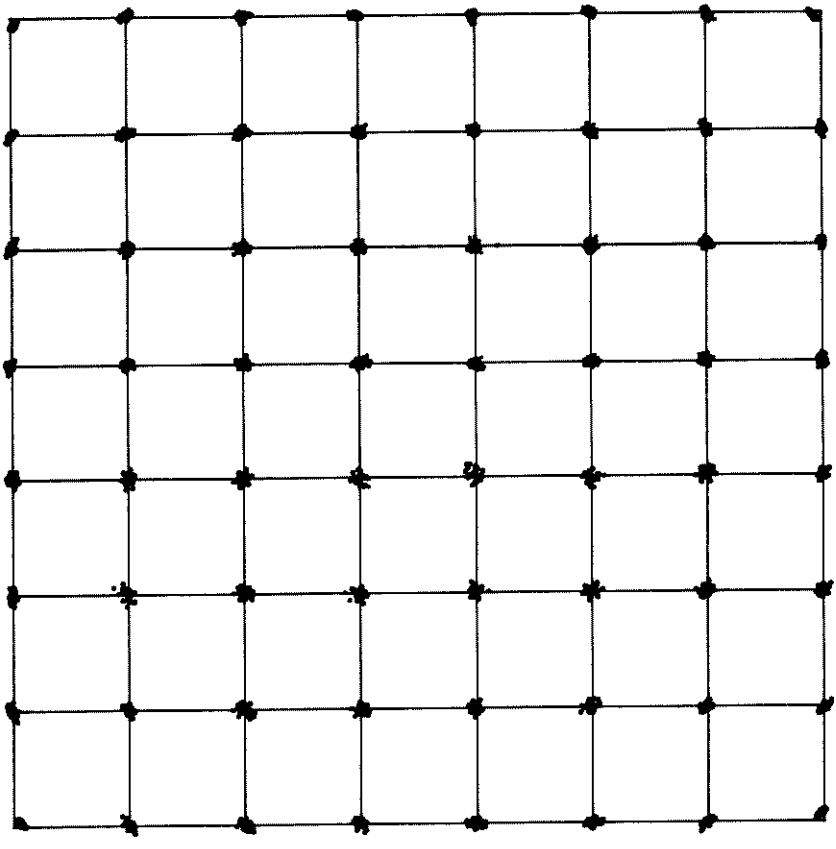


Fig. 7—Same as Fig. 6, but with the step sizes reduced by a factor of 10, i.e., $\alpha_n = 0.05$ and $\beta_n = 0.1$. About 100 iteration steps were needed for convergence.

Table II—Number of bits of the D/A and A/D converters for the four quadrants of Figs. 5, 6, and 7

| Quadrant Number | Number of Bits of the D/A Converter | Number of Bits of the A/D Converter |
|---|---|---|
| 1 | 9 | 9 |
| 2 | 9 | 7 |
| 3 | 8 | 6 |
| 4 | 8 | 8 |

start. The digital nature of the linearizer allows both the signal constellation and the average transmitted power to be changed through software.

The proposed linearizer is only suitable for amplifiers with essen-

tially memoryless and single-valued nonlinearities. Consequently, pulse shaping is not permitted before amplification. In our approach the task of pulse shaping is relegated to the combination of the transmit RF filter, a receiver-based filter, and an automatic equalizer. The latter is presumed to be required anyway to deal with multipath fading, especially for signals having a large number of levels.

Our principal conclusion is that high-power RF amplifiers can operate at saturation, where they are most efficient, provided that an adaptive predistortion linearizer is used. We have demonstrated that such a system is feasible and have provided a theory and a methodology for assessing its performance.

## REFERENCES


1. G. J. Foschini and J. Salz, "Digital Communications over Fading Radio Channels," B.S.T.J., *62*, No. 2, Part 1 (February 1983), pp. 429–56.
2. R. P. Hecken and R. C. Heidt, "Predistortion Linearizer of the AR6A Transmitter," ICC '80 Conf. Rec., *2* (June 1980), pp. 33.1.1–6.
3. T. Nojima and Y. Okamato, "Predistortion Nonlinear Compensator for Microwave SSB-AM System," ICC '80 Conf. Rec., *2* (June 1980), pp. 33.2.1–6.
4. A. Egger, M. Horn, and T. Vien. "Broadband Linearization of Microwave Power Amplifiers," Proc. of the 10th European Microwave Conf., September 1980, pp. 490–4.
5. C. Brenson, M. Palazo, and R. Neyer, "Linearizing T.W.T. Amplifiers in Satellite Transponders—System Aspects and Practical Implementation," Proc. of the AIAA 8th Commun. Satellite Conf., April 1980, pp. 80–9.
6. G. Sato, H. Shibazaki, T. Asai, and T. Kurokawa, "A Linearizer for Satellite Communications," ICC '80 Conf. Rec., *2* (June 1980), pp. 33.3.1–5.
7. G. Satoh and T. Mizuno, "Impact of a New TWTA Linearizer Upon QPSK/TDMA Transmission Performance," IEEE J. Select. Areas in Comm., special issue on Digital Satellite Communications, *SAC-1*, No. 1 (January 1983), pp. 39–45.
8. D. D. Falconer, "Adaptive Equalization of Channel Nonlinearities in QAM Data Transmission Systems," B.S.T.J., *57*, No. 7 (September 1978), pp. 2589–611.
9. S. Benedetto and E. Biglieri, "Nonlinear Equalization of Digital Satellite Channels," IEEE J. Select Areas in Comm., special issue on Digital Satellite Communications, *SAC-1*, No. 1 (January 1983), pp. 57–62.
10. W. J. Weber, III, "The Use of TWT Amplifiers in *M*-Ary Amplitude and Phase-Shift Keying Systems," ICC '75 Conf. Rec., *3* (June 1975), 36.17–21.
11. R. C. Davis and W. Boyd, "Adaptive Predistortion Technique for Linearizing a Power Amplifier for Digital Data Systems," U.S. Patent 4,291,277, issued September 22, 1981.
12. T. L. Osborne and P. D. Karabinis, unpublished work.
13. L. J. Greenstein and D. Vitello, "Required Transmit Filter Bandwidth in Digital Radio Systems," IEEE Trans. Comm., *COM-29*, No. 9 (September 1981), pp. 1405–8.
14. H. Wang and C. Ren, private communications.
15. J. K. Plourde and C. Ren, "Application of Dielectric Resonators in Microwave Components," IEEE Trans. Microwave Theory Tech., *MTT-29*, No. 8 (August 1981), pp. 754–70.
16. A. L. Berman and C. H. Mahle, "Nonlinear Phase Shift in Traveling-Wave Tubes as Applied to Multiple Access Communications Satellites," IEEE Trans. Comm. Tech., *COM-18*, No. 1 (February 1970), pp. 37–48.
17. A. A. M. Saleh, "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers," IEEE Trans. Comm., *COM-29*, No. 11 (November 1981), pp. 1715–20.
18. A. E. Albert and L. A. Gardner, Jr., "Stochastic Approximation and Nonlinear Regression," Research Monograph No. 42, Cambridge, MA: MIT Press, 1967.
19. R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the Design of Gradient Algorithms for Digitally Implemented Adaptive Filters," IEEE Trans. Circuit Theory, *CT-20*, No. 2 (March 1973), pp. 125–36.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 4, April 1983
*Printed in U.S.A.*

# An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition

By S. E. LEVINSON, L. R. RABINER, and M. M. SONDHI

(Manuscript received September 10, 1982)

*In this paper we present several of the salient theoretical and practical issues associated with modeling a speech signal as a probabilistic function of a (hidden) Markov chain. First we give a concise review of the literature with emphasis on the Baum-Welch algorithm. This is followed by a detailed discussion of three issues not treated in the literature: alternatives to the Baum-Welch algorithm; critical facets of the implementation of the algorithms, with emphasis on their numerical properties; and behavior of Markov models on certain artificial but realistic problems. Special attention is given to a particular class of Markov models, which we call "left-to-right" models. This class of models is especially appropriate for isolated word recognition. The results of the application of these methods to an isolated word, speaker-independent speech recognition experiment are given in a companion paper.*

## I. INTRODUCTION

It is generally agreed that information in the speech signal is encoded in the temporal variation of its short-duration power spectrum. To decode the signal, then, requires techniques for both estimation of power spectra and tracking their changes in time. This paper is concerned with the application of the theory of probabilistic functions of a (hidden) Markov chain to modeling the inherent nonstationarity of the speech signal for the purposes of automatic speech recognition (ASR).

The use of hidden Markov models for ASR was proposed by Baker[1,2] and, independently, by a group at IBM.[3–12] The theory on which their work rests is due to Baum et al.[13–17] Its first appearance in the literature occurred several years before Baker's studies and has since been

explored in some detail.[18,19] Our previous work in ASR has used temporal alignment procedures based on dynamic programming techniques,[20] and we hoped that through studying the new (to us) body of material we could improve the performance and/or capabilities of our present ASR systems.

Our initial goal, therefore, was to understand the theory of hidden Markov models sufficiently well to enable us to implement a new ASR system that could be compared directly to our existing ones. We have, in fact, been able to accomplish that goal, and a description and the results of our experiments are reported in a companion paper.[21] In the course of our studies, we have collected and integrated a number of loosely related mathematical techniques pertinent to Markov modeling. We have also modified and adapted these techniques to the specific ASR problems we wished to study. Our purpose in writing this tutorial, then, is to present this synthesis in a way that will be enlightening to those not familiar with the theory of hidden Markov models. We also hope that this treatment will provide for a better understanding of our companion paper. Finally, we hope to make the presentation general enough so that the theory is seen to be applicable to more than the problem of ASR.

We shall proceed as follows. We begin by defining probabilistic functions of a (hidden) Markov[22] chain and then show how they may be used in a natural way to model the speech signal. Once this is done, our task is reduced to solving two specific and well-defined mathematical problems: (i) computing the parameters of a proposed model conditioned on a sequence of observations assumed to have been generated by the model, and (ii) calculating the probability that a given set of observations was produced by a particular model.

First we review the solution to these problems as originally given by Baum,[13] who treated them as problems in statistical estimation. Lest the problems be too narrowly construed, we look at them as problems of classical constrained optimization. This allows us to give a partial geometrical interpretation to the Baum-Welch algorithm and to relate it to other studies of the problem by Baum and Eagon[14] and Baum and Sell.[17] It also makes clear that there are other methods of solution available that may, in certain instances, have advantages over the Baum-Welch algorithm. Finally, we discuss the dynamic programming algorithm of Viterbi[23] as an alternative to the so-called "forward-backward" method of Baum[13] for computing the probability of a sequence of observations conditioned on a specific model.

The treatment of these problems in the existing literature, and as recounted here, can lull a prospective user of the theory into a false sense of security. The equations look innocuous enough but, in reality, they overlook two problems that, though uninteresting from a theo-

retical standpoint, are of great significance for a robust implementation. We believe it is worthwhile to address, first, a numerical problem arising from the evaluation of certain frequently occurring algebraic expressions, and then an experimental difficulty precipitated by the inescapable reality of finite training-data set.

The numerical problem arises because, regardless of the method of solution chosen, one is required to evaluate a product of stochastic matrices involving a number of factors proportional to the number of observations. In any real computer, this will ultimately result in underflow. Fortunately, the computation can be scaled using a technique that subsequently will be seen to have some very useful properties.

The problem of insufficient training data can be ameliorated by changing the constraints on the optimization problem. This can be simply and directly accomplished in the classical methods. We show that the Baum-Welch algorithm, too, can be modified to produce the same result. Both of these methods appear to be simpler in implementation than the technique proposed by Jelinek and Mercer.[9]

Finally, under the heading of implementational considerations, we discuss techniques for model averaging. These can be used both for block processing of observations in case one is subject to storage limitations, and for increasing model stability under some circumstances.

The speech recognition experiment that we had in mind was on a speaker-independent, isolated word recognition system with a small vocabulary. Oddly enough, this is a simpler task than those to which the theory had already been applied by Baker[1] and the IBM group.[3-12] Perhaps our choice of a manageable problem is responsible for the degree of success reported in the companion paper.[21] We determined that for our ASR task it is advantageous to use a particular kind of hidden Markov model, which we call a left-to-right model. In such a model there are strong temporal constraints on the Markov chain. First, any state, once left, cannot be later revisited. Second, there is a final absorbing state in which all observation sequences are assumed to terminate. These restrictions on the sequences affect both parameter estimation and probability computation procedures. We show how the Baum-Welch algorithm, the Viterbi algorithm, and the classical methods can all be adapted for use with left-to-right models.

We conclude our presentation with several sample solutions to some artificial but nontrivial problems that illustrate concepts treated in the foregoing discussion.

## II. A REVIEW OF THE THEORY

A probabilistic function of a (hidden) Markov chain is a stochastic

process generated by two interrelated mechanisms, an underlying Markov chain having a finite number of states, and a set of random functions, one of which is associated with each state. At discrete instants of time, the process is assumed to be in some state and an observation is generated by the random function corresponding to the current state. The underlying Markov chain then changes states according to its transition probability matrix. The observer sees only the output of the random functions associated with each state and cannot directly observe the states of the underlying Markov chain; hence the term hidden Markov model.

In principle, the underlying Markov chain may be of any order and the outputs from its states may be multivariate random processes having some continuous joint probability density function. In this discussion, however, we shall restrict ourselves to consideration of Markov chains of order one, i.e., those for which the probability of transition to any state depends only upon that state and its predecessor. We shall also limit the discussion to processes whose observations are drawn from a discrete finite alphabet according to discrete probability distribution functions associated with the states.

It is quite natural to think of the speech signal as being generated by such a process. We can imagine the vocal tract as being in one of a finite number of articulatory configurations or states. In each state a short (in time) signal is produced that has one of a finite number of prototypical spectra depending, of course, on the state. Thus, the power spectra of short intervals of the speech signal are determined solely by the current state of the model, while the variation of the spectral composition of the signal with time is governed predominantly by the probabilistic state transition law of the underlying Markov chain. For speech signals derived from a small vocabulary of isolated words, the model is reasonably faithful. The foregoing is, of course, an oversimplification intended only for the purpose of motivating the following theoretical discussion.

Let us say that the underlying Markov chain has $N$ states $q_1, q_2, \cdots, q_N$ and the observations are drawn from an alphabet, $\mathbf{V}$, of $M$ prototypical spectra, $v_1, v_2, \cdots, v_M$. The underlying Markov chain can then be specified in terms of an initial state distribution vector $\pi' = (\pi_1, \pi_2, \cdots, \pi_N)$ and a state transition matrix, $A = [a_{ij}]$ $1 \leq i, j \leq N$. Here, $\pi_i$ is the probability of $q_i$ at some arbitrary time, $t = 0$, and $a_{ij}$ is the probability of transiting to state $q_j$ given current state, $q_i$, that is $a_{ij} = \text{prob}(q_j \text{ at } t + 1 | q_i \text{ at } t)$.

The random processes associated with the states can be collectively represented by another stochastic matrix $B = [b_{jk}]$ in which for $1 \leq j \leq N$ and $1 \leq k \leq M$, $b_{jk}$ is the probability of observing symbol $v_k$ given current state $q_j$. We denote this as $b_{jk} = \text{prob}(v_k \text{ at } t | q_j \text{ at } t)$. Thus a

hidden Markov model, $\mathbf{M}$, is identified with the parameter set $(\pi, A, B)$.

To use hidden Markov models to perform speech recognition we must solve two specific problems: observation sequence probability estimation, which will be used for classification of an utterance; and model parameter estimation, which will serve as a procedure for training models for each vocabulary word. Both problems proceed from a sequence, $\mathbf{O}$, of observations $O_1O_2 \cdots O_T$ where each $O_t$ for $1 \leq t \leq T$ is some $v_k \in \mathbf{V}$.

Our particular classification problem is as follows. We wish to recognize utterances known to be selected from some vocabulary, $W$, of words $w_1, w_2, \cdots, w_V$. We are given an observation sequence, $\mathbf{O}$, derived from the utterance of some unknown $w_i \in W$ and a set of $V$ models $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_V$. We must compute $P_i = \text{prob}(\mathbf{O}|\mathbf{M}_i)$ for $1 \leq i \leq V$. We will then classify the unknown utterance as $w_i$ iff $P_i \geq P_j$ for $1 \leq j \leq V$.

The training problem is simply that of determining the models $\mathbf{M}_i = (\pi_i, A_i, B_i)$ for $1 \leq i \leq V$ given training sequences $\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \cdots, \mathbf{O}^{(V)}$, where $\mathbf{O}^{(i)}$ is known to have been derived from an utterance of word $w_i$ for $1 \leq i \leq V$.

One could, in principle, compute $\text{prob}(\mathbf{O}|\mathbf{M})$ by computing the joint probability prob $(\mathbf{O}, \mathbf{s}|\mathbf{M})$ for each state sequence, s, of length $T$, and summing over all state sequences. Obviously this is computationally intractable. Fortunately, however, there is an efficient method for computing $P$. Let us define the function $\alpha_t(i)$ for $1 \leq t \leq T$ as $\text{prob}(O_1O_2 \cdots O_t$ and $q_i$ at $t|\mathbf{M})$. According to the definition $\alpha_1(i) = \pi_i b_i(O_1)$, where $b_i(O_1)$ is understood to mean $b_{ik}$ iff $O_1 \equiv v_k$; then we have the following recursive relationship for the "forward probabilities":

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(O_{t+1}) \qquad 1 \leq t \leq T-1. \tag{1}$$

Similarly, we define another function, $\beta_t(j) = \text{prob}(O_{t+1}O_{t+2} \cdots O_T|q_j$ at $t$ and $\mathbf{M})$. We set $\beta_T(j) = 1\ \forall\ j$ and then use the backward recursion

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(O_{t+1})\beta_{t+1}(j) \qquad T-1 \geq t \geq 1 \tag{2}$$

to compute the "backward probabilities."

The two functions can be used to compute $P$ according to

$$P = \text{Prob}(\mathbf{O}|\mathbf{M}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j) \tag{3}$$

for any $t$ such that $1 \leq t \leq T-1$. Equations (1) to (3) are from Baum[13] and are sometimes referred to as the "forward-backward" algorithm.

Setting $t = T - 1$ in (3) gives

$$P = \sum_{i=1}^{N} \alpha_T(i) \tag{4}$$

so that $P$ can be computed from the forward probabilities alone. A similar formula for $P$ can be obtained from the backward probabilities by setting $t = 1$. These and several other formulas in this section may be compactly written in matrix notation (see Appendix A). For instance,

$$P = \pi' B_1 A B_2 A \cdots A B_T \mathbf{1}, \tag{5}$$

where $\mathbf{1}$ is the $N$-vector $(1, 1, 1, \cdots, 1)'$ and

$$B_t = \begin{pmatrix} b_1(O_t) & & & \\ & b_2(O_t) & & \mathbf{O} \\ & & \ddots & \\ \mathbf{O} & & & b_N(O_t) \end{pmatrix} \tag{6}$$

for $1 \le t \le T$. From (5) it is clear that $P$ is a homogeneous polynomial in the $\pi_i$, $a_{ij}$, and $b_{jk}$. Any of eqs. (3) through (5) may be used to solve the classification problem. The forward and backward probabilities will prove to be convenient in other contexts.

When we compute $P$ with the forward-backward algorithm, we are including the probabilities of all possible state sequences that may have generated $\mathbf{O}$. Alternatively, we may define $P$ as the maximum over all state sequences $i = i_0, i_1, \cdots i_T$ of the joint probability $P(\mathbf{O}, i)$. This distinguished state sequence and the corresponding probability of the observation sequence can be simultaneously computed by means of the Viterbi[23] algorithm. This dynamic programming technique proceeds as follows: Let $\phi_1(i) = \pi_i b_i(O_1)$ for $1 \le i \le N$. Then we can perform the following recursion for $2 \le t \le T$ and $1 \le j \le N$

$$\phi_t(j) = \max_{1 \le i \le N} [\phi_{t-1}(i) a_{ij}] b_j(O_t) \tag{7a}$$

and

$$\psi_t(j) = i^*, \tag{7b}$$

where $i^*$ is a choice of an index $i$ that maximizes $\phi_{t-1}(i)$.

The result is that $P = \max_{1 \le i \le N} [\phi_T(i)]$. Also the maximum likelihood state sequence can be recovered from $\psi$ as follows. Let $q_T = i^*$, where $i^*$ maximizes $P$. Then for $T \ge t \ge 2$, $q_{t-1} = \psi_t(q_t)$. If one only wishes to compute $P$, the linked list, $\psi$, need not be maintained as in (7b). Only the recursion (7a) is required.

The problem of training a model, unfortunately, does not have such a simple solution. In fact, given any finite observation sequence as training data, we cannot optimally train the model. We can, however,

choose $\pi$, $A$, and $B$ such that prob($\mathbf{O}|\mathbf{M}$) is locally maximized. For an asymptotic analysis of the training problem the reader should consult Baum and Petrie.[15]

We can use the forward and backward probabilities to formulate a solution to the problem of training by parameter estimation. Given some estimates of the parameter values we can compute, for example, that the expected number of transitions, $\gamma_{ij}$, from $q_i$ to $q_j$, conditioned on the observation sequence is just

$$\gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \tag{8}$$

Then, the expected number of transitions, $\gamma_i$ out of $q_i$, given $\mathbf{O}$, is

$$\gamma_i = \sum_{j=1}^{N} \gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i), \tag{9}$$

the last step of which is based on (2). The ratio $\gamma_{ij}/\gamma_i$ is then an estimate of the probability of state $q_j$, given that the previous state was $q_i$. This ratio may be taken as a new estimate, $\bar{a}_{ij}$, of $a_{ij}$. That is,

$$\bar{a}_{ij} = \frac{\gamma_{ij}}{\gamma_i} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}. \tag{10}$$

Similarly, we can make a new estimate of $b_{jk}$ as the frequency of occurrence of $v_k$ in $q_j$ relative to the frequency of occurrence of any symbol in state $q_j$. Stated in terms of the forward and backward probabilities we have

$$\bar{b}_{jk} = \frac{\sum_{t \ni O_t = v_k} \alpha_t(j) \beta_t(j)}{\sum_{t=1}^{T} \alpha_t(j) \beta_t(j)}. \tag{11}$$

Finally, new values of the initial state probabilities may be obtained from

$$\bar{\pi}_i = \frac{1}{P} \alpha_1(i) \beta_1(i). \tag{12}$$

As we shall see in the next section, the reestimates are guaranteed to increase $P$, except at a critical point.

### 2.1 Proof of the reestimation formula

The reestimation formulas (10), (11), and (12) are instances of the Baum-Welch algorithm. Although it is not at all obvious, each application of the formulas is guaranteed to increase $P$ except if we are at a critical point of $P$, in which case the new estimates will be identical

to their current values. Several proofs of this rather surprising fact are given in the literature.[13,17] Because we shall need to modify it later to cope with the finite sample size problem, we shall briefly sketch Baum's proof[13] here. The proof is based on the following two lemmas:

*Lemma 1: Let* $u_i$, $i = 1, \cdots, S$ *be positive real numbers, and let* $v_i$, $i = 1, \cdots, S$ *be nonnegative real numbers such that* $\sum_i v_i > 0$. *Then from the concavity of the log function it follows that*

$$\ln\left(\frac{\sum v_i}{\sum u_i}\right) = \ln\left[\sum_i \left(\frac{u_i}{\sum_k u_k}\right) \cdot \frac{v_i}{u_i}\right]$$

$$\geq \sum_i \frac{u_i}{\sum_k u_k} \ln\left(\frac{v_i}{u_i}\right)$$

$$= \frac{1}{\sum_k u_k}\left[\sum_i (u_i \ln v_i - u_i \ln u_i)\right]. \tag{13}$$

Here every summation is from 1 to $S$.

*Lemma 2: If* $c_i > 0$ $i = 1, \cdots, N$, *then subject to the constraint* $\sum_i x_i = 1$, *the function*

$$F(\mathbf{x}) = \sum_i c_i \ln x_i \tag{14}$$

*attains its unique global maximum when*

$$x_i = \frac{c_i}{\sum_i c_i}. \tag{15}$$

The proof follows from the observation that by the Lagrange method

$$\frac{\partial}{\partial x_i}\left[F(\mathbf{x}) - \lambda \sum_i x_i\right] = \frac{c_i}{x_i} - \lambda = 0. \tag{16}$$

Multiplying by $x_i$ and summing over $i$ gives $\lambda = \sum_i c_i$, hence the result.

Now in Lemma 1, let $S$ be the number of state sequences of length $T$. For the $i$th sequence let $u_i$ be the joint probability

$$u_i = \text{Prob}[\text{state sequence } i, \text{ observation } \mathbf{O} \mid \text{model } \mathbf{M}]$$

$$= P(i, \mathbf{O} \mid \mathbf{M}).$$

Let $v_i$ be the same joint probability conditioned on model $\bar{\mathbf{M}}$. Then

$$\sum_i u_i = p(\mathbf{O} \mid \mathbf{M}) \triangleq P(\mathbf{M})$$

$$\sum_i v_i = p(\mathbf{O} \mid \bar{\mathbf{M}}) \triangleq P(\bar{\mathbf{M}}) \tag{17}$$

and the lemma gives

$$\ln \frac{P(\bar{\mathbf{M}})}{P(\mathbf{M})} \geq \frac{1}{P(\mathbf{M})} \cdot [Q(\mathbf{M}, \bar{\mathbf{M}}) - Q(\mathbf{M}, \mathbf{M})], \tag{18}$$

where

$$Q(\mathbf{M}, \bar{\mathbf{M}}) \triangleq \sum_i u_i \ln v_i. \tag{19}$$

Thus, if we can find a model $\bar{\mathbf{M}}$ that makes the right-hand side of (18) positive, we have a way of improving the model $\mathbf{M}$. Clearly, the largest guaranteed improvement by this method results for $\bar{\mathbf{M}}$, which maximizes $Q(\mathbf{M}, \bar{\mathbf{M}})$ [and hence maximizes the right-hand side of (18)]. The remarkable fact proven in Ref. 13 is that $Q(\mathbf{M}, \bar{\mathbf{M}})$ attains its maximum when $\bar{\mathbf{M}}$ is related to $\mathbf{M}$ by the reestimation formulas (10) through (12). To show this let the $s$th-state sequence be $s_0, s_1, \cdots, s_T$, and the given observation sequence be $O_{k_1}, \cdots, O_{k_T}$. Then

$$\ln v_s = \ln p(s, \mathbf{O}|\bar{\mathbf{M}}) = \ln \bar{\pi}_{s_0} + \sum_{t=0}^{T-1} \ln \bar{a}_{s_t s_{t+1}} + \sum_{t=0}^{T-1} \ln \bar{b}_{s_{t+1}}(O_{t+1}). \tag{20}$$

Substituting this in (19) for $Q(\mathbf{M}, \bar{\mathbf{M}})$, and regrouping terms in the summations according to state transitions and observed symbols, it can be seen that

$$Q(\mathbf{M}, \bar{\mathbf{M}}) = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} \ln \bar{a}_{ij} + \sum_{j=1}^{N} \sum_{k=1}^{M} d_{jk} \ln \bar{b}_j(k) + \sum_{i=1}^{N} e_i \ln \bar{\pi}_i. \tag{21}$$

Here

$$c_{ij} = \sum_{s=1}^{S} p(s, \mathbf{O}|\mathbf{M}) n_{ij}(s) \tag{22a}$$

$$d_{jk} = \sum_{s=1}^{S} p(s, \mathbf{O}|\mathbf{M}) m_{jk}(s) \tag{22b}$$

$$e_i = \sum_{s=1}^{S} p(s, \mathbf{O}|\mathbf{M}) r_i(s), \tag{22c}$$

and for the $s$th-state sequence

$n_{ij}(s)$ = number of transitions from state $q_i$ to $q_j$

$m_{jk}(s)$ = number of times symbol $k$ is generated in state $q_j$

$r_i(s)$ = 1 if initial state is $q_i$

= 0 otherwise.

Thus, $c_{ij}$, $d_{jk}$, and $e_i$ are the expected values of $n_{ij}$, $m_{jk}$, $r_i$, respectively, based on model $\mathbf{M}$.

The expression (21) is now a sum of $2N + 1$ independent expressions of the type maximized in Lemma 2. Hence, $Q(\mathbf{M}, \bar{\mathbf{M}})$ is maximized if

$$\bar{a}_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} \tag{23a}$$

$$\bar{b}_j(k) = \frac{d_{jk}}{\sum_k d_{jk}} \tag{23b}$$

$$\bar{\pi}_i = \frac{e_i}{\sum_i e_i}. \tag{23c}$$

These are recognized as the reestimation formulas.

### 2.2 Solution by optimization techniques

Lest the reader be led to believe that the reestimation formulas are peculiar to stochastic processes, we shall examine them briefly from several different points of view. Note that the reestimation formulas update the model in such a way that the constraints

$$\sum_{i=1}^{N} \pi_i = 1 \tag{24a}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{for} \quad 1 \le i \le N \tag{24b}$$

and

$$\sum_{k=1}^{M} b_{jk} = 1 \quad \text{for} \quad 1 \le j \le N \tag{24c}$$

are automatically satisfied at each iteration. The constraints are, of course, required to make the hidden Markov model well defined. It is thus natural to look at the training problem as a problem of constrained optimization of $P$ and, at least formally, solve it by the classical method of Lagrange multipliers. For simplicity, we shall restrict the discussion to optimization with respect to $A$. Let $Q$ be the Lagrangian of $P$ with respect to the constraints (24b). We see that

$$Q = P + \sum_{i=1}^{N} \lambda_i \left( \sum_{j=1}^{N} a_{ij} - 1 \right), \tag{25}$$

where the $\lambda_i$ are the as yet undetermined Lagrange multipliers.

At a critical point of $P$ on the interior of the manifold defined by (24a through c), it will be the case that for $1 \le i, j \le N$

$$\frac{\partial Q}{\partial a_{ij}} = \frac{\partial P}{\partial a_{ij}} + \lambda_i = 0. \tag{26}$$

Multiplying (26) by $a_{ij}$ and summing over $j$ we get

$$\sum_{j=1}^{N} a_{ij} \frac{\partial P}{\partial a_{ij}} = -\left[\sum_{j=1}^{N} a_{ij}\right] \lambda_i = -\lambda_i = \frac{\partial P}{\partial a_{ij}}, \tag{27}$$

where the right-hand side of (27) follows from substituting (24b) for the sum of $a_{ij}$ and then replacing $\lambda_i$ according to (26). From (27) it may be seen that $P$ is maximized when

$$a_{ij} = \frac{a_{ij} \dfrac{\partial P}{\partial a_{ij}}}{\displaystyle\sum_{k=1}^{N} a_{ik} \frac{\partial P}{\partial a_{ik}}}. \tag{28}$$

A similar argument can be made for the $\pi$ and $B$ parameters.

While it is true that solving (28) for $a_{ij}$ is analytically intractable, it can be used to provide some useful insights into the Baum-Welch reestimation formulas and alternatives to them for solving the training problem. Let us begin by computing $\partial P/\partial a_{ij}$ by differentiating (3), according to the formula for differentiating a product,

$$\frac{\partial P}{\partial a_{ij}} = \sum_{t=1}^{T-1} \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j). \tag{29}$$

Substituting the right-hand side of (29) for $\partial P/\partial a_{ij}$ in (28), we get

$$a_{ij} = \frac{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\displaystyle\sum_{j=1}^{N} \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}. \tag{30}$$

Then changing the order of summation in the denominator of (30) and substituting in the right-hand side of (2) we get

$$a_{ij} = \frac{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\displaystyle\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}. \tag{31}$$

The right-hand side of (31) is thus seen to be identical to that of the reestimation formula (10). Thus, at a critical point, the reestimation formula (10) solves the equations (31). Similarly, if we differentiate (3) with respect to $\pi_i$ and $b_{jk}$ we get

$$\begin{aligned} \frac{\partial P}{\partial \pi_i} &= \sum_{j=1}^{N} b_i(O_1) a_{ij} b_j(O_2) \beta_2(j) \\ &= b_i(O_1) \beta_1(i) \end{aligned} \tag{32}$$

and

$$\frac{\partial P}{\partial b_{jk}} = \sum_{t \ni O_t = v_k} \sum_{i=1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j) + \delta(O_1, v_k) \pi_j \beta_1(j), \tag{33}$$

respectively. In (33) $\delta$ is understood to be the Kronecker $\delta$ function.

By substituting (32) and (33) into their respective analogs of (28), we obtain the reestimation formulas (12) and (11), respectively, at a critical point. Thus it appears that the reestimation formulas may have more general applications than might appear from their statistical motivation.

Equation (28) suggests that we define a transformation, $\mathbf{T}$, of the parameter space onto itself as

$$\mathbf{T}(x)_{ij} = \frac{x_{ij} \dfrac{\partial P}{\partial x_{ij}}}{\sum_{k=1}^{N} x_{ik} \dfrac{\partial P}{\partial x_{ik}}}, \tag{34}$$

where $\mathbf{T}(\mathbf{x})_{ij}$ is understood to mean the $i, j$th coordinate of the image of $\mathbf{x}$ under $\mathbf{T}$. The parameter space is restricted to be the manifold such that $x_{ij} \geq 0$ for $1 \leq i, j \leq N$ and $\sum_{j=1}^{N} x_{ij} = 1$ for $1 \leq i \leq N$. Thus, the reestimation formulas (10), (11), and (12) are a special case of the transformation (34), with $P$ a particular homogeneous polynomial in the $x_{ij}$ having positive coefficients. Here the $x_{ij}$ include the $\pi_i$, the $a_{ij}$, and the $b_{jk}$. Baum and Eagon[14] have shown that for any such polynomial $P[\mathbf{T}(\mathbf{x})] > P(\mathbf{x})$ except if $\mathbf{x}$ is a critical point of $P$. Thus the transformation, $\mathbf{T}$, is appropriately called a growth transformation. The conditions under which $\mathbf{T}$ is a growth transformation were relaxed by Baum and Sell[17] to include all polynomials with positive coefficients. They further proved that $P$ increases monotonically on the segment from $\mathbf{x}$ to $\mathbf{T}(\mathbf{x})$. Specifically, they showed that $P[\eta \mathbf{T}(\mathbf{x}) + (1 - \eta)\mathbf{x}] \geq P(\mathbf{x})$ for $0 \leq \eta \leq 1$. Other properties of the transformation (34) have been explored by Passman[18] and Stebe.[19] There may be still less restrictive general criteria on $P$ for $\mathbf{T}$ to be a growth transformation.

We can give $\mathbf{T}(\mathbf{x})$ a simple geometric interpretation. For the purposes of this discussion we shall restrict ourselves to $\mathbf{x} \in \mathbf{R}^N$, $x_i \geq 0$ for $1 \leq i \leq N$, and the single constraint $G(\mathbf{x}) = \sum_{i=1}^{N} x_i - 1 = 0$. We do so without loss of generality, since constraints such as those of (24a, b, and c) are disjoint, i.e., no pair of constraints has any common variables. As shown in Fig. 1, given any $\mathbf{x}$ satisfying $G(\mathbf{x}) = 0$, $\mathbf{T}(\mathbf{x})$ is the intersection of the vector $\mathbf{X}$, or its extension, with the hyperplane $\sum_{i=1}^{N} x_i - 1 = 0$, where $\mathbf{X}$ has components $x_i \dfrac{\partial P}{\partial x_i}$ for $1 \leq \mathrm{i} \leq N$.

This may be shown by observing that a line in the direction of $\mathbf{X}$ passing through the origin has the equation $\mathbf{y} = r\mathbf{X}$, where $r$ is a non-

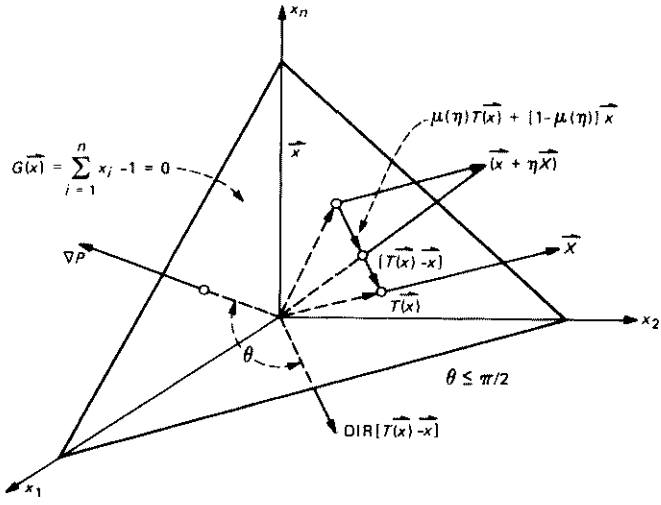



Fig. 1—Geometrical relationship of the quantities involved in the reestimation formulas.

negative scalar. Component-wise this is equivalent to

$$y_i = r x_i \frac{\partial P}{\partial x_i} \quad \text{for} \quad 1 \leq i \leq N. \tag{35}$$

We can find that $r$ for which $\mathbf{y}$ intersects the hyperplane $G(\mathbf{x}) = 0$ by summing over $i$. Thus

$$\sum_{i=1}^{N} y_i = r \sum_{i=1}^{N} x_i \frac{\partial P}{\partial x_i} = 1, \tag{36}$$

since $\mathbf{y}$ lies on the hyperplane $G(\mathbf{x}) = 0$. Rearranging (36) we have

$$r = \frac{1}{\sum_{i=1}^{N} x_i \frac{\partial P}{\partial x_i}} \tag{37}$$

and

$$y_i = \frac{x_i \frac{\partial P}{\partial x_i}}{\sum_{j=1}^{N} x_j \frac{\partial P}{\partial x_j}}. \tag{38}$$

Furthermore, as also shown in Fig. 1, the vector $[\mathbf{T}(\mathbf{x}) - \mathbf{x}]$ is the set of intersections of the vector $(\mathbf{x} + \eta\mathbf{X})$ with the hyperplane $G(\mathbf{x}) = 0$ for $0 \leq \eta \leq +\infty$ with $\mathbf{T}(\mathbf{x})$ corresponding to $\eta = +\infty$ and $\mathbf{x}$ to $\eta = 0$.

Finally, in view of the result of Baum and Sell, quoted above, the vector $\mathbf{T}(\mathbf{x}) - \mathbf{x}$ must also have a positive projection on $\nabla P$. This,

too, is easily seen. If $P$ is a polynomial with positive coefficients, then $\partial P/\partial x_i \geq 0$ for $1 \leq i \leq N$. From the definition of T it is clear that

$$\mathbf{T}(\mathbf{x})_i \geq x_i \quad iff \quad \frac{\partial P}{\partial x_i} \geq \sum_{j=1}^{N} x_j \frac{\partial P}{\partial x_j} = r, \tag{39}$$

where $r$ is some constant. Then it must be true that

$$\sum_{i=1}^{N} [\mathbf{T}(\mathbf{x})_i - x_i] \left( \frac{\partial P}{\partial x_i} - r \right) \geq 0 \tag{40}$$

since both factors in each summand are of the same sign. Rearranging (40) we have

$$\sum_{i=1}^{N} [\mathbf{T}(\mathbf{x})_i - x_i] \frac{\partial P}{\partial x_i} \geq r \sum_{i=1}^{N} [\mathbf{T}(\mathbf{x})_i - x_i] = 0. \tag{41}$$

The right-hand side is zero since $\sum_{i=1}^{N} \mathbf{T}(\mathbf{x})_i = \sum_{i=1}^{N} x_i = 1$. Thus $[\mathbf{T}(\mathbf{x}) - \mathbf{x}] \cdot \nabla P \geq 0$, proving that a step of the transformation has a positive projection along the gradient of $P$.

This merely guarantees that we can move an infinitesimal amount in the direction of $[\mathbf{T}(\mathbf{x}) - \mathbf{x}]$ while increasing $P$. The theorem of Baum and Eagon, however, guarantees much more, namely that we can take a finite step and be assured of increasing $P$. We may, in fact, be able to continue past $\mathbf{T}(\mathbf{x})$ while still increasing $P$. We are unable, at present, to give a geometrical interpretation of this fact.

While the reestimation formulas provide an elegant method for maximizing $P$, their success depends critically on the constraint set (24a, b, and c). As we will suggest later, in some cases there may be advantages in using classical optimization methods.

The principle of the classical methods is to search along the projection of $\nabla P$ on the constraint space, $G$, for a local maximum. The method of Rosen,[24] for example, uses only $\nabla P$ and a crude search strategy. The method of Davidon is one of many quasi-Newton techniques that uses the Fletcher-Powell[25] approximation to the inverse of the Hessian of $P$ and an exact line search with adaptive step size. There are many collections of general purpose subroutines for constrained optimization that can be used to solve the training problem. We have successfully used a version of the Davidon procedure from the Harwell Subroutine Library.[26] However, for the constraints that $\pi$, $A$, and $B$ be stochastic, the computation can be greatly simplified.

We illustrate this by outlining the gradient search algorithm for the case where $P$ is a function of the variables $x_1, \cdots, x_N$ subject to the constraints $x_i \geq 0$ for $1 \leq i \leq N$ and $\sum_{i=1}^{N} x_i - 1 = 0$. For convenience we will call the last constraint $G_1$, and the inequality constraints on $x_1, \cdots, x_N$ as $G_2, \cdots, G_{N+1}$, respectively.

An initial starting point $\mathbf{x}$ is chosen and the "active" constraints

identified. For our case $G_1$ is always active. For $i > 1$, $G_i$ is active if $x_{i-1} = 0$. Let $G_{n_j}$, $j = 1, \cdots, \ell$ be the active constraints (with $n_1 = 1$) at the initial point. Let $Q = P + \sum_{j=1}^{\ell} \lambda_j G_{n_j}$. Then according to the Kuhn-Tucker theorem,[27] the Lagrangian multipliers, $\lambda_j$, are determined by demanding that $\nabla Q$ be orthogonal to $\nabla G_{n_j}$ for $1 \le j \le \ell$. Now

$$\begin{aligned} \nabla Q &= \nabla P + \sum_{j=1}^{\ell} \lambda_j \nabla G_{n_j} \\ &= \nabla P + \Gamma\lambda, \end{aligned} \tag{42}$$

where $\Gamma$ is the $N \times \ell$ matrix with $\Gamma_{ij} = (\nabla G_{n_j})_i = \partial G_{n_j}/\partial x_i$, and $\lambda$ is the vector with components $\lambda_j$ for $j = 1, \cdots, \ell$. Thus the Kuhn-Tucker requirement is equivalent to

$$\Gamma'\nabla Q = 0 \tag{43}$$

or, from (42),

$$\lambda = -(\Gamma'\Gamma)^{-1}\Gamma'\nabla P. \tag{44}$$

For our special constraints we have

$$\Gamma_{i1} = 1 \quad \text{for} \quad 1 \le i \le N \tag{45}$$

and, for $j \neq 1$

$$\Gamma_{ij} = \begin{cases} 1 & \text{if} \quad i = n_j - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{46}$$

With $\Gamma$ defined this way

$$\Gamma'\Gamma = \begin{pmatrix} N & 1 & 1 & \cdots & 1 \\ 1 & 1 & & & \\ 1 & & 1 & \mathbf{O} & \\ \vdots & \mathbf{O} & & \ddots & \\ 1 & & & & 1 \end{pmatrix} \tag{47}$$

and $(\Gamma'\Gamma)^{-1}$ may be shown to be

$$(\Gamma'\Gamma)^{-1} = \frac{1}{N - \ell + 1} \begin{pmatrix} 1 & -1 & -1 & \cdots & -1 \\ -1 & N - \ell & 1 & \cdots & 1 \\ -1 & 1 & N - \ell & 1 & 1 \\ \vdots & & & \ddots & 1 \\ -1 & 1 & 1 & & N - \ell \end{pmatrix}. \tag{48}$$

Substituting (48) into (44) gives $\lambda$. When this $\lambda$ is substituted back into (42), it turns out that the resulting vector $\nabla Q$ can be computed by the following simple steps:

(*i*) Compute $\nabla P$ and let $S$ be the sum of all components of $\nabla P$ *except* $(\nabla P)_{n_{j-1}}$, $j = 2, \cdots, \ell$.

(*ii*) Then

$$(\nabla Q)_i = 0 \qquad i = n_j, \qquad j = 2, \cdots, \ell$$
$$= (\nabla P)_i - \frac{S}{N - \ell + 1} \qquad \text{otherwise.}$$

Finally, the values of $P$ are searched along the line

$$\mathbf{x}(\eta) = \mathbf{x} + \eta \frac{\nabla Q}{\|\nabla Q\|} \tag{49}$$

for a maximum with respect to $\eta$. The procedure is repeated at this new point.

In applying this technique to the actual training problem, there will be $2N + 1$ stochasticity constraints analogous to $G_1$ and a corresponding number of positivity constraints analogous to $G_2$, $G_3$, $\cdots$ $G_{N+1}$. In this case we have the option of treating all the parameters and their associated constraints together, or we may divide them into disjoint subsets and determine search directions for each subset independently.

Notice that this derivation does not require $P$ to be of any special form. This may prove to be an advantage since the Baum-Welch algorithm is not applicable to all $P$. Furthermore, the constraints may be changed. Although, as we shall see later, the Baum-Welch algorithm can be somewhat generalized in this respect, it does not generalize to work with arbitrary linear constraints.

## III. CONSIDERATIONS FOR IMPLEMENTATION

From the foregoing discussion it might appear that solutions to the problems of hidden Markov modeling can be obtained by straightforward translation of the relevant formulas into computer programs. Unfortunately, for all but the most trivial problems, the naive implementation will not succeed for two principal reasons. First, any of the methods of solution presented here for either the classification or the training problem require evaluation of $\alpha_t(i)$ and $\beta_t(i)$ for $1 \le t \le T$ and $1 \le i \le N$. From the recursive formulas for these quantities, (1) and (2), it is clear that as $T \to \infty$, $\alpha_T(i) \to 0$, and $\beta_1(i) \to 0$ in exponential fashion. In practice, the number of observations necessary to adequately train a model and/or compute its probability will result in underflow on any real computer if (1) and (2) are evaluated directly. Fortunately, there is a method for scaling these computations that not only solves the underflow problem but also greatly simplifies several other calculations.

The second problem is more serious, more subtle, and admits of a less gratifying, though still effective, solution. Baum and Petrie[15] have shown that the maximum likelihood estimates of the parameters of a hidden Markov process are consistent estimates (converge to the true

values as $T \rightarrow \infty$) of the parameters. The practical implication of the theorem is that, in training, one should use as many observations as possible which, as we have noted, make scaling necessary. In reality, of course, the observation sequence will always be finite. Then the following situation can arise. Suppose a given training sequence of length $T$ results in $b_{jk} = 0$. (It is, in fact, possible for a local maximum of $P$ to lie on a boundary of the parameter manifold.) Suppose further that we are subsequently asked to compute the probability that a new observation sequence was generated by our model. Even if the new sequence was actually generated by the model, it can be such that $\alpha_{t-1}(i)a_{ij}$ is nonzero for only one value of $j$ and that $O_t = v_k$, whence $\alpha_t(j) = 0$ and the probability of the observation then becomes zero. This phenomenon is fatal to a classification task; yet, the smaller $T$ is, the more likely is its occurrence. Jelinek and Mercer[9] have dealt with this problem in a slightly different context. Here, we offer the much simpler solution of constraining the parameter values so that $x_{ij} \geq \epsilon_{ij} > 0$.

Finally, in this section we discuss the related problem of model stability. Baum and Eagon[14] note that successive applications of the reestimation formulas converge to a connected component of the local maximum set of $P$. In case there are only a finite number of such extrema, the point of convergence is unique to within a renaming of the states. The component of the local maximum set to which the iteration converges as well as which of the $N!$ labelings of the states is determined by the initial estimates of the parameters. If we wish to average several models resulting from several different starting points to achieve model stability, we must be able to match the states of models whose states are permuted. We have devised a solution to this problem based on a minimum-weight bipartite matching algorithm.[28]

### 3.1 Scaling

The principle on which we base our scaling is to multiply $\alpha_t(i)$ by some scaling coefficient independent of $i$ so that it remains within the dynamic range of the computer for $1 \leq t \leq T$. We propose to perform a similar operation on $\beta_t(i)$ and then, at the end of the computation, remove the total effect of the scaling.

We illustrate the procedure for (10), the reestimation formula for the state transition probabilities. Let $\alpha_t(i)$ be computed according to (1) and then be multiplied by a scaling coefficient, $c_t$, where say,

$$c_t = \left[ \sum_{i=1}^{N} \alpha_t(i) \right]^{-1} \tag{50}$$

so that $\sum_{i=1}^{N} c_t\alpha_t(i) = 1$ for $1 \leq t \leq T$. Then, as we compute $\beta_t(i)$ from (2), we form the product $c_t\beta_t(i)$ for $T \geq t \geq 1$ and $1 \leq i \leq N$. In terms

of the scaled forward and backward probabilities, the right-hand side of (10) becomes

$$\frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) D_{t+1}}{\sum_{t=1}^{T-1} \sum_{\ell=1}^{N} C_t \alpha_t(i) a_{i\ell} b_\ell(O_{t+1}) \beta_{t+1}(\ell) D_{t+1}}, \tag{51}$$

where

$$C_t = \prod_{\tau=1}^{t} c_\tau \tag{52}$$

and

$$D_t = \prod_{\tau=t}^{T} c_\tau .$$

This results from the individual scale factors being multiplied together as we perform the recursions of (1) and (2).

Now note that each summand in both the numerator and the denominator has the coefficient $C_t D_{t+1} = \prod_{\tau=1}^{T} c_\tau$. These coefficients can be factored out and canceled so that (51) has the correct value $\bar{a}_{ij}$ as specified by (10). The reader can verify that this technique may be equally well applied to the reestimation formulas (11) and (12). It should also be obvious that, in practice, the scaling operation need not be performed at every observation time. One can use any scaling interval for which underflow does not occur. In this case, the scale factors corresponding to values of $t$ within any interval are set to unity.

While the above described scaling technique leaves the reestimation formulas invariant, (3) and (4) are still useless for computing $P$. However, $\log P$ can be recovered from the scale factors as follows. Assume that we compute $c_t$ according to (50) for $t = 1, 2, \cdots T$. Then

$$C_T \sum_{i=1}^{N} \alpha_T(i) = 1 \tag{53}$$

and from (53) it is obvious that $C_T = 1/P$. Thus, from (52) we have

$$\prod_{t=1}^{T} c_t = \frac{1}{P}. \tag{54}$$

The product of the individual scale factors cannot be evaluated but we can compute

$$\log P = -\sum_{t=1}^{T} \log c_t. \tag{55}$$

If one chooses to use the Viterbi algorithm for classification, then $\log P$ can be computed directly from $\pi$, $A$, and $B$ without regard for

the scale factors. Initially, we let $\phi_1(i) = \log[\pi_i b_i(O_1)]$ and then modify (7a) so that

$$\phi_t(j) = \max_{1 \le i \le N} [\phi_{t-1}(i) + \log a_{ij}] + \log[b_j(O_t)]. \tag{56}$$

In this case $\log P = \max_{1 \le i \le N} [\phi_T(i)]$.

If the parameters of the model are to be computed by means of classical optimization techniques, we can make the computation better conditioned numerically by maximizing $\log P$ rather than $P$. The scaling method of (50) makes this straightforward.

First note that if we are to maximize $\log P$, then we will need the partial derivatives of $\log P$ with respect to the parameters of the model. So, for example, we will need

$$\frac{\partial}{\partial a_{ij}} (\log P) = \frac{1}{P} \frac{\partial P}{\partial a_{ij}} = C_T \frac{\partial P}{\partial a_{ij}}. \tag{57}$$

Substituting the right-hand side of (29) for $\partial P / \partial a_{ij}$ in the right-hand side of (57) yields

$$\begin{aligned} \frac{\partial}{\partial a_{ij}} (\log P) &= C_T \sum_{t=1}^{T-1} \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) \\ &= \sum_{t=1}^{T-1} C_t \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) D_{t+1} \\ &= \sum_{t=1}^{T-1} \left( \prod_{\tau=1}^{t} c_\tau \right) \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) \left( \prod_{\tau=t+1}^{T} c_\tau \right). \end{aligned} \tag{58}$$

So that if we evaluate (29) formally, using not the true values of the forward and backward probabilities but the scaled values, then we will have the correct value of the partial derivatives of $\log P$ with respect to the transition probabilities. A similar argument can be made for the other parameters of the model and, thus, the scaling method of (50) provides a means for the direct evaluation of $\nabla(\log P)$, which is required for the classical optimization algorithms. Later we shall see that the combination of maximizing $\log P$ and this scaling technique simplifies the solution of the left-to-right Markov modeling problem as well.

### *3.2 Finite training sets*

We now turn our attention to solving the problems created by finite-training-set size. As we noted earlier, the effect of this problem is that observation sequences generated by a putative model will have zero probability conditioned on the model parameters. Since the cause of the difficulty is the assignment of zero to some parameters, usually

one or more symbol probabilities, it is reasonable to try to solve the problem by constraining the parameters to be positive.

We can maximize $P$ subject to the new constraints $a_{ij} \geq \epsilon > 0$; $b_{jk} \geq \epsilon > 0$, most easily using the classical methods. In fact, the algorithm described earlier based on the Kuhn-Tucker theorem is unchanged except that the procedure for determining the active constraints is based on $\epsilon$ rather than zero.

While the Lagrangian methods are perfectly adequate, it is also possible to build the new constraints into the Baum-Welch algorithm. We can show how this is done by making a slight modification to the proof of the algorithm given earlier (Section 2.1). Recall that the proof of the Baum-Welch algorithm was based on maximization of $2N + 1$ expressions of the type maximized in Lemma 2, eq. (14). Since these expressions involve disjoint sets of variables chosen from $A$, $B$, $\pi$, it suffices to consider any one of the maximizations. In fact, it suffices to show how Lemma 2 gets modified. Thus we wish now to maximize

$$F(\mathbf{x}) = \sum_i c_i \ln x_i \tag{59}$$

subject to the constraints

$$\sum_i x_i = 1 \tag{60a}$$

and

$$x_i \geq \epsilon, \qquad i = 1, \cdots N. \tag{60b}$$

(From the following discussion it will be obvious that a trivial generalization allows $\epsilon$ to depend on $i$.)

Now without the inequality constraints (60b), Lemma 2 showed that $F(\mathbf{x})$ attains its unique global maximum when $x_i = c_i / \sum_i c_i$. Suppose now that this global maximum occurs outside the region specified by the inequality constraints (60b). Specifically, let

$$\bar{x}_i = \frac{c_i}{\sum_{j=1}^{N} c_j} \geq \epsilon \quad \text{for} \quad i = 1, \cdots, N - \ell \tag{61a}$$

$$< \epsilon \quad \text{for} \quad i = N - \ell + 1, \cdots, N. \tag{61b}$$

From the concavity of $F(\mathbf{x})$ it follows that the maximum, subject to the inequality constraints, must occur somewhere on the boundary specified by the violated constraints (61b). Now it is easily shown that if $\bar{x}_i$ for some $i > N - \ell$ is replaced by $\epsilon$, then the global maximum over the rest of the variables occurs at values *lower* than those given above. From this we conclude that we must set

$$\bar{x}_i = \epsilon \quad \text{for} \quad i > N - \ell \tag{62}$$

and maximize

$$\tilde{F}(\mathbf{x}) = \sum_{i=1}^{N-\ell} c_i \ln x_i \tag{63}$$

subject to the constraint $\sum_{i=1}^{N-\ell} x_i = 1 - \ell\epsilon$. But this, analogously to Lemma 2, occurs when

$$\bar{x}_i = (1 - \ell\epsilon) \frac{c_i}{\sum_{j=1}^{N-\ell} c_j} \qquad i \leq N - \ell. \tag{64}$$

If these new values of $\bar{x}_i$ satisfy the constraints, we are done. If one or more become lower than $\epsilon$, they too must be set equal to $\epsilon$, and $\ell$ augmented appropriately.

Thus the modified Baum-Welch algorithm is as follows. Suppose we wish to constrain $b_{jk} \geq \epsilon$ for $1 \leq j \leq N$ and $1 \leq k \leq M$. We first evaluate $B$ using the reestimation formulas. Assume that some set of the parameters in the $j$th row of $B$ violates the constraint so that $b_{jk_i} < \epsilon$ for $1 \leq i \leq \ell$. Then set $\bar{b}_{jk_i} = \epsilon$ for $1 \leq i \leq \ell$ and readjust the remaining parameters according to (64) so that

$$\bar{b}_{jk} = (1 - \ell\epsilon) \frac{b_{jk}}{\sum_{i=1}^{N-\ell} b_{ji}} \qquad \forall k \notin \{k_i \mid 1 \leq i \leq \ell\}. \tag{65}$$

After performing the operation of (65) for each row of $B$, the resulting $\bar{B}$ is the optimal update with respect to the desired constraints. The method can be extended to include the state transition matrix if so desired. There is no advantage to treating $\pi$ in the same manner since, for any single observation sequence, $\bar{\pi}$ will always be a unit vector with exactly one nonzero component. In any case, (65) may be applied at each iteration of the reestimation formulas, or once as a post-processing stage after the Baum-Welch algorithm has converged.

### *3.3 Combining models*

The final implementational issue that we shall consider in this section is that of combining models for improved stability. There are several circumstances under which it may be desirable to combine several models into one. In spectral estimation, for example, to compute a long-term average spectrum of a stationary signal, it may be convenient to average a number of spectra computed over shorter intervals. It seems quite natural to apply similar block-processing techniques to the Markov modeling problem if the source is assumed to be ergodic. We may, for example, divide a long sequence of observations into contiguous subsequences, estimate model parameters for each subsequence, and combine the results.

Whether or not the source is ergodic, we may still attempt to increase the robustness of our model by averaging the parameter estimates derived from multiple initial values and/or independent observation sequences.

In any case, the difficulty that will be encountered is that even if there are finitely many isolated local maxima of $P$, they are only unique to within a renaming of the states. For two different observation sequences, $q_i$ and $q_j$ may be topologically equivalent, but $i \neq j$. We might try to avoid this problem by using the final parameter values for one observation sequence as the initial values for the next in hopes that this will restrict the search to a neighborhood of a single local maximum. This method, unfortunately, is not reliable. A better approach is that of finding a renaming of the states that minimizes, in some sense, the difference between two models.

Suppose $\mathbf{b}_j$ and $\bar{\mathbf{b}}_j$, $1 \leq j \leq N$ are, respectively, the rows of two estimates of $B$. Let $p(j)$ be a permutation of the state index, $j$, and let $d(\cdot,\cdot)$ be some distance metric. Then we seek the permutation, $p$, of $(q_1, q_2, \cdots, q_N)$ such that

$$D = \sum_{j=1}^{N} d[\mathbf{b}_j, \bar{\mathbf{b}}_{p(j)}] \tag{66}$$

is minimized. The naive solution is to try all possible $N!$ permutations and select the best one in the sense of (66). However, for $N > 10$ the computation becomes intractable. The problem can be brought within reach, however, by transforming it into a minimum-weight bipartite-graph-matching problem on $2N$ vertices. In the literature on combinatorial optimization (see, e.g., Ref. 28), several algorithms are available for accomplishing such a match in a number of operations that grow as $N^3$. In Appendix B, we describe one such algorithm based on an outline provided to us by R. E. Tarjan.

## IV. LEFT-TO-RIGHT HIDDEN MARKOV MODELS

For the purposes of isolated word recognition, it is useful to consider a special class of absorbing Markov chains that leads to what we call left-to-right models. These models have the following properties:

(*i*) The first observation is produced while the Markov chain is in a distinguished state called the starting state, designated $q_1$.

(*ii*) The last observation is generated while the Markov chain is in a distinguished state called the final or absorbing state, designated $q_N$.

(*iii*) Once the Markov chain leaves a state, that state cannot be revisited at a later time.

The simplest form of a left-to-right model is shown in Fig. 2, from which the origin of the term left-to-right becomes clear.

In this section we shall consider two problems associated with these

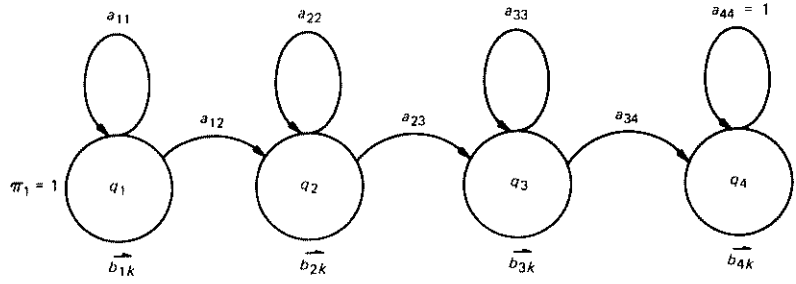



Fig. 2—The simplest form of left-to-right model.

special hidden Markov models. Note that a single, long-observation sequence is useless for training such models, because once the state $q_N$ is reached, the rest of the sequence provides no further information about earlier states. The appropriate training data for such a model are a *set* of observation sequences obtained by several starts in state $q_1$. In the case of isolated word recognition, for instance, several independent utterances of the same word provide such a set. We wish, therefore to modify the training algorithm to handle such training data. We also wish to compute the probability that a single given observation sequence, $O_1$, $O_2$ $\cdots$, $O_T$, was produced by the model, with the assumption that $O_1$ was produced in state $q_1$ and $O_T$ in state $q_N$. The three conditions mentioned above can be satisfied as follows:

Condition $(i)$ will be satisfied if we set $\pi = (1, 0, \cdots, 0)$ and do not reestimate it. Condition $(ii)$ can be imposed by setting

$$\beta_T(j) = \begin{cases} 1 & \text{for } j = N \\ 0 & \text{otherwise.} \end{cases} \tag{67}$$

Condition $(iii)$ can be guaranteed in the Baum-Welch algorithm by initially setting $a_{ij} = 0$ for $j < i$ (and in fact for any other combination of indices that specify transitions to be disallowed). It is clear from (34) that any parameter once set to zero will remain zero. For the gradient methods the appropriate $a_{ij}$'s are just set to zero and only the remaining parameters are adjusted.

The modification of the training procedure is as follows: Let us denote by $\mathbf{O} \equiv [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \cdots \mathbf{O}^{(K)}]$ the set of observation sequences, where $\mathbf{O}^{(k)} = O_1^{(k)} O_2^{(k)} \cdots O_{T_k}^{(k)}$ is the $k$th sequence. We treat the observation sequences as independent of each other and then we adjust the parameters of the model $\mathbf{M}$ to maximize

$$P = \prod_{k=1}^{K} \text{Prob}(\mathbf{O}^{(k)} | \mathbf{M}).$$
$$= \prod_{k=1}^{K} P_k. \tag{68}$$

Since the Baum-Welch algorithm computes the frequency of occurrence of various events, all we need to do is to compute these frequencies of occurrence in each sequence separately and add them together. Thus the new reestimation formulas may be written as

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \tag{69}$$

and

$$\bar{b}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t, \ni O_t(k)=v_j} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)}. \tag{70}$$

As noted above, $\pi$ is not reestimated.

Scaling these computations requires some care since the scale factors for each individual set of forward and backward probabilities will be different. One way of circumventing the problem is to remove the scale factors from each summand before adding. We can accomplish this by returning the $1/P$ factor [which appears in (8) and (9) and was cancelled to obtain (10)] to the reestimation formula. Using the reestimation formula for the transition probabilities as an example, (69) becomes

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}. \tag{71}$$

If the right-hand side of (71) is evaluated using the scaled values of the forward and backward probabilities, then each term in the inner summation will be scaled by $C_t^k D_{t+1}^k$, which will then be cancelled by the same factor which multiplies $P_k$. Thus, using the scaled values in computing (69) results in an unscaled $\bar{a}_{ij}$. The procedure is easily extended to computation of the symbol probabilities. Also note that for the purposes of classification only one subsequence is to be considered so that either (55) or (56) may be used unaltered to compute $P$.

To apply Lagrangian techniques to left-to-right models we note that upon taking logarithms of (68) we have

$$\log P = \sum_{k=1}^{K} \log P_k. \tag{72}$$

The derivatives needed to maximize $\log P$ in (72) can be obtained by evaluating expressions for the derivatives of each individual subse-

quence and summing. For example, for $a_{ij}$ we have [cf. (57) and (58)]

$$\frac{\partial}{\partial a_{ij}}(\log P) = \sum_{k=1}^{K} \frac{\partial}{\partial a_{ij}}(\log P) = C_T^k \sum_{t=1}^{T_k-1} \alpha_t^k(i) b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j). \quad (73)$$

As in all previous cases an analogous formula may be derived for the other parameters.

In practice, $A$ and $B$ for left-to-right models are especially sparse. Some of the zero values are so by design but others are dependent on $\mathbf{O}$. Parameters of this type will be found one at a time by standard line search strategies. We have found that the convergence of the Lagrangian techniques can be substantially accelerated by taking large enough steps so that several positivity constraints become binding. The corresponding variables are then clamped and (65) is applied before beginning the next iteration.

## V. NUMERICAL EXAMPLES

In this section, we give some instructive examples of the behavior of several of the algorithms discussed above. The algorithms were all coded in FORTRAN 77 on a Data General MV-8000, which uses a 32-bit floating point word. The data used in the tests came from either a Monte Carlo simulation of a hidden Markov chain or from a portion of a newspaper text that was edited to include only the 26 characters of the English alphabet and a special character denoting an interword space. The simulations have the valuable property that the model is known a priori, so that simple models may be used for checking program correctness while the more complicated ones can elicit some subtle and important numerical and methodological characteristics of the algorithms.

In our experiments we used the following procedure to generate observation sequences by means of a random number generator whose output is uniform on [0, 1] and specified values of $\pi$, $A$, $B$, $T$, and $V$:

(*i*) Partition the unit interval proportionally to the components of $\pi$. Generate a random number and select a start state, $q_i$, according to the subinterval in which the number falls. Set $t = 1$.

(*ii*) Partition the unit interval proportionally to the components of the $i$th row of $B$. Generate a random number and select a symbol, $v_k$, according to the subinterval in which the number falls. Set $O_t = v_k$.

(*iii*) Partition the unit interval proportionally to the components of the $i$th row of $A$. Generate a random number and select the next state, $q_j$, according to the subinterval in which the number falls.

(*iv*) Increment $t$. If $t \leq T$ set $q_i = q_j$ and repeat (*ii*) through (*iv*); otherwise stop.

Using this observation generator, several two- and three-state Markov models were simulated. These simulations were used to verify that

the parameter estimation algorithms were working correctly and to study the effects of the scaling interval on the accuracy of the algorithms. In this study we found that all scaling intervals that were sufficiently short to prevent underflow yielded numerically identical results. Thus one can, at one extreme, scale the forward and backward probabilities after each observation or, at the other, wait until a threshold signaling that underflow is imminent is exceeded and only then perform the scaling operation.

We next proceeded to study a pair of four-state ($N = 4$), four-symbol ($M = 4$) models shown below and referred to as SRC44 and SRC45.

SRC44:

$$A = \begin{bmatrix} 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

$$\pi = [0.25 \quad 0.25 \quad 0.25 \quad 0.25]$$

and

SRC45:

$$A = \begin{bmatrix} 0 & 0 & 0.25 & 0.75 \\ 0.15 & 0 & 0 & 0.85 \\ 0.2 & 0.8 & 0 & 0 \\ 0 & 0.22 & 0.78 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0.25 & 0.75 & 0 & 0 \\ 0 & 0.15 & 0.85 & 0 \\ 0 & 0 & 0.1 & 0.9 \\ 0.2 & 0 & 0 & 0.8 \end{bmatrix}$$

$$\pi = [0.25 \quad 0.25 \quad 0.25 \quad 0.25].$$

The state transition diagrams for these models are shown in Fig. 3.

Model SRC44 is a balanced model in the sense that all permissible transitions and symbols are equally likely, whereas model SRC45 is skewed in that it distinctly favors some transitions and symbols over others.

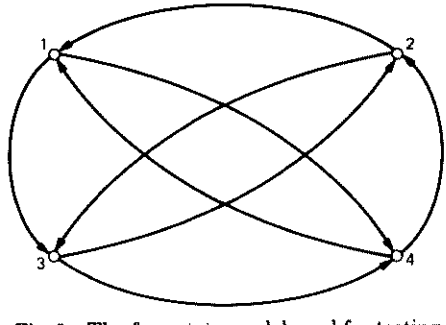



Fig. 3—The four-state model used for testing.

For each of these sources we processed observation sequences ranging in length from 100 to 4000 with the Baum-Welch algorithm. Initial values for $A$, $B$, and $\pi$ were chosen at random and the algorithm was terminated in one of two ways; either when the change in log $P$ from one iteration to the next fell below an arbitrary threshold, or when the number of iterations exceeded a specified maximum value. The maximum number of iterations was varied from 100 to 1000. For each estimate, $\hat{B}$, of the source matrix $B$, a measure of estimation error

$$\|\hat{B} - B\| = \left\{ \frac{1}{M \cdot N} \sum_{j=1}^{N} \sum_{k=1}^{M} \left[ \hat{b}_{jk} - b_{p(j)k} \right]^2 \right\}^{1/2} \tag{74}$$

was computed, where $p(j)$ is the state permutation that minimizes the

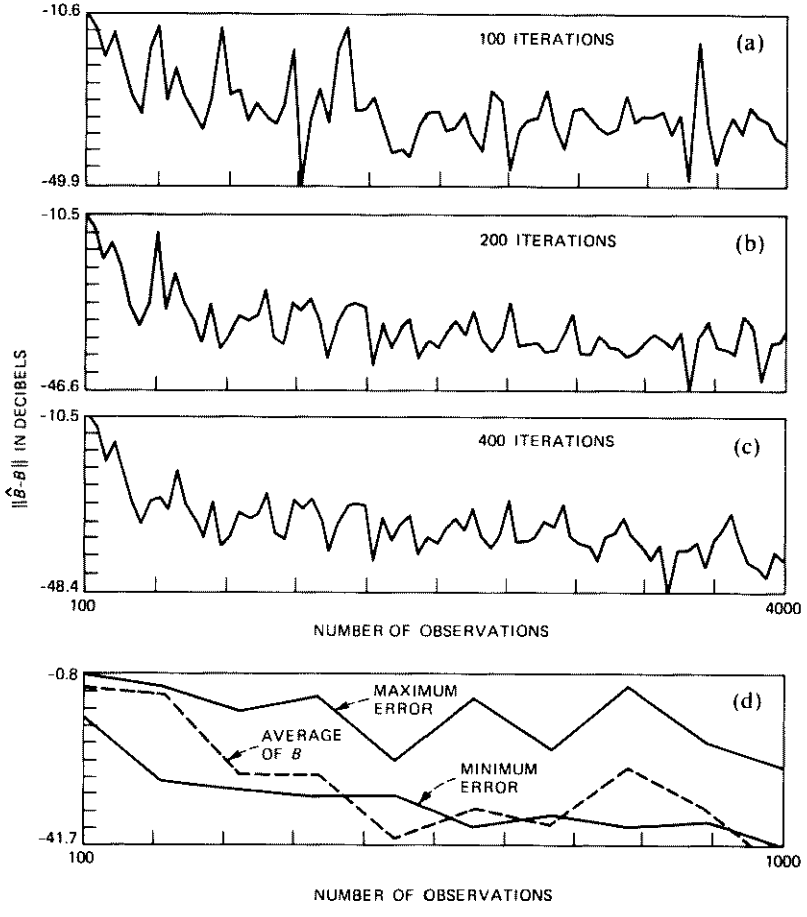



Fig. 4—Estimation error as a function of number of observations for source SRC44 for: (a) 100 iterations maximum, (b) 200 iterations, (c) 400 iterations, and (d) 10 random initial starts with 200 iterations maximum.

estimation error. The technique of minimum bipartite matching (see Appendix B) was used to determine the optimum state permutation.

Plots of the quantity $\|\hat{B} - \mathrm{B}\|$ (on a log scale) versus $T$, the number of observations, are given in Fig. 4 for source SRC44. Separate results are shown for 100 iterations maximum (part a) of the $BW$ reestimation procedure, 200 iterations (part b), and 400 iterations (part c). Also shown in Fig. 4d are the results of using 10 random initial starts with 200 iterations maximum. Shown in Fig. 4d are the maximum and minimum estimation errors for each $T$ and the estimation error for the average of all 10 $B$ matrices. (The reader should note that $T$ goes to 4000 in parts a through c, but only to 1000 in part d of Fig. 4.)

The curves given in Fig. 4 show several very interesting properties of the reestimation procedure. First we see that as the number of observations increases, a slow decrease in the average estimation error is obtained. However, we can see that statistical fluctuations (owing to different initial guesses for the model parameters) are often of larger magnitude than the slowly decreasing components of the curve. As the maximum number of iterations increases, the magnitude of the statistical fluctuations decreases, especially for larger values of $T$.

The curves of Fig. 4d show that although there is a wide range in the value of the estimation error for multiple starting choices, averaging the $B$ matrices (after appropriate state alignment) leads to estimation errors comparable with the best single estimates.

Figure 5 shows a similar set of results for the Markov source SRC45. Figure 5a shows a curve of estimation error versus number of observations for a maximum of 400 iterations, Fig. 5b shows the same curve with a maximum of 1000 iterations, Fig. 5c shows the curve when the initial estimates of both $A$ and $B$ are set to the source values exactly, and Fig. 5d shows maximum and minimum estimation errors for 10 random starting points.

Although the general trends of the data in Fig. 5 are similar to those of Fig. 4, there are several key differences. From Fig. 5b it can be seen that even for 1000 iterations, the variation in model estimates is enormous (36-dB variations). This result suggests that it is significantly more difficult to estimate parameters of a skewed Markov model than those of a fairly uniform model. The curves of Fig. 5c, in which the initial conditions were set to the source generator values, show that extremely good solutions could be obtained if the reestimation procedure could start in the neighborhood of the "exact" solution. Obviously this situation (i.e., starting near the correct parameter values) is not enforceable for real data.

The curves of Fig. 5d, in which multiple estimates of the Markov model are averaged, show that averaging the individual parameter estimates does not lead to a low error estimate for SRC45. This is

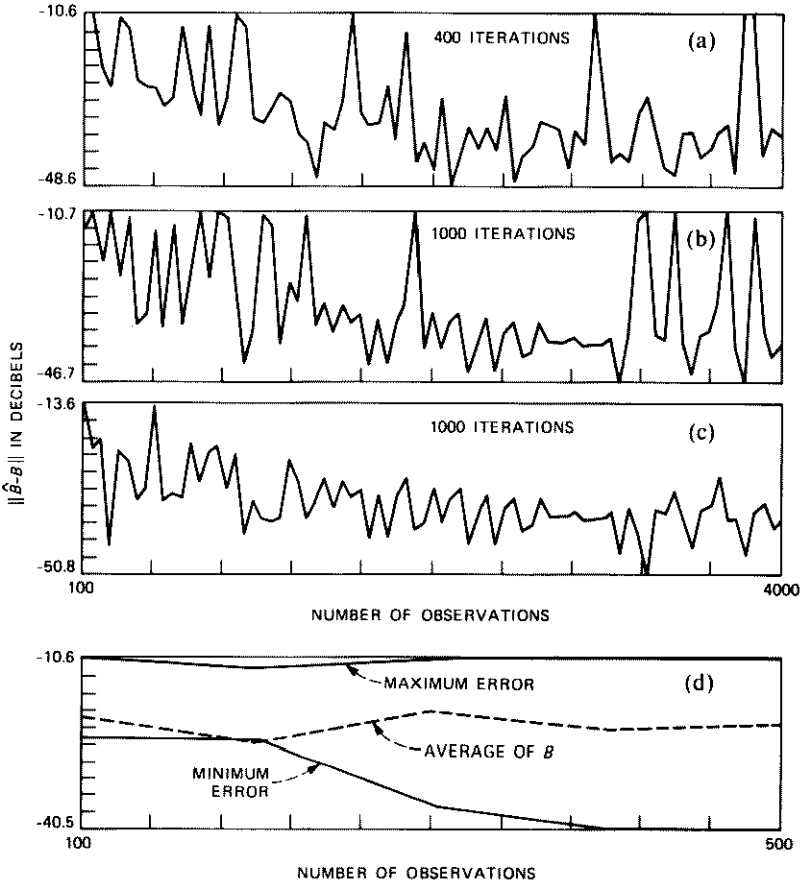



Fig. 5—Estimation error as a function of number of observations for source SRC45 for: (a) 400 iterations maximum, (b) 1000 iterations maximum, (c) initial estimates of $A$ and $B$ set to source values, and (d) maximum and minimum estimation errors for 10 random starting points.

undoubtedly because of the parameter estimates with high errors that occur and which have an undue influence on the average.

### *5.1 Left-to-right Markov source estimation*

The second series of experiments dealt with the left-to-right Markov models, as would be appropriate for our intended application to isolated word recognition. Figure 6 shows four such models. For each of these models (denoted as SRC195, SRC295, SRC395, and SRC495 in Fig. 6), the specifications were:

$$N = 5, M = 9, \pi = \{1, 0, 0, 0, 0\}$$

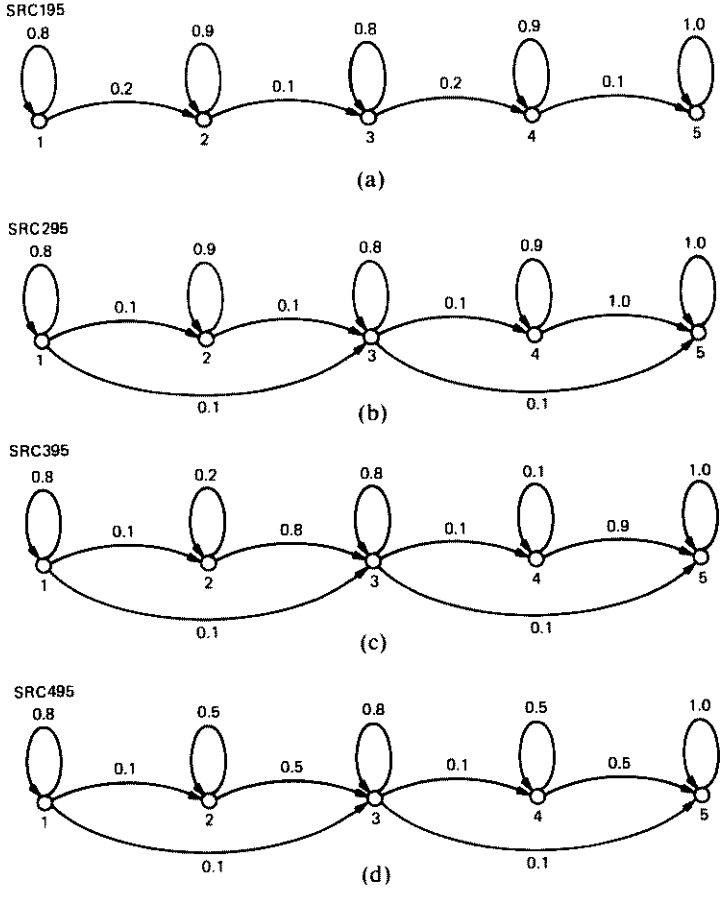



Fig. 6—Left-to-right models used for testing for: (a) SRC195, (b) SRC295, (c) SRC395, and (d) SRC495.

and

$$B = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \end{bmatrix}.$$

The state transition probabilities were those shown in Fig. 6. The SRC195 model is a left-to-right model. The SRC295 model allows a transition between states 1 and 3 and states 3 and 5, as well as transitions between sequentially numbered states. Both the SRC395 and SRC495 models include states whose self-transition probabilities

($a_{ii}$) are very small. We will see below that the average occupancy of such states is only 1 to 2 observations. For these non-ergodic models the concept of occupancy of the transient (i.e., non-absorbing) states is important.

If we denote the probability of a transition from a state to itself as $p$, then the probability of a transition out of that state at time $T + 1$ (assuming the state was entered at time $t = 0$) is

$$\text{Prob}(q_i \text{ at } t = 0, 1, 2, \cdots, T \quad \text{and} \quad q_j \neq q_i \text{ at } T + 1) = p^T(1 - p)$$

for models of the form of SRC195. Hence the average occupancy of a state is given by

$$\bar{d} = \sum_{t=1}^{\infty} (t + 1)p^t(1 - p)$$

$$= \frac{1}{1 - p}. \tag{75}$$

For $p = 0.9$ we get $\bar{d} = 10$, for $p = 0.8$ we get $\bar{d} = 5$, for $p = 0.5$ we get $\bar{d} = 2$, and for $p = 0.1$ we get $\bar{d} = 1.1$. Standard formulas are available for computing average state occupancies for arbitrary transition matrices. We will not consider them here; however, it is clear that states 2 and 4 in models SRC395 and SRC495 are of low occupancy.

To test the reestimation procedure on the Markov sources of Fig. 6, a set of $K$ sequences were generated for each model, where $K$ was the set (10, 25, 50, 100). Each sequence was generated using the Markov sequence generator described earlier, modified slightly to ensure that each sequence terminated in state 5 (the final state) and stayed there for 5 observations. The sequences were, however, of variable duration, depending on the exact sequence of state transitions that occurred.

The results showed that for sequences generated from model SRC195, the correct model parameters (to within small estimation errors) were obtained for all values of $K$ from 10 to 100. For sequences generated from model SRC295 only the 10 observation training sequence yielded grossly incorrect model parameter estimates. All other sequences ($K$ = 25, 50, 100) yielded the correct parameter values.

For both source models SRC395 and SRC495, however, no completely correct parameter estimates were obtained. In particular, the states whose expected occupancy was small (i.e., states 2 and 4 in both models) were merged with either the preceding or the following state (or both), while other states whose expected occupancy was larger were often split into 2 states, each with the same set of output symbols. These experiments indicate that it is difficult to reliably estimate parameters of a state, in a left-to-right model, whose average occupancy is very much smaller than that of the states to which it is connected.

### 5.2 Tests on a non-Markov source

In the experiments described above, the observations were in fact generated by a known probabilistic function of a Markov chain. A more difficult test of these techniques was performed in which the observations were 5000 characters from a newspaper article edited to contain only the letters of the alphabet and spaces. We used the observation sequence to train a four-state, 27-symbol model. Of course, the "true" underlying model is not known, as it was in the tests previously described. Nor is it likely that the four-state model is complex enough to model the richness of structure of written English. Even if it were, it is unlikely that 5000 characters is sufficient to capture the structure. Unfortunately, these are exactly the limitations with which the experimenter will be faced in trying to model "real" processes. Our hope was that the text analysis problem would reveal some of the ambiguities that will be encountered in making hidden Markov models of natural phenomena.

The text was first analyzed using the Baum-Welch reestimation formulas with a randomly chosen starting point. The algorithm converged in 310 iterations with $\log P = -1.317 \times 10^4$. For purposes of comparison, we analyzed the same data with a quasi-Newton optimization routine, VE01A, from the Harwell Library.[26] It required 125 iterations to obtain a maximum value of $\log P$ of $-1.356 \times 10^4$. In this case some care was required with the parameter values. The finite precision arithmetic occasionally results in a parameter value of $-10^{-7}$, which appears to satisfy the positivity constraints. Such a value is fatal to the computation of $\log P$ since it will result in an attempt to take a logarithm of a negative quantity. Fortunately, this condition is readily detected in the scaling routine and corrected by setting the offending parameter to zero.

Finally, we applied the Lagrangian technique described earlier to the same observation sequence but with a different set of initial parameter values. After 136 iterations, a still different model with $\log P = -1.327 \times 10^4$ is obtained. These results illustrate some important features of hidden Markov modeling. The computational methods used to obtain the models are roughly equivalent. All of the resulting models capture some of the structure of the data being analyzed. There are many different possible models with very little evidence for selecting the "best" one. For even very simple models, the likelihood function is too complicated to attribute the selection of one model or another by one algorithm or another to its properties. Finally, we note that all of the algorithms tested make large improvements to $P$ during the early iterations and only slight incremental improvements later. In fact, the last half of the iterations provides no significant change to the model. We have used a convergence criterion

of $10^{-7}$ on the relative increment between iterations. This may be relaxed substantially, resulting in many fewer iterations with no attendant degradation in the model.

## VI. CONCLUSION

We have presented some of the salient theoretical and practical issues associated with modeling data by probabilistic functions of a Markov chain. In our presentation we have concentrated on three issues: alternatives to the Baum-Welch reestimation algorithm, critical facets of implementation, and behavior of Markov models on certain artificial but realistic signals.

We have observed that, while most of the discussion of parameter estimation for Markov models in the open literature is devoted to the Baum-Welch algorithm, classical optimization techniques are not only a viable alternative but may even be preferable in some cases. In particular, classical techniques are virtually unrestricted by the forms of either the likelihood function or the constraints. The reestimation formulas may be growth transformations for a wider class of functions and constraints than has heretofore been proven; however, it is not likely that a universal reestimation formula exists. For applications to continuous density functions (c.f. Liporace[29]), the classical techniques may have still other advantages.

The open literature has provided only a perfunctory, if any, discussion of some crucial numerical and implementational problems associated with Markov modeling. We have given details of methods of dealing with floating point loss of significance, finite-training-set size, and model stability. Wherever possible we have made our techniques formal and algorithmic.

Finally, we have given several examples of the behavior of Markov modeling techniques on some reasonably realistic data. The most important lesson that can be drawn from these experiments is that even under ideal conditions (i.e., when the data are associated with a known hidden Markov process) and all the more so under realistic conditions, the computed models may contain artifacts and may not faithfully represent the inherent structure of the data. Thus, great caution and empirical validation is required in using these techniques.

Despite this caveat, hidden Markov models may be beneficial in studying many diverse problems. In our companion paper[21] we recount a successful application of this body of theory to a problem in automatic speech recognition.

## REFERENCES

1. J. K. Baker, "The DRAGON System—An Overview," IEEE Trans. Acoustics, Speech and Signal Processing, *ASSP-23* (February 1975), pp. 24–9.

2. J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," in *Speech Recognition*, R. Reddy, ed., New York: Academic Press, 1975.
3. L. R. Bahl and F. Jelinek, "Decoding for Channels with Insertions, Deletions and Substitutions, with Applications to Speech Recognition," IEEE Trans. Information Theory, *IT-21* (July 1975), pp. 404–11.
4. L. R. Bahl, J. K. Baker, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Automatic Recognition of Continuously Spoken Sentences from a Finite State Grammar," Proc. ICASSP (April 1978), pp. 418–21.
5. L. R. Bahl, J. K. Baker, P. S. Cohen, N. R. Dixon, F. Jelinek, R. L. Mercer, and H. F. Silverman, "Preliminary Results on the Performance of a System for the Automatic Recognition of Continuous Speech," Proc. ICASSP (April 1976), pp. 425–9.
6. L. R. Bahl, J. K. Baker, P. S. Cohen, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Recognition of a Continuously Read Natural Corpus," Proc. ICASSP (April 1978), pp. 422–4.
7. L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further Results on the Recognition of a Continuously Read Natural Corpus," Proc. ICASSP (April 1980), pp. 872–5.
8. L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Recognition Results with Several Experimental Acoustic Processors," Proc. ICASSP (April 1979), pp. 249–51.
9. F. Jelinek and R. L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," in *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, eds., Amsterdam, The Netherlands: North Holland Publishing Company, 1980.
10. F. Jelinek, "Fast Sequential Decoding Algorithm Using a Stack," IBM J. Res. Develop. (1969), pp. 675–85.
11. F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," IEEE Trans. Information Theory, *IT-21* (May 1975), pp. 250–6.
12. F. Jelinek, "Continuous Speech Recognition by Statistical Methods," Proc. IEEE, *64* (April 1976), pp. 532–56.
13. L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process," Inequalities, *3* (1972), pp. 1–8.
14. L. E. Baum and J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology," Bull. AMS, *73* (1967), pp. 360–3.
15. L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," Ann. Math. Stat., *37* (1966), pp. 1559–63.
16. L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Ann Math. Stat., *41* (1970), pp. 164–71.
17. L. E. Baum and G. R. Sell, "Growth Transformations for Functions on Manifolds," Pac. J. Math., *27* (1968), pp. 211–27.
18. D. S. Passman, "The Jacobian of a Growth Transformation," Pac. J. Math., *44* (1973), pp. 281–90.
19. P. Stebe, "Invariant Functions of an Iterative Process for Maximization of a Polynomial," Pac. J. Math., *43* (1972), pp. 765–83.
20. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," IEEE Trans. Commun., *COM-29* (May 1981), pp. 621–59.
21. L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent Isolated Word Recognition," B.S.T.J., this issue.
22. A. A. Markov, "An Example of Statistical Investigation in the Text of 'Eugene Onyegin' Illustrating Coupling of Tests in Chains," Proc. Acad. Sci. St. Petersburg, *7* (1913), pp. 153–62.
23. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm," IEEE Trans. Information Theory, *IT-13* (April 1967), pp. 260–9.
24. J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming—Part I: Linear Constraints," J. Soc. Indust. Appl. Math., *8* (1960), pp. 181–217.
25. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," Computer J., *6* (1963), pp. 163–8.

26. M. J. Hopper, ed., *Harwell Subroutine Library: A Catalog of Subroutines, 55,* Oxfordshire, England: A.E.R.E. Harwell, 1979.
27. M. R. Hestenes, *Optimization Theory,* New York: Wiley, 1975.
28. E. Lawler, *Combinatorial Optimization,* New York: Holt Rinehart and Winston, 1976.
29. L. R. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," IEEE Trans. Information Theory, *IT-28* (September 1982), pp. 729–34.

## APPENDIX A

Several of the formulas derived in the text are much more compact in matrix notation. Let $'$ denote matrix transposition, as usual, and let the column vectors $\pi$ and $\mathbf{1}$, and the matrices $A$, $B_t$, $t = 1, \cdots, T$ be defined as in the text. Also let $\alpha_t$ and $\beta_t$ be column vectors with components $\alpha_t(i)$, $i = 1, \cdots, N$ and $\beta_t(i)$, $i = 1, \cdots, N$, respectively. Then the recursion for $\alpha_t$ is

$$\alpha_{t+1} = B_{t+1}A'\alpha_t, \qquad t = 1, \cdots, T-1. \tag{76}$$

The recursion for $\beta_t$ is

$$\beta_t = A\,B_{t+1}\beta_{t+1} \qquad t = T-1, \cdots, 1. \tag{77}$$

The starting values are

$$\alpha_1 = B_1\pi$$

$$\beta_T = 1. \tag{78}$$

The probability $P$ is given by

$$P = \beta_t'\alpha_t \qquad \text{for any } t \text{ in } (1, T). \tag{79}$$

The special cases $t = 1$ and $t = T$ give

$$P = \pi' B_1\beta_1 \tag{80}$$

and

$$\begin{aligned} P &= \mathbf{1}'\alpha_T \\ &= \mathbf{1}'B_T A' B_{T-1} \cdots A' B_1\pi. \end{aligned} \tag{81}$$

In each of these formulas $P$ can be regarded as the trace of a $1 \times 1$ matrix, which [as expanded in (81)], is a product of several matrices. The fact that the trace of a product of matrices is invariant to a cyclic permutation of the matrices can be used to advantage in finding the gradient of $P$. Define $\nabla_A P$ as the matrix whose $ij$th component is $\partial P/\partial a_{ij}$. Similarly, define $\nabla_B P$ and $\nabla_\pi P$. Then it is straightforward to show that

$$\begin{aligned} \nabla_\pi P &= B_1\beta_1 \\ \nabla_A P &= \sum_{t=1}^{T-1} \alpha_t\beta_{t+1}' B_{t+1} \\ (\nabla_B P)_{jk} &= \sum_{t \ni O_t = k} (A'\alpha_{t-1})_j (\beta_t)_j. \end{aligned} \tag{82}$$

In the last equation, if $O_1 = v_k$ then the corresponding term in the sum is just $\pi'\beta_1$.

## APPENDIX B

As we mentioned in Section 3.3, it is frequently necessary to compare (or average) two different estimates $B$ and $\tilde{B}$ of the symbol matrix. The optimization procedures, in general, relabel the states; therefore the rows of $\tilde{B}$ may, in general, be permuted relative to those of $B$. Before comparison, therefore, the optimum permutation must be found. This is defined as one that minimizes the distance $D$ defined in eq. (66). The problem of finding this permutation can be converted to a network optimization problem called "bipartite weighted matching." To this end define $w_{ij}$ as the distance of the $i$th row of $B$ from the $j$th row of $\tilde{B}$. As we have done in Fig. 7, draw two sets $P$ and $Q$ of $N$ vertices each. For $1 \leq i, j \leq N$, draw an edge from the $i$th vertex in $P$ to the $j$th vertex in $Q$, and label this edge with the weight $w_{ij}$. The resulting graph is a complete weighted bipartite graph, and the problem is to find an $N$-match (i.e., one to one matching with $N$ edges) such that it has minimum weight.

Suppose that $\mathbf{Z}_k$ is a $k$-match ($k \leq N$). With respect to this match make the following definitions:

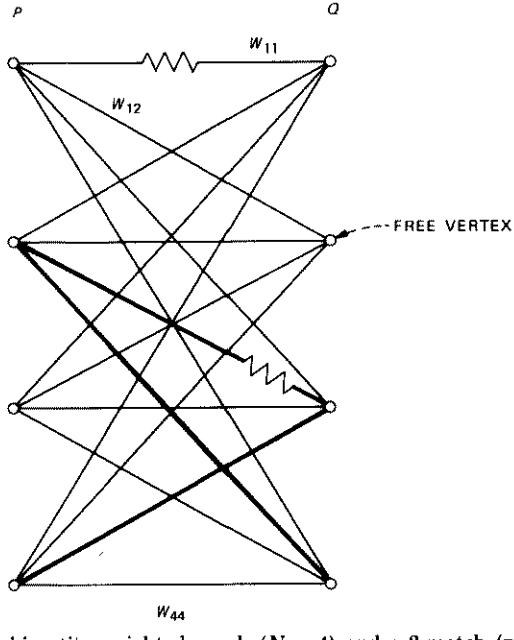



Fig. 7—Complete bipartite weighted graph ($N = 4$) and a 2-match (matched edge →). Path shown in heavy lines is an alternating path for the 2-match. It also happens to be an augmenting path.

(*i*) A matched edge is an edge in $\mathbf{Z}_k$.

(*ii*) A free vertex is one that is not on a matched edge.

(*iii*) An alternating path is a path along edges that alternately belong to $\mathbf{Z}_k$ and do not belong to $\mathbf{Z}_k$. [N.B. the number of matched edges, $m$, on an alternating path may have any value $m \leq k$. The degenerate case $m = 0$ is also valid.]

(*iv*) An augmenting path is an alternating path between two free vertices. Again note that a single edge connecting two free vertices is a valid augmenting path.

An augmenting path has the structure

$$p_1 U q_1 M p_2 U q_2 \cdots U q_j. \tag{83}$$

Here $U$ represents an unmatched edge, $M$ represents a matched edge, and $p_1$ and $q_j$ are the only free vertices on the path. (Here $p_i$ is the $i$th $P$-vertex along the path. The $q_i$ are similarly defined.) Note that the number of $U$'s on an augmenting path is exactly one more than the number of $M$'s. Hence the total number of edges in an augmenting path is always odd.

Suppose we are given a $k$-match $\mathbf{Z}_k$ and an augmenting path **ap** of length $2j + 1$. Then we can obtain a $(k + 1)$-match $\mathbf{Z}_{k+1}$ by a complementary labeling of the edges of **ap** (i.e., by changing every $U$ to $M$ and vice versa). If $w_1, w_2, \cdots, w_{2j+1}$ are the weights along **ap**, then the weight $\mathbf{Z}_{k+1}$ exceeds that of $\mathbf{Z}_k$ by the amount $w_1 - w_2 + w_3 - \cdots + w_{2j+1}$.

This method of obtaining a $(k + 1)$-match from a $k$-match has the following key property (proof given below): Suppose $\mathbf{M}_k$ is a minimum-weight $k$-match. Let **apm** be an augmenting path for $\mathbf{M}_k$ with minimum incremental weight. Then the match $\mathbf{M}_{k+1}$ obtained from $\mathbf{M}_k$ and **apm** is a minimum weight $(k + 1)$-match.

Assuming this property for the moment, a minimum-weight $N$-match can be determined by the following $N$-step algorithm:

For $k = 1, 2, \cdots, N$ generate $\mathbf{M}_k$ by finding an optimum augmenting path for $\mathbf{M}_{k-1}$. (Note that $\mathbf{M}_0$ is an empty set.)

We now show that finding an optimum augmenting path is equivalent to a shortest path problem. With reference to Fig. 8 let us generate a directed graph by directing each edge in $\mathbf{M}_k$ to the left and each unmatched edge to the right. Also let us multiply the weights of all matched edges by $-1$. Finally, add two vertices labeled **s** and **t**. Connect **s** to all *free* vertices in $P$ by right-going edges of zero weight. Similarly, connect all *free* vertices of $Q$ to **t** by right-going edges of zero weight. In this directed graph any path from **s** to **t** is an augmenting path for the matching $\mathbf{M}_{k-1}$. Hence, if we interpret the weights as lengths, it is clear than an optimum augmenting path is a shortest path from **s** to **t**.

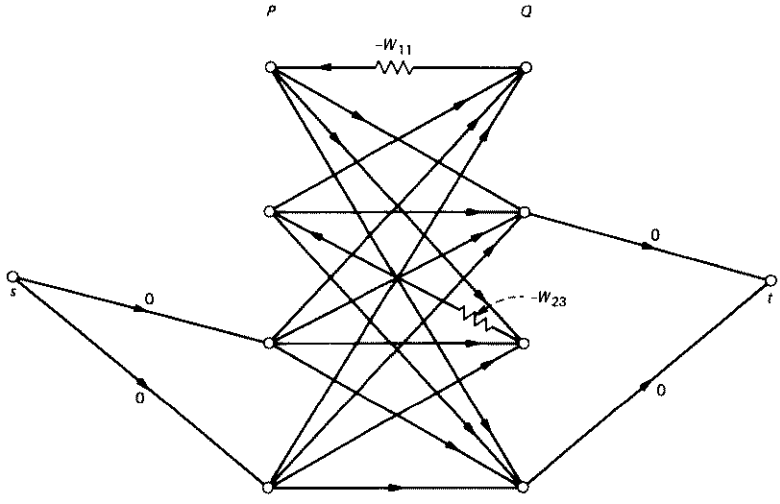



Fig. 8—Directed graph obtained from Fig. 7. Every path from **s** to **t** is an augmenting path of the 2-match (except for the dummy edges of 0 weight from $s$ and to $t$). Interpreting weights as lengths, the length of a path from **s** to **t** is the incremental weight of the corresponding augmenting path.

A shortest path problem can be solved in polynomial time. However, the problem can be solved much more easily and efficiently if the path lengths are all nonnegative. In that case the problem can be solved in $N^2$ time by Dijkstra's method.[28]

It is possible to avoid negative distances in our problem by the method of assigning a "potential" $f(v)$ to each vertex. Suppose that at step $k$ the graph has nonnegative weights, and the edges corresponding to $\mathbf{M}_{k-1}$ have zero weight. Then use Dijkstra's method to find shortest paths to all vertices (including **t**) from **s**. Define $f(v)$ for the vertex **v** as its shortest distance from **s**. Next, modify the weight $w_{ij}$ of the edge from $i$ to $j$ to

$$w'_{ij} = w_{ij} + f(i) - f(j). \tag{84}$$

It is easily seen that this procedure leaves all weights nonnegative, does not alter shortest paths, and all shortest paths have weight 0. Reversing a shortest path from **s** to **t** gives us a matching $\mathbf{M}_k$, and the new graph has nonnegative weights and zero weights for the matched edges. Now $\mathbf{M}_0$ trivially has the postulated properties. Therefore, at every step the graph will have these properties.

An important property of the above procedure is that if a vertex $p$ (or $q$) is on some matched edge in $\mathbf{M}_k$, then it will be on some matched edge in $\mathbf{M}_{k+1}$ also. In writing the actual computer program, this property simplifies the bookkeeping considerably.

Another important property concerns the vertex potentials. Suppose

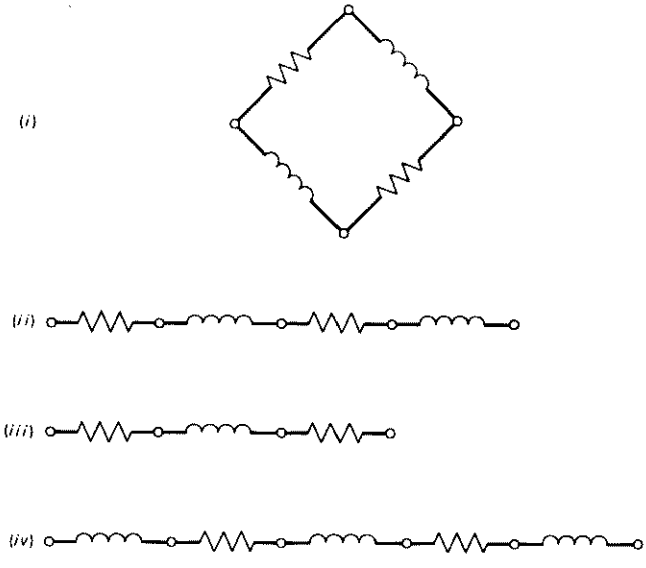



Fig. 9—Types of paths in the symmetric difference between $\mathbf{M}_k$ and $\mathbf{N}_{k+1}$. ($\wwv \in \mathbf{M}_k$, $\sim\sim \in \mathbf{N}_{k+1}$.) There must be exactly one more path of type (*iv*) than of type (*iii*).

$F(v)$ is the sum of the potentials assigned to vertex $v$ at each of the $N$ steps. Suppose $w_{ij}$ and $d_{ij}$ are the original and final weights, respectively, of the edge connecting vertex $i$ to vertex $j$. Then

$$d_{ij} = w_{ij} + F(i) - F(j) \geq 0, \tag{85}$$

and $d_{ij} = 0$ for every edge in the final $N$-match. The numbers $F(v)$ thus provide a simple proof that the final match is indeed an optimum match.

We turn now to a proof of the key property mentioned above. Let $\mathbf{M}_k$ be an optimum $k$-match and let $N_{k+1}$ be *any optimum* $k+1$-match. Then we will show that there is a $k+1$-match obtained from an augmenting path of $\mathbf{M}_k$ such that its weight is equal to that of $N_{k+1}$. For this purpose define $\mathbf{S}$ as the set of edges in the symmetric difference of $\mathbf{M}_k$ and $\mathbf{N}_{k+1}$. (Recall that the symmetric difference of two sets, $\mathbf{A}$ and $\mathbf{B}$, is the set of elements that belong either to $\mathbf{A}$ or to $\mathbf{B}$ but not to both.)

From the geometry of a bipartite graph three properties of the set $\mathbf{S}$ are obvious:

(*i*) The number of edges in $\mathbf{S}$ which belong to $\mathbf{N}_{k+1}$ must exceed the number that belongs to $\mathbf{M}_k$ by exactly 1.

(*ii*) The edges on any path in $\mathbf{S}$ must alternate between $\mathbf{M}_k$ and $\mathbf{N}_{k+1}$.

(*iii*) A vertex on an edge in **S** cannot be shared with any edge in $\mathbf{M}_k$ or $\mathbf{N}_{k+1}$ that does not belong to **S**.

From the first two of these properties it follows that there can be four types of paths in **S** (see Fig. 9).

(*i*) Circuits, each consisting of an even number of edges;

(*ii*) Open paths, each with an even number of edges;

(*iii*) Open paths, each with an odd number of edges beginning with an edge in $\mathbf{M}_k$;

(*iv*) Open paths, each with an odd number of edges beginning with an edge in $\mathbf{N}_{k+1}$. The number of such paths must be exactly one more than the number of paths of type (*iii*).

It is easily seen that the incremental weight of every path of type (*i*) or (*ii*) must be exactly zero. (If it is negative then the weight of $\mathbf{M}_k$ can be decreased by a complementary labeling of the path; if positive then the weight of $\mathbf{N}_{k+1}$ can be decreased in the same way. But this contradicts the hypothesis that $\mathbf{M}_k$ and $\mathbf{N}_{k+1}$ are optimum matches.)

In view of this and property (3), $\mathbf{N}_{k+1}$ can be modified by replacing its edges on all paths of types (*i*) and (*ii*) by the corresponding edges of $\mathbf{M}_k$. The modified $k + 1$-match has exactly the same weight as that of $N_{k+1}$; however, the symmetric difference between $\mathbf{M}_k$ and the modified $k + 1$-match has no paths of type (*i*) or (*ii*).

The same argument applies to pairs of paths, one path each of types (*iii*) and (*iv*). Thus a final modified $k + 1$ match is obtained whose symmetric difference with $\mathbf{M}_k$ is exactly one path of type (*iv*). The weight of this final modified $k + 1$-match is exactly the same as that of the original $\mathbf{N}_{k+1}$, and is obtained from an augmenting path of $\mathbf{M}_k$.

We have written a subroutine that implements the above procedure. The timing, from a number of test runs, is approximately $0.063N^3$ ms central processing unit (CPU) time on the MV-8000. Thus for $N = 40$ about 4 seconds of CPU time is needed.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 4, April 1983
*Printed in U.S.A.*

# On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition

By L. R. RABINER, S. E. LEVINSON, and M. M. SONDHI

(Manuscript received September 10, 1982)

*In this paper we present an approach to speaker-independent, isolated word recognition in which the well-known techniques of vector quantization and hidden Markov modeling are combined with a linear predictive coding analysis front end. This is done in the framework of a standard statistical pattern recognition model. Both the vector quantizer and the hidden Markov models need to be trained for the vocabulary being recognized. Such training results in a distinct hidden Markov model for each word of the vocabulary. Classification consists of computing the probability of generating the test word with each word model and choosing the word model that gives the highest probability. There are several factors, in both the vector quantizer and the hidden Markov modeling, that affect the performance of the overall word recognition system, including the size of the vector quantizer, the structure of the hidden Markov model, the ways of handling insufficient training data, etc. The effects, on recognition accuracy, of many of these factors are discussed in this paper. The entire recognizer (training and testing) has been evaluated on a 10-word digits vocabulary. For training, a set of 100 talkers spoke each of the digits one time. For testing, an independent set of 100 tokens of each of the digits was obtained. The overall recognition accuracy was found to be 96.5 percent for the 100-talker test set. These results are comparable to those obtained in earlier work, using a dynamic time-warping recognition algorithm with multiple templates per digit. It is also shown that the computation and storage requirements of the new recognizer were an order of magnitude less than that required for a conventional pattern recognition system using linear prediction with dynamic time warping.*

## I. INTRODUCTION

There currently exist two standard approaches to isolated word recognition, namely, feature extraction methods and statistical pattern recognition models. A statistical pattern recognition approach has the property of being a nonparametric approach to recognition and therefore is widely used in most commercial and industrial recognizers.[1-6] The feature-based approach to recognition has been primarily used in the (computationally) less expensive systems, and as a basis for recognition of continuous speech (in conjunction with segmentation and labeling algorithms).[4-9]

In the past few years a new approach to speech processing has been proposed, namely, using probabilistic functions of Markov models. This approach has been applied at the Institute for Defense Analyses for speaker recognition,[10] and at Carnegie Mellon University and IBM to solve problems in continuous speech recognition[11,12] with good success. Based on its success in these related areas of speech processing, a question that arises naturally is how well these probabilistic models would work on problems in isolated word recognition.

It is the prime purpose of this paper to provide an answer to the question posed above. Before discussing the approach we have taken to get at the answer, we must first describe the structure of a word recognition system based on (hidden) Markov models (HMM). As in most recognition systems we assume we have a labeled training set of data from which we build a series of Markov models, one for each vocabulary word. Then when we want to recognize an unknown token, we compute a probability score for each word HMM on that token, and choose as the recognized word the one corresponding to the model with the highest probability score (i.e., the most likely word HMM). Techniques for training and scoring such HMMs are discussed both here and in the companion paper.[3]

In a conventional pattern recognition system the unknown test token is time-aligned in turn to each reference pattern via some form of time-warping procedure, typically, dynamic time warping (DTW). By contrast, no such direct alignment is performed in the HMM system; only an indirect time alignment is obtained based on the probabilistic scoring. Thus it is interesting to study the relationship between probabilistic scoring and DTW as applied to isolated word recognition. As we shall see, there is no simple relationship. We will point out several similarities and differences in the two approaches.

The organization of this paper is as follows. In Section II we briefly review the conventional DTW word recognizer based on LPC modeling, since this will be the focus of comparison throughout the paper. In Section III we review the basic ideas behind the use of HMMs for isolated word recognition. It is the purpose of this section to establish

notation and terminology that will define the basic parameters of interest in the HMM system. Section III shows that one inherent feature of the HMM recognizer (as we have implemented it) is that the models need a discrete, finite set of observations (input data) to obtain the best model parameters for each word in the vocabulary. A vector quantizer (VQ) was used to transform the continuous set of linear predictive coefficient (LPC) vectors into a finite observation set. Therefore, in Section IV we describe the key ideas behind vector quantization of LPC sets, and discuss the particular implementation that we used. In Section V we describe the overall structure of the HMM isolated word recognizer. In Section VI we describe a series of experiments used to evaluate the performance of the HMM word recognizer and compare it to the performance of a DTW recognizer on the same vocabulary. The effects, on performance, of several parameter variations in the HMM and VQ are also described in this section. In Section VII we discuss the results of the performance evaluation and comparison experiments. The strengths and weaknesses of the HMM word recognizer are discussed, along with computational and storage comparisons of HMM and DTW word recognizers. We attempt, in this section, to determine the fundamental relationships between the HMM and DTW systems.

## II. REVIEW OF CONVENTIONAL DTW WORD RECOGNIZER BASED ON LPC MODELING

Figure 1 shows a block diagram of the LPC-based isolated word recognizer.[2,3] The input speech signal, $s(n)$, recorded over a standard dialed-up telephone line, is bandpass-filtered between 100 and 3200 Hz, and digitized at a 6.67-kHz rate. The first step in the processing is

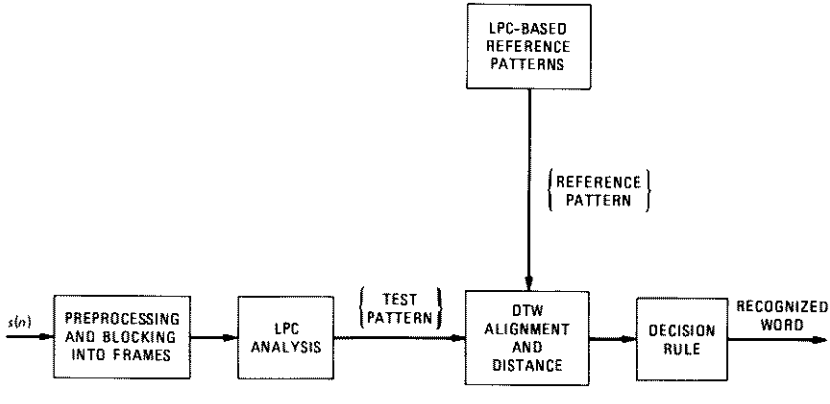



Fig. 1—Block diagram of conventional LPC-based word recognizer using a standard dynamic time-warping algorithm for registering test and reference patterns.

the preprocessing block, a first-order digital network, which provides a high-frequency pre-emphasis to the speech. The pre-emphasized signal is blocked into frames of 45 ms (300 samples) with each consecutive frame spaced 15 ms (100 samples) apart. An 8-pole LPC analysis (autocorrelation method) is performed on each frame of the word (after isolating it with an endpoint detector[14]), thus creating the test pattern. This test pattern is compared with each reference pattern using a DTW alignment algorithm that simultaneously provides a distance score associated with the alignment. The distance scores for all the reference patterns are sent to a decision rule, which provides a classification of the spoken word, and possibly an ordered (by distance) set of the best $n$ candidates.

The word reference patterns for the recognizer of Fig. 1 are created by a training algorithm. For speaker-trained applications, typically a single reference pattern is created for each word in the vocabulary using a robust training algorithm.[15] For speaker-independent applications, a set of $Q$ reference patterns is created for each vocabulary word using a clustering procedure.[16,17] Typically, about 12 templates per word are sufficient for recognizing words from a fairly homogeneous adult population of native American talkers.

## III. BASICS OF HMM FOR WORD RECOGNITION

We assume we have a finite sequence, $\mathbf{O}$, of observations,

$$\mathbf{O} = O_1 O_2 \cdots O_T, \tag{1}$$

where each observation is a discrete symbol drawn from a finite alphabet of symbols. (For the system we will be describing, the observations are the indices of the LPC vectors obtained from an LPC vector quantizer.) We further assume that the sequence of observations may be modeled as a probabilistic function of an underlying Markov chain whose state transitions are not directly observable; hence the name "Hidden Markov Model." Figure 2 shows such a model, $\mathbf{M}$, which is characterized by the following:

($i$) $N$ = the number of states in the model. For the model of Fig. 2, $N = 5$.

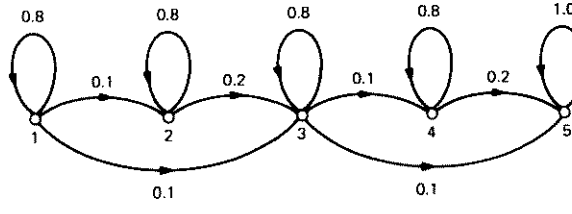



Fig. 2—A typical state diagram for a 5-state Markov model.

(*ii*) $M$ = the number of output symbols in the discrete alphabet of the model. For the present example, $M = 5$.

(*iii*) $A = \{a_{ij}\}$, the transition matrix of the underlying Markov chain. Here, $a_{ij}$ is the probability of making a transition to state $j$, given that the model is in state $i$. For the model of Fig. 2 we have

$$A = \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}.$$

Note that only 11 of the 25 $a_{ij}$'s are nonzero.

(*iv*) $B = \{B_{jk}\} = \{b_j(k)\}$, the model output symbol probability matrix, where $b_j(k)$ is the probability of outputting symbol $k$, given that the model is in state $j$. For the example chosen,

$$B = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}.$$

(*v*) $\pi = \{\pi_i\}$, $i = 1, 2, \cdots, N$, the initial state probability vector. For the left-to-right models of the type shown in Fig. 2, we assume the system always begins in state 1, i.e., $\pi_1 = 1$, $\pi_i = 0$, $i \neq 1$.

Isolated word recognition using HMM consists of two phases, training and recognition (or classification). In the training phase, the training set of observations is used to derive a set of reference models of the above type, one for each word in the vocabulary. In the classification phase, the probability of generating the test observation is computed for each reference model. The test is classified as the word whose model gives the highest probability. The computations in each of these phases are fairly straightforward.

Let us begin with the classification phase. Given the observation sequence, $\mathbf{O}$, and a model, $\mathbf{M}$ (i.e., $N$, $M$, $A$, $B$, and $\pi$), the probability of $\mathbf{O}$ having been generated by model $\mathbf{M}$ is

$$P(\mathbf{O}|\mathbf{M}) = \sum_{i_1, i_2 \cdots i_T} \pi_{i_1} b_{i_1}(O_1) a_{i_1 i_2} \cdots a_{i_{T-1} i_T} b_{i_T}(O_T). \tag{2}$$

The summation in eq. (2) is more readily computed by defining a forward partial probability, $\alpha_t(i)$, as

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t \text{ and state } i \text{ at time } t|\mathbf{M}). \tag{3}$$

This leads to the recursion

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \qquad t = 1, 2, \cdots, T-1 \tag{4}$$

by which eq. (2) can be expressed as

$$P(\mathbf{O}|\mathbf{M}) = P = \sum_{j=1}^{N} \alpha_T(j). \tag{5}$$

In the training phase an initial estimate of the model is made, and $P$ is computed for the training observation sequence according to eq. (5). Next the model is iteratively adjusted to increase $P$. The iterations are stopped when $P$ stops increasing significantly, or when some other stopping criterion is met (e.g., when the number of iterations exceeds a limit).

One remarkable algorithm for improving a trial model is the Baum-Welch reestimation algorithm.[18] However, maximizing $P$ can also be looked upon as a constrained optimization problem, for which many algorithms have been proposed. In the companion paper in this issue of the Journal,[13] we discuss the relative merits of these procedures.

We now discuss a number of factors that influence the performance of HMM recognizers.

### 3.1 Initial estimates of A and B

One factor of interest for the HMM recognizer is the choice of initial estimates for the elements of the matrices $A$ and $B$. The problem here is that although the training procedure is guaranteed to reach a critical point of $P$, the value of $P$ obtained is typically a local maximum. Hence, alternative starting values of $A$ and $B$ could yield models with higher (or lower) values of $P$. For our simulations we have chosen to start the training models with essentially random choices for the nonzero elements of both $A$ and $B$, normalized to satisfy the constraints

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad i = 1, 2, \cdots, N \tag{6a}$$

$$\sum_{k=1}^{M} b_j(k) = 1 \qquad j = 1, 2, \cdots, N. \tag{6b}$$

An alternative starting condition could be

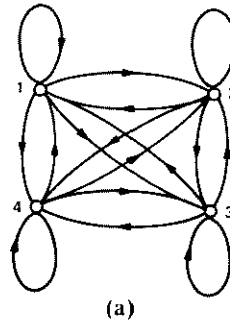
$$a_{ij} = 1/N + \epsilon \tag{7a}$$

$$b_j(k) = 1/M + \epsilon, \tag{7b}$$

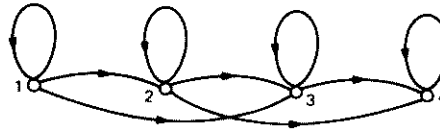
where $\epsilon$ is a uniformly distributed random variable whose peak is much smaller than either $1/N$ or $1/M$. [Again the $a_{ij}$'s and $b_j(k)$'s of eq. (7) must be normalized using eq. (6) prior to running the optimization.]

### 3.2 HMM structures and the number of states

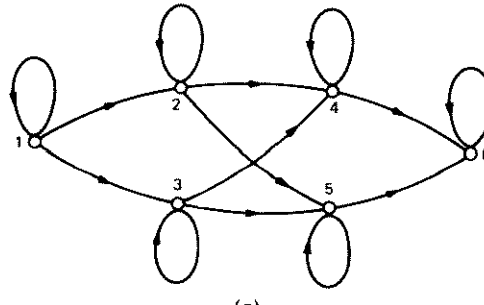
A second factor affecting the determination of optimum HMMs for each vocabulary word is the model structure and the number of states.




(a)




(b)




(c)

Fig. 3—State diagrams for: (a) unconstrained Markov model with four states, (b) constrained serial Markov model with four states, and (c) constrained parallel Markov model with six states.

We have considered three types of model structures, namely unconstrained, constrained serial, and constrained parallel. Typical examples of each of these models are shown in Fig. 3. In the unconstrained models (shown in Fig. 3a) a transition from any state to any other state can be made—i.e., all $a_{ij}$'s are allowed to be nonzero. Both the constrained serial models (shown in Fig. 3b) and the constrained parallel models (shown in Fig. 3c) are left-to-right models, i.e., the state transition matrix $A$ is upper triangular. The serial models generally proceed sequentially through the states (although individual states can be skipped over), whereas the parallel models allow multiple paths through the model, with each path skipping one or more model states. For example, there are four distinct paths through the model of

Fig. 3c, 1-2-4-6, 1-2-5-6, 1-3-4-6 and 1-3-5-6, each of which traverses four of the six model states.

Each of the model structures of Fig. 3 can be generalized to include an arbitrary number of states. Recall, however, that the number of free parameters of the Markov model is on the order of $N^2$ (for the $A$ matrix) plus $NM$ (for the $B$ matrix). Hence, if $N$ gets too large, accurate and reliable determination of the optimum $A$'s and $B$'s may become difficult for a fixed-size training set. However, within these constraints we have investigated models with as few as two states, and as many as 20 states. There appears to be no good theoretical way to choose the number of states needed for a word model, since the states need not be physically related to any single observable phenomenon.

### 3.3 Multiple observation sequences

A third factor affecting the determination of the optimum HMM for each vocabulary word is the observation sequence used for training. Since we are interested in obtaining speaker-independent models, the observation sequence, $\mathbf{O}$, actually consists of several independent sequences $\mathbf{O}^{(k)}$, $k = 1, 2, \cdots, K$, where $\mathbf{O}^{(k)}$ is the training sequence for talker $k$, and $K$ is the total number of talkers used for training. Typically, a value of $K = 100$ has been used in our clustering work for speaker-independent training. The way in which we handle multiple sequences is to calculate $P(\mathbf{O}^{(k)}|\mathbf{M})$, using eq. (5), for each sequence, and maximize the product of the probabilities, i.e.,

$$P = \prod_{k=1}^{K} P(\mathbf{O}^{(k)}|\mathbf{M}). \tag{8}$$

The implementation of the computation of eq. (8) is straightforward for the Baum-Welch reestimation procedure, as well as for the gradient methods.[13] Thus the fact that the training data consist of multiple sequences causes no problem in estimating the optimum HMM parameters.

### 3.4 Constraints on A, B matrices during training

As we show in Fig. 3 we have considered three general HMM structures. For the unconstrained structure the $A$ and $B$ matrix elements are allowed to assume any value consistent with the stochasticity constraints. For the constrained serial models we have considered two general constraints, namely:

$$\text{SC1: } a_{ij} = 0 \quad \text{for} \quad j < i \quad \text{and} \quad j \geq i + 3 \text{ (double skip allowed)} \tag{9a}$$

$$\text{SC2: } a_{ij} = 0 \quad \text{for} \quad j < i \quad \text{and} \quad j \geq i + 2 \text{ (single skip allowed).} \tag{9b}$$

These two cases are illustrated in Fig. 4 for a 5-state model. Constraints SC1 allow single- or double-state jumps when exiting a given state,

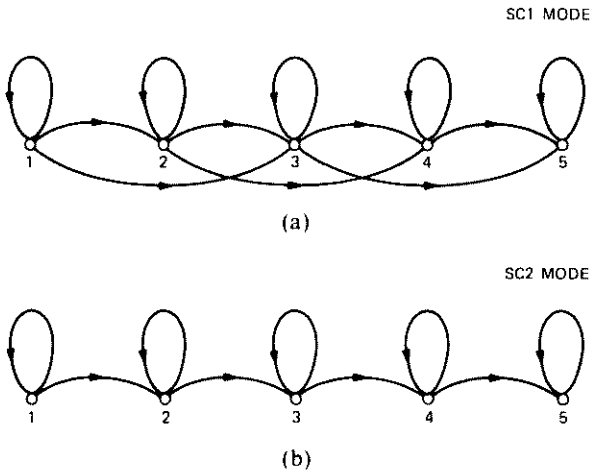



Fig. 4—Markov models for two types of serial constraints: (a) single and double transitions allowed, and (b) only single transitions allowed.

whereas constraints SC2 only allow single jumps when exiting a state. Hence, models of the type shown in Fig. 4a have somewhat more flexibility than those of the type shown in Fig. 4b.

For the constrained parallel models we have only considered transition matrices of the type illustrated in Fig. 3c, i.e., a given state can only exit to either of a pair of states in the next column of the grid.

For the most part we have not constrained the $B$ matrix. However, one problem arises if the $B$ matrix is left completely unconstrained. The problem is that a finite training sequence of length $T$ may result in $b_j(k) = 0$. In classification it can then be the case that $\alpha_{t-1}(i)\, a_{ij}$ is nonzero for only one value of $j$, and $O_t = k$, then the probability of that sequence arising from the model with $b_j(k) = 0$ is $P = 0$; hence a recognition error must occur. This is the so-called missing or inadequate training data problem. We handle this problem (see Ref. 13 for a justification) by using post-estimation constraints on the $b_j(k)$'s of the form

$$b_j(k) \geq \epsilon, \tag{10}$$

where $\epsilon$ is a suitably chosen threshold value. All $b_j(k)$'s are compared to the $\epsilon$ threshold and those that are below $\epsilon$ are replaced by $\epsilon_j$ for each $j$. After this replacement, each $b_j(k)$ that was not changed to the $\epsilon$ value is rescaled by the quantity $1 - R_j\epsilon$ [where $R_j$ is the number of $b_j(k)$'s changed for a given $j$] to properly normalize the $b_j(k)$'s.

It should be clear that all the constrained HMMs are left-right models in that the observations must begin in state 1, must proceed from state to state in a monotonically increasing manner, and must

end in state $N$. Thus, temporal information in the observation sequence is coded directly into the left-to-right HMM.

### 3.5 Multiple estimates of A, B, and averaging

As we mentioned earlier, the reestimation and gradient optimization procedures are guaranteed to find a critical point of $P$ for each HMM. However, in practice, a large number of such points exist in the parameter space. Thus, different initial conditions on $A$ and $B$ may lead to different solutions. To understand the variability in model parameters as well as its effects on overall recognition performance, a series of HMMs were obtained for each word by selecting $R$ random starting sets for $A$ and $B$, and solving for the optimum $A$ and $B$ in each case. By scoring each of the $R$ models individually, we can obtain an indication of the statistical variability in performance score owing to uncertainty in $A$ and $B$.

An alternative procedure to using multiple HMMs for each word, obtained from different random starting values for $A$ and $B$, is to average the $R$ sets of $A$ and $B$ to give an averaged model for each word. The effects of such averaging on word recognition accuracy will be discussed in Section 6.3.

### 3.6 Scoring of observation sequences

One way to score a given observation sequence, $\mathbf{O}$, is to use the iterative calculation of eqs. (4) and (5). We call this the Baum-Welch score, $P_{BW}$. For left-to-right models, eq. (5) is modified as

$$P_{BW} = \alpha_T(N) \tag{11}$$

because the sequence is constrained to end in state $N$.

An alternative scoring procedure for the observation sequence, $\mathbf{O}$, given the model, $\mathbf{M}$, is the Viterbi algorithm,[20] which may be compactly stated as:

($i$) Initialization—$\delta_1(i) = \log[\pi_i b_i(O_1)]$, $i = 1, 2, \cdots, N$

($ii$) Recursion—for $2 \le t \le T$, $1 \le j \le N$

$$\delta_t(j) = \max_{1 \le i \le N} \{\delta_{t-1}(i) + \log[a_{ij} b_j(O_t)]\}$$

($iii$) Termination—$P_{VI} = \delta_T(N)$ for left-to-right models

$$= \sum_{j=1}^{N} \delta_T(j) \text{ for unconstrained models.}$$

The above algorithm is a form of the well-known dynamic programming method and can be shown to have the property of determining the state sequence $\mathbf{i} = i_1 i_2 \cdots i_T$, which maximizes

$$P(\mathbf{i}|\mathbf{O}, \mathbf{M}).$$

It is easily shown that both the Baum-Welch and Viterbi scoring procedures require roughly the same amount of computation. The major differences are in the interpretation of the resulting solutions.

## IV. VECTOR QUANTIZATION OF LPC COEFFICIENTS

In Section III we noted that in our implementation of HMMs for isolated word recognition the inputs to the model are assumed to be sequences of discrete symbols chosen from a finite alphabet. We obtained these discrete symbols by using the method of vector quantization[19,21] of the LPC vectors measured as described in Section II. In this section we review the theory of vector quantization and discuss its implementation for isolated word recognition.

### *4.1 Theory of vector quantization*

Assume we have a training set of LPC vectors, $\mathbf{a}_i$, $i = 1, 2, \cdots, I$, which are a good representation of the types of LPC vectors that occur when the words in the vocabulary are pronounced by a wide range of talkers. The main idea behind vector quantization is to determine the optimum set of codebook LPC vectors, $\hat{\mathbf{a}}_m$, $m = 1, 2, \cdots, M$, such that for a given $M$, the average distortion in replacing each of the training set vectors, $\mathbf{a}_i$, by the closest codebook entry, $\hat{\mathbf{a}}_m$, is minimum.

More formally stated, if we define $d(\mathbf{a}_R, \mathbf{a}_T)$ as the distance between two LPC vectors, $\mathbf{a}_R$ and $\mathbf{a}_T$, then the goal of vector quantization is to find the set, $\hat{\mathbf{a}}_m$, such that

$$\|D_M\| = \min_{\hat{\mathbf{a}}_m} \left\{ \frac{1}{I} \sum_{i=1}^{I} \min_{1 \le m \le M} [d(\hat{\mathbf{a}}_m, \mathbf{a}_i)] \right\} \tag{12}$$

is satisfied. The quantity $\|D_M\|$ is the average distortion (distance) of the vector quantizer.

The way in which eq. (12) is solved, for a given value of $M$, is due to Juang et al.[21] The algorithm first finds the optimum solution for $M = 2$ (two codebook entries), then splits each optimum LPC vector into two components, and finds the optimum solution for $\hat{M} = 2 \cdot M$. This procedure iterates until $M$ is as large as desired. A flow diagram of the details of the codebook generation procedure is given in Ref. 21. The local distance used in our system was the likelihood distance,[2]

$$d(\mathbf{a}_R, \mathbf{a}_T) = \frac{\mathbf{a}_R V_T \mathbf{a}_R'}{\mathbf{a}_T V_T \mathbf{a}_T'} - 1, \tag{13}$$

where $V_T$ is the autocorrelation matrix of the sequence that gave rise to LPC vector $\mathbf{a}_T$.

### 4.2 Implementation of the vector quantizer

To train the vector quantizer we used a set of 39708 LPC vectors, obtained by using all the vectors in one complete set of 10 isolated digits uttered by each of 100 talkers (50 male, 50 female). Applying the algorithm of Juang et al.,[21] we generated vector quantizers of size $M = 2$, 4, 8, 16, 32, 64, and 128. During the course of running the algorithm, several performance criteria were monitored, including:

(*i*) Average distortion, $\|D_M\|$, of eq. (12)

(*ii*) Sigma ratio (cluster separation) of the resulting codebook entries (clusters), defined as

$$\sigma = \frac{\frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{M-1}\right)\sum_{j=1}^{M} d(\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j)}{\|D_M\|}, \tag{14}$$

where the numerator is the average intercluster distance, and $\|D_M\|$ is the average intracluster distance.

(*iii*) Cluster cardinality, $N_i$, defined as the number of tokens in the $i$th cluster (i.e., the cluster represented by the $i$th codebook entry).

(*iv*) Cluster distortion, $\tilde{d}_i$, defined as the average distortion (distance) for the $i$th cluster.

It should be clear that the average distortion, $\|D_M\|$, satisfies the relation

$$\|D_M\| = \frac{1}{I}\sum_{i=1}^{M} \tilde{d}_i \cdot N_i \tag{15}$$

and that the cluster occupancy satisfies the relation

$$\sum_{i=1}^{M} N_i = I. \tag{16}$$

Results of running the VQ algorithm on the training set of 39708 vectors are given in Figs. 5 through 8. Figure 5 shows plots of $\|D_M\|$ versus $M$ (on a log scale) (part a), and the $\sigma$-ratio versus $M$ (part b) for values of $M$ from 2 to 128. We can see that for values of $M \geq 32$ the average distortion falls below 0.3, and that for $M = 64$ the value of $\|D_M\| \approx 0.2$. If we use the conventional recognizer of Fig. 1, the average distance between repetitions of a word (after DTW alignment) has been found to be on the order of 0.3 to 0.4;[16] hence, values of $\|D_M\| < 0.3$ imply smaller error for the VQ than for interreplication variations of words. The $\sigma$-ratio plot shows ratios greater than 10 for $M \geq 32$; hence, extremely good cluster separation is achieved in the vector quantizer for these values of $M$.

Figures 6 through 8 show a detailed analysis of the statistics of the vector quantizer output for $M = 128$. Figure 6a shows the cluster

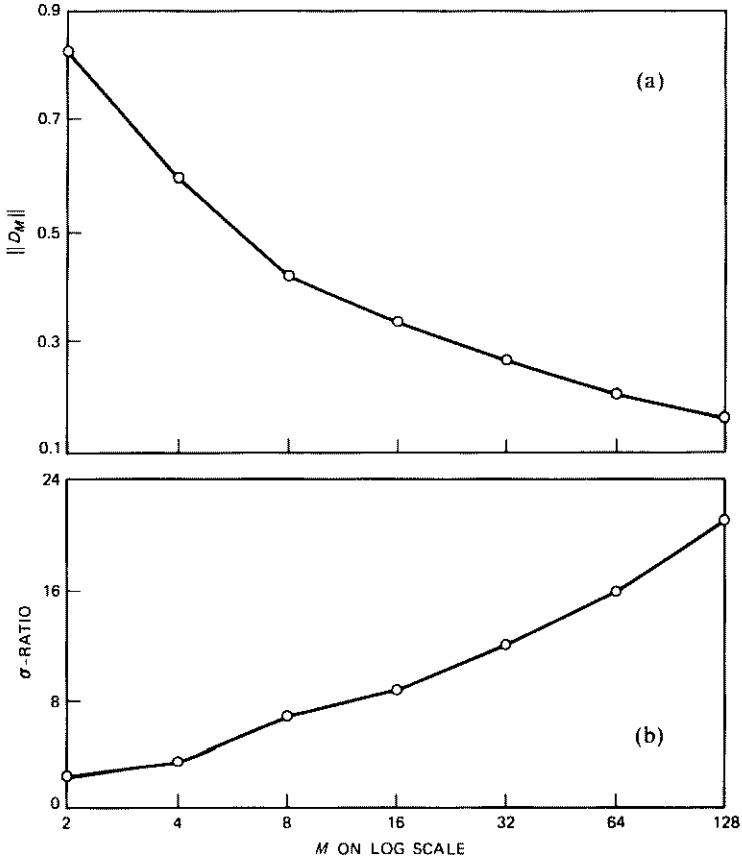



Fig. 5—Plots of vector quantizer performance versus size of codebook for: (a) average distortion, and (b) sigma ratio.

cardinality as a function of the VQ index, and Fig. 6b a histogram of cluster cardinality. The largest cluster has 857 tokens, whereas the smallest cluster has 119 tokens; hence, a spread of over 7 to 1 in cluster occupancy is obtained. The average cluster cardinality, for this case, is 310 tokens, as denoted by the dashed line in Fig. 6a. The histogram of cluster cardinality indicates that the vast majority of clusters have fewer than the average number of tokens.

Figure 7a shows the cluster distortion as a function of the VQ index, and Fig. 7b shows a histogram of cluster distortions. The largest distortion for any cluster is 0.303, whereas the smallest distortion is 0.047; hence, a spread of more than 6 to 1 is observed in cluster distortions. The dashed lines in Fig. 7 denote the average cluster distortion, which in this case is 0.165.

Finally, Fig. 8 shows a plot of the total cluster distortion, defined as

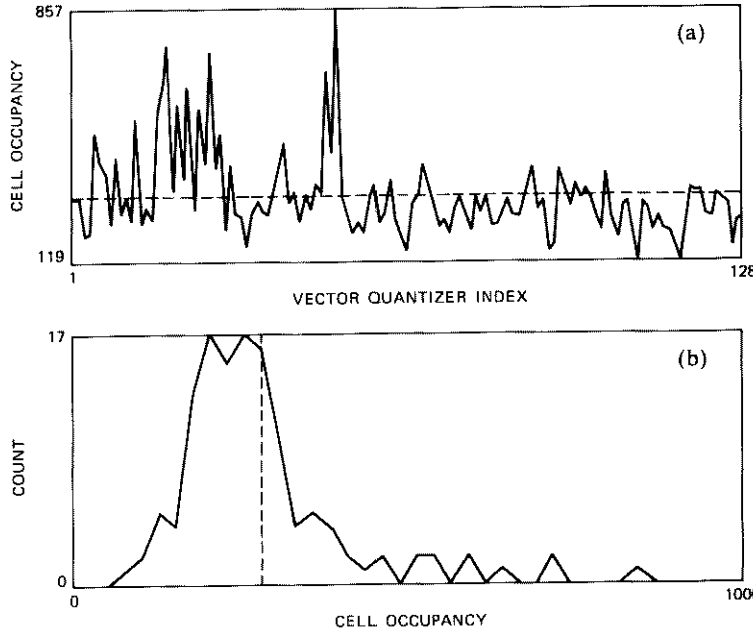



Fig. 6—Plots of (a) cell occupancy versus codebook index, and (b) histogram of cell occupancy for 128-codeword vector quantizer.

$N_i\tilde{d}_i$, versus VQ index. The range of total cluster distortion is from 25 to 72; hence, a spread of less than 3 to 1 is obtained. It is conjectured that the "ideal" vector quantizer seeks to determine the set of "optimum" codebook vectors such that the total cluster distortion is as close to uniform as possible. Hence, clusters with large average distortions should have low cardinality, whereas clusters with small average distortion should have high cardinality. It can be seen from Figs. 6 through 8 that the total cluster distortion statistics are much closer to uniform than are either the cardinality or the average cluster distortion statistics.

Based on the results shown in Figs. 5 through 8, it was decided to implement the HMM recognizer using a $M = 64$ VQ, since the small decrease in average distortion from $M = 64$ to $M = 128$ did not justify the increased computation owing to the larger codebook.

Figure 9 shows some properties of the LPC vectors in the codebook for $M = 64$. Shown in this figure are plots of the first few resonances of the 64 codebook entries (part a), and plots of first versus second resonance (part b), first versus third resonance (part c), and second versus third resonance (part d). As we anticipated, typical vowel resonances for the digits are seen clearly in the plots (e.g., high front

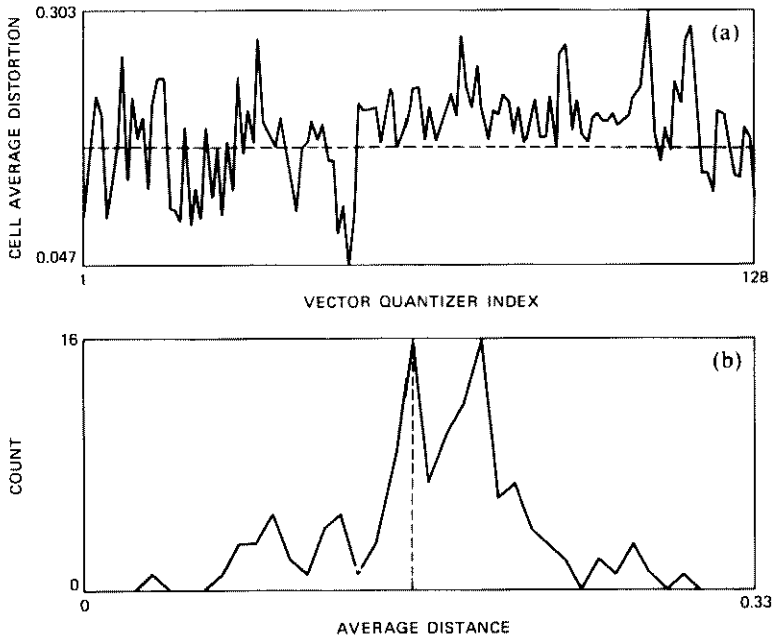



Fig. 7—Plots of (a) cell average distortion versus codebook index, and (b) histogram of cell average distortion for 128-codeword vector quantizer.

vowels, low back vowels, etc), along with characteristic resonances of transient and other nonvoiced sounds. Detailed examination of the spectra of the 64 codebook entries did not provide any further enlightenment as to the VQ properties.

## V. OVERALL HMM/VQ ISOLATED WORD RECOGNIZER

A block diagram of the overall HMM/VQ isolated word recognizer is given in Fig. 10. The recognizer operates as a speaker-independent word recognizer, which runs first in a training mode, to provide the codebook entries of the VQ, and the model coefficients of each word HMM.

In the classification mode the LPC sets of the unknown word are first sent through the vector quantizer (to give a finite set of VQ indices) and then scored on each word HMM (using either the Viterbi scoring or the Baum-Welch scoring) to give a probability score for each word model. The decision rule chooses the word whose model gives the highest probability.

In the next section we describe the results of several tests designed to measure the performance of the HMM/VQ word recognizer and to compare it with that of a conventional LPC/DTW recognizer.

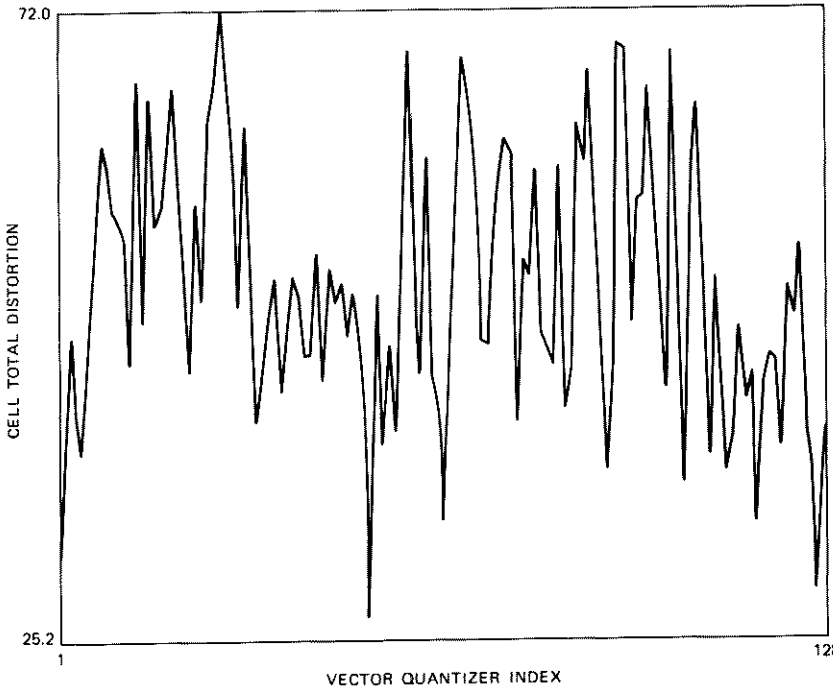



Fig. 8—Plot of cell total distortion versus codebook index for 128-codeword vect quantizer.

## VI. EVALUATION EXPERIMENTS AND RESULTS

Several evaluation tests were performed on the HMM/VQ and LPC/DTW isolated word recognizers. For most conditions a single test set of data, denoted as TS1, was used, consisting of one replication of each of the 10 digits by a set of 100 talkers. These talkers were the same ones used to train the recognizer; however, the test replication was recorded many days after the training replication. A second test set of data, denoted as TS2, was used in a couple of tests. This test set consisted of 20 replications of each of the 10 digits by a set of 10 new talkers (5 male, 5 female). Thus, TS2 contained twice as many test tokens as TS1, but represented only one-tenth the number of talkers; however, none of the talkers was included in the training set for either the VQ or the HMMs.

The results presented in this section are the output of a series of recognition tests in which one or more features of the HMM/VQ recognizer were varied. Following the presentation of the results of each of the individual experiments, we shall endeavor to provide a measure of coherency to the results.

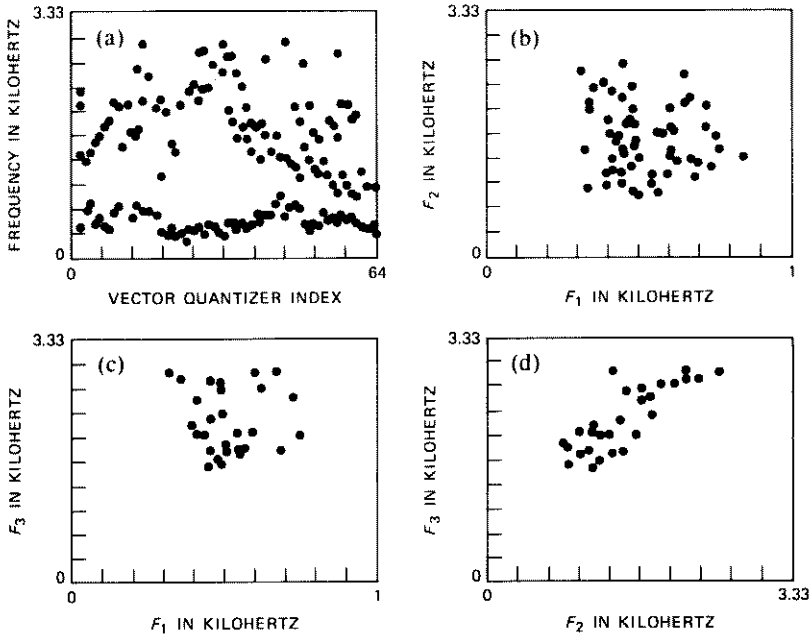



Fig. 9—Plots of locations of first three resonances of codebook vectors: (a) plotted as a function of the vector index, (b) plotted in the $F_1 - F_2$ plane, (c) plotted in the $F_1 - F_3$ plane, and (d) plotted in the $F_2 - F_3$ plane.

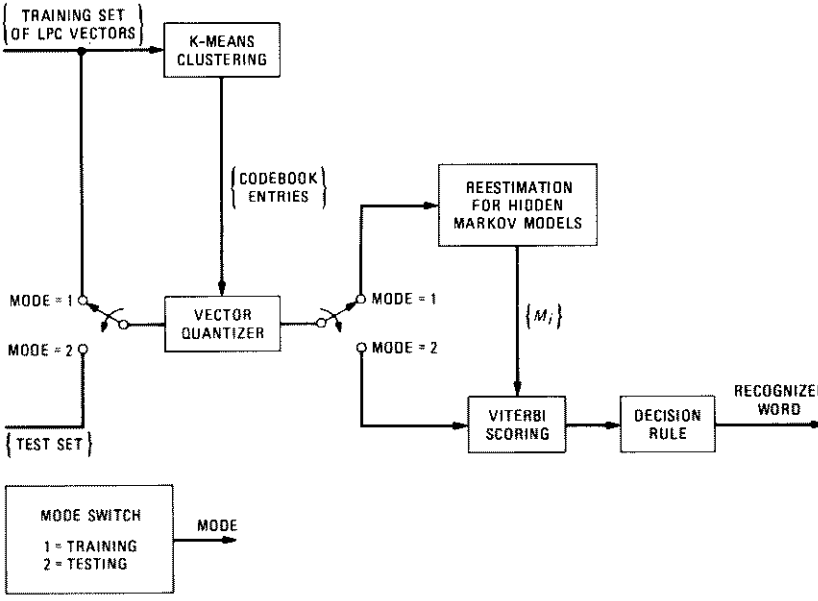



Fig. 10—Overall block diagram of the hidden Markov model—vector-quantizer isolated word recognizer.

### 6.1 Effects of constrained and unconstrained HMMs

The first set of experiments sought to understand the effects of placing constraints on the $A$ and $B$ matrices on the performance of the overall recognizer. As such, the HMM was trained for an $N = 5$ state model with the following constraints:

| | |
|---|---|
| UC model: | No constraints placed on $A$; epsilon constraints [of the type given in eq. (10)] placed on $B$. |
| CO model: | The constraints of eq. (9a) placed on $A$; epsilon constraints of eq. (10) placed on $B$. |
| UC.35 model: | Same as UC model but all training sequences with VQ distortions greater than 0.35 were eliminated. |
| CO.35 model: | Same as CO model but all training sequences with VQ distortions greater than 0.35 were eliminated. |

For each of the above four models, the 1000-digit sequences of TS1 were used to measure the overall error rate as a function of the $\epsilon$ constant parameter. The results are given in Fig. 11, which shows plots of error rate versus $\epsilon$ (on a log scale) for each of the four models. Several trends clearly emerge from these results. First, we can see that a nonzero value of $\epsilon$ is an absolute *necessity* for obtaining good performance. Whenever a symbol, $k$ (a VQ index), appears in a test word in a state, $j$, where $b_j(k) = 0$, the probability for that word model is multiplied by the $\epsilon$ value. If $\epsilon = 0$ then the word model is eliminated from consideration and an error occurs. For finite, nonzero values of $\epsilon$, however, such errors need not, and generally will not, occur. Hence,

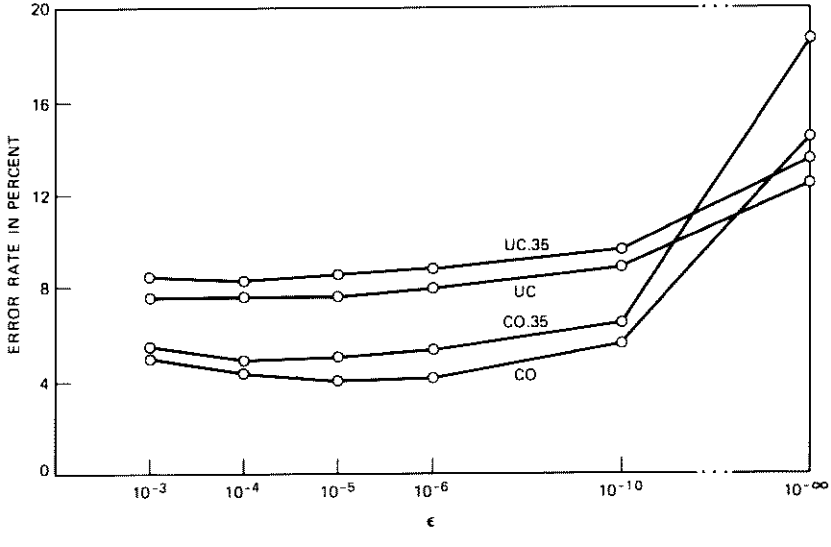



Fig. 11—Plots of average word error rate versus the minimum value of the symbol probability matrix, $\epsilon$, for four types of hidden Markov models, for TS1 data.

even for $\epsilon$ on the order of $10^{-10}$, the error rate is substantially smaller than for $\epsilon = 0$.

A second clear trend that can be seen in Fig. 11 is that the constrained serial models performed consistently better than the unconstrained models for all nonzero values of $\epsilon$. This result implies that the extra freedom of the unconstrained models tends to raise the probability scores for incorrect words more than for correct words. This is somewhat reminiscent of the fact that opening up the search region of a conventional DTW search helps the wrong words much more than it does the correct words.[16]

It can also be seen from Fig. 11 that it is *always* preferable to train with all available sequences. This result suggests that the more training data given to the HMM model estimation algorithm the better the estimates of the HMM parameters, even if some of the data are less than ideal.

The final trend that emerges from the curves of Fig. 11 is that there is a large range of values of $\epsilon$ for which essentially identical performance results. For example, in the range $10^{-10} \leq \epsilon \leq 10^{-3}$, for model CO, the recognition error rate changes by less than 1.6 percent. Thus, so long as $\epsilon$ is in this broad range, the exact value of $\epsilon$ is not overly significant.

Based on the results of this first series of experiments, we applied the following restrictions:

(*i*) Consider only constrained HMMs

(*ii*) Constrain $B$ matrix entries such that $b_j(k) \geq \epsilon = 10^{-5}$

(*iii*) Use all possible training sequences for the HMMs.

Before we proceed to the next series of experiments, some comments should be made about practical methods of implementing constraints on the $B$ matrix. In Ref. 13 we show how the constraints on the $b_j(k)$ coefficients, of the type given in eq. (10), can be incorporated directly into either the gradient or the Baum-Welch reestimation algorithm. We have also tried a post-normalization technique in which no constraints were placed directly on the $B$ matrix. Following convergence, the $B$ matrix was examined and all entries whose values were below $\epsilon$ were reset to the value $\epsilon$, and the rows of the matrix were suitably renormalized to sum to 1.0. Our recognition results indicate identical performance for both the direct and the post-normalization constraint methods. Hence, there appears to be no advantage to constraining the $B$ matrix directly.

### *6.2 Markov model with variable number of states*

The second set of experiments consider the effects on recognition accuracy of using the constrained serial HMMs with different numbers of states. In particular we computed the optimum SC1 model (see Fig.

4) for each digit where the number of states varied from 2 to 9. We also computed the optimum SC1 model for a 20-state model for each digit.

To evaluate these different models a recognition test was conducted in which each digit was represented by a single $N$-state HMM, where $N$ took on the values 2 to 9, and 20. The results of this experiment are given in Fig. 12a, which shows word error rate versus number of states in the HMM for the data of TS1. We see a steady but slow decrease in the average word error rate in this curve. We also see a statistical fluctuation in the curve owing to the sampling variability in the $A$'s and $B$'s for each word HMM. (We will return to this issue later in this section.)

A second recognition test was conducted in which each word was represented by *all* the word models with number of states up to some maximum value, *NMAX*. The results of this recognition test (again using TS1 data) are shown in Fig. 12b. A somewhat smoother curve of

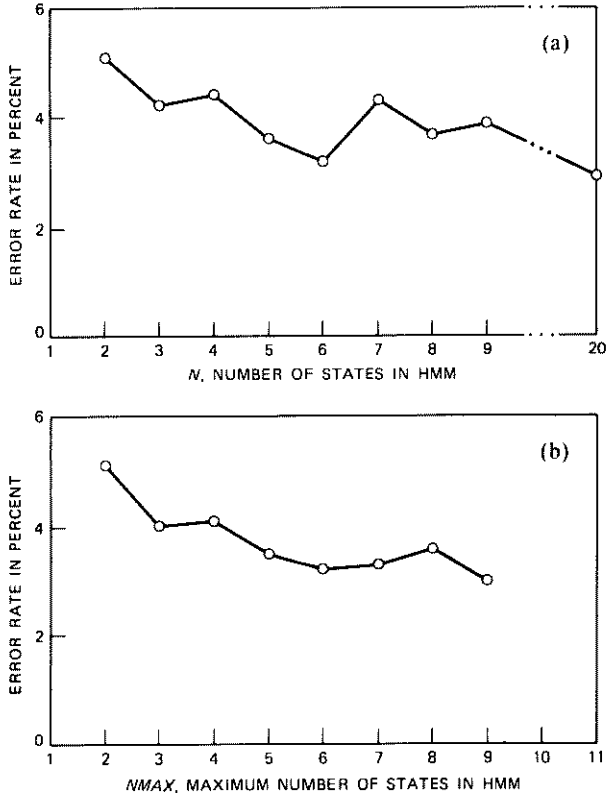



Fig. 12—Plots of (a) error rate versus the number of states in the HMM, and (b) error rate versus the maximum number of states in the HMMs for TS1 data.

errors versus *NMAX* is obtained in Fig. 12b than for the individual models of Fig. 12a; however, the general behavior of both curves is similar.

Figure 13 shows a breakdown of the error rates for each digit for the experiment in which a single HMM, with $N$ states, was used for each digit. There is a highly complex interaction between error rate and model size for all digits. Thus it cannot be argued, for instance, that digits like zero and seven (two syllables) need more states in their models than digits like two or one (monosyllables).

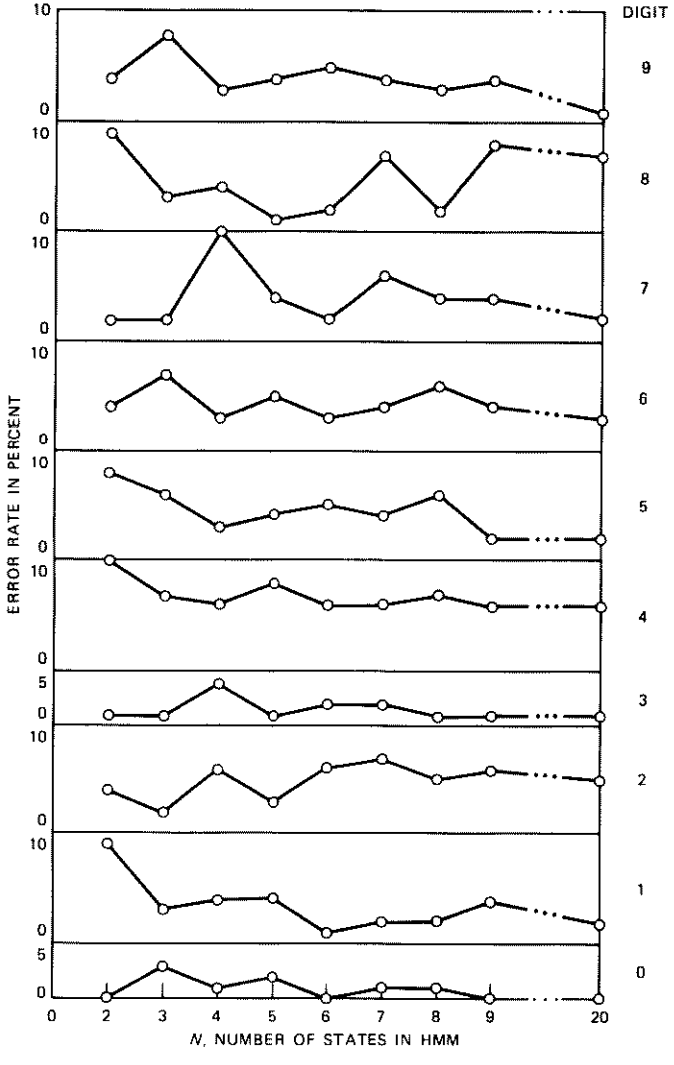



Fig. 13—Individual plots of digit error rates versus the number of states in the HMMs.

From the results shown in Figs. 12 and 13, it was concluded that there is very little gain in using HMMs with more than five or six states when the SC1 structure for each model is being used. It was also concluded that no simple relationship existed between word (digit) accuracy, number of sounds (syllables, etc) in the word, and number of states needed in the word HMM.

### *6.3 Effects of random starting points*

The third set of experiments was concerned with the statistical variability in the performance scores resulting from statistical variability in the parameters of the word HMMs because of different random initial estimates of the parameters. To quantify this effect, a 5-state SC1 model was generated for each digit using 10 different random starting sets of model parameters. Thus, for each digit, 10 "equivalent" HMMs were created.

A recognition test was then run, using TS1 data, in which each of the 10 models was tested separately. Also tested was the case in which all 10 models were used for each digit, as well as the case in which a single model was used for each digit, where the model parameters were obtained by averaging the parameter estimates for each of the 10 word models. The results of these recognition tests are given in Fig. 14, which shows word error rate versus the random start number. Also shown, as single isolated values, are the error scores for the average and the combined 10 state runs. The dashed line in Fig. 14 is the average error rate of the 10 individual models.

From the data of Fig. 14, we can see that the 10 individual models all performed identically to within ±1 percent; hence, the expected statistical variability in error rate scores, due to random starts, should

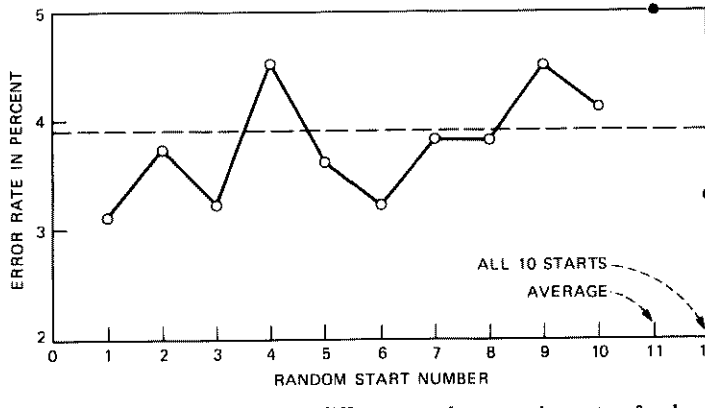



Fig. 14—Plots of error rate for 10 different random starting sets of values of the HMM parameters. Also shown are individual error rates for an averaged model and for combining 10 random start models.

be on the order of ±1 percent. We can also see that the performance of the "averaged" model (obtained by averaging HMM parameters for each word) was somewhat poorer than that of any of the 10 individual models. The fact that using all 10 word models for each digit gives an error rate comparable to that of the *best* word models indicates that multiple word models provide a small gain that comes at the cost of greatly increased computation.

Overall, the data of Fig. 14 suggest that a single model per word should be adequate for most purposes, and that the effects of different random starts on the overall error performance are small.

### *6.4 Parallel constrained HMMs*

The last factor investigated was the effect of using constrained parallel HMMs for each digit. The main idea was that a true parallel structure could model the effects of using a multiplicity of word models in much the same way as multiple templates are used in the conventional LPC/DTW word recognizer.

Figure 15a shows the 5-state constrained serial HMM that was used in previous experiments along with a 7-state, constrained parallel HMM (Fig. 15b), and an 8-state, constrained parallel model (Fig. 15c). The 7-state parallel structure was intended to represent four distinct 5-state word models, in that there were four sets of paths through the model, namely, 1-2-5-7, 1-2-6-7, 1-3-5-7, and 1-3-6-7. The 8-state parallel structure was intended to represent eight distinct 5-state word models in that there were eight sets of paths through this model, namely, 1-2-4-6-8, 1-2-4-7-8, 1-2-5-6-8, 1-2-5-7-8, 1-3-4-6-8, 1-3-4-7-8, 1-3-5-6-8, and 1-3-5-7-8.

Each of the three HMM structures of Fig. 15 was used to generate a word model for each of the digits. The three sets of models were then tested using TS1 data. The results showed each of the three systems obtained the same word error rate (3.5 percent) to within ±0.1 percent. These results indicated that there was really no advantage in using the parallel structure.

### *6.5 Comparison with LPC/DTW recognizers*

To provide some basis of comparison for the performance of the HMM/VQ recognizer with that of more conventional word recognizers, the data of TS1 was tested on the LPC/DTW recognizer of Fig. 1. The reference set consisted of 12 templates per digit, generated from a clustering analysis of the 100 tokens of each digit in the training set (the same training set used to train each word HMM). The decision rule was the nearest neighbor rule (KNN = 1).[17]

Tables I and II show average word recognition accuracies for the following three recognizers:

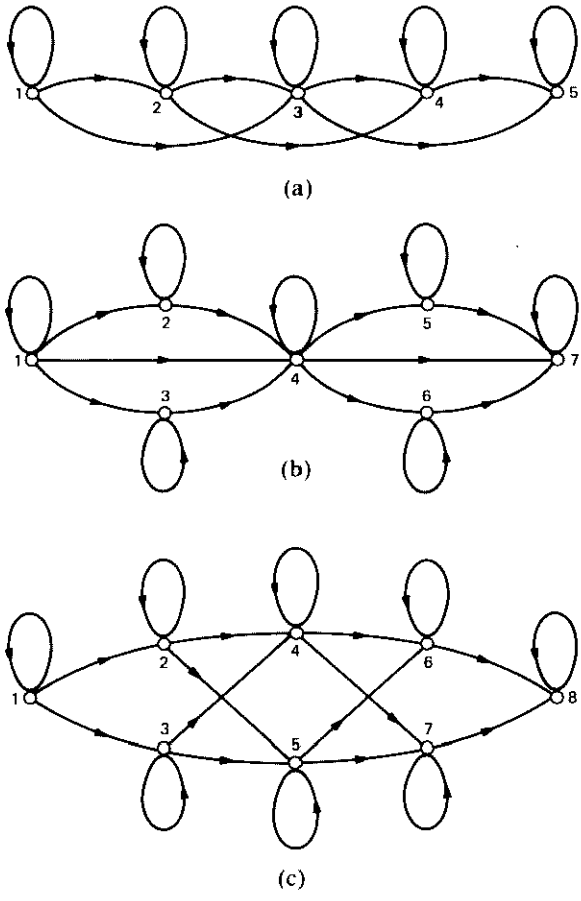



Fig. 15—State diagrams for (a) simple constrained serial 5-state model, (b) constrained parallel 7-state model, and (c) constrained parallel 8-state model.

(*i*) HMM/VQ using a constrained serial structure with five states for the HMM and the 64-element VQ.

(*ii*) LPC/DTW, the conventional recognizer.

(*iii*) LPC/DTW/VQ, the conventional recognizer with both reference and test patterns quantized using the same VQ used in the HMM case.

The results shown in Table I are for the 1000 digits of TS1, whereas those shown in Table II are for the 2000 digits (10 talkers) of TS2. The results given in Table I show that for TS1, both the LPC/DTW and HMM recognizers, when using the VQ, achieved essentially the same digit accuracy; however, the LPC/DTW system without the VQ achieved a 2-percent higher word accuracy. For TS2 in Table II the

Table I—Comparison of results on HMM/VQ and LPC/DTW word recognizers for [average word accuracy (%)] TS1, 100 talkers, 10 digits per talker

| | Recognizer | | |
|---|---|---|---|
| Digit | HMM/VQ | LPC/DTW | LPC/DTW/VQ |
| 0 | 98 | 99 | 99 |
| 1 | 98 | 98 | 99 |
| 2 | 96 | 100 | 96 |
| 3 | 99 | 99 | 97 |
| 4 | 93 | 97 | 96 |
| 5 | 97 | 96 | 93 |
| 6 | 96 | 100 | 94 |
| 7 | 99 | 100 | 94 |
| 8 | 92 | 98 | 96 |
| 9 | 95 | 98 | 96 |
| Average | 96.3 | 98.5 | 96.5 |

Table II—Comparison of results on HMM/VQ and LPC/DTW word recognizers for TS2, 10 talkers, 200 digits per talker

| | Recognizer | | |
|---|---|---|---|
| Talker | HMM/VQ | LPC/DTW | LPC/DTW/VQ |
| 1 | 74.5 | 96 | 87 |
| 2 | 99.5 | 100 | 99.5 |
| 3 | 94 | 99 | 97 |
| 4 | 89 | 99 | 91 |
| 5 | 95.5 | 100 | 100 |
| 6 | 95.5 | 99 | 99.5 |
| 7 | 100 | 100 | 99.5 |
| 8 | 91.5 | 96 | 93 |
| 9 | 91.5 | 99.5 | 93.5 |
| 10 | 96.5 | 98.5 | 98 |
| Average | 92.8 | 98.7 | 95.5 |

results show that the VQ led to a 3.2-percent reduction in accuracy for the LPC/DTW system, and an additional 2.7-percent loss in accuracy for the HMM system. A good deal of the loss in accuracy, however, was contributed by talker 1, whose accuracy was 87 percent for the LPC/DTW/VQ recognizer, and 74.5 percent for the HMM/VQ recognizer. With only 10 talkers, the influence of a single talker on the overall accuracy may be substantial.

An analysis of the actual errors of all three of the recognizers of Tables I and II shows the following:

(*i*) Of the 37 tokens misclassified by recognizer HMM/VQ in TS1, 31 were correctly identified by the LPC/DTW recognizer, and 27 were correctly identified by the LPC/DTW/VQ recognizer.

(*ii*) The vast majority (25) of the 37 errors made by recognizer HMM/VQ were cases in which the probability of the correct word was

lower than the probability of the incorrect word by a factor of $e^6$ or larger.

The results show that when the LPC/DTW recognizer (either with or without the VQ) has incorrectly identified the word, most of the time the HMM recognizer has correctly identified the word. Hence, a side result of this work is that the HMM/VQ model can be combined with an LPC/DTW model to provide word accuracies greater than either individual recognizer could obtain. In particular, for the data of TS1, the combined recognizer could have achieved a 99.4-percent word accuracy by using appropriate decision logic[22] on all cases in which both recognizers did not agree.

The same sort of trends are noted in the errors of TS2. Of the 124 errors made by the HMM/VQ recognizer, 113 were correct in the LPC/DTW or equivalently of the 26 errors made by the LPC/DTW recognizer, 15 were correct in the HMM/VQ system. Hence, again a combined system could potentially achieve an accuracy of about 99.5 percent on TS2 data. Considering that even the talkers in the TS2 data were different from those in the training set, this accuracy appears to be quite remarkable.

In summary, comparisons between the HMM/VQ and LPC/DTW recognizers indicate that without the VQ, the LPC/DTW recognizer achieves from 2- to 6-percent higher accuracy than the HMM/VQ system; with the VQ the differential in accuracy is from 0 to 3 percent.

## VII. DISCUSSION

In this paper we showed that the techniques of vector quantization of LPC vectors and hidden Markov modeling can be combined in a simple, straightforward manner to implement a speaker-independent, isolated word recognizer. With adequate training of the vector quantizer and the Markov model estimation algorithm, a digits vocabulary can be recognized, with accuracies of from 93 to 96 percent across a wide variety of talkers. Direct comparisons with a conventional linear predictive coding recognizer using dynamic time warping for time alignment with multiple templates for each vocabulary word showed that the HMM/VQ recognizer performs only a little worse (0.2 percent in one test, 2.7 percent in another) than the LPC/DTW recognizer when using the VQ. Without the VQ, the LPC/DTW recognizer was about 2 to 3 percent better than when the VQ was used.

Several general conclusions can be drawn from the results. The first is that the HMM/VQ recognizer performed exceedingly well on the difficult task of speaker-independent recognition of isolated digits. The fact that the overall performance of the HMM/VQ recognizer was somewhat poorer than the LPC/DTW/VQ recognizer appears to be primarily because of the insufficiency of the HMM training data.

Although 100 training sequences per word are adequate for a clustering analysis, as used in the conventional recognizer, they appear to be inadequate for obtaining good HMM models for these words. This suggestion is made plausible by considering what is being estimated in the HMM for each word. For an $N$-state Markov model, with $M$ finite symbols per state, a total of $N^2 + NM$ parameters must be estimated. (Of course with constraints there are somewhat fewer parameters.) For $N = 5$, $M = 64$, we need 345 parameters to be estimated from about $100 \times 40$ frames of VQ indices. The "curse of dimensionality" would imply that this amount of training data is woefully inadequate. [We have seen one consequence of this inadequacy in that we had to use the $\epsilon$-constraints of eq. (10) on any $b_j(k)$ whose value fell below the $\epsilon$ threshold.] In view of this, the fact that we achieve the results we are getting is rather remarkable.

A second conclusion that can be drawn from the results is that the use of the VQ on the LPC sets leads to a small, but not insignificant, degradation in performance of both the HMM/VQ and LPC/DTW/VQ recognizers as compared to the conventional LPC/DTW system. This suggests the need for using more than 64 vectors in the codebook or resorting to continuous models of the LPC parameters.

The results have shown that the errors made by the HMM/VQ and LPC/DTW recognizers are largely disjoint. Here there exists the potential of using some fairly standard techniques to combine the two recognizers into one whose accuracy is as good as the best of both recognizers on any given word.[22] This topic merits further consideration.

The experimentation with various forms of the Markov models used in the recognizer showed fairly conclusively that:

(*i*) Constrained models (with constrained transition matrices) performed consistently better than unconstrained models.

(*ii*) A finite minimum constraint on the state symbol probability matrix was a necessity for good system performance.

(*iii*) The effects of different random starting values for the HMM parameters were negligible in evaluating overall recognizer performance.

(*iv*) The required number of states in each word HMM needed to be on the order of 5. More states did not lead to significant improvements in performance.

(*v*) Parallel HMM structures yielded no real improvements over cascade structures, thereby indicating that an equivalent of multiple HMMs is not readily obtainable by simply changing the model structure.

(*vi*) The Viterbi scoring and the Baum-Welch scoring of test sequences give essentially identical performance.

### 7.1 Computational considerations of the HMM/VQ recognizer

It is worthwhile estimating the storage required and the computation needed to process an unknown test utterance using the HMM/VQ recognizer, and to compare them to the requirements of the conventional LPC/DTW recognizer. Our intention is to provide only a rough estimate of computational expense. A number of straightforward reductions in computation can be achieved for each recognizer through the judicious use of table storage. Also we ignore overhead owing to index computation, etc.

The first extra step in the HMM/VQ recognizer (after conventional LPC analysis) is vector quantization of the unknown test pattern. If we assume there are $T$ frames in the test word, and $M$ codebook entries in the VQ, then we need a total of

$$C_1 = M \cdot T(p + 1) \tag{17}$$

multiplications* (where $p$ = LPC order) to perform the $M \cdot T$ dot products required to get the best codebook entry for each frame.

Evaluation of the word HMM score, using the Viterbi scoring method with the constrained $A$ matrix, requires approximately

$$C_2 = T \cdot N \cdot 3 \tag{18}$$

multiplications and logarithms per word model, where $N$ is the number of states in the model, and the factor 3 accounts for the number of valid transitions into a given state. For a vocabulary of $V$ words a total of

$$C_* = M \cdot T \cdot (p + 1) + V \cdot T \cdot N \cdot 3 \tag{19a}$$

$$C_{\log} = V \cdot T \cdot N \cdot 3 \tag{19b}$$

multiplications and logarithms are required. For $M = 64$, $T = 40$, $V = 10$, $N = 5$, and $p = 8$, eq. (19) gives $C_* = 29040$ multiplies and $C_{\log} = 6000$ logarithms.

For a conventional LPC/DTW recognizer with $Q$ templates per vocabulary word, the computation for DTW processing is

$$C_3 \cong Q \cdot V \cdot T^2/3 \cdot (p + 1), \tag{20}$$

which for $Q = 12$ and other parameters the same as above gives $C_3 = 576000$ multiplications. Hence, the HMM/VQ recognizer requires about 17 times less computation (assuming logarithms are equivalent to multiplications) than the LPC/DTW recognizer.

With regard to storage, the HMM/VQ recognizer requires

* In this simplified analysis we neglect additions and comparisons of data and use multiplication count as the measure of computation.

$$S_1 = M(p + 1) \tag{20a}$$

$$S_2 = (M \cdot N + 3N) \cdot V, \tag{20b}$$

where $S_1$ is the storage (in words*) for the VQ codebook entries, and $S_2$ is the storage (in words) for the set of $V$ word HMMs. For the given values of the parameters, the total storage is $S = S_1 + S_2 = 4106$ words.

For the conventional LPC/DTW recognizer the storage is

$$S_3 = Q \cdot V \cdot T \cdot (p + 1), \tag{20c}$$

which gives 43,200 words for the assumed parameter values. Again we see a 10 to 1 reduction for the HMM/VQ model over the LPC/DTW model.

It should be noted that in our analysis of computation we have not included the computation of LPC coefficients. This computation must be performed in "real-time" and is independent of the vocabulary size, $V$. Hence, for a sufficiently large vocabulary the computation for scoring each word dominates the overall computation, and the rough analysis given above is appropriate. Furthermore, the computation for coding each LPC vector into the nearest codebook entry [eq. (17)] is also independent of vocabulary size and often could be performed in the "real-time" part of the recognizer. For such implementations the gain in computation of the HMM/VQ recognizer, over the conventional LPC/DTW recognizer, is even higher than our simple analysis predicts. Finally, it is straightforward to show that if we compare the computation of the HMM/VQ recognizer with that of the LPC/DTW/VQ recognizer, assuming that the VQ is done in real time and that tabular computation of products and logarithms is used, then by comparing the number of additions, the computational advantage of the HMM/VQ system over the LPC/DTW/VQ system still holds.

### *7.2 Some comments on the relationships between DTW and HMMs*

Contemporary research on speech recognition has produced two algorithmic procedures for dealing with the nonstationarity of the speech signal: temporal alignment techniques, and Markov modeling. These methods display certain superficial similarities (e.g., both use dynamic programming methods, can be cast in a Bayesian framework, and have a state transition network associated with them), as a result of which it has occasionally been claimed that they are identical. To the best of our knowledge, the experiments reported here represent the first direct comparison of the two methods. From these experiments it is abundantly clear that the methods are not identical. While

* Words of storage refer to unquantized floating point data.

their overall performances are comparable, they appear to make different errors and involve different amounts of computation and different complexities of training.

In all of these respects the two methods reflect the dichotomy between parametric and nonparametric methods of pattern recognition in the sense of Patrick.[23] In Markov modeling we assume that there is a family of models of a particular structure differing only in the values of their parameters. We use a large training set to estimate these parameters and assume that, if correctly done, the parameters will capture the structure of the data. The training procedure is computationally expensive but need be done only once. After training is complete, relatively little computation is required to determine whether an unknown observation was generated by the model.

Temporal alignment methods are opposite in the following ways. We assume that there is an underlying structure to the training data but its form is unknown. We attempt to capture that structure by simply storing one or more samples and measuring their "distance" to an unknown sample with a metric that is sensitive to the distinctive features of the categories that we seek to identify. The metric is monotonically related to the class conditional density functions so that minimum distance corresponds to maximum model probability. In this case training is a computationally simple data collection and storage process. Probability computation, on the other hand, is very costly since we must measure the distance to every prototype in the training set.

All of these characteristics are made manifest by our experiments. What remains to be determined is whether the parity of these methods extends to more difficult problems of speech recognition. We hope to answer that question by further experimentation.

## VIII. SUMMARY

We have described the results of an extensive investigation into the applicability of the techniques of vector quantization and hidden Markov modeling to speaker-independent, isolated word recognition. We have shown that, when properly designed, the resulting recognition system produces highly accurate word recognition on a vocabulary of isolated digits. We have also discussed the effects of variations of model parameters on system performance. Our experiments show that the resulting recognizer requires about 10 times less storage, and about 17 times less computation for classifying a test utterance than does an equivalent recognizer using LPC coding and dynamic time warping. These economies are obtained at the expense of only a slight increase in error rate.

## REFERENCES


1. T. B. Martin, "Practical Applications of Voice Input to Machines," Proc. IEEE, *64* (April 1976), pp. 487–501.
2. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. on Acoustics, Speech, and Signal Processing, *ASSP-23*, No. 1 (February 1975), pp. 67–72.
3. L. R. Rabiner and S. E. Levinson, "Isolated Connected Word Recognition—Theory and Selected Applications," IEEE Trans. on Communications, *COM-29*, No. 5 (May 1981), pp. 621–59.
4. N. R. Dixon and T. B. Martin, Eds., *Automatic Speech and Speaker Recognition*, New York: IEEE Press, 1979.
5. W. Lea, Ed., *Trends in Speech Recognition*, Englewood Cliffs, NJ: Prentice Hall, 1980.
6. D. R. Reddy, Ed., *Speech Recognition*, New York: Academic Press, 1974.
7. A. Newell et al., "Speech Understanding Systems," Final Report of a Study Group, Carnegie Mellon Report, May 1971.
8. M. R. Sambur and L. R. Rabiner, "A Speaker-Independent Digit Recognition System," B.S.T.J., *54*, No. 1 (January 1975), pp. 81–102.
9. T. B. Martin, "Acoustic Recognition of a Limited Vocabulary in Continuous Speech," Ph.D Dissertation, Univ. of Penn., 1970.
10. A. B. Poritz, "Linear Predictive Hidden Markov Models and the Speech Signal," Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 82, Paris, France (May 1982), pp. 1291–4.
11. J. K. Baker, "The Dragon System—An Overview," IEEE Trans. on Acoustics, Speech, and Signal Processing, *ASSP-23*, No. 1 (Feburary 1975), pp. 24–9.
12. F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a Linguistic Decoder for the Recognition of Continuous Speech," IEEE Trans. on Information Theory, *IT-21* (May 1975), pp. 250–6.
13. S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," B.S.T.J., this issue.
14. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," IEEE Trans. on Acoustics, Speech, and Signal Processing, *ASSP-29*, No. 4 (August 1981), pp. 777–85.
15. L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," J. Acoust. Soc. Amer., *68*, No. 5 (November 1980), pp. 1271–6.
16. S. E. Levinson, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Iteractive Clustering Techniques for Selecting Speaker Independent Reference Templates for Isolated Word Recognition," IEEE Trans. on Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 2 (April 1979), pp. 134–41.
17. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. on Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 4 (August 1979), pp. 336–49.
18. L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Annals of Mathematical Statistics, *41*, No. 1 (1970), pp. 164–71.
19. A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech Coding Based Upon Vector Quantization," IEEE Trans. on Acoustics, Speech and Signal Processing, *ASSP-28*, No. 5 (October 1980), pp. 562–74.
20. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. on Information Theory, *IT-13* (April 1967), pp. 260–9.
21. B. H. Juang, D. Y. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," IEEE Trans. on Acoustics, Speech, and Signal Processing, *ASSP-30*, No. 2 (April 1982), pp. 294–304.
22. M. K. Brown and L. R. Rabiner, "On the Use of Energy in LPC Based Recognition of Isolated Words," B.S.T.J., *61*, No. 10, Part 1 (December 1982), pp. 2971–87.
23. E. A. Patrick, *Fundamentals of Pattern Recognition*, Englewood Cliffs, NJ: Prentice Hall Inc., 1972.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 4, April 1983
*Printed in U.S.A.*

# A Lithographic Mask System for MOS Fine-Line Process Development

By J. M. ANDREWS

(Manuscript received September 23, 1982)

*A mask set, incorporating a group of seven test chips, has been designed for fine-line process development and process control. Although six lithographic levels are available, the masks are generally intended to be used only in subsets of two or three levels to minimize the delay encountered in obtaining electrical test results for whichever processes require investigation. The mask levels serve a variety of purposes for special process development experiments. Available structures include: metal-oxide-semiconductor capacitors, p-n junctions, guarded and unguarded Schottky barrier diodes, ohmic contacts, van der Pauw patterns, insulated gate field-effect transistors, gated diodes, resistors for sheet resistance and linewidth variations, and tapped electromigration test strings. It is not anticipated that a process engineer should ever need more than a maximum of four levels to achieve an appropriate experimental structure for process development. It is not the purpose of these masks to establish fine-line design rules. The masks are intended to be used primarily with standard photolithographic processing, and most device structures have been designed to tolerate up to 5 μm in misalignment errors. However, certain selected features have been coded in a diminishing sequence to a minimum of 1.0 μm for special fine-line investigations. A salient feature of this mask system is the option to interleave rapid turnaround photolithographic steps with fine-line X-ray patterning; therefore, some mask levels have been reissued for X-ray lithography.*

## I. INTRODUCTION

In the past, the development of new silicon integrated circuit processes was impeded by the fact that an adequate set of simple test structures usually could not be fabricated without resorting to the full set of six lithographic levels required by the Poon Tester Chips.[1] This set sometimes requires several months to fabricate if X-ray lithography

is used. If device wafers could not be sacrificed, the processing engineer had to resort to simulating device structures, either by using metal dots of fixed areas on unpatterned oxides or deposited films, or by other schemes such as the use of offset circular windows using a pair of photolithographic steps.[2,3]

A set of photolithographic masks has been designed and is now available to fill the process development gap. The goal has been to provide the processing engineer with the means to simulate critical processing steps by introducing a monitor wafer, prepared in advance by one photolithographic step, and usually requiring only one more lithographic step of any type to obtain a structure ready for electrical testing.

The full set of fine-line process development masks consists of six photolithographic levels, but these have been designed to be utilized in subsets of 2, 3, or 4 levels only. Available structures include metal-oxide-semiconductor (MOS)* capacitors, contact windows, guarded and unguarded Schottky diodes, van der Pauw patterns, insulated gate field-effect transistors (IGFETs), gated diodes, and tapped electromigration test strings. Also available are large areas accessible for direct probing, for evaporation of MOS dots, or for Auger, scanning electron microscope (SEM), and transmission electron microscope (TEM) studies.

Section II contains a complete description of each of the devices that are available in the MOS mask set. The organization of the devices among each of seven test chips, the chip designations, the device assignments, wafer layout, design rules, and the alignment features are discussed in Section III. Section IV consists of a detailed description of each mask level separately, including the primary purpose for which the level is intended and the features available. Section IV also includes specific recommendations regarding which levels can be omitted with respect to the particular devices required or the experimental intent. Specific applications to silicon integrated circuit processing are discussed in Section V, and the MOS mask system is summarized in Section VI.

## II. DEVICE DESCRIPTION

Most of the test structures in the lithographic mask system are MOS capacitors, because such two-terminal devices are easily fabricated. Normally, only two photolithographic operations are needed, and the first could usually be done on a large number of wafers before specific experiments are planned. Furthermore, the same patterns can be used

* Acronyms and abbreviations are defined fully in the Glossary at the end of this paper. Tables and figures are located at the end of the text.

to fabricate p-n junctions, Schottky diodes, or ohmic contacts to the substrate by simply omitting the gate oxidation. All such devices, with dimensions ranging from 1 to 500 $\mu$m, are contained on a single chip designated A. Larger devices, with dimensions ranging from 1000 to 4000 $\mu$m, are included on chips B and C1 through C4. The aim has been to provide maximum experimental flexibility by including a wide range of available device dimensions, which permits electrically active areas to span more than seven orders of magnitude. A summary of all devices contained in the MOS mask system is presented in Table I in the form of a device key, listing MOS device, nominal dimension, the chip assignment, and the appropriate probing pad number. Detailed descriptions of each device follow in Subsections 2.1 to 2.5. Dimensional data on all device structures are contained in the pad keys shown in Tables II through V.

### *2.1 Sixfold MOS capacitor group (HEXCAP)*

The design of the lithographic mask system has evolved from a sixfold set of MOS capacitors (HEXCAP) that can be implemented at almost any stage of device processing to provide electrical characterization of dielectric layers.

#### *2.1.1 FOXCAP and GOXCAP*

The field oxide capacitor (FOXCAP) and gate oxide capacitor (GOXCAP) are shown in Fig. 1. For this pair of capacitors, the dimension $L_1$ increases through the sequence 50, 100, 200, 500, 1000, 2000, and 4000 $\mu$m. In all cases the dimension $L_2 = L_1 + 10$ $\mu$m, to conform with relaxed design rules to minimize registration errors (see Section 3.3). All FOXCAP and GOXCAP devices with $L_1 \leq 500$ $\mu$m are contained on the A chip, with peripheral probing pad locations coinciding with the Poon Tester[1] and the Process Monitor[4] chips. To facilitate MOS characterization of thick oxides or deposited dielectrics, a limited number of HEXCAP devices with $L_1$ equal to 1000 and 2000 $\mu$m were assigned to a larger chip B with area equal to that of ten standard Poon chips. The largest FOXCAP and GOXCAP devices on the C1 chips (see Section 3.1.3) are not square. Rectangular GOXCAP dimensions $L_3$ and $L_4$ for the largest devices have been selected so that

$$L_3 L_4 = (4000\ \mu m)^2. \tag{1}$$

The rectangular structures of FOXCAP and GOXCAP are shown in Fig. 16. Specific values for $L_3$ and $L_4$ were selected so that the overall C1 chip dimensions could be adjusted to accommodate the contact metallization test chip (METEST) (see Sections 3.1.4 and 3.2).

In each HEXCAP group, the portions of the polysilicon (POLY) areas that overlay the gate and source and drain (GASAD) areas are

equal. Thus, the capacitors formed over gate oxide ($C_{GOX}$) are equal among all elements of each HEXCAP group except FOXCAP. Furthermore, the total POLY areas are equal for FOXCAP, GOXCAP, and the devices with comb-shaped electrodes (POLYCOMB) to facilitate use with automatic testing programs. Thus, constant parasitic capacitance contributed by the field oxide ($C_{FOX}$) in parallel with $C_{GOX}$ has been maintained among GOXCAP, GOXCOMB, and POLYCOMB (see following Section 2.1.2) for convenience in software development. Detailed dimensional data are contained in the pad keys, Tables II, III, and IV.

It should be clear from Fig. 1 that omission of the gate oxide in GOXCAP results in a structure suitable for ohmic contact, Schottky barrier diode, and p-n junction experiments. For this reason the dimension $L_1$ also decreases through the sequence 20, 15, 10, 7, 5, 4, 3, 2, 1.5, and 1.0 $\mu$m on the A chip. It is not expected, however, that the smallest GOXCAP windows would be routinely resolved by standard photolithographic processes. To resolve the smallest features, the GASAD mask level has been reissued for X-ray lithography. Because the X-ray target is nearly a point source, the finite separation between X-ray mask and the silicon substrate results in a slight magnification of the X-ray image. To compensate for the magnified X-ray image, a small demagnification of all features on the X-ray mask was required to make the X-ray lithographic level compatible with previous or subsequent optical lithographic levels. Alignment in the X-ray exposure facility, however, is done optically. Therefore the spacing between alignment features on the X-ray mask must remain the same as on the optical mask. With these modifications, X-ray lithography can be interleaved with more easily performed photolithographic steps.

The use of the GOXCAP structure to form a guarded Schottky diode is illustrated in Fig. 2, in which a guard ring with width $W$ is diffused into the substrate before GASAD lithography. Guard-ring width options available in the mask set decrease through the sequence 10, 7, 5, 3, 2, and 0 $\mu$m (see Section 4.1). In all cases the guard ring has been centered on the GASAD boundary so that half its width extends into the contacting metal and reduces the effective contact area (see Fig. 2), so that:

$$A_{\text{eff}} = (L_1 - W)^2 \quad (\mu\text{m}^2). \tag{2}$$

For the cases in which $W \geq L_1$, the structure simply reduces to a p-n junction. Square or rectangular features have also been included in the GUARDRING level under the FOXCAP structures to provide the option of fabricating buried channel MOS capacitors[5–8] or to investigate the profiles of various ion implantations, such as those used to control threshold or punchthrough or for depletion loads.

### 2.1.2 GOXCOMB and POLYCOMB

To study peripheral effects or defects, structures with expanded gate-oxide/field-oxide periphery were included as the third HEXCAP element. This structure, abbreviated GOXCOMB, is shown in Fig. 3. On the A chip, the GOXCOMB structures comprise rectangular elements of width $d_1 = 25$ μm, which are spaced $d_2 = 10$ μm apart. The total gate oxide areas of the two GOXCOMB structures on the A chip are $(200\ \mu m)^2$ and $(500\ \mu m)^2$. To further enhance peripheral effects, GOXCOMB structures have been included on the B and C2 chips with $d_1 = 5$ μm and $d_2 = 10$ μm. Because of the decreased filling factor associated with reducing $d_1$ to 5 μm, it was not practical to keep the total gate oxide areas equal to the areas of the associated 1000-, 2000-, and 4000-μm GOXCAP structures; the actual areas are listed in Tables III and IV.

The POLYCOMB structure is similar to GOXCOMB except that the increased perimeter or peripheral expansion occurs at the gate-oxide/polysilicon boundary. The structural detail of POLYCOMB, the fourth HEXCAP element, is shown in Fig. 4. The chip assignments for POLYCOMB are the same as for GOXCOMB except for the 4000-μm structure, which is on the C3 chip. The same values for $d_1$ and $d_2$ apply to both COMB structures.

### 2.1.3 OVLAP and NOVLAP with FIELD PLATE

In some cases it is desirable to minimize parasitic capacitance in an MOS structure, i.e., the parallel capacitance composed of the area in which the gate electrode overlaps field oxide. The last pair of HEXCAP devices has been designed to minimize parasitic capacitance, consistent with the design rules discussed in Section 3.3. The OVLAP capacitor, shown on the left in Fig. 5, has a 5-μm overlap of the gate electrode onto the surrounding field oxide. The NOVLAP capacitor, shown on the right, has a gate electrode that has been retracted 5 μm from the GASAD boundary. The portion of the gate electrode that covers gate oxide in the OVLAP capacitor is equal to the area of the gate electrode in the NOVLAP capacitor. Both OVLAP and NOVLAP must be probed directly, because they are completely surrounded by a field plate that can be used to control the surface potential near the edges of each capacitor.

If a metallic silicide is formed in place of gate oxide, the resulting structure allows investigation of the effects of overlying metallization when excessive metallic penetration at contact window edges is suspected.[9] For these investigations it may be useful to include the GUARDRING option prior to silicide formation.

### 2.2 Sheet resistance group (SADSHEET and POLYSHEET)

Two three-terminal structures on the A chip can be used to obtain sheet resistance data from lines 400 $\mu$m long. Thus, accurate sheet resistance measurements can be made, even though the linewidths may deviate from the coded values because of unknown degrees of overetching or other process variations.

The left side of Fig. 6 shows the structure for measuring polysilicon sheet resistance (POLYSHEET). It consists of two 400-$\mu$m lines in series, but the coded linewidths are different: $W_1 = 5$ $\mu$m and $W_2 = 8$ $\mu$m. After processing, the actual linewidths may differ from the coded linewidths by a constant amount, $\epsilon$. Assume that a positive value for $\epsilon$ corresponds to a linewidth loss from the coded value, $W_i$. If the resistance of each line is measured, it is possible to solve for both the sheet resistance and a constant linewidth loss, $\epsilon$. It can be shown that

$$R_S = \frac{R_1 R_2}{R_1 - R_2} \cdot \frac{W_2 - W_1}{L} = 7.5 \times 10^{-3} \frac{R_1 R_2}{R_1 - R_2} \ (\Omega/\square) \tag{3}$$

and

$$\epsilon = \frac{R_1 W_1 - R_2 W_2}{R_1 - R_2} = \frac{5R_1 - 8R_2}{R_1 - R_2} \quad (\mu\text{m}). \tag{4}$$

A GUARDRING structure is included beneath some of the POLYSHEET lines. If the GUARDRING option were used, for example, to etch channels of various widths in a field oxide, the resulting POLYSHEET structure would provide information on poly-Si linewidths within oxide channels or straddling oxide steps.

Source-and-drain sheet resistance and linewidth variations can be determined from measurements on the structure shown on the right side of Fig. 6. The coded dimensions of the SADSHEET lines are exactly the same as for POLYSHEET, and eqs. (3) and (4) apply.

When the GUARDRING level precedes GASAD, some of the SADSHEET lines are imbedded into the guard-ring diffusion. Such a structure could be useful in determining the sheet resistance of a metallic silicide line for cases in which a low Schottky barrier height between the silicide and the substrate would interfere with electrical measurements.

### 2.3 van der Pauw group (VANDERPAUW)

The four-terminal symmetric structure shown in Fig. 7 has been provided on the A chip to make accurate determinations of polycrystalline silicon (poly-Si) sheet resistance in a way that is independent of the actual shape of the resistive pattern.[10] The poly-Si lines have been extended to the probing pads so that window and metallization lithography is not necessarily required. However, the option to have

overlying metallization on the lines leading to the van der Pauw pattern, with windows to the underlying poly-Si, is available for unusual circumstances in which polysilicon sheet resistance may be very high.

The other VANDERPAUW structures are on the C4 chip, and consist of GUARDRING and GASAD patterns. The combinations POLY/GUARDRING and POLY/GASAD VANDERPAUW patterns are also included on the C4 chip to enable measurements of inversion layer sheet resistance[11] and to investigate CHANSTOP performance.

### *2.4 IGFET group*

It is not anticipated that the IGFET group of devices will be utilized as often as the HEXCAP group of MOS capacitors, because the IGFETs require a minimum of four mask levels (GASAD, POLY, WINDOW, and METAL). For this reason, all IGFETs have been relegated to the B chip that has more available terminals than the A chip, although it may be less convenient for automatic probing.

Most of the IGFETs are included in one group with common sources and gates. The structure of the IGFET with $L = 20$ $\mu$m is shown in Fig. 8. All gates are 100 $\mu$m wide, and the gate lengths $L$ descend through the sequence 20, 15, 10, 8, 6, 5, 4, 3, 2, 1.5, and 1.0 $\mu$m. It is not anticipated, however, that the shortest gates will be resolved with ordinary photolithographic processing. Therefore, the POLY mask level may also be reissued for X-ray lithography. Even with wide variations in processing, the range of gate lengths provides a means to determine the true (electrically active) channel length from a plot of $\beta^{-1}$ versus the coded value for $L$, where $\beta$ is the transconductance of the IGFET.

The GUARDRING option is available on all elements of the IGFET group. The guard ring straddles the GASAD feature on the three sides that are not adjacent to the gate, as shown in Fig. 9. Such a structure may be useful to minimize edge leakage when Schottky barrier sources and drains are investigated.

In some cases it is useful to make C-V (capacitance measured as a function of voltage) measurements of the gate electrode in an active IGFET. But practical IGFETs are generally designed to minimize gate capacitance in order to maximize switching speed, and the true gate capacitance is difficult to separate from parasitic capacitance. Therefore, four large-gate IGFETs have also been included on the B chip with gate dimensions descending through the sequence 500, 300, 200, and 100 $\mu$m square.

### *2.5 Gated diode group (GATODE)*

The measurement of the depleted surface recombination velocity

$s_0$[12] is especially useful in evaluating the effectiveness of a low-temperature anneal to reduce surface state density[13] and to investigate the effects of radiation damage.[14,15] In the determination of $s_0$ it is necessary to directly control the surface potential near an MOS capacitor by means of a third electrode. This option has been made available by means of the gated diode group (GATODE), with dimensions decreasing through the sequence 500, 300, 200, and 100 $\mu$m square. The structure of the 100-$\mu$m gated diode is shown in Fig. 10. Obviously, a nearly equivalent structure could be realized by simply shorting the source to the drain of one of the large area IGFETs. The GATODE structures on the B chip differ from the IGFETs, however, in that the source-and-drain diffusions completely surround the gate electrode except for a 10-$\mu$m tab that connects the gate electrode to the probing pad. The GATODE structures are inverted from the usual gate-controlled diode in the sense that the p-n junction surrounds the gate electrode, whereas, the original gate-controlled diode structure consisted of an MOS capacitor in the form of a ring that surrounded a p-n junction.[16,17] The advantage of the GATODE structure is that better control of minority carrier production is possible when primary interest is centered on the properties of deeply depleted MOS capacitors.[18,19]

### *2.6 Contact metallization test chip (METEST)*

Electromigration studies generally require high current densities, of the order of $10^6$ A/cm$^2$, to achieve accelerated aging at a practical rate.[20] In the vicinity of contact windows, electromigration has been difficult or impossible to study, because the only test structure available has been the 100-window arrays on the Poon Tester A and C chips.[1] At the required current density, the sum of the voltage drops accumulated over a 100-window array often exceeds the breakdown voltage of the p-n junction that exists beneath each pair of windows. The contact metallization test chip D (METEST) has been designed to avoid large accumulated voltage drops by means of a tapped string of metal-to-diffusion windows, as shown in Fig. 11. Each tapped string is composed of series combinations of 1, 2, and 4 contact cells. Structural detail of one such contact cell is shown in Fig. 12 for a window size of 7 $\mu$m. With the D chip, a reliability engineer can select 2, 4, 6, 8, 10, or 14 windows in series, depending upon the particular breakdown characteristics of the structure. Each tapped string has been reproduced for a variety of contact window sizes, decreasing through the sequence 7, 5, 3, 2, 1.5, and 1.0 $\mu$m square. It is not expected, however, that the smallest windows would be routinely resolved by standard photolithographic processes. Therefore, the POLYCON mask level, which contains the contact windows for the METEST structures, has been reissued for X-ray lithography. The X-ray alignment features

have been modified appropriately to make the X-ray lithographic level compatible with prior GASAD and subsequent POLY optical lithographic levels.

There has been a tendency to avoid rectangular contacts with large aspect ratios, i.e., $L/W > 3$. The reason is related to photolithographic exposure problems with very small contact windows, such as $W \leq 2$ $\mu$m. When additional contact area is required, parallel strings of square contact windows are often preferred to large, rectangular contacts. For this reason, the smallest contact windows on the D chip have been repeated in multiples of 4, 6, and 8 parallel windows for the 2-, 1.5-, and 1.0-$\mu$m windows, respectively. Structural detail of a multiple-window contact cell is shown in Fig. 13 for 4-window, 2-$\mu$m contacts. Obviously, the current does not divide evenly among the windows in such a multiple-window structure, but the extra contact windows can be regarded as providing an experimental backup when the first window fails. The multiple window strings also tend to increase continuity probability when working close to the limit of lithographic resolution.

## III. ORGANIZATION

The lithographic mask system for fine-line process development has been organized on the wafer so that the simplest structures with the most convenient dimensions are available together on chip A. The included structures are HEXCAP, SADSHEET, POLYSHEET, and VANDERPAUW (see Section II). Perhaps the most unusual feature of the mask organization has stemmed from the enormous range of device sizes that have been made available to maximize experimental flexibility. Thus, structural dimensions ranging from 1 $\mu$m to 4000 $\mu$m are all present together on the same wafer. Furthermore, the largest areas can be used for direct probing, for evaporated MOS dots, or can be easily cleaved for Auger, SEM, TEM, X-ray, and other analytical investigations. It is the large range of device sizes (4-1/2 orders of magnitude) which has dictated chip designation and wafer layout.

### *3.1 Test chip designation*

#### *3.1.1 Chip A (1600 × 4096 $\mu$m)*

Most of the devices with dimensions ranging between 1.0 and 500 $\mu$m are included on the A chip. A composite view of the POLY and WINDOW levels of the A chip is shown in Fig. 14. Both the dimensions of this chip and the placement of the 36 probing pads have been selected to coincide with the Poon Tester[1] and the Process Monitor[4] chips to facilitate automated probing with existing probe cards.

#### *3.1.2 Chip B (8000 × 8192 $\mu$m)*

The dimensions of the B chip are integral multiples (5 × 2) of the A

chip to facilitate wafer layout (see Section 3.2), and the area is equal to that occupied by ten A chips. A composite view of the POLY, WINDOW, and METAL levels of the B chip is shown in Fig. 15. The size of the B chip has been selected to accommodate HEXCAP groups measuring 1000 and 2000 $\mu$m square, in the case of FOXCAP, GOXCAP, OVLAP, and NOVLAP. The GOXCOMB and POLYCOMB capacitors are nearly square, and have been laid out so that the equivalent areas are equal to the areas of the square capacitors (see Table III). Please note that to provide adequate resolution for illustration in Fig. 15, the width and spacing of the tines in POLYCOMB have been magnified 3X, and the number of tines has been accordingly reduced by a factor of 3 so that the overall dimensions remain unchanged. Accurate dimensional data on POLYCOMB can be measured from Fig. 15 by scaling down detail 3X. Areas and perimeters are listed in Table III. Also the gap between the field plate and the OVLAP and NOVLAP capacitors has been widened 3X. The IGFET arrays and gated diodes were assigned to the B chip for two reasons: (*i*) at least four photolithographic steps are required to realize completed devices, so it is anticipated that these will not be used as frequently as the two-level structures on the A chip; (*ii*) the IGFET and gated diode arrays require 23 additional probing pads that are not available on the A chip. To provide adequate resolution for illustration, the spacing between gates and sources and drains has been increased 3X in Fig. 15. In the gated diodes the space between gates and junctions has also been increased 3X.

#### *3.1.3 Chip C (6400 × 8192 $\mu$m)*

The largest MOS capacitors, with areas measuring $(4000\ \mu m)^2$, had to be allocated to four separate chips. The C1 chip contains FOXCAP and GOXCAP capacitors with areas equal to $(4000\ \mu m)^2$. A composite view of the POLY and WINDOW levels is shown in Fig. 16. The C2 chip contains a rectangular GOXCOMB structure with area somewhat reduced from the rectangular devices on chip C1; the exact coded areas are listed in Table IV. A composite view of the POLY and WINDOW levels is shown in Fig. 17. The C3 chip contains a rectangular POLYCOMB structure with area somewhat reduced from the rectangular devices on chip C1; the exact coded areas are listed in Table IV. A composite view of the POLY and WINDOW levels is shown in Fig. 18. To provide adequate resolution for illustration in Fig. 18, the width and spacing of the tines have been magnified 3X, and the number of tines has been accordingly reduced by a factor of 3 so that the overall dimensions remain unchanged. Accurate dimensional data can be measured from Fig. 18 by scaling down detail 3X. Areas and perimeters are listed in Table IV.

The C4 chip contains both OVLAP and NOVLAP rectangular capacitors with areas measuring $(4000\ \mu m)^2$ and surrounded by a field plate. A composite view of GUARDRING, GASAD, POLY, and WINDOW levels is shown in Fig. 19. For the purpose of illustration the gap between the field plate and the OVLAP and NOVLAP capacitors has been widened 3X in Fig. 19. All C chips measure $6400 \times 8192\ \mu m$. When combined with the D chip (see next section), the overall dimensions of the combination are exactly equal to the dimensions of the B chip or a $2 \times 5$ array of A chips.

#### *3.1.4 Chip D (1600 × 4096 μm) METEST*

A composite view of the POLY and WINDOW levels of the contact metallization test chip D (METEST) is shown in Fig. 20. Both the dimensions and the locations of the 36 probing pads of the METEST chip have been selected to coincide with the Poon Tester[1] and the Process Monitor[4] chips to facilitate automatic probing. It is expected that the tapped strings with 2-, 3-, 5-, and 7-$\mu$m windows will be used most extensively. These strings have been terminated on probing pad numbers 3, 4, 5, 6 (3 $\mu$m), 9, 10, 11, 12 (2 $\mu$m), 21, 22, 23, 24 (7 $\mu$m), and 27, 28, 29, 30 (5 $\mu$m) to coincide with existing metallization probing cards.

Detailed dimensional data for the D chip are listed in Table V. Table V differs from Tables II through IV in many respects, because the tapped strings were not intended to provide capacitance data. There are no GUARDRING features. Entries tabulated under GASAD tub refer to features straddling the indicated pad numbers, although all tubs within each string are connected in parallel after metallization with the POLY level. Entries tabulated under POLYCON window refer to the total cross-sectional area of a single tub input or output. However, current density is not expected to be uniform over any given window and especially among multiple windows. Entries tabulated under POLY are intended to aid in estimating string resistances from the sheet resistance of the metallization layer provided by the POLY pattern. Taps and ties are defined in Fig. 11, and the equivalent numbers of squares straddling each pair of pad numbers are indicated. The equivalent number of squares for the contact areas were not included, because these depend upon the sheet resistance of the underlying tubs.

### *3.2 Wafer layout*

The location of each of the test chips described in the preceding section is shown in Fig. 21. The A chip is the most numerous, totaling 130 and arranged in blocks of 10 to form the cross-shaped pattern coded AX in Fig. 21. The number X, following A, denotes the width of

the guard ring when the GUARDRING (N35) option is selected. The symbol A0 denotes unguarded devices. The arrangement of the A chips, which have been laid out to permit automatic probing with existing facilities, is obviously intended to reveal horizontal and vertical parametric trends on test wafers.

There are only twelve B chips, which contain 1000- and 2000-$\mu$m devices. When the GUARDRING (N35) option is selected, there are only two chips for each guard-ring width, i.e., 2, 3, 5, 7, and 10 $\mu$m. As in the case of the A chip, guard-ring width is represented by X in the notation BX, shown in Fig. 21. The asterisks denote undefined guard-ring diffusions or implantations that cover the entire chip for evaluation of guard-ring performance without a parallel Schottky diode.

There are six C1 and six C4 chips on the test wafer. In each case, three are unguarded and three have 10-$\mu$m guard rings, *viz.* C1-0 and C1-10 in Fig. 21. The C2 and C3 chips contain large GOXCOMB and POLYCOMB structures, respectively. Two are unguarded, *viz.* C2-0, and two have guard rings, *viz.* C2-10, in Fig. 21. In the case of GOXCOMB, the guard-ring width does not permit interdigitating the individual COMB elements, so the guard-ring option provides a buried diffused tub beneath the GASAD structure. As in the B chip, the asterisks denote undefined guard-ring diffusions or implantations.

The contact metallization test chip is denoted D in Fig. 21. There are a total of 40, which have been divided equally among the four quadrants of the wafer.

At the top and bottom and left and right are four alignment patterns that have been designed to permit aligning any mask to any other in any order (see Section 3.4). Also associated with each alignment pattern are two TEM test chips that have been specially designed to facilitate transmission electron microscopic (TEM) analysis.[21] (See Section 3.5.)

### *3.3 Design rules*

The lithographic mask system has been designed specifically for fine-line process development and process control. Since the masks are generally to be used with standard photolithographic processing, it obviously would be inappropriate to interpret data in terms of fine-line design rules. Consequently, most device structures have been designed to tolerate up to 5 $\mu$m in misalignment errors. Figure 22 is an example of such relaxed design rules, showing the source-gate structure of a typical element from the IGFET group (see Section 2.4). The same structure also applies to the junction contacts of the gated diode group (GATODE) (see Section 2.5). In general, all contact windows have a minimum width of 5 $\mu$m, and all overlapping regions are a minimum of 5 $\mu$m.

A number of exceptions exist, however. Most prevalent are the GOXCAP group (see Section 2.1.1), which has been deliberately continued to a minimum size of 1.0 $\mu$m to enable special experiments with nonstandard photolithographic processes or X-ray lithography. A similar philosophy has been applied to the entire contact metallization test chip D (METEST), which has been fully described in Section 2.6.

### *3.4 Alignment features*

It is hoped that maximum flexibility has been achieved by the use of a modified version of the standard Perkin-Elmer projection (PEP) alignment features. These modified PEP (MOPEP) features are shown in Figs. 23a through f and are presented in the anticipated "normal" order or suggested sequences, i.e. GUARDRING, GASAD, POLYCON, POLY, WINDOW, and METAL. The upper set of MOPEP features in each of Figs. 23a through f corresponds to the "normal" processing sequence. Unlike alignment procedures for virtually all device codes, each mask in this lithographic system must be aligned to the immediately preceding level, because levels prior to the one immediately preceding introduce overlapping patterns. But the unique feature of the MOPEP alignment features is that any number of levels can be skipped. For example, it has been anticipated that a popular sequence may be GASAD followed by POLY only. The alignment feature remaining on the test wafer after GASAD lithography is shown in Fig. 23b. Alignment of POLY to GASAD corresponds to a "normal" processing sequence, so the second MOPEP feature in the upper half of Fig. 23d would have to be aligned to the second (right-hand) MOPEP feature in the upper half of Fig. 23b. The left-hand MOPEP feature in Fig. 23b is simply ignored, because the GUARDRING level was omitted.

The lower set of MOPEP alignment features in Figs. 23a through f have been included to enable an "inverted" processing sequence. Such an "inverted" processing sequence might be required for some unique or novel structure that was not originally intended or anticipated. For example, POLY features can be defined on the surface of an unpatterned field oxide. After oxidation, or deposition of an intermediate dielectric layer, it might be necessary to define additional conductive features of either poly-Si or metal directly over the original poly-Si features. This capability is available by using a "GASAD" level with reverse tone (see Table VI), consisting of opaque features within a transparent background, to produce conductive patterns in polysilicon or metal. Alignment is carried out by inserting the central MOPEP feature in the lower half of Fig. 23b into the right-hand MOPEP feature on the lower half of Fig. 23d, which would be the pattern left

on the test wafer after POLY lithography. (Tone reversal is not applied to MOPEP features.)

To be consistent with the relaxed 5-$\mu$m design rules discussed in the preceding section, the right-hand MOPEP feature in the upper halves of Figs. 23a through f are all 25 $\mu$m wide. All of the other MOPEP features in the upper halves of Fig. 23b through e and all MOPEP features in the upper half of Fig. 23f are 20 $\mu$m wide. A similar scheme that provides 2.5-$\mu$m frames for alignment in inverted order applies to the lower halves of Figs. 23a through f. The 2.5-$\mu$m alignment frame has resulted from a compromise that should offset the effects of photolithographic processing variations, but proper alignment does require some judgment on behalf of the alignment operator to optimize registration of sequential mask levels.

### *3.5 TEM test chip*

Sample preparation techniques for transmission electron microscopy (TEM) usually produce sections sufficiently thin over a region that may vary between 40 and 100 $\mu$m. All morphological features essential for process evaluation can be translated into the area for TEM study by a special test pattern 1 mm wide and approximately 6.7 mm long, and with a structural period of 29.5 $\mu$m. The TEM test pattern is shown in Fig. 24, showing gate oxide within regions defined by GASAD, contact windows formed by POLYCON, layers of poly-Si defined by POLY, and subsequent P-glass and metallization levels. Thus all windows, steps, and other peripheral features normally encountered in fine-line process development are reproduced over 200 times within each TEM test chip. Two TEM test chips have been placed symmetrically with respect to each MOPEP alignment feature (see Fig. 21), and no active device areas have been sacrificed. A total of eight TEM test chips have been incorporated into th · lithographic mask system, and the feature boundaries of each TEM test chip are oriented orthogonal to a ⟨100⟩ cleavage plane so that the cross section shown in Fig. 24 can be readily obtained from widely separated areas of the wafer. The TEM test chip shown in Fig. 24 differs from the one published by Sheng and Marcus,[21] partially because the chip in Fig. 24 was designed for a fine-line process that does not involve selective oxidation.

## IV. MASK LEVELS

The six mask levels that comprise the normal tone portion of the lithographic mask system are listed in the upper part of Table VI and are intended to be used with positive photoresist. The suggested sequence reflects the primary purpose for which each level was intended; a few examples are shown in Table VII. Each mask level

contains four full sets of MOPEP alignment features, corresponding to "normal" and "inverted" processing sequences. Thus, it is possible to align any mask level to any other, thereby permitting novel device structures with processing sequences that have not been anticipated. (See Section 3.4 for alignment details.) The three levels that have been issued in reverse tone are shown in the lower part of Table VI. They have been intended for use with negative photoresist, uniform gold metallization, selective oxidation or other special processes.

### *4.1 GUARDRING*

The GUARDRING level would generally be omitted for most MOS processing investigations. In a sense, it may be viewed as being analogous to the isolation tub diffusion that occurs in CMOS processing prior to GASAD. The principal purpose of the guard ring is to provide p-n junctions that straddle the boundaries of Schottky barrier diodes to electrically isolate metallization edges that often obscure barrier characterization. The GUARDRING level is unusual because it comprises a subset of six patterns that provide guard-ring widths of 10, 7, 5, 3, 2, and 0 $\mu$m. The location of each of these subsets is indicated in Fig. 21 by the final hyphened integers X, i.e., C1-X. Each guard ring is located such that it frames each GASAD boundary symmetrically. In the case of the smallest GASAD features, the area enclosed by the larger GUARDRING features vanishes, and a p-n junction is formed, which is useful for evaluating guard-ring performance. The asterisks denote undefined diffused or implanted areas that cover the entire chip. These chips can also be used to characterize the guard rings independently from the other features or to evaluate ion implantation profiles (see Section 2.1.1).

### *4.2 GASAD*

The GASAD level is normally the first level that would be used for MOS process development and monitoring. The principal purpose of the GASAD level is to open up areas in the field oxide in preparation for a possible ion implantation, for threshold control, followed by gate oxidation. If gate oxidation is omitted, however, the GASAD level provides a range of areas for investigations of contact resistance, Schottky barrier diodes, and p-n junctions. Most of the patterns provided by GASAD are square and progressively increase in size through the sequence 1.0, 1.5, 2, 3, 4, 5, 7, 10, 15, 20, 50, 100, 200, 500, 1000, 2000, and 4000 $\mu$m. At 200 $\mu$m and above, comb-like structures (GOXCOMB) are included in the GASAD level for investigation of peripheral effects and defects. The 200- and 500-$\mu$m GOXCOMB structures consist of a series of 25-$\mu$m slots separated by 10 $\mu$m of field oxide (see Fig. 3). All of the slots are interconnected to guarantee

equalization of surface potential. The choice of the relatively large slots assures that all of the MOS capacitors within HEXCAP will have nearly equal gate areas despite typical variations in linewidth owing to processing variables. Obviously, GOXCOMB can be utilized for aggravating edge effects or defects in device structures. In the largest GOXCOMB structures (1000, 2000, and 4000 $\mu$m) the slots have been reduced to 5 $\mu$m, separated by 10 $\mu$m of field oxide, to further increase the aggravation caused by peripheral electric fields and edge-related defects.

It is anticipated that in most MOS process monitoring and development applications it would be possible to use wafers that had been previously patterned using the GASAD mask. Thus, the most important device structures would be complete after only one additional photolithographic step (see POLY, Section 4.4).

### *4.3 POLYCON*

For most MOS monitoring or process development applications the POLYCON level would be omitted. The main reason for including this level has been to provide windows to the underlying GASAD tub diffusions or ion implantations that are required by the metallization test chip D (METEST). However, when investigation of p-n junctions with large areas or extended peripheries is required, the GASAD mask is used to define the diffused or ion-implanted regions. In this case the POLYCON level provides the required contact windows to junctions formed by the GASAD features. Evaluation of p-n junctions thus requires a minimum of three mask levels, because the POLY level must be used for metallization. If it should be necessary to investigate the effects of high-temperature processing after poly-Si deposition, such as oxidations and/or insulating depositions, additional mask levels WINDOW and METAL might be required.

### *4.4 POLY*

It has been anticipated that the combination of GASAD followed by POLY would be the most widely used sequence of mask levels for MOS process development. For this reason, the POLY patterns have been extended to include the probing pads so that the poly-Si can be probed directly. Typical probe-spreading resistance measurements are of the order of 50Ω in a poly-Si film with a sheet resistance of 20 Ω/□, *providing that the probe tips are sufficiently hard and sharp enough to pierce 50Å of native oxide that typically occurs on the surface of $n^+$-poly-Si.* For this purpose tungsten carbide probes are recommended with tip radii of $5 \times 10^{-4}$ cm or less. The use of palladium probes with planar tips $5 \times 10^{-3}$ cm in diameter has been found to be unsatisfactory for probing $n^+$-poly-Si. Unfortunately, many probe

cards designed for automatic probing cannot be used for probing $n^+$-poly-Si directly, because erratic probe contact resistance may range over many orders of magnitude, sometimes exceeding 10 megohms. Regular cleaning and inspection of probe tips are mandatory when probing $n^+$-poly-Si. Ion implantation causes an amorphous layer of semi-insulating material to occur near the surface of $n^+$-poly-Si. Thus, any $n^+$-poly-Si that has been exposed to ion implantation must be annealed at 950°C in $N_2$ for 30 minutes to avoid excessive probe resistance. In automatic probe stations with existing probe cards, the POLY mask level may be used to pattern aluminum, with or without an intervening poly-Si layer, to reduce probe resistance. Alternatively, the WINDOW and METAL mask levels can be used (see next sections).

At 200 $\mu$m and above, comb-like structures (POLYCOMB) are included in the POLY level for investigating edge effects or defects (see Fig. 4). The width and separation of individual elements are the same as for GOXCOMB (see GASAD, Section 4.2).

### *4.5 WINDOW*

Occasionally, it is necessary to investigate changes in device characteristics resulting from oxidations, insulating depositions, or annealing after poly-Si definition. For this purpose the WINDOW level has been provided, which opens 90-$\mu$m square windows over each probing pad. In most cases the poly-Si could then be probed directly at the contact pads without resorting to aluminum deposition and lithography, providing the recommendations contained above in Section 4.4 are followed. In the case of the OVLAP and NOVLAP MOS capacitors surrounded by field plates, the capacitors must be probed directly. For this purpose, the WINDOW level also contains large contacting areas over each capacitor surrounded by a field plate. Occasionally, it may be beneficial to access the van der Pauw pattern with aluminum metallization. Therefore, the WINDOW level also contains four contacts oriented directly over the leads at the edges of the van der Pauw structures to provide conduction to overlying metal lines leading to the probing pads. The WINDOW level is also required for contacts to the sources and drains of IGFETs, the junction terminal of the GATODEs, and to SADSHEET.

### *4.6 METAL*

As in the case of the GUARDRING and POLYCON levels, it is expected that the METAL level could be omitted in most MOS monitoring or process development applications. The principal exceptions include IGFETs and gated diodes, in which structures the re-

quired electrical continuity could not have been provided by the POLY level alone.

In most cases the metal has been excluded from any areas where poly-Si is in contact with gate oxide. The reason for this exclusion stems from experimental evidence that aluminum, sintered into polycrystalline silicon, sometimes deteriorates the dielectric breakdown strength of underlying gate oxides, especially if the gate oxide is very thin (i.e., $\leq$ 250Å). Exception occurs for all of the MOS capacitors surrounded by field plates, which must be probed directly. These OVLAP and NOVLAP capacitors thus provide experimental structures appropriate for cases in which it is necessary to compare the breakdown voltages in MOS capacitors with and without overlying metallization.

## V. APPLICATIONS

Most basic circuit elements utilized by unipolar semiconductor integrated electronics can be easily fabricated with an appropriate subset of the lithographic mask system. These elements fall into five general classifications: MOS capacitors, p-n junctions, contacts, sheet resistors, and IGFETs. The contact class can be further subdivided to include guarded and unguarded Schottky diodes, ohmic contacts, and contact metallization test cells. The sheet resistor class can be subdivided to include polysilicon sheet resistors (POLYSHEET), source and drain sheet resistors (SADSHEET), and van der Pauw patterns in the GUARDRING, GASAD, and POLY levels. The IGFET class also includes gated diodes. These thirteen subdivisions of unipolar device structures are shown in the first column of Table VII. The remaining six columns show the *required* mask levels needed to realize a particular device structure. Other mask levels are generally optional, but some may be required for certain experiments.

At the present time the lithographic mask system is in wide use, and more than fifty experiments have been initiated in the Advanced Large-Scale Integration (LSI) Development Laboratory using one or more levels. References 22 through 26 contain published experiments that have utilized this mask system.

## VI. SUMMARY

Fine-line MOS process characterization, determination of base-line parameters, and new process development can be efficiently carried out using the lithographic mask system. By selecting an appropriate subset of photo- or X-ray lithographic mask levels, most unipolar semiconductor circuit elements can be fabricated in an enormous range of sizes. X-ray and trilevel lithographic processes are used only when

absolutely required, and these can be interleaved with photolithographically defined patterns. Any mask level can be aligned to any other, and any number of mask levels can be skipped. Registration tolerance is 5 $\mu$m for most device structures. Most experimental device structures can be completed and ready for electrical evaluation in a fraction of the time required to fabricate the elements on the Poon Tester chips,[1] which are included within the array of fine-line device chips and require six X-ray lithographic levels to complete.

## VII. ACKNOWLEDGMENTS

The author is indebted to W. T. Lynch for initiating this project and for his continued interest, suggestions, and encouragement. The initial ideas were refined and implemented as a result of numerous discussions with E. N. Fuls, R. A. Kushner, E. I. Povilonis, and N. C. Wittwer. The universal alignment capability, resulting in modified PEP (MO-PEP) features, grew out of discussions with N. C. Wittwer, K. G. Schaffner, and R. Colish. The modification of X-ray alignment features to permit mixing of X-ray and photolithography was done by G. A. Coquin. The design of the metallization test chip (METEST) was carried out in collaboration with A. K. Sinha and S. Vaidya. Discussions with J. R. Brews and G. W. Taylor contributed to the design of the gated diodes and the option of buried channel capacitors. The TEM test chip was contributed by R. B. Marcus. Chip layouts were carried out by A. W. Russo, M. L. Oeser, J. P. Boyle, and M. Wagner of the Engineering Information Department. The wafer array was prepared by S. R. Juice, also of the Engineering Information Department. Mask fabrication was done in the Murray Hill Mask Lab; the cooperation of J. G. Skinner, L. J. Heilos, and other members of the Photomask and Pattern Generation Department is gratefully acknowledged. Final mask inspection was done by D. P. Rosenfeld of the Lithographic Systems Development Department.

Table I—Device key*

| MOS Device | Nominal Dimension | Chip Assignment | Pad No(s). |
|---|---|---|---|
| HEXCAP Group | | | |
| 1. FOXCAP | 4000 | C1 | 93 |
| | 2000 | B | 41 |
| | 1000 | B | 3 |
| | 500 | A | 2 |
| | 200 | A | 32 |
| | 100 | A | 26 |
| | 50 | A | 19 |
| 2. GOXCAP | 4000 | C1 | 42 |
| | 2000 | B | 49 |
| | 1000 | B | 7 |
| | 500 | A | 4 |
| | 200 | A | 31 |
| | 100 | A | 29 |
| | 50 | A | 20 |
| | 20 | A | 25 |
| | 15 | A | 1 |
| | 10 | A | 3 |
| | 7 | A | 5 |
| | 5 | A | 7 |
| | 4 | A | 8 |
| | 3 | A | 10 |
| | 2 | A | 11 |
| | 1.5 | A | 12 |
| | 1.0 | A | 14 |
| 3. GOXCOMB | 4000 | C2 | 67 |
| | 2000 | B | 106 |
| | 1000 | B | 34 |
| | 500 | A | 6 |
| | 200 | A | 30 |
| 4. POLYCOMB | 4000 | C3 | 68, 69, 70, 71 |
| | 2000 | B | 95, 96 |
| | 1000 | B | 22 |
| | 500 | A | 9 |
| | 200 | A | 29 |
| 5/6. OVLAP and NOVLAP with FIELD PLATE | 4000 | C4 | 14 |
| | 2000 | B | 75 |
| | 1000 | B | 13 |
| | 500 | A | 13 |
| | 200 | A | 33 |
| | 100 | A | 28 |
| | 50 | A | 18 |
| SHEET Group | | | |
| 1. SADSHEET | 80SQ | A | 15 |
| | COM | A | 16 |
| | 50SQ | A | 17 |
| 2. POLYSHEET | 80SQ | A | 36 |
| | COM | A | 35 |
| | 50SQ | A | 34 |
| VANDERPAUW Group | | | |
| 1. POLYSI | 250 | A | 21 |
| | | | 22 |
| | | | 23 |
| | | | 24 |
| 2. GASAD | 250 | C4 | 101 |
| | | | 102 |
| | | | 103 |
| | | | 104 |
| 3. GUARDRING | 250 | C4 | 96 |
| | | | 97 |
| | | | 98 |
| | | | 99 |

* All dimensions are in micrometers. Exact areas and perimeters are listed in Tables II, III, and IV.

Table I–Device key*(*Continued*)

| MOS Device | Nominal Dimension | Chip Assignment | Pad No(s). |
|---|---|---|---|
| 4. POLYSI and GUARDRING | 250 | C4 | 87<br>88<br>89<br>90<br>91<br>92<br>93<br>94 |
| 5. POLYSI and GASAD | 250 | C4 | 75<br>76<br>77<br>78<br>79<br>80<br>81<br>82 |
| **IGFET Group** | **Channel** | | |
| **Terminal** | **Width Length** | | |
| 1. Drain | 100 20 | B | 87 |
| | 100 15 | B | 86 |
| | 100 10 | B | 85 |
| | 100 8 | B | 84 |
| | 100 6 | B | 83 |
| | 100 5 | B | 82 |
| | 100 4 | B | 81 |
| | 100 3 | B | 80 |
| | 100 2 | B | 79 |
| | 100 1.5 | B | 78 |
| | 100 1.0 | B | 77 |
| Common Gate | 100 | B | 88 |
| Common Source | 100 | B | 76 |
| 2. Source | 500 500 | B | 53 |
| Gate | | B | 54 |
| Drain | | B | 55 |
| 3. Source | 300 300 | B | 56 |
| Gate | | B | 57 |
| Drain | | B | 58 |
| 4. Source | 200 200 | B | 59 |
| Gate | | B | 60 |
| Drain | | B | 61 |
| 5. Source | 100 100 | B | 62 |
| Gate | | B | 63 |
| Drain | | B | 64 |
| **GATED DIODE Group** | **Channel** | | |
| **Terminal** | **Width and Length** | | |
| 1. Junction | 500 | B | 65 |
| Gate | | B | 67 |
| 2. Junction | 300 | B | 68 |
| Gate | | B | 69 |
| 3. Junction | 200 | B | 70 |
| Gate | | B | 71 |
| 4. Junction | 100 | B | 72 |
| Gate | | B | 73 |
| **METEST Group** | **Window Size** | | |
| 1. Uni-Window | 2 | D | 9–12 |
| | 3 | D | 3–6 |
| | 5 | D | 27–30 |
| | 7 | D | 21–24 |
| 2. Quad-Window | 2 | D | 7, 8, 25, 26 |
| 3. Hex-Window | 1.5 | D | 1, 2, 35, 36 |
| 4. Oct-Window | 1 | D | 31–34 |

* All dimensions are in micrometers. Exact areas and perimeters are listed in Tables II, III, and IV.

## Table II—Pad key for chip A*

| Pad No. | MOS Device | Nom. Dim. | GASAD Area | GASAD Perimeter | POLYCON Area | POLY Area | POLY Perimeter | POLY Overlap FOX | POLY Overlap GOX | Window Area | Metal Area | GUARDRING (Width: 2) Area | GUARDRING (Width: 2) Perimeter | GUARDRING (Width: 2) Enclosed Area |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | GOXCAP | 15 | 225 | 60 | 81 | 11,750 | 540 | 11,525 | 225 | 8,100 | 10,000 | 120 | 120 | 169 |
| 2 | FOXCAP | 500 | 0 | 0 | 0 | 271,350 | 2,490 | 271,350 | 0 | 8,100 | 10,000 | 262,144 | 2,048 | 0 |
| 3 | GOXCAP | 10 | 100 | 40 | 16 | 11,300 | 530 | 11,200 | 100 | 8,100 | 10,000 | 80 | 80 | 64 |
| 4 | GOXCAP | 500 | 250,000 | 2,000 | 100 | 271,350 | 2,490 | 21,350 | 250,000 | 8,100 | 10,000 | 4,000 | 4,000 | 248,004 |
| 5 | GOXCAP | 7 | 49 | 28 | 0 | 11,054 | 524 | 11,005 | 49 | 8,100 | 10,000 | 56 | 56 | 25 |
| 6 | GOXCOMB | 500 | 250,000 | 20,050 | 100 | 361,705 | 2,818 | 111,705 | 250,000 | 8,100 | 10,000 | 40,976 | 40,100 | 229,954 |
| 7 | GOXCAP | 5 | 25 | 20 | 0 | 10,900 | 520 | 10,875 | 25 | 8,100 | 10,000 | 40 | 40 | 9 |
| 8 | GOXCAP | 4 | 16 | 16 | 0 | 10,826 | 518 | 10,810 | 16 | 8,100 | 10,000 | 32 | 32 | 4 |
| 9 | POLYCOMB | 500 | 350,455 | 2,368 | 100 | 271,350 | 20,702 | 21,350 | 250,000 | 8,100 | 10,000 | 4,736 | 4,736 | 348,091 |
| 10 | GOXCAP | 3 | 9 | 12 | 0 | 10,754 | 516 | 10,745 | 9 | 8,100 | 10,000 | 24 | 24 | 1 |
| 11 | GOXCAP | 2 | 4 | 8 | 0 | 10,684 | 514 | 10,680 | 4 | 8,100 | 10,000 | 16 | 16 | 0 |
| 12 | GOXCAP | 1.5 | 2.25 | 6 | 0 | 10,649.75 | 513 | 10,647.5 | 2.25 | 8,100 | 10,000 | 12.25 | 14 | 0 |
| 13 | FIELD PLATE | 500 | 0 | 0 | 0 | 111,500 | 8,020 | 111,500 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| — | OVLAP | 500 | 250,000 | 2,000 | 240,100 | 260,100 | 2,040 | 10,100 | 250,000 | 240,100 | 260,100 | 4,000 | 4,000 | 248,004 |
| — | NOVLAP | 500 | 260,100 | 2,040 | 240,100 | 250,000 | 2,000 | 0 | 250,000 | 240,100 | 250,000 | 4,080 | 4,080 | 258,064 |
| 14 | GOXCAP | 1.0 | 1.0 | 4 | 0 | 10,616 | 512 | 10,615 | 1.0 | 8,100 | 10,000 | 9 | 12 | 0 |
| 15 | SADSHEET | 80SQ | 3,250 | 990 | 225 | 10,000 | 400 | 10,000 | 0 | 8,325 | 11,225 | 1,980 | 1,980 | 2,264 |
| 16 | SADSHEET | COM | | | 450 | 10,000 | 400 | 10,000 | 0 | 8,550 | 12,450 | | | |
| 17 | SADSHEET | 50SQ | 4,450 | 984 | 225 | 10,000 | 400 | 10,000 | 0 | 8,325 | 11,225 | 1,968 | 1,968 | 3,470 |
| 18 | FIELD PLATE | 50 | 0 | 0 | 0 | 32,750 | 1,670 | 32,750 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| — | OVLAP | 50 | 2,500 | 200 | 1,600 | 3,600 | 240 | 1,100 | 2,500 | 1,600 | 3,600 | 400 | 400 | 2,304 |
| — | NOVLAP | 50 | 3,600 | 240 | 1,600 | 2,500 | 200 | 0 | 2,500 | 1,600 | 2,500 | 480 | 480 | 3,364 |
| 19 | FOXCAP | 50 | 0 | 0 | 0 | 14,850 | 690 | 14,850 | 0 | 8,100 | 10,000 | 3,844 | 248 | 0 |
| 20 | GOXCAP | 50 | 2,500 | 200 | 100 | 14,850 | 690 | 12,350 | 2,500 | 8,100 | 10,000 | 400 | 400 | 2,304 |
| 21 | VANDERPAUW | 250 | 0 | 0 | 0 | 137,675 | 5,210 | 137,675 | 0 | 8,685 | 30,912.5 | 63,504 | 1,008 | 0 |
| 22 | VANDERPAUW | 250 | 0 | 0 | 0 | 137,675 | 5,210 | 137,675 | 0 | 8,587.5 | 15,587.5 | 63,504 | 1,008 | 0 |
| 23 | VANDERPAUW | 250 | 0 | 0 | 0 | 137,675 | 5,210 | 137,675 | 0 | 8,437.5 | 11,212.5 | 63,504 | 1,008 | 0 |
| 24 | VANDERPAUW | 250 | 0 | 0 | 0 | 137,675 | 5,210 | 137,675 | 0 | 8,602.5 | 17,462.5 | 63,504 | 1,008 | 0 |
| 25 | GOXCAP | 20 | 400 | 80 | 100 | 12,250 | 550 | 11,850 | 400 | 8,100 | 10,000 | 160 | 160 | 324 |
| 26 | FOXCAP | 100 | 0 | 0 | 0 | 23,350 | 890 | 23,350 | 0 | 8,100 | 10,000 | 12,544 | 448 | 0 |
| 27 | GOXCAP | 100 | 10,000 | 400 | 100 | 23,350 | 890 | 13,350 | 10,000 | 8,100 | 10,000 | 800 | 800 | 9,604 |
| 28 | FIELD PLATE | 100 | 0 | 0 | 0 | 37,750 | 2,520 | 37,750 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| — | OVLAP | 100 | 10,000 | 400 | 8,100 | 12,100 | 440 | 2,100 | 10,000 | 8,100 | 12,100 | 800 | 800 | 9,604 |
| — | NOVLAP | 100 | 12,100 | 440 | 8,100 | 10,000 | 400 | 0 | 10,000 | 8,100 | 10,000 | 880 | 880 | 11,664 |
| 29 | POLYCOMB | 200 | 56,350 | 950 | 100 | 55,350 | 3,782 | 15,350 | 40,000 | 8,100 | 10,000 | 1,900 | 1,900 | 55,404 |
| 30 | GOXCOMB | 200 | 40,000 | 3,250 | 100 | 67,600 | 1,400 | 27,600 | 40,000 | 8,100 | 10,000 | 6,500 | 6,472 | 36,754 |
| 31 | GOXCAP | 200 | 40,000 | 800 | 100 | 55,350 | 1,290 | 15,350 | 40,000 | 8,100 | 10,000 | 1,600 | 1,600 | 39,204 |
| 32 | FOXCAP | 200 | 0 | 0 | 0 | 55,350 | 1,290 | 55,350 | 0 | 8,100 | 10,000 | 44,944 | 848 | 0 |
| 33 | FIELD PLATE | 200 | 0 | 0 | 0 | 56,787.5 | 4,043 | 56,787.5 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| — | OVLAP | 200 | 40,000 | 800 | 36,100 | 44,100 | 840 | 4,100 | 40,000 | 36,100 | 44,100 | 1,600 | 1,600 | 39,204 |
| — | NOVLAP | 200 | 44,100 | 840 | 36,100 | 40,000 | 800 | 0 | 40,000 | 36,100 | 40,000 | 1,680 | 1,680 | 43,264 |
| 34 | POLYSHEET | 50SQ | 0 | 0 | 0 | | | | 0 | 8,100 | 10,000 | 4,020 | 814 | 0 |
| 35 | POLYSHEET | COM | 0 | 0 | 0 | 35,216 | 2,778 | 35,216 | 0 | 8,100 | 10,000 | | | |
| 36 | POLYSHEET | 80SQ | 0 | 0 | 0 | | | | 0 | 8,100 | 10,000 | 2,800 | 824 | 0 |

* All dimensions are in micrometers.

## Table II—Pad key for chip A* (*Continued*)

| PAD No. | MOS device | Nom. Dim. | GUARDRING (Width: 3) | | | GUARDRING (Width: 5) | | | GUARDRING (Width: 7) | | | GUARDRING (Width: 10) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Area | Perimeter | Enclosed Area | Area | Perimeter | Enclosed Area | Area | Perimeter | Enclosed Area | Area | Perimeter | Enclosed Area |
| 1 | GOXCAP | 15 | 180 | 120 | 144 | 300 | 120 | 100 | 420 | 120 | 64 | 600 | 120 | 25 |
| 2 | FOXCAP | 500 | 263,169 | 2,052 | 0 | 265,225 | 2,060 | 0 | 267,289 | 2,068 | 0 | 270,400 | 2080 | 0 |
| 3 | GOXCAP | 10 | 120 | 80 | 49 | 200 | 80 | 25 | 280 | 80 | 9 | 400 | 80 | 0 |
| 4 | GOXCAP | 500 | 6,000 | 4,000 | 247,009 | 10,000 | 4,000 | 245,025 | 14,000 | 4,000 | 243,049 | 20,000 | 4,000 | 240,100 |
| 5 | GOXCAP | 7 | 84 | 56 | 16 | 140 | 56 | 4 | 196 | 56 | 0 | 289 | 68 | 0 |
| 6 | GOXCOMB | 500 | 60,150 | 40,100 | 219,934 | 100,250 | 40,100 | 199,900 | 140,350 | 40,100 | 179,874 | 200,560 | 22,404 | 149,850 |
| 7 | GOXCAP | 5 | 60 | 40 | 4 | 100 | 40 | 0 | 144 | 48 | 0 | 225 | 60 | 0 |
| 8 | GOXCAP | 4 | 48 | 32 | 1 | 81 | 36 | 0 | 121 | 44 | 0 | 196 | 56 | 0 |
| 9 | POLYCOMB | 500 | 7,104 | 4,736 | 346,912 | 11,840 | 4,736 | 344,560 | 16,576 | 4,736 | 342,216 | 23,680 | 4,736 | 338,715 |
| 10 | GOXCAP | 3 | 36 | 24 | 0 | 64 | 32 | 0 | 100 | 40 | 0 | 169 | 52 | 0 |
| 11 | GOXCAP | 2 | 25 | 20 | 0 | 49 | 28 | 0 | 81 | 36 | 0 | 144 | 48 | 0 |
| 12 | GOXCAP | 1.5 | 20.25 | 18 | 0 | 42.25 | 26 | 0 | 72.25 | 34 | 0 | 132.25 | 46 | 0 |
| 13 | FIELD PLATE | 500 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | OVLAP | 500 | 6,000 | 4,000 | 247,009 | 10,000 | 4,000 | 245,025 | 14,000 | 4,000 | 243,049 | 20,000 | 4,000 | 240,100 |
| – | NOVLAP | 500 | 6,120 | 4,080 | 257,049 | 10,200 | 4,080 | 255,025 | 14,280 | 4,080 | 253,009 | 20,400 | 4,080 | 250,000 |
| 14 | GOXCAP | 1.0 | 16 | 16 | 0 | 36 | 24 | 0 | 64 | 32 | 0 | 121 | 44 | 0 |
| 15 | SADSHEET | 80SQ | 2,970 | 1,980 | 1,774 | 4,950 | 1,090 | 800 | 6,130 | 1,090 | 648 | 7,850 | 1,150 | 450 |
| 16 | SADSHEET | COM | | | | | | | | | | | | |
| 17 | SADSHEET | 50SQ | 2,952 | 1,968 | 2,983 | 4,920 | 1,968 | 2,015 | 6,888 | 1,968 | 1,055 | 9,020 | 1,024 | 450 |
| 18 | FIELD PLATE | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | OVLAP | 50 | 600 | 400 | 2,209 | 1,000 | 400 | 2,025 | 1,400 | 400 | 1,849 | 2,000 | 400 | 1,600 |
| – | NOVLAP | 50 | 720 | 480 | 3,249 | 1,200 | 480 | 3,025 | 1,680 | 480 | 2,809 | 2,400 | 480 | 2,500 |
| 19 | FOXCAP | 50 | 3,969 | 252 | 0 | 4,225 | 260 | 0 | 4,489 | 268 | 0 | 4,900 | 280 | 0 |
| 20 | GOXCAP | 50 | 600 | 400 | 2,209 | 1,000 | 400 | 2,025 | 1,400 | 400 | 1,849 | 2,000 | 400 | 1,600 |
| 21 | VANDERPAUW | 250 | 64,009 | 1,012 | 0 | 65,025 | 1,020 | 0 | 66,049 | 1,028 | 0 | 67,600 | 1,040 | 0 |
| 22 | VANDERPAUW | 250 | 64,009 | 1,012 | 0 | 65,025 | 1,020 | 0 | 66,049 | 1,028 | 0 | 67,600 | 1,040 | 0 |
| 23 | VANDERPAUW | 250 | 64,009 | 1,012 | 0 | 65,025 | 1,020 | 0 | 66,049 | 1,028 | 0 | 67,600 | 1,040 | 0 |
| 24 | VANDERPAUW | 250 | 64,009 | 1,012 | 0 | 65,025 | 1,020 | 0 | 66,049 | 1,028 | 0 | 67,600 | 1,040 | 0 |
| 25 | GOXCAP | 20 | 240 | 160 | 289 | 400 | 160 | 225 | 560 | 160 | 169 | 800 | 160 | 100 |
| 26 | FOXCAP | 100 | 12,769 | 452 | 0 | 13,225 | 460 | 0 | 13,689 | 468 | 0 | 14,400 | 480 | 0 |
| 27 | GOXCAP | 100 | 1,200 | 800 | 9,409 | 2,000 | 800 | 9,025 | 2,800 | 800 | 8,649 | 4,000 | 800 | 8,100 |
| 28 | FIELD PLATE | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | OVLAP | 100 | 1,200 | 800 | 9,409 | 2,000 | 800 | 9,025 | 2,800 | 800 | 8,649 | 4,000 | 800 | 8,100 |
| – | NOVLAP | 100 | 1,320 | 880 | 11,449 | 2,200 | 880 | 11,025 | 3,080 | 880 | 10,609 | 4,400 | 880 | 10,000 |
| 29 | POLYCOMB | 200 | 2,850 | 1,900 | 54,934 | 4,750 | 1,900 | 54,000 | 6,650 | 1,900 | 53,074 | 9,500 | 1,900 | 51,700 |
| 30 | GOXCOMB | 200 | 9,360 | 6,458 | 35,134 | 16,250 | 6,430 | 31,900 | 22,750 | 28,674 | 6,402 | 55,200 | 4,120 | 23,850 |
| 31 | GOXCAP | 200 | 2,400 | 1,600 | 38,809 | 4,000 | 1,600 | 38,025 | 5,600 | 1,600 | 37,249 | 8,000 | 1,600 | 36,100 |
| 32 | FOXCAP | 200 | 45,369 | 852 | 0 | 46,225 | 860 | 0 | 47,089 | 868 | 0 | 48,400 | 880 | 0 |
| 33 | FIELD PLATE | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | OVLAP | 200 | 2,400 | 1,600 | 38,809 | 4,000 | 1,600 | 38,025 | 5,600 | 1,600 | 37,249 | 8,000 | 1,600 | 36,100 |
| – | NOVLAP | 200 | 2,520 | 1,680 | 42,849 | 4,200 | 1,680 | 42,025 | 5,880 | 1,680 | 41,209 | 8,400 | 1,680 | 40,000 |
| 34 | POLYSHEET | 50SQ | 4,422 | 826 | 0 | 5,226 | 830 | 0 | 6,030 | 834 | 0 | 7,236 | 840 | 0 |
| 35 | POLYSHEET | COM | | | | | | | | | | | | |
| 36 | POLYSHEET | 80SQ | 3,200 | 816 | 0 | 4,000 | 820 | 0 | 4,800 | 824 | 0 | 6,000 | 830 | 0 |

* All dimensions are in micrometers.

## Table III—Pad key for chip B*

| Pad No. | MOS Device | Nom. Dim. | GASAD Area | GASAD Peri-meter | POLYCON Area | POLY Area | POLY Peri-meter | POLY Overlap FOX | POLY Overlap GOX | Window Area | Metal Area | GUARDRING (Width: 2) Area | GUARDRING Peri-meter | GUARDRING Enclosed Area |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | FOXCAP | 1,000 | 0 | 0 | 0 | 1,031,350 | 4,490 | 1,031,350 | 0 | 8,100 | 10,000 | 1,024,144 | 4,048 | 0 |
| 7 | GOXCAP | 1,000 | 1,000,000 | 4,000 | 100 | 1,031,350 | 4,490 | 31,350 | 1,000,000 | 8,100 | 10,000 | 8,000 | 8,000 | 996,004 |
| 13 | FIELD PLATE | 1,000 | 0 | 0 | 0 | 209,000 | 15,170 | 209,000 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| – | OVLAP | 1,000 | 1,000,000 | 4,000 | 980,100 | 1,020,100 | 4,040 | 20,100 | 1,000,000 | 980,100 | 1,020,100 | 8,000 | 8,000 | 996,004 |
| – | NOVLAP | 1,000 | 1,020,100 | 4,040 | 980,100 | 1,000,000 | 4,000 | 0 | 1,000,000 | 980,100 | 1,000,000 | 8,080 | 8,080 | 1,016,064 |
| 22 | POLYCOMB | 1,000 | 2,027,970 | 5,766 | 90 | 706,350 | 270,927 | 31,325 | 675,025 | 8,100 | 10,000 | 11,532 | 11,532 | 2,022,208 |
| 34 | GOXCOMB | 1,000 | 675,000 | 270,020 | 90 | 2,038,220 | 6,256 | 1,363,220 | 675,000 | 8,100 | 10,000 | 540,020 | 540,020 | 404,994 |
| 41 | FOXCAP | 2,000 | 0 | 0 | 0 | 4,051,350 | 8,490 | 4,051,350 | 0 | 8,100 | 10,000 | 4,048,144 | 8,048 | 0 |
| 49 | GOXCAP | 2,000 | 4,000,000 | 8,000 | 100 | 4,051,350 | 8,490 | 51,350 | 4,000,000 | 8,100 | 10,000 | 16,000 | 16,000 | 3,992,004 |
| 53 | IGFET: SOURCE | | 0 | 0 | 0 | | | | | 10,500 | 18,250 | 1,080 | 1,084 | 0 |
| 54 | IGFET: GATE | 500 | 275,000 | 2,100 | 0 | 265,500 | 2,500 | 15,500 | 250,000 | 8,100 | 10,000 | | | |
| 55 | IGFET: DRAIN | | | | 0 | | | | | 10,500 | 18,250 | 1,080 | 1,084 | 0 |
| 56 | IGFET: SOURCE | | 0 | 0 | 0 | | | | | 9,500 | 18,647.5 | 680 | 684 | 0 |
| 57 | IGFET: GATE | 300 | 105,000 | 1,300 | 0 | 103,500 | 1,700 | 13,500 | 90,000 | 8,100 | 10,000 | | | |
| 58 | IGFET: DRAIN | | | | 0 | | | | | 9,500 | 18,647.5 | 680 | 684 | 0 |
| 59 | IGFET: SOURCE | | | | 0 | | | | | 9,000 | 16,487.5 | 480 | 484 | 0 |
| 60 | IGFET: GATE | 200 | 50,000 | 900 | 0 | 52,500 | 1,300 | 12,500 | 40,000 | 8,100 | 10,000 | | | |
| 61 | IGFET: DRAIN | | | | 0 | | | | | 9,000 | 16,487.5 | 480 | 484 | 0 |
| 62 | IGFET: SOURCE | | | | 0 | | | | | 8,500 | 14,987.5 | 280 | 284 | 0 |
| 63 | IGFET: GATE | 100 | 15,000 | 500 | 0 | 21,500 | 900 | 11,500 | 10,000 | 8,100 | 10,000 | | | |
| 64 | IGFET: DRAIN | | | | 0 | | | | | 8,500 | 14,987.5 | 280 | 284 | 0 |
| 65 | GATOD: JUNC. | 500 | 278,200 | 2,110 | 0 | | | | | 10,600 | 18,475 | 4,220 | 4,220 | 276,094 |
| 67 | GATOD: GATE | | | | 0 | 263,475 | 2,517 | 13,375 | 250,100 | 8,100 | 10,000 | | | |
| 68 | GATOD: JUNC. | 300 | 107,200 | 1,310 | 0 | | | | | 9,600 | 15,475 | 2,620 | 2,620 | 105,894 |
| 69 | GATOD: GATE | | | | 0 | 102,350 | 1,672 | 12,250 | 90,100 | 8,100 | 10,000 | | | |
| 70 | GATOD: JUNC. | 200 | 51,700 | 910 | 0 | | | | | 9,100 | 13,975 | 1,820 | 1,820 | 50,794 |
| 71 | GATOD: GATE | | | | 0 | 52,190 | 1,618 | 12,090 | 40,100 | 8,100 | 10,000 | | | |
| 72 | GATOD: JUNC. | 100 | 16,200 | 510 | 0 | | | | | 8,100 | 8,600 | 1,020 | 1,020 | 15,694 |
| 73 | GATOD: GATE | | | | 0 | 21,600 | 1,100 | 11,500 | 10,100 | 8,100 | 10,000 | | | |
| 75 | FIELD PLATE | 2,000 | 0 | 0 | 0 | 436,500 | 30,220 | 436,500 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| – | OVLAP | 2,000 | 4,000,000 | 8,000 | 3,960,100 | 4,040,100 | 8,040 | 40,100 | 4,000,000 | 3,960,100 | 4,040,100 | 16,080 | 16,080 | 3,992,004 |
| – | NOVLAP | 2,000 | 4,040,100 | 8,040 | 3,960,100 | 4,000,000 | 8,000 | 0 | 4,000,000 | 3,960,100 | 4,000,000 | 16,080 | 16,080 | 4,032,064 |
| 76 | IGFET: COMSOURCE | | 0 | 0 | 0 | 10,000 | 400 | 10,000 | 0 | 12,500 | 61,100 | 280 | 284 | 0 |
| 77 | IGFET: DRAIN | 1.0 | 5,100 | 302 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,935 | | | |
| 78 | IGFET: DRAIN | 1.5 | 5,150 | 303 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,875 | | | |
| 79 | IGFET: DRAIN | 2 | 5,200 | 304 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,870 | | | |
| 80 | IGFET: DRAIN | 3 | 5,300 | 306 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,820 | | | |
| 81 | IGFET: DRAIN | 4 | 5,400 | 308 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,770 | | | |
| 82 | IGFET: DRAIN | 5 | 5,500 | 310 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,720 | | | |
| 83 | IGFET: DRAIN | 6 | 5,600 | 312 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,670 | | | |
| 84 | IGFET: DRAIN | 8 | 5,800 | 316 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,565 | | | |
| 85 | IGFET: DRAIN | 10 | 6,000 | 320 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,465 | | | |
| 86 | IGFET: DRAIN | 15 | 6,500 | 330 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 16,225 | | | |
| 87 | IGFET: DRAIN | 20 | 7,000 | 340 | 0 | 10,000 | 400 | 10,000 | 0 | 8,500 | 15,980 | | | |
| 88 | IGFET: COMGATE | | 0 | 0 | 0 | 31,650 | 5,595 | 24,100 | 7,550 | 8,100 | 10,000 | 280 | 284 | 0 |
| 95 | POLYCOMB | 2,000 | 8,028,720 | 11,334 | 90 | 2,717,840 | 1,072,598 | 51,350 | 2,666,490 | 8,100 | 10,000 | 22,668 | 22,668 | 8,017,390 |
| 106 | GOXCOMB | 2,000 | 2,676,190 | 1,070,486 | 90 | 8,038,970 | 11,824 | 5,362,780 | 2,676,190 | 8,100 | 10,000 | 2,140,972 | 2,140,972 | 1,605,708 |

* All dimensions are in micrometers.

## Table III—Pad key for chip B* (*Continued*)

| PAD No. | MOS Device | Nom. Dim. | GUARDRING (Width: 3) | | | GUARDRING (Width: 5) | | | GUARDRING (Width: 7) | | | GUARDRING (Width: 10) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Area | Perimeter | Enclosed Area | Area | Perimeter | Enclosed Area | Area | Perimeter | Enclosed Area | Area | Perimeter | Enclosed Area |
| 3 | FOXCAP | 1,000 | 1,026,169 | 4,052 | 0 | 1,030,225 | 4,060 | 0 | 1,034,289 | 4,068 | 0 | 1,040,400 | 4,080 | 0 |
| 7 | GOXCAP | 1,000 | 12,000 | 8,000 | 994,009 | 20,000 | 8,000 | 990,025 | 28,000 | 8,000 | 986,049 | 40,000 | 8,000 | 980,100 |
| 13 | FIELD PLATE | 1,000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | OVLAP | 1,000 | 12,000 | 8,000 | 994,009 | 20,000 | 8,000 | 990,025 | 28,000 | 8,000 | 986,049 | 40,000 | 8,000 | 980,100 |
| – | NOVLAP | 1,000 | 12,120 | 8,080 | 1,014,049 | 20,200 | 8,080 | 1,010,025 | 28,280 | 8,080 | 1,006,009 | 40,400 | 8,080 | 1,000,000 |
| 22 | POLYCOMB | 1,000 | 17,298 | 11,532 | 2,019,330 | 28,830 | 11,532 | 2,013,580 | 40,362 | 11,532 | 2,007,838 | 57,660 | 11,532 | 1,999,240 |
| 34 | GOXCOMB | 1,000 | 810,030 | 540,020 | 269,994 | 2,013,580 | 5,746 | 0 | 2,019,330 | 5,754 | 0 | 2,027,970 | 5,766 | 0 |
| 41 | FOXCAP | 2,000 | 4,052,169 | 8,052 | 0 | 4,060,225 | 8,060 | 0 | 4,068,289 | 8,068 | 0 | 4,080,400 | 8,080 | 0 |
| 49 | GOXCAP | 2,000 | 24,000 | 16,000 | 3,988,009 | 40,000 | 16,000 | 3,980,025 | 56,000 | 16,000 | 3,972,049 | 80,000 | 16,000 | 3,960,100 |
| 53 | IGFET: SOURCE | | 1,620 | 1,086 | 0 | 2,700 | 1,090 | 0 | 3,780 | 1,094 | 0 | 5,400 | 1,100 | 0 |
| 54 | IGFET: GATE | 500 | | | | | | | | | | | | |
| 55 | IGFET: DRAIN | | 1,620 | 1,086 | 0 | 2,700 | 1,090 | 0 | 3,780 | 1,094 | 0 | 5,400 | 1,100 | 0 |
| 56 | IGFET: SOURCE | | 1,020 | 686 | 0 | 1,700 | 690 | 0 | 2,380 | 694 | 0 | 3,400 | 700 | 0 |
| 57 | IGFET: GATE | 300 | | | | | | | | | | | | |
| 58 | IGFET: DRAIN | | 1,020 | 686 | 0 | 1,700 | 690 | 0 | 2,380 | 694 | 0 | 3,400 | 700 | 0 |
| 59 | IGFET: SOURCE | | 720 | 486 | 0 | 1,200 | 490 | 0 | 1,680 | 494 | 0 | 2,400 | 500 | 0 |
| 60 | IGFET: GATE | 200 | | | | | | | | | | | | |
| 61 | IGFET: DRAIN | | 720 | 486 | 0 | 1,200 | 490 | 0 | 1,680 | 494 | 0 | 2,400 | 500 | 0 |
| 62 | IGFET: SOURCE | | 420 | 286 | 0 | 700 | 290 | 0 | 980 | 294 | 0 | 1,400 | 300 | 0 |
| 63 | IGFET: GATE | 100 | | | | | | | | | | | | |
| 64 | IGFET: DRAIN | | 420 | 286 | 0 | 700 | 290 | 0 | 980 | 294 | 0 | 1,400 | 300 | 0 |
| 65 | GATOD: JUNC. | 500 | 6,330 | 4,220 | 275,044 | 10,550 | 4,220 | 272,950 | 14,770 | 4,220 | 270,864 | 21,100 | 4,220 | 267,750 |
| 67 | GATOD: GATE | | | | | | | | | | | | | |
| 68 | GATOD: JUNC. | 300 | 3,930 | 2,620 | 105,244 | 6,550 | 2,620 | 103,950 | 9,170 | 2,620 | 102,664 | 13,100 | 1,620 | 100,750 |
| 69 | GATOD: GATE | | | | | | | | | | | | | |
| 70 | GATOD: JUNC. | 200 | 2,730 | 1,820 | 50,344 | 4,550 | 1,820 | 49,450 | 6,370 | 1,820 | 48,564 | 9,100 | 1,820 | 47,250 |
| 71 | GATOD: GATE | | | | | | | | | | | | | |
| 72 | GATOD: JUNC. | 100 | 1,530 | 1,020 | 15,644 | 2,550 | 1,020 | 14,950 | 3,570 | 1,020 | 14,464 | 5,100 | 1,020 | 13,750 |
| 73 | GATOD: GATE | | | | | | | | | | | | | |
| 75 | FIELD PLATE | 2,000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | OVLAP | 2,000 | 24,000 | 16,000 | 3,988,009 | 40,000 | 16,000 | 3,980,025 | 56,000 | 16,000 | 3,972,049 | 80,000 | 16,000 | 3,960,100 |
| – | NOVLAP | 2,000 | 24,120 | 16,080 | 4,028,049 | 40,200 | 16,080 | 4,020,025 | 56,280 | 16,080 | 4,012,009 | 80,400 | 16,080 | 4,000,000 |
| 76 | IGFET: COMSOURCE | | 420 | 286 | 0 | 700 | 290 | 0 | 980 | 294 | 0 | 1,400 | 300 | 0 |
| 77 | IGFET: DRAIN | 1.0 | | | | | | | | | | | | |
| 78 | IGFET: DRAIN | 1.5 | | | | | | | | | | | | |
| 79 | IGFET: DRAIN | 2 | | | | | | | | | | | | |
| 80 | IGFET: DRAIN | 3 | | | | | | | | | | | | |
| 81 | IGFET: DRAIN | 4 | | | | | | | | | | | | |
| 82 | IGFET: DRAIN | 5 | | | | | | | | | | | | |
| 83 | IGFET: DRAIN | 6 | | | | | | | | | | | | |
| 84 | IGFET: DRAIN | 8 | | | | | | | | | | | | |
| 85 | IGFET: DRAIN | 10 | | | | | | | | | | | | |
| 86 | IGFET: DRAIN | 15 | | | | | | | | | | | | |
| 87 | IGFET: DRAIN | 20 | | | | | | | | | | | | |
| 88 | COMGATE | | 420 | 286 | 0 | 700 | 290 | 0 | 980 | 294 | 0 | 1,400 | 300 | 0 |
| 95 | POLYCOMB | 2,000 | 34,002 | 22,668 | 8,011,728 | 56,670 | 22,668 | 8,000,410 | 79,338 | 22,668 | 7,989,100 | 113,340 | 22,668 | 7,972,150 |
| 106 | GOXCOMB | 2,000 | 3,211,458 | 2,140,972 | 1,070,470 | 8,000,410 | 11,314 | 0 | 8,011,728 | 11,322 | 0 | 8,028,720 | 11,334 | 0 |

* All dimensions are in micrometers.

## Table IV—Pad key for chips C1 through C4*

| Pad No. | MOS Device | Chip | Nom. Dim. | GASAD Area | GASAD Perimeter | Polycom Area | POLY Area | POLY Perimeter | POLY Overlap FOX | POLY Overlap GOX | Window Area | Metal Area | GUARDRING (Width: 10) Area | GUARDRING (Width: 10) Perimeter | GUARDRING (Width: 10) Enclosed Area |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 42 | GOXCAP | C1 | 4,000 | 16,000,000 | 16,400 | 100 | 16,093,350 | 16,890 | 93,350 | 16,000,000 | 8,100 | 10,000 | 164,000 | 32,800 | 15,918,100 |
| 43 | FOXCAP | C1 | 4,000 | 0 | 0 | 0 | 16,093,350 | 16,890 | 16,093,350 | 0 | 8,100 | 10,000 | 16,164,400 | 16,480 | 0 |
| 67 | GOXCOMB | C2 | 4,000 | 14,400,000 | 5,760,010 | 90 | 43,221,500 | 27,020 | 28,821,500 | 14,400,000 | 8,100 | 10,000 | 43,211,250 | 26,530 | 0 |
| 68-71 | POLYCOMB | C3 | 4,000 | 43,211,250 | 26,530 | 90 | 14,493,350 | 5,762,167 | 93,350 | 14,400,000 | 32,400 | 40,000 | 265,300 | 53,060 | 43,078,700 |
| 14 | FIELD PLATE | C4 | 4,000 | 0 | 0 | 0 | 715,500 | 56,600 | 715,500 | 0 | 8,100 | 10,000 | 0 | 0 | 0 |
| – | OVLAP | C4 | 4,000 | 16,000,000 | 16,400 | 15,918,100 | 16,082,100 | 16,440 | 82,100 | 16,000,000 | 15,918,100 | 16,082,100 | 164,000 | 32,800 | 15,918,100 |
| – | NOVLAP | C4 | 4,000 | 16,082,100 | 16,440 | 15,918,100 | 16,000,000 | 16,400 | 0 | 16,000,000 | 15,918,100 | 16,000,000 | 164,400 | 32,880 | 16,000,000 |
| | DOUBLE VANDERPAUW | | | | | | | | | | | | | | |
| 75 | GASAD | C4 | 250 | 67,812.5 | 1,425 | 585.0 | 53,252.5 | 3,865 | | | 8,685.0 | 43,382.5 | 0 | 0 | 0 |
| 76 | POLY | C4 | 260 | 0 | 0 | 0 | 169,975 | 7,430 | 101,875 | 68,100 | 8,685.0 | 34,675.0 | 0 | 0 | 0 |
| 77 | GASAD | C4 | 250 | 67,812.5 | 1,425 | 487.5 | 29,597.5 | 1,971 | | | 8,587.5 | 19,715.0 | 0 | 0 | 0 |
| 78 | POLY | C4 | 260 | 0 | 0 | 0 | 169,975 | 7,430 | 101,875 | 68,100 | 8,587.5 | 13,125.0 | 0 | 0 | 0 |
| 79 | GASAD | C4 | 250 | 67,812.5 | 1,425 | 337.5 | 15,077.5 | 811 | | | 8,437.5 | 5,202.5 | 0 | 0 | 0 |
| 80 | POLY | C4 | 260 | 0 | 0 | 0 | 169,975 | 7,430 | 101,875 | 68,100 | 8,437.5 | 3,062.5 | 0 | 0 | 0 |
| 81 | GASAD | C4 | 250 | 67,812.5 | 1,425 | 502.5 | 16,952.5 | 961 | | | 8,602.5 | 7,072.5 | 0 | 0 | 0 |
| 82 | POLY | C4 | 260 | 0 | 0 | 0 | 169,975 | 7,430 | 101,875 | 68,100 | 8,602.5 | 15,325.0 | 0 | 0 | 0 |
| 87 | GUARDRING | C4 | 250 | 0 | 0 | 585.0 | 55,600 | 4,048 | 55,600 | 0 | 8,685.0 | 55,725 | 67,812.5 | 1,425 | 0 |
| 88 | POLY | C4 | 260 | 0 | 0 | 0 | 172,875 | 7,662 | 172,875 | 0 | 8,685.0 | 43,875 | 0 | 0 | 0 |
| 89 | GUARDRING | C4 | 250 | 0 | 0 | 487.5 | 29,425 | 1,954 | 29,425 | 0 | 8,587.5 | 29,550 | 67,812.5 | 1,425 | 0 |
| 90 | POLY | C4 | 260 | 0 | 0 | 0 | 172,875 | 7,662 | 172,875 | 0 | 8,587.5 | 23,550 | 0 | 0 | 0 |
| 91 | GUARDRING | C4 | 250 | 0 | 0 | 337.5 | 14,300 | 794 | 14,300 | 0 | 8,437.5 | 14,425 | 67,812.5 | 1,425 | 0 |
| 92 | POLY | C4 | 260 | 0 | 0 | 0 | 172,875 | 7,662 | 172,875 | 0 | 8,437.5 | 11,675 | 0 | 0 | 0 |
| 93 | GUARDRING | C4 | 250 | 0 | 0 | 502.5 | 16,800 | 944 | 16,800 | 0 | 8,602.5 | 16,925 | 67,812.5 | 1,425 | 0 |
| 94 | POLY | C4 | 260 | 0 | 0 | 0 | 172,875 | 7,662 | 172,875 | 0 | 8,602.5 | 26,675 | 0 | 0 | 0 |
| | VANDERPAUW | | | | | | | | | | | | | | |
| 96 | GUARDRING | C4 | 250 | 0 | 0 | 585.0 | 31,350 | 2,108 | 31,350 | 0 | 8,685.0 | 31,325 | 67,225 | 1,378 | 0 |
| 97 | GUARDRING | C4 | 250 | 0 | 0 | 487.5 | 16,025 | 882 | 16,025 | 0 | 8,587.5 | 16,025 | 67,225 | 1,378 | 0 |
| 98 | GUARDRING | C4 | 250 | 0 | 0 | 337.5 | 11,650 | 532 | 11,650 | 0 | 8,437.5 | 11,650 | 67,225 | 1,378 | 0 |
| 99 | GUARDRING | C4 | 250 | 0 | 0 | 502.5 | 17,900 | 1,032 | 17,900 | 0 | 8,602.5 | 17,900 | 67,225 | 1,378 | 0 |
| 101 | GASAD | C4 | 250 | 67,075 | 1,366 | 585.0 | 31,387.5 | 2,111 | 30,087.5 | 1,300.0 | 8,685.0 | 31,387.5 | 0 | 0 | 0 |
| 102 | GASAD | C4 | 250 | 67,075 | 1,366 | 487.5 | 16,062.5 | 885 | 14,875.0 | 1,187.5 | 8,587.5 | 16,062.5 | 0 | 0 | 0 |
| 103 | GASAD | C4 | 250 | 67,075 | 1,366 | 337.5 | 11,687.5 | 535 | 10,775.0 | 912.5 | 8,437.5 | 11,687.5 | 0 | 0 | 0 |
| 104 | GASAD | C4 | 250 | 67,075 | 1,366 | 502.5 | 17,937.5 | 1,035 | 16,762.5 | 1,175.0 | 8,602.5 | 17,937.5 | 0 | 0 | 0 |

* All dimensions are in micrometers.

Table V—Pad key for METEST chip D*

| PAD NO. | GASAD | | | POLYCON | | | POLY | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TUB | | | WINDOW | | | TAPS | | | TIES | | | CONTACTS | |
| | SIZE | NO. | AREA | SIZE | NO. | AREA* | $W_1$ | $L_1$ | $SQ_1$ | $W_2$ | $L_2$ | $SQ_2$ | $W_3$ | $L_3$ |
| 1,2 | 20×78 | 4 | 6,240 | 1.5 | 6 | 13.5 | 25 | 187 | 7.48 | 5 | 50 | 10 | 11.5 | 272 |
| 3,4 | 20×36 | 2 | 1,440 | 3 | 1 | 9 | 25 | 125 | 5 | 5 | 153 | 30.6 | 13 | 52 |
| 4,5 | 20×36 | 1 | 720 | 3 | 1 | 9 | 25 | 125 | 5 | 5 | 189 | 37.8 | 13 | 26 |
| 5,6 | 20×36 | 4 | 2,880 | 3 | 1 | 9 | 25 | 125 | 5 | 5 | 81 | 16.2 | 13 | 104 |
| 7,26 | 20×70 | 4 | 5,600 | 2 | 4 | 16 | 25 | 125 | 5 | 5 | 746.5 | 149.3 | 12 | 240 |
| 7,8 | 20×70 | 7 | 9,800 | 2 | 4 | 16 | 25 | 125 | 5 | 5 | 1788.0 | 357.6 | 12 | 420 |
| 8,25 | 20×70 | 2 | 2,800 | 2 | 4 | 16 | 25 | 125 | 5 | 5 | 886.5 | 177.3 | 12 | 120 |
| 9,10 | 20×34 | 2 | 1,360 | 2 | 1 | 4 | 25 | 125 | 5 | 5 | 157 | 31.4 | 12 | 48 |
| 10,11 | 20×34 | 1 | 680 | 2 | 1 | 4 | 25 | 125 | 5 | 5 | 191 | 38.2 | 12 | 24 |
| 11,12 | 20×34 | 4 | 2,720 | 2 | 1 | 4 | 25 | 125 | 5 | 5 | 89 | 17.8 | 12 | 96 |
| 13,14 | 20×33 | 2 | 1,320 | 1.5 | 1 | 2.25 | 25 | 125 | 5 | 5 | 159 | 31.8 | 11.5 | 46 |
| 14,15 | 20×33 | 1 | 660 | 1.5 | 1 | 2.25 | 25 | 125 | 5 | 5 | 222 | 44.4 | 11.5 | 23 |
| 15,16 | 20×33 | 4 | 2,640 | 1.5 | 1 | 2.25 | 25 | 125 | 5 | 5 | 93 | 18.6 | 11.5 | 92 |
| 17,18 | 20×32 | 2 | 1,280 | 1.0 | 1 | 1 | 25 | 125 | 5 | 5 | 161 | 32.2 | 11 | 44 |
| 18,19 | 20×32 | 1 | 640 | 1.0 | 1 | 1 | 25 | 125 | 5 | 5 | 223 | 44.6 | 11 | 22 |
| 19,20 | 20×32 | 4 | 2,560 | 1.0 | 1 | 1 | 25 | 125 | 5 | 5 | 97 | 19.4 | 11 | 88 |
| 21,22 | 20×44 | 2 | 1,760 | 7 | 1 | 49 | 25 | 125 | 5 | 5 | 137 | 27.4 | 17 | 68 |
| 22,23 | 20×44 | 1 | 880 | 7 | 1 | 49 | 25 | 125 | 5 | 5 | 181 | 36.2 | 17 | 34 |
| 23,24 | 20×44 | 4 | 3,520 | 7 | 1 | 49 | 25 | 125 | 5 | 5 | 49 | 9.8 | 17 | 136 |
| 25,26 | 20×70 | 1 | 1,400 | 2 | 4 | 16 | 25 | 125 | 5 | 5 | 155 | 31 | 12 | 60 |
| 27,28 | 20×40 | 2 | 1,600 | 5 | 1 | 25 | 25 | 125 | 5 | 5 | 145 | 29 | 15 | 60 |
| 28,29 | 20×40 | 1 | 800 | 5 | 1 | 25 | 25 | 125 | 5 | 5 | 185 | 37 | 15 | 30 |
| 29,30 | 20×40 | 4 | 3,200 | 5 | 1 | 25 | 25 | 125 | 5 | 5 | 65 | 13 | 15 | 120 |
| 31,32 | 20×74 | 2 | 2,960 | 1.0 | 8 | 8 | 25 | 125 | 5 | 5 | 77 | 15.4 | 11 | 128 |
| 32,33 | 20×74 | 1 | 1,480 | 1.0 | 8 | 8 | 25 | 125 | 5 | 5 | 143.5 | 28.7 | 11 | 64 |
| 33,34 | 20×74 | 4 | 5,920 | 1.0 | 8 | 8 | 25 | 171 | 6.84 | 5 | 50 | 10 | 11 | 256 |
| 35,36 | 20×78 | 2 | 3,120 | 1.5 | 6 | 13.5 | 25 | 125 | 5 | 5 | 69 | 13.8 | 11.5 | 136 |
| 36,1 | 20×78 | 1 | 1,560 | 1.5 | 6 | 13.5 | 25 | 125 | 5 | 5 | 139.5 | 27.9 | 11.5 | 68 |

* All dimensions are in micrometers.
† Cross-sectional area of tub input or output.

Table VI—Mask levels

| Suggested Sequence | Mask Level | Tone | Note | Features | Background |
|---|---|---|---|---|---|
| 1 | GUARDRING | Normal | 1 | Clear | Opaque |
| 2 | GASAD | Normal | 1 | Clear | Opaque |
| 3 | POLYCON | Normal | 1 | Clear | Opaque |
| 4 | POLY | Normal | 1 | Opaque | Clear |
| 5 | WINDOW | Normal | 1 | Clear | Opaque |
| 6 | METAL | Normal | 1 | Opaque | Clear |
| $\bar{2}$ | R. T. GASAD | Reverse | 2 | Opaque | Clear |
| $\bar{4}$ | R. T. POLY | Reverse | 2 | Clear | Opaque |
| $\bar{6}$ | R. T. METAL | Reverse | 2 | Clear | Opaque |

1. For use with positive photoresist.
2. For use with negative photoresist, uniform gold metallization, selective oxidation, or other special processes.

Table VII—Experimental devices

| Device Structures | | Required Mask Levels | | | | | |
|---|---|---|---|---|---|---|---|
| | | GUARDRING | GASAD | POLYCON | POLY | WINDOW | METAL |
| MOS Capacitors | | | * | | * | | |
| Schottky Diodes: | Guarded | * | * | | * | | |
| | Unguarded | | * | | * | | |
| Ohmic Contacts | | | * | * | * | | |
| p-n Junctions | | | * | * | * | | |
| IGFETs | | | * | | * | * | * |
| Gated Diodes | GATODE | | * | | * | * | * |
| SADSHEET | | | * | | | * | * |
| POLYSHEET | | | | | * | | |
| VANDERPAUW: | POLY | | | | * | | |
| | GASAD | | * | * | * | | |
| | GUARDRING | * | | * | * | | |
| Contact Metallization Test Cells | METEST | | * | * | * | | |

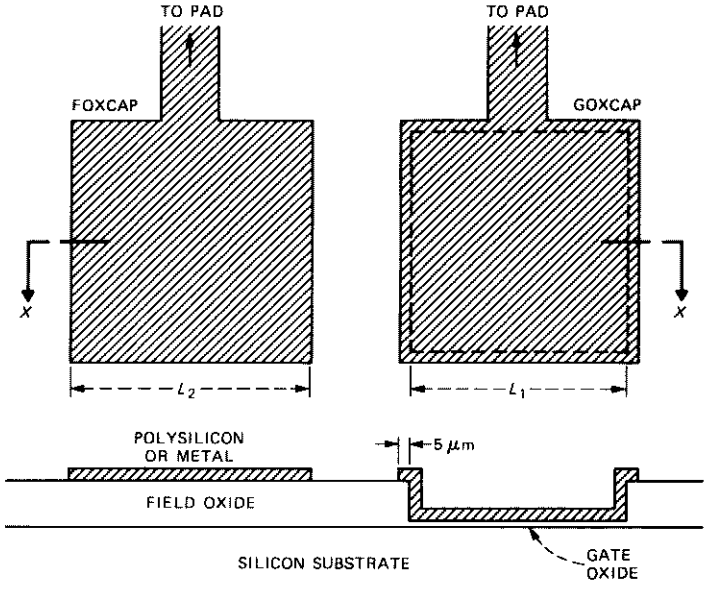



Fig. 1—FOXCAP and GOXCAP MOS capacitors (chips A through C).

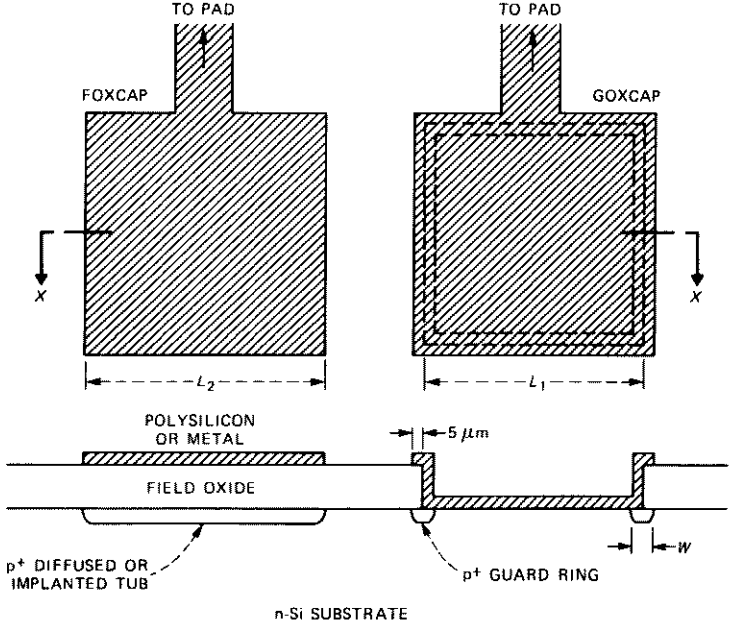



Fig. 2—Guarded Schottky diode and buried channel capacitor formed with FOXCAP and GOXCAP features (chips A through C).

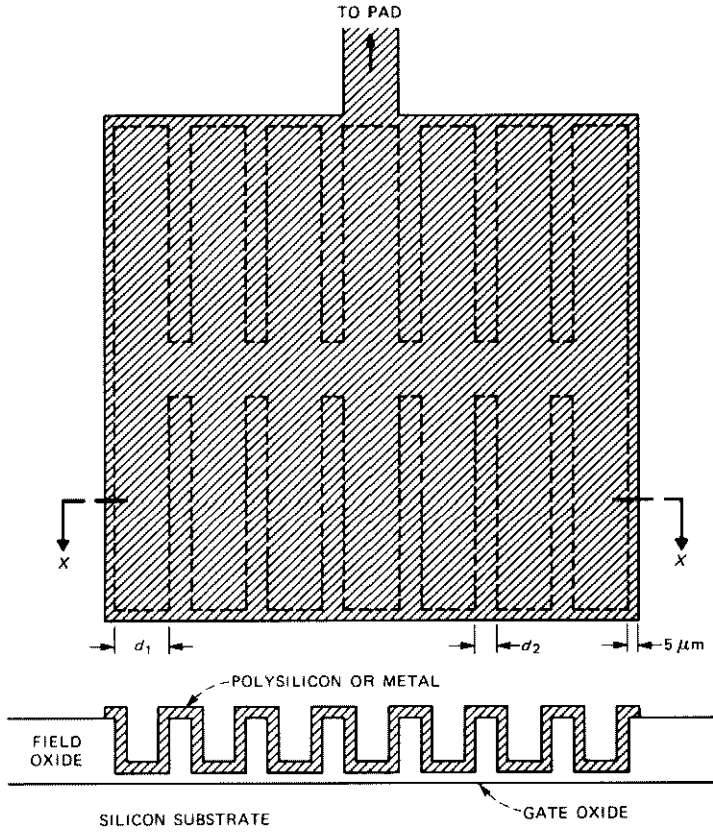



Fig. 3—GOXCOMB MOS capacitor (chips A through C).

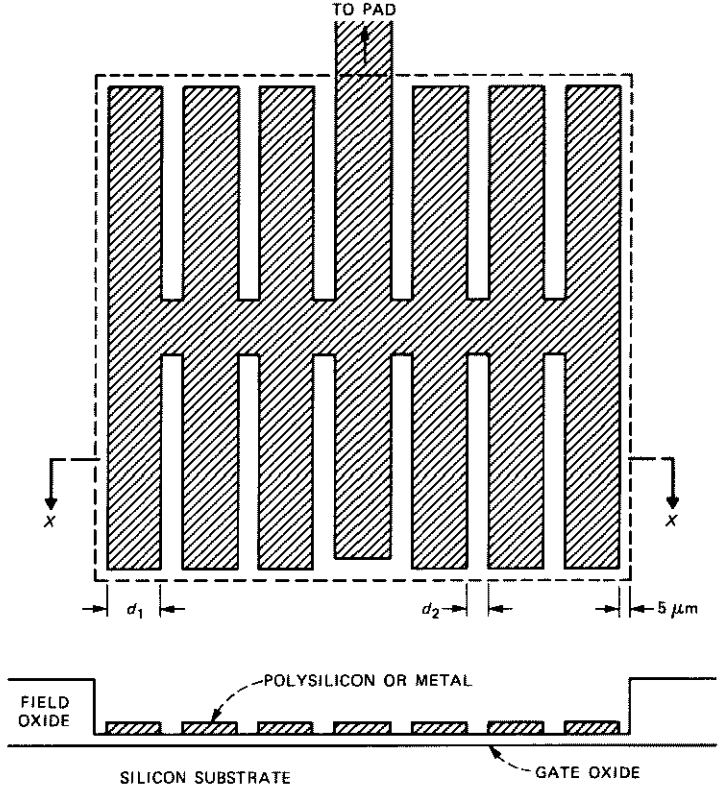



Fig. 4—POLYCOMB MOS capacitor (chips A through C).

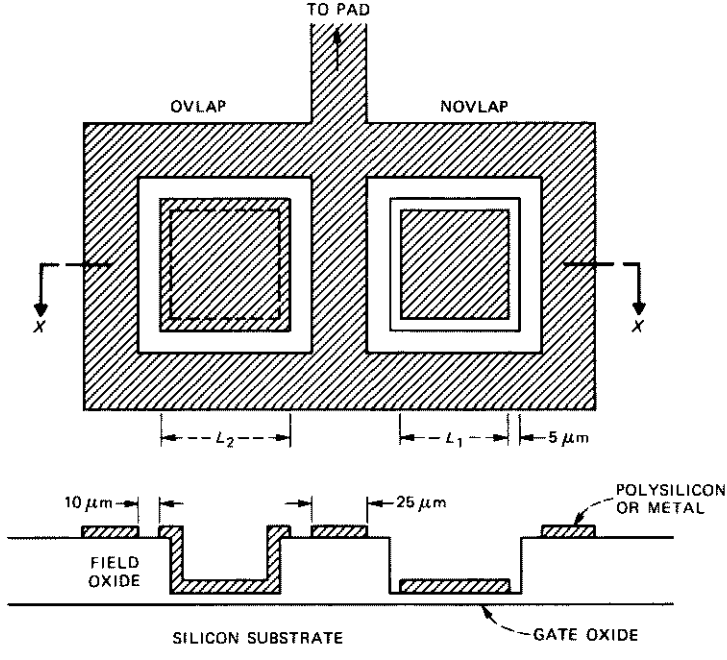



Fig. 5—OVLAP and NOVLAP MOS capacitors with field plate to control surrounding surface potential (chips A through C).

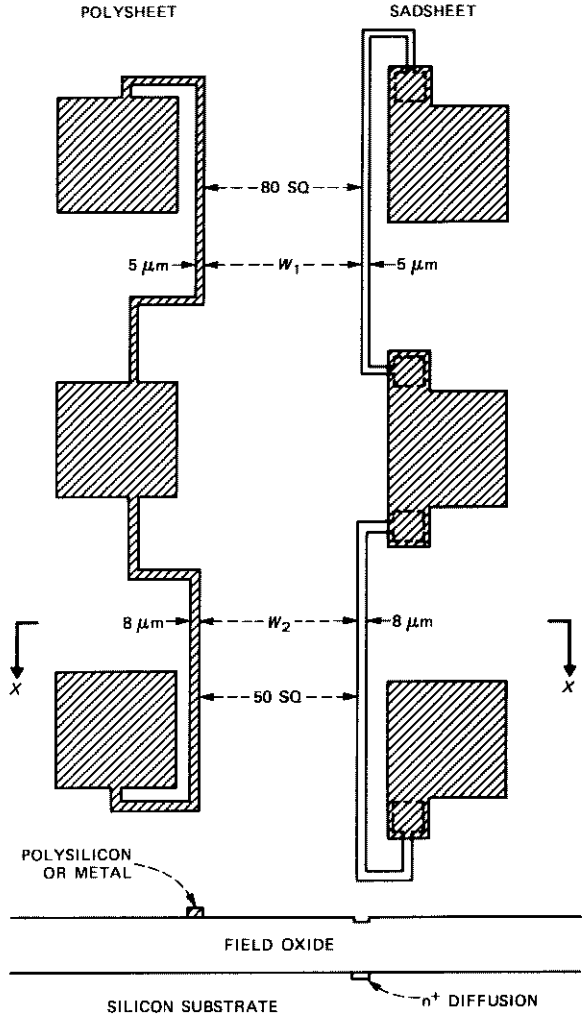



Fig. 6—POLYSHEET and SADSHEET sheet resistance and linewidth features (chip A).

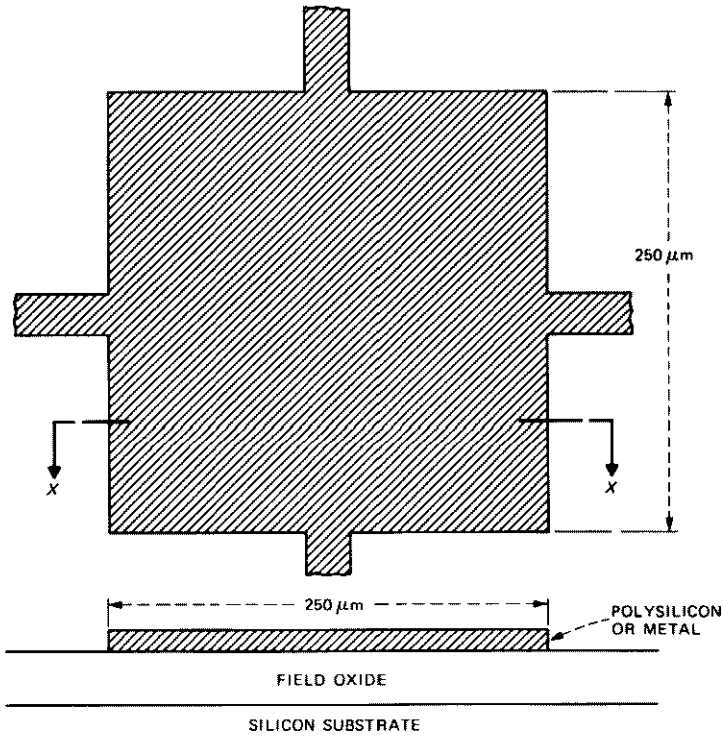



Fig. 7—Van der Pauw pattern (chip A and chip C4).

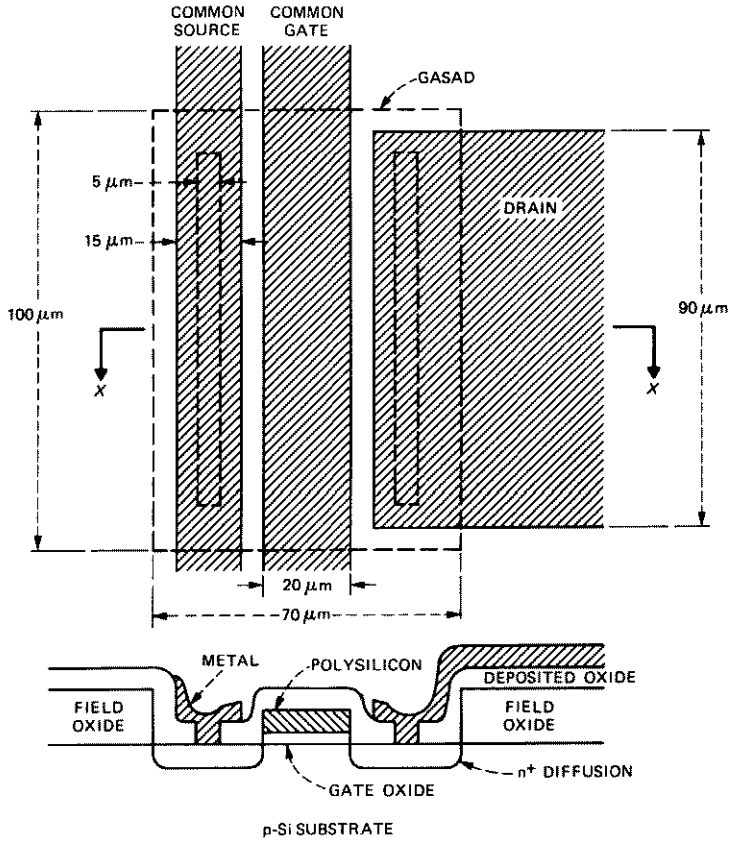



Fig. 8—Standard IGFET with common sources and drains (chip B).

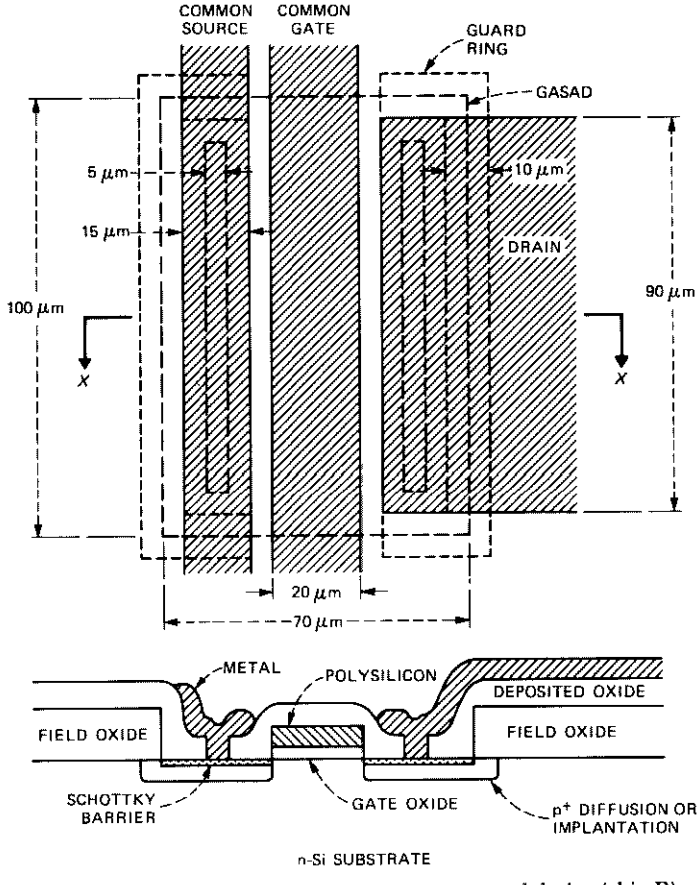



Fig. 9—Guarded IGFET with common sources and drains (chip B).

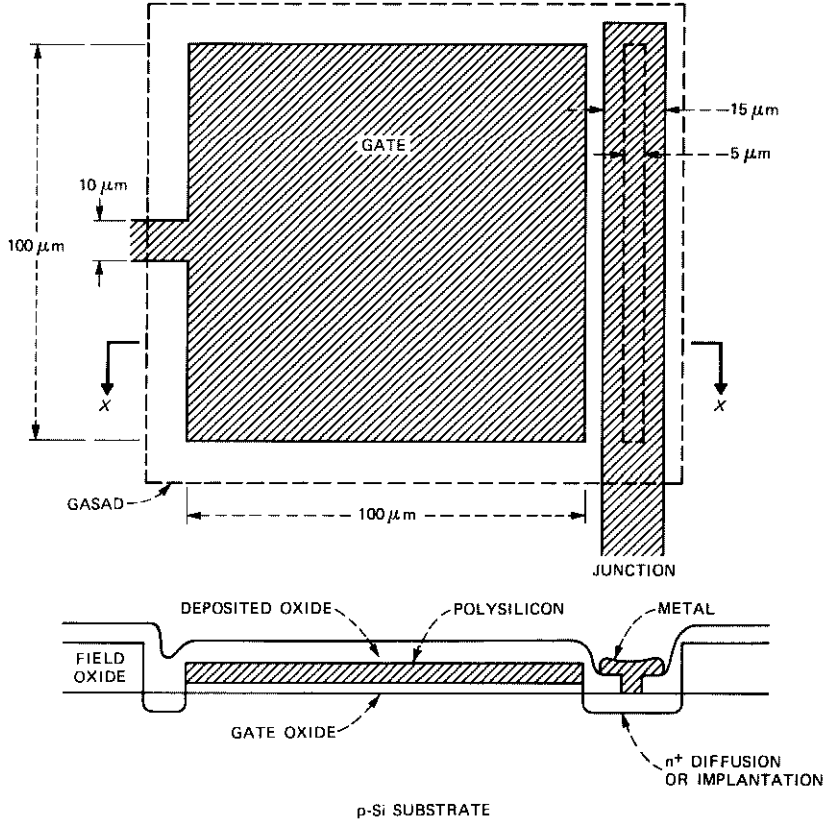



Fig. 10—Gated diode with $n^+$ diffusion or implantation completely surrounding the gate (GATODE chip B).

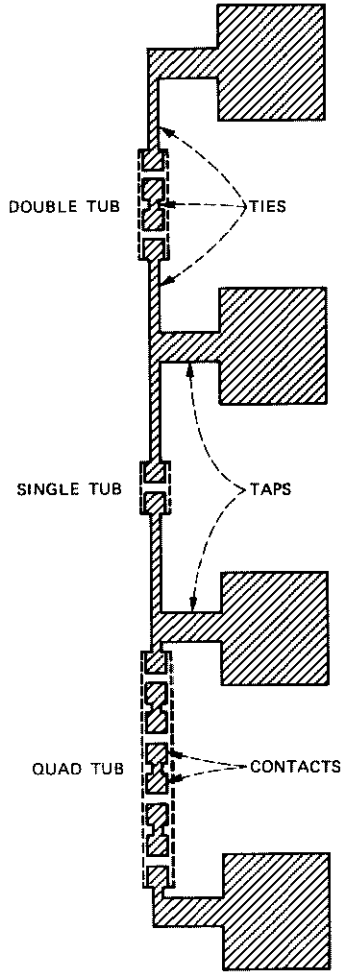



Fig. 11—Tapped string for the metallization test chip (METEST chip D).

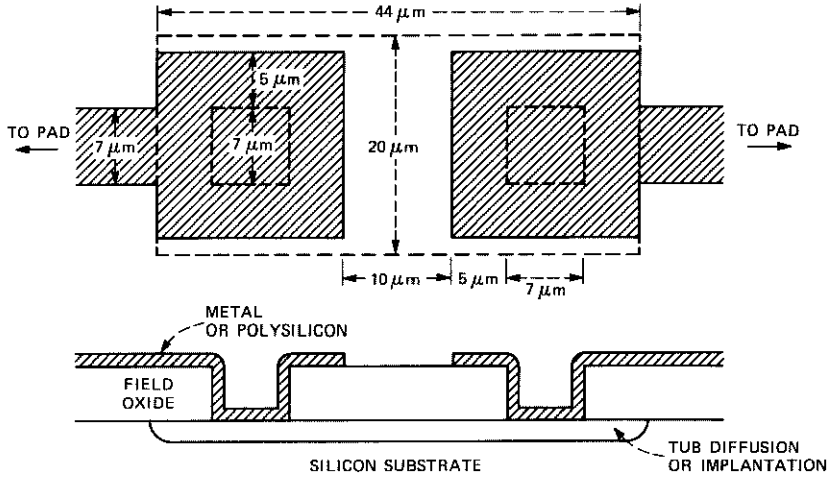
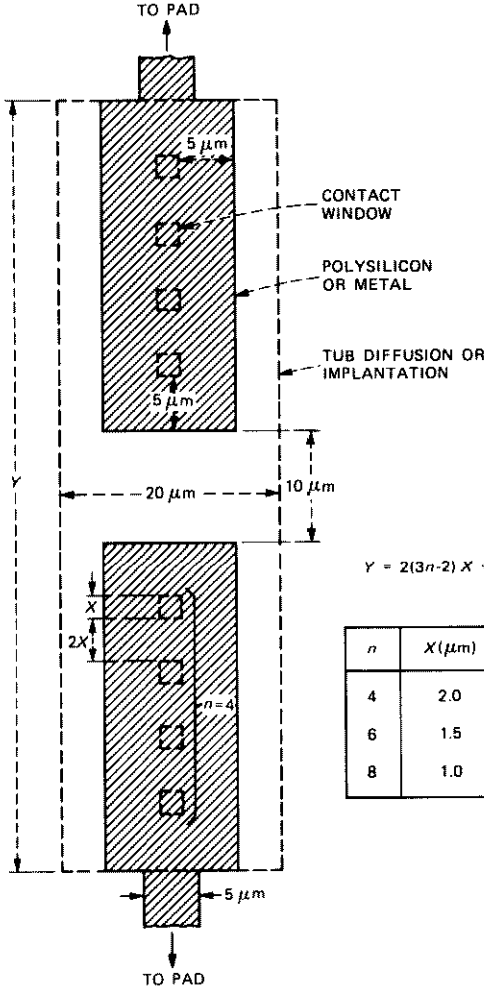



Fig. 12—Dual contact cell for the metallization test chip (METEST chip D).




$Y = 2(3n-2)X + 30$

| $n$ | $X$ (μm) | $Y$ (μm) |
|---|---|---|
| 4 | 2.0 | 70 |
| 6 | 1.5 | 78 |
| 8 | 1.0 | 74 |

Fig. 13—Multiple contact cell for the metallization test chip (METEST chip D).

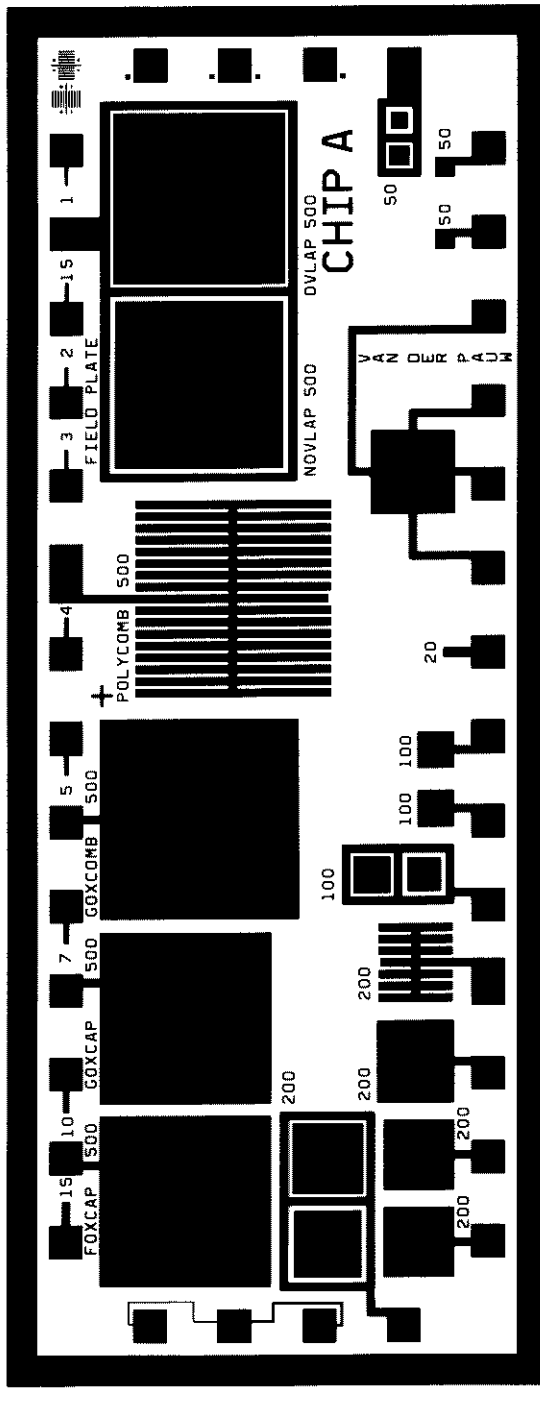



Fig. 14—Composite POLY and WINDOW levels for the A chip.

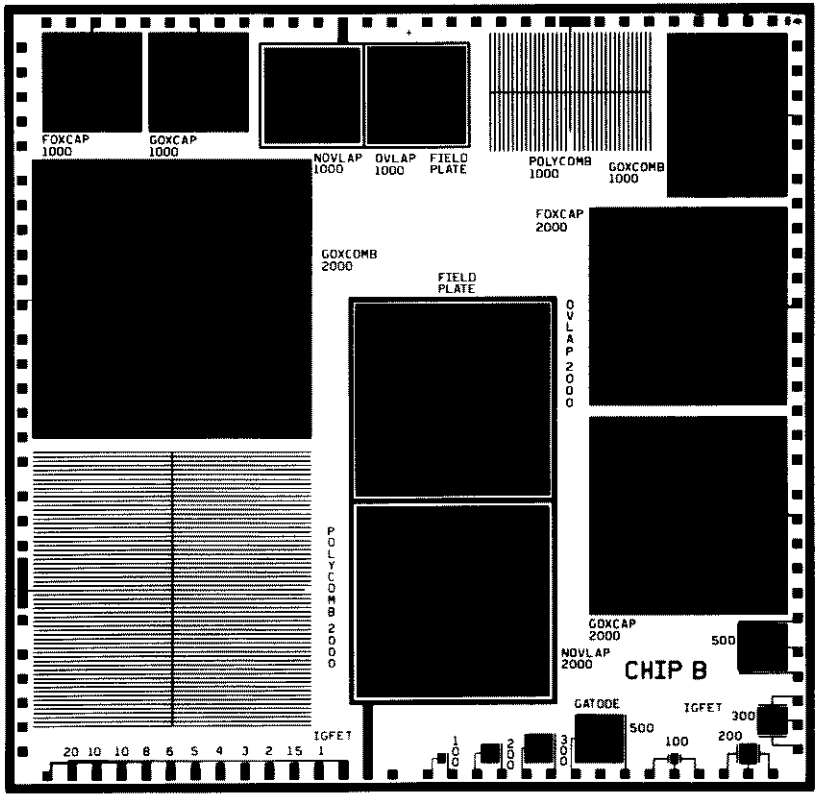



Fig. 15—Composite POLY, WINDOW, and METAL levels for the B chip. Some of the detail has been enlarged 3X to achieve adequate resolution for this illustration.

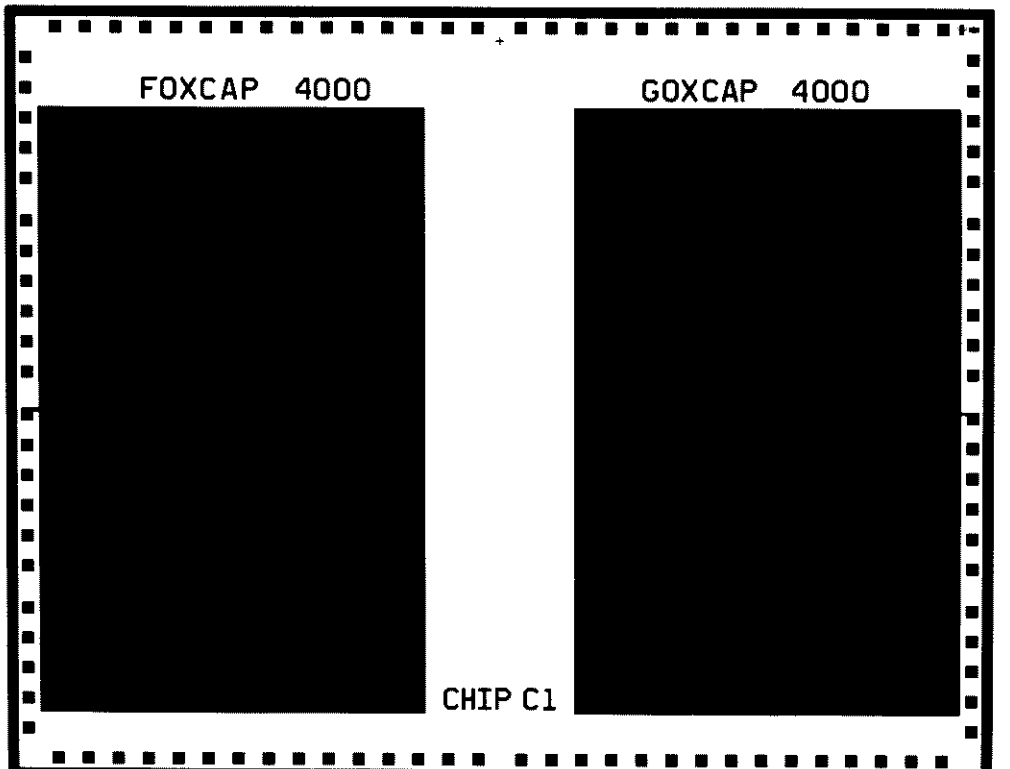



Fig. 16—Composite POLY and WINDOW levels for the C1 chip.

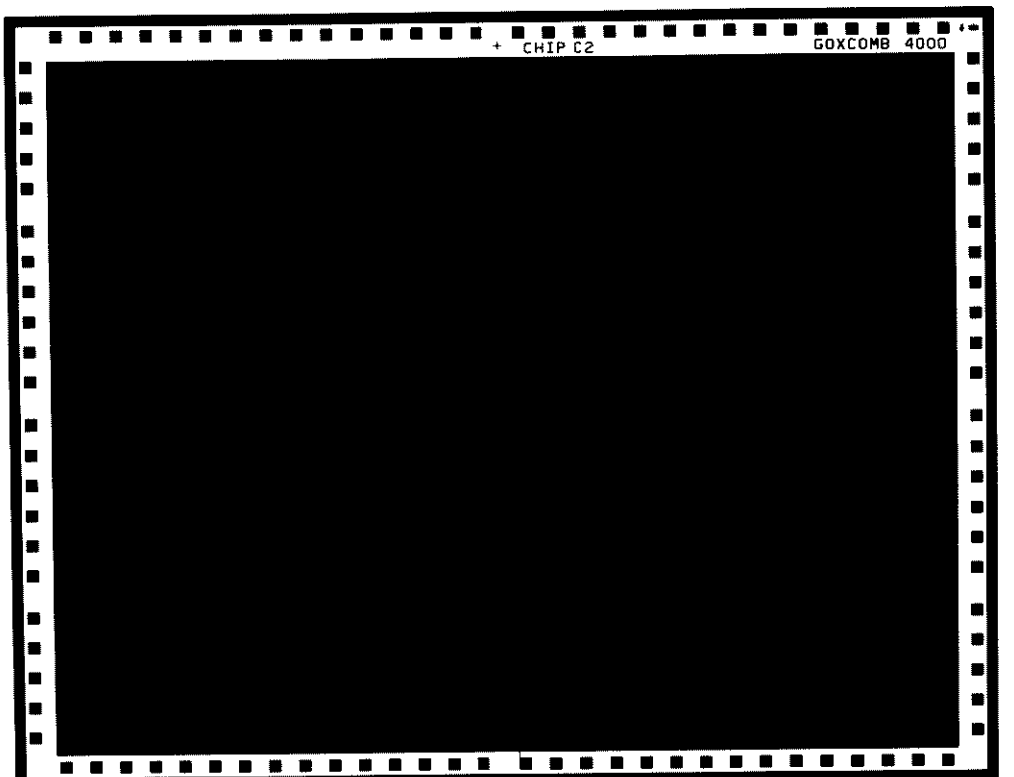



Fig. 17—Composite POLY and WINDOW levels for the C2 chip.

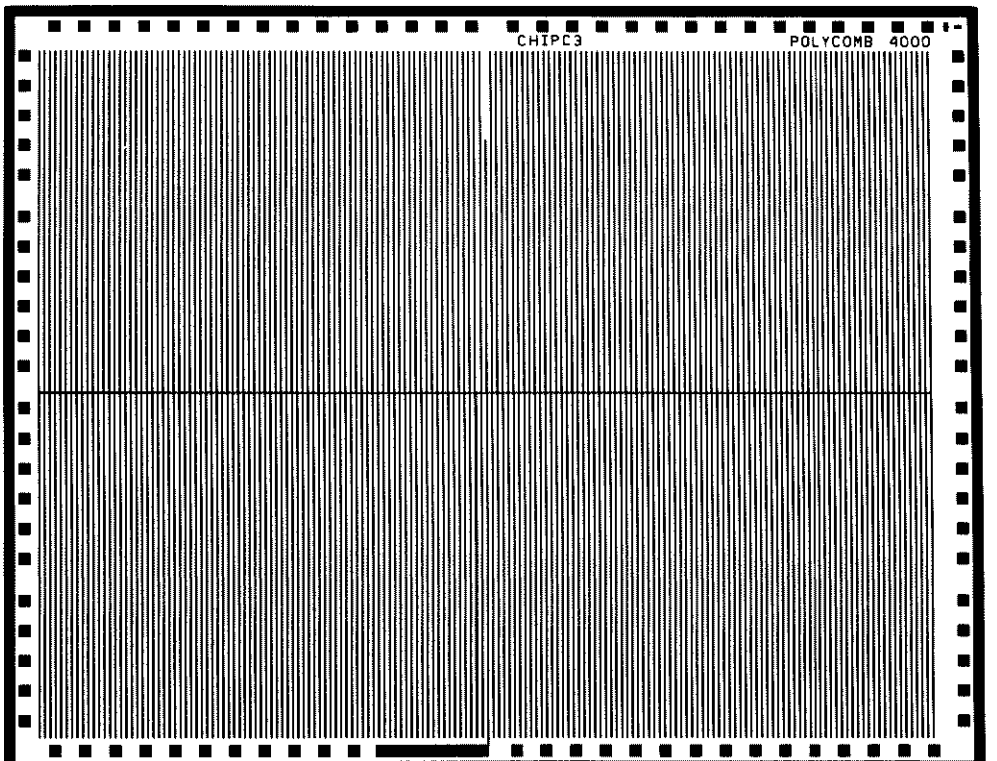



Fig. 18—Composite POLY and WINDOW levels for the C3 chip. Detail has been enlarged 3X to achieve adequate resolution for this illustration.

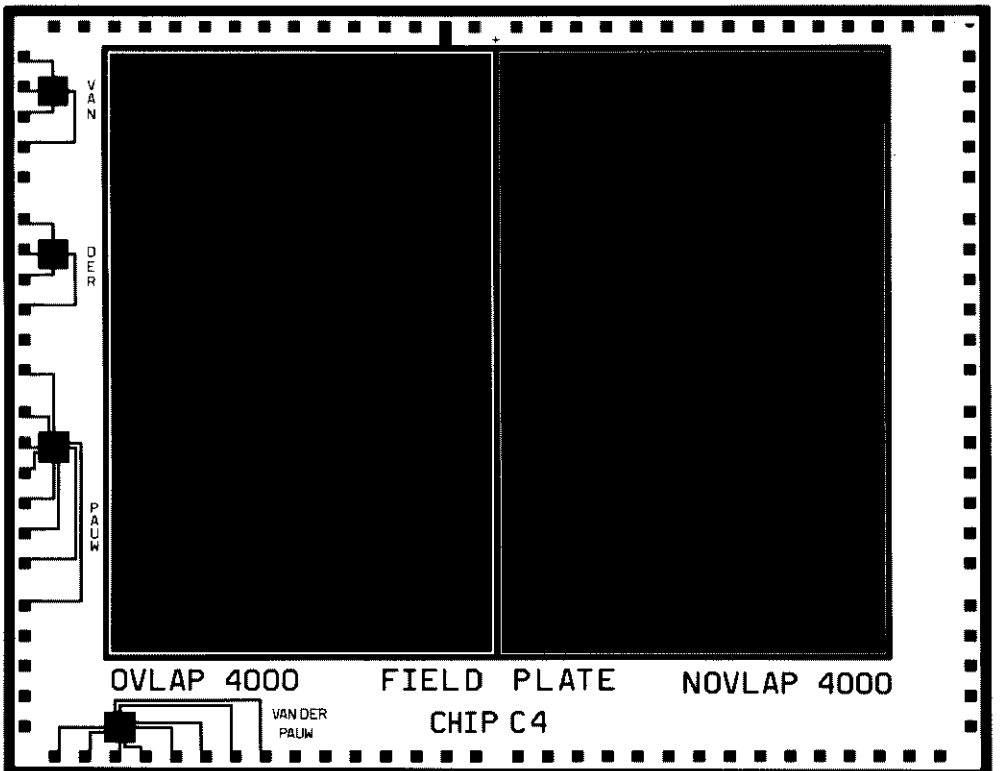



Fig. 19—Composite GUARDRING, GASAD, POLY, and WINDOW levels for the C4 chip. The frames surrounding OVLAP and NOVLAP have been enlarged 3X to achieve adequate resolution for this illustration.

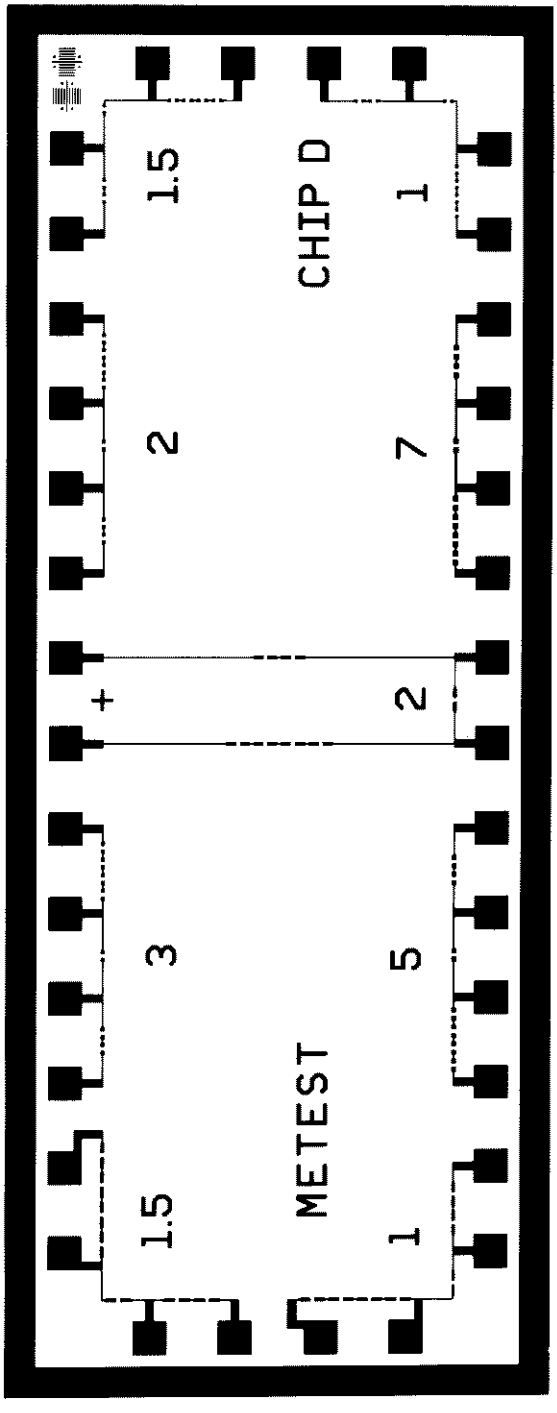



Fig. 20—Composite POLY and WINDOW levels for the D chip.

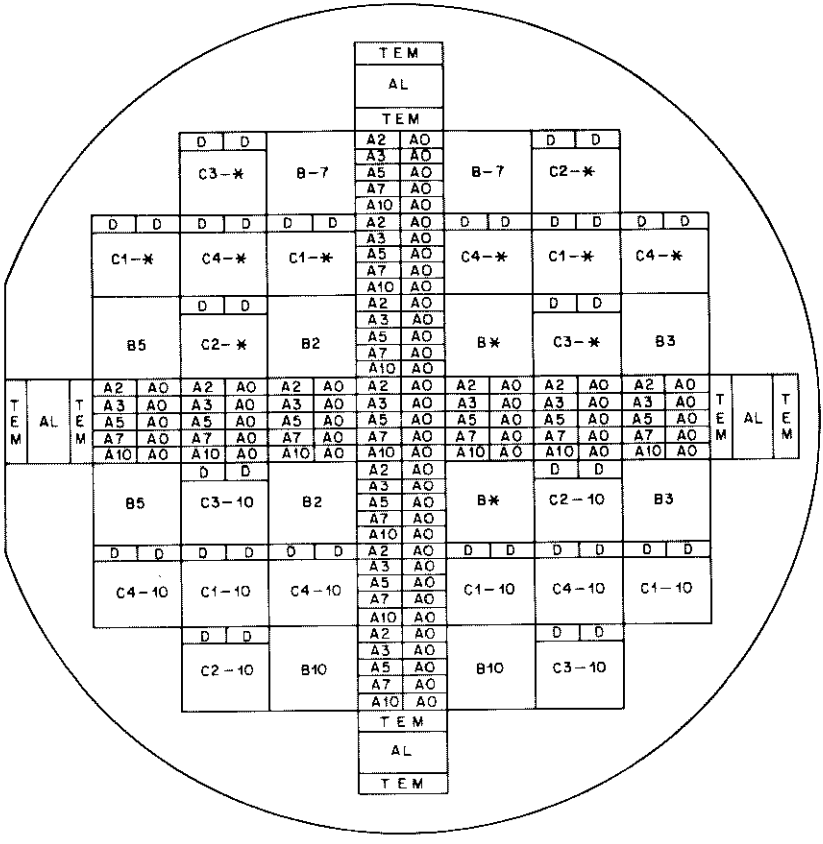



Fig. 21—Chip layout on the fine-line process development wafer.

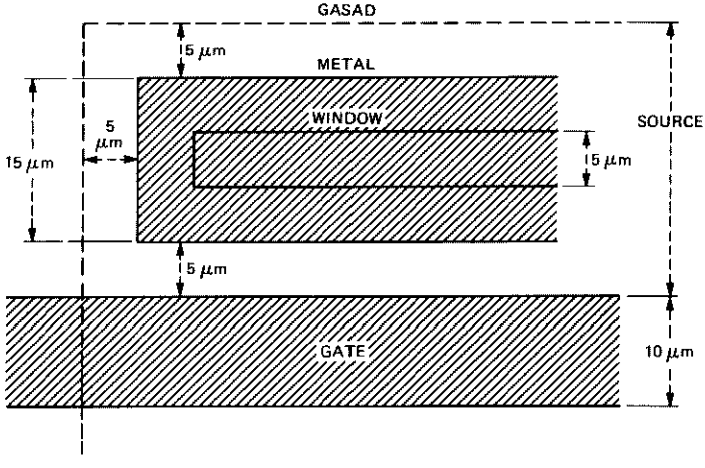



Fig. 22—IGFET design rules for fine-line process development system.

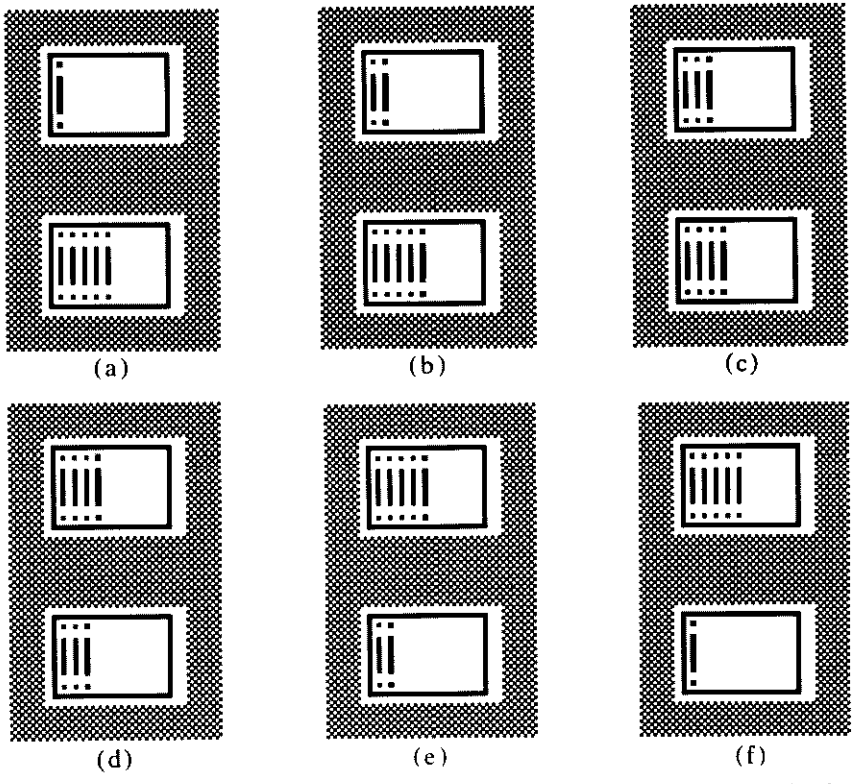



Fig. 23—MOPEP alignment features. (a) GUARDRING level. (b) GASAD level. (c) POLYCON level. (d) POLY level. (e) WINDOW level. (f) METAL level.

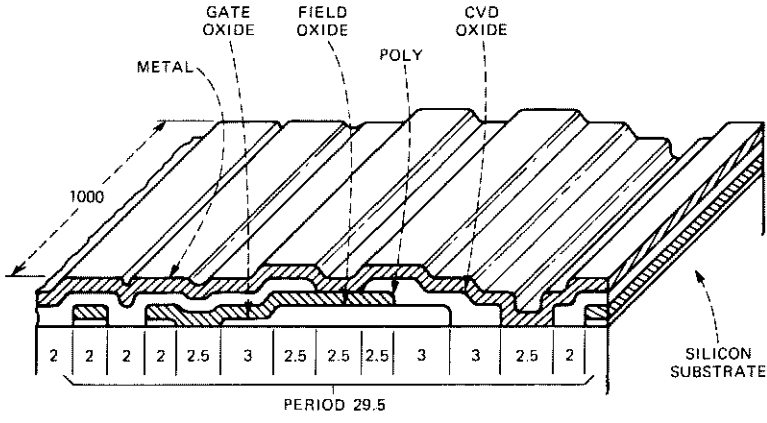



Fig. 24—TEM test chip.

## GLOSSARY

| | |
|---|---|
| CHANSTOP | channel stopping implantation or diffusion to avoid inversion of the silicon surface at the Si-$SiO_2$ interface |
| C-V | capacitance measured as a function of voltage |
| FOXCAP | field oxide capacitor |
| GASAD | gate and source and drain feature delineated in the field oxide prior to gate oxidation. Also, the second photolithographic level in the set of fine-line process development masks. |
| GATODE | gated diode, essentially an IGFET (see below) with common source and drain. |
| GOXCAP | gate oxide capacitor |
| GOXCOMB | a gate oxide feature with a comb-shaped structure |
| GUARDRING | electrically guarded structure, fabricated by ion implantation or diffusion, which straddles and surrounds the boundary of a metallization feature, forming a closed ring. |
| HEXCAP | six-fold or hexadic capacitor group |
| IGFET | insulated gate field-effect transistor |
| LSI | large-scale integration |
| METAL | metallization pattern, the final photolithographic level in the set of fine-line process development masks. |
| METEST | metallization test structure consisting of tapped strings with contacts to underlying diffused tubs. |
| MOPEP | modified Perkin Elmer projection alignment features |
| MOS | metal-oxide-semiconductor sandwich structure used |

| | |
|---|---|
| | for electrical characterization of device fabrication processes |
| NOVLAP | conductive pad not overlapping field oxide and forming the top level of a metal-oxide-semiconductor capacitor |
| OVLAP | conductive pad overlapping field oxide and forming the top level of a metal-oxide-semiconductor capacitor |
| PEP | Perkin Elmer projection alignment features |
| POLY | polycrystalline silicon which, when patterned, forms a conductive electrode for electrical tests. Also, an intermediate photolithographic mask level in the set of fine-line process development masks. |
| POLYCOMB | a polycrystalline silicon feature with a comb-shaped structure |
| POLYCON | polycrystalline contact to underlying silicon substrate. Also, an intermediate photolithographic mask level in the set of fine-line process development masks. |
| POLYSHEET | polycrystalline silicon feature for sheet resistance and linewidth loss measurements |
| SADSHEET | structure formed during gate and source and drain (GASAD) lithography to determine source and drain sheet resistance and linewidth loss |
| SEM | scanning electron microscope |
| TEM | transmission electron microscope |
| VANDER-PAUW | a symmetric structure introduced by L. J. van der Pauw[10] to determine the electrical resistivity of thin conductive layers |
| WINDOW | next to the last photolithographic mask level in the set of fine-line process development masks to form source and drain contacts to insulated gate field-effect transistors, junction contacts in gated diodes, and access to polycrystalline silicon features when the poly Si is covered by an intermediate dielectric. |

## REFERENCES


1. H. C. Poon and R. L. Johnston, unpublished work.
2. J. M. Andrews, J. Electron. Mater., *8* (1979), pp. 227–47.
3. R. A. Kushner, private communication.
4. D. M. Boulin, unpublished.
5. A. M. Mohsen and F. J. Morris, Solid-St. Electron., *18* (1975), pp. 407–16.
6. J. R. Brews and D. Kahng, unpublished work; U.S. Patent No. 4,135,289, applied for August 23, 1977, issued January 23, 1979.
7. B. V. Gokhale and J. Zeidenbergs, Technical Digest of the 1974 International Electron Devices Meeting. Abs. No. 23.1, p. 529.
8. G. W. Taylor, Solid-St. Electron., *19* (1976), pp. 495–503.
9. John Andrews, E. F. Labuda, and Paul Norton, The Electrochemical Society Extended Abstracts, Fall Meeting, Pittsburgh, PA, Oct. 15–20, 1978, Abstract 203, p. 543.

10. L. J. van der Pauw, Philips Res. Repts., *13* (1958), pp. 1–9.
11. B. R. Chawla, A. P. Gnadinger, and W. Rosenzweig, unpublished work.
12. R. F. Pierret, Solid-St. Electron., *17* (1974), pp. 1257–69.
13. P. L. Castro and B. E. Deal, J. Electrochem. Soc., *118* (1971), pp. 280–86.
14. D. J. Fitzgerald and A. S. Grove, Proc. IEEE, *54* (1966), pp. 1601–2.
15. E. H. Snow, A. S. Grove, and D. J. Fitzgerald, Proc. IEEE, *55* (1967), pp. 1168–85.
16. A. S. Grove and D. J. Fitzgerald, Solid-St. Electron., *9* (1966), pp. 783–806.
17. D. J. Fitzgerald and A. S. Grove, Surface Sci., *9* (1968), pp. 347–69.
18. G. W. Taylor, private communication.
19. J. R. Brews, private communication.
20. S. Vaidya, D. B. Fraser, and W. S. Lindenberger, J. Appl. Phys., *51* (1980), pp. 4475–82.
21. T. T. Sheng and R. B. Marcus, J. Electrochem. Soc., *127* (1980), pp. 737–43.
22. S. Vaidya, A. K. Sinha, and J. M. Andrews, J. Electrochem. Soc., *130* (1983), pp. 496–501.
23. J. M. Andrews, P. A. Heimann, and R. A. Kushner, Proc. First Int. Symp. on Very Large Scale Integration Science and Technology, Detroit, MI, October 17–21, 1982, Abstract 147, pp. 68–75.
24. C. J. Koeneke, S. M. Sze, R. M. Levine, and E. Kinsbron, IEDM Tech. Digest (1981), pp. 367–70.
25. P. A. Heimann, P. S. D. Lin, and T. T. Sheng, Electrochem. Soc. Extended Abstr., Fall Meeting, Detroit, MI, October 17–21, 1982, Abstract 195, pp. 309–10.
26. P. A. Heimann and J. E. Olsen, J. Appl. Phys. *53* (1982), pp. 546–9.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 4, April 1983
*Printed in U.S.A.*

# A Circuit That Changes the Word Rate of Pulse Code Modulated Signals

By J. C. CANDY and O. J. BENJAMIN

(Manuscript received November 2, 1982)

*In this paper we describe a circuit that accepts pulse code modulated signals sampled at about 8 kHz and resamples them at any desired rate up to 512 kHz. When the sampling satisfies Nyquist's criterion, the distortion introduced is at least 35 dB below the signal level. The circuit uses a digital low-pass filter to interpolate sample values, and it may be integrated as about 2500 gates on a 5 mm$^2$ chip.*

## I. INTRODUCTION

It often is impractical to synchronize all of the clocks of an extensive digital network. Consequently, data will arrive at connections out of synchronism and special circuits are needed to bring them into time with the local clock. For irregular bursts of data, synchronism can be easily obtained using buffer memories, but for continuous streams of data, such methods are useful only for very small discrepancies in clock frequencies. A case of particular interest in telephone networks is the transmission and processing of pulse code modulation (PCM). Changing the word rate of such data can introduce objectionable noise into the signal. We describe a circuit that uses digital filters to contain this noise.

## II. RESAMPLING

We know that when an analog signal, $x(t)$, having spectral density $X(\omega)$ and bandwidth $\omega_0$ is sampled at regular intervals, $\tau$, the spectral density of the sampled signal can be expressed as the sum of images

$$X'(\omega) = \sum_{-\infty}^{\infty} X\left(\omega + \frac{2\pi n}{\tau}\right). \tag{1}$$

When $\omega_0\tau < \pi$, the original signal can be recovered by filtering out the images for $n \neq 0$.

If we were to change the sampling rate[1] by first holding each sample value constant throughout its period and resampling at the new period, $\tau_1 < \pi/\omega_0$, the spectral density of the resampled signal can be expressed as

$$X''(\omega) = \sum_n \sum_k X\left(\omega + \frac{2\pi n}{\tau} + \frac{2\pi k}{\tau_1}\right) H\left(\omega + \frac{2\pi n}{\tau}\right), \tag{2}$$

where

$$H(\omega) = \frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}} = \text{sinc}\,(f\tau). \tag{3}$$

When we try to recover the original signal from $X''(\omega)$ by filtering out the components for $n \neq k \neq 0$, it is contaminated by cross modulation products for which

$$0 < \left(\frac{n}{\tau} + \frac{k}{\tau_1}\right) < f_0 = \frac{\omega_0}{2\pi}. \tag{4}$$

Notice that when $\tau$ and $\tau_1$ are integer multiples of one another, this condition never holds and reflected noise is absent from the baseband.

An obvious means of eliminating in-band cross-modulation products from eq. (2) is to smooth out the high-frequency components of the sampled signal $x'(t)$ before resampling it. This has been accomplished by replacing the sample and hold represented by $H(\omega)$ in eq. (3) by a better low-pass filter that interpolates new sample values from the old ones.[1,2] Another implementation of this method is demodulation with a digital/analog (D/A) converter, analog smoothing, and remodulation with an analog/digital (A/D) converter. The method[1] presented here raises the sample rate to a high multiple of the original rate, smooths the sample values with digital filters, holds the smoothed samples, and resamples them at the desired rate.

## III. A CIRCUIT FOR RESAMPLING 8-kHz PCM

We will describe a circuit for resampling a 3.5-kHz signal that has been pulse code modulated at a nominal 8-kHz rate using 16 bits per word. The new sampling can be at any rate up to 512 kHz and even higher rates can be accommodated by minor modification of the circuits. The technique first raises the sampling rate 16 times to 128 kHz using digital interpolating filters to smooth out all unnecessary images of the signal, leaving only those that are adjacent to the new

rate and its harmonics. The high-frequency code is placed in a holding register from which the output is gated at the desired rate. We will see that the filter action of this hold, sinc $(f/128)$, provides more than 30-dB attenuation of the cross-modulation products that fold into baseband.

Because the complexity of digital filters increases sharply with increasing sampling rate, it pays to raise the rate in stages.[1] Figure 1 illustrates the process. Our first stage raises the sampling rate four times from 8 to 32 kHz, employing a low-pass filter that cuts off sharply to attenuate spectral images between 4 and 28 kHz. The second stage raises the rate to 128 kHz by simple linear interpolation, and the third stage is a holding register. Most of the circuits used originally were

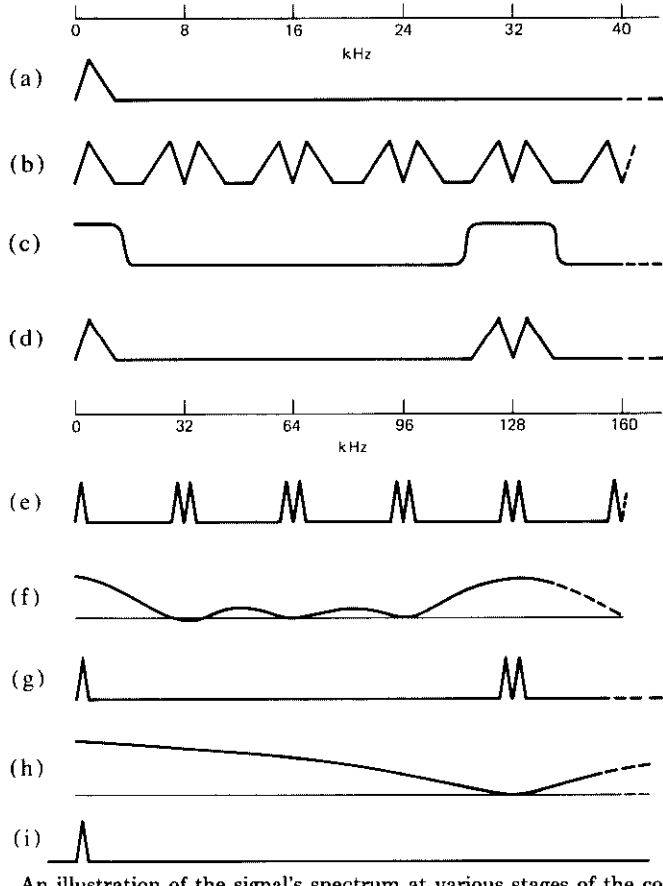



Fig. 1—An illustration of the signal's spectrum at various stages of the conversion: (a) The original signal. (b) The signal sampled at 8 kHz. (c) The response of the low-pass filter. (d) and (e) The output from the low-pass filter. (f) The response of the linear interpolator. (g) The output of the interpolator. (h) The response of the holding circuit. (i) The held signal.

designed to be part of an oversampled coder/decoder (codec).[3] We shall summarize their relevant properties in the following sections.

## IV. THE LOW-PASS FILTER

The low-pass filter shown in Fig. 2 processes data at 32 kilowords per second. Its $z$-transform response is given by

$$F(z) = \frac{1}{32}\frac{1-z^{-4}}{1-z^{-1}} \cdot \frac{1-\frac{5}{4}z^{-1}+z^{-2}}{1-\frac{19}{16}z^{-1}+\frac{31}{64}z^{-2}} \cdot \frac{1-\frac{3}{4}z^{-1}+z^{-2}}{1-\frac{23}{16}z^{-1}+\frac{55}{64}z^{-2}}. \quad (5)$$

Each word of its 8-kHz input signal is repeated four times and fed into two second-order sections. This filter attenuates the images of the signal in the range 4 to 28 kHz by more than 34 dB. Its output is a good approximation of pulse code modulation at 32 kHz. The spectral response of the entire resampler is shown in Fig. 3. The zeros at 4.5 and 6 kHz are introduced by the second-order sections, and those at 8 and 16 kHz by the repetition of input words.

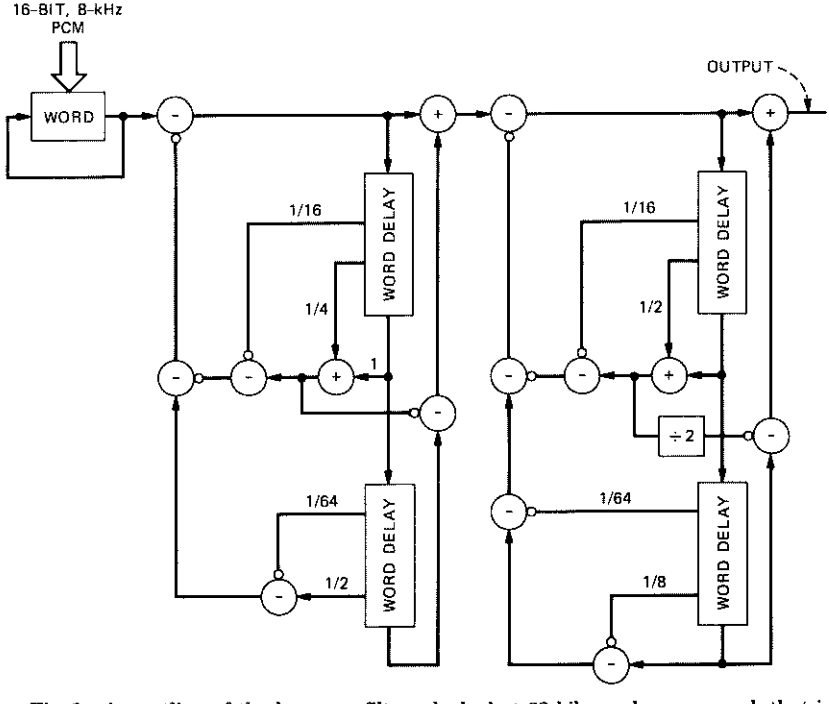



Fig. 2—An outline of the low-pass filter, clocked at 32 kilowords per second, that is used to raise the sampling rate from 8 to 32 kHz.

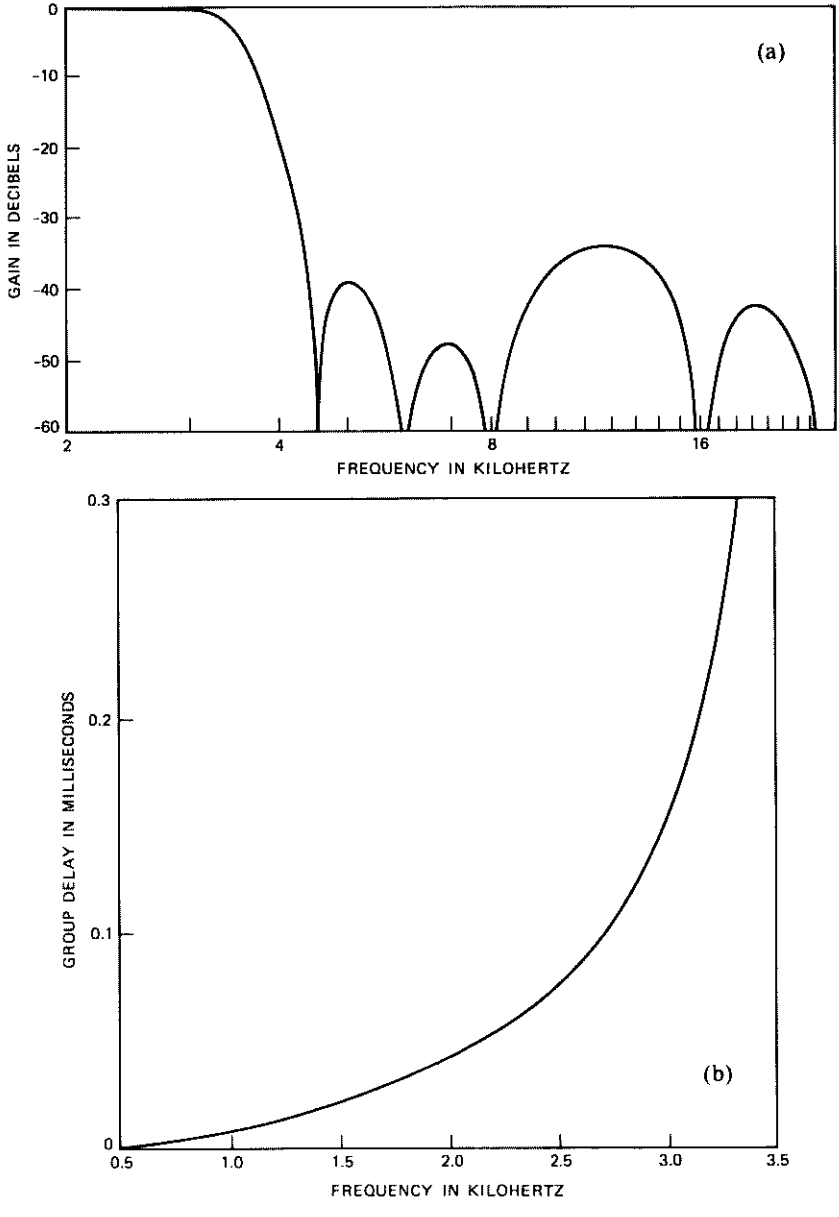



Fig. 3—Spectral response of the resampler. (a) The calculated gain of the cascaded filters used in the resampler. (b) The group delay of the filters.

## V. LINEAR INTERPOLATION

Simple linear interpolation of three sample values increases the sampling rate from 32 to 128 kHz. This process has frequency responses that can be expressed as

$$I(\omega) = \left(\frac{\text{sinc } (f/32)}{\text{sinc } (f/128)}\right)^2. \tag{6}$$

Its attenuation of spectral images in the range 28 to 36, 60 to 68, and 92 to 100 kHz exceeds 40 dB. The small amount of droop introduced into baseband is compensated in the low-pass filter so that the entire circuit has inband gain in the range −0.41 to −0.57 dB. The circuit implementation shown as Fig. 4 is based on the results

$$y(n\tau) = y[(n-1)\tau] + \tfrac{1}{4}\,\Delta x(n\tau)$$

$$\Delta x(n\tau) = x(4n\tau) - x[4(n-1)\tau]$$

and

$$y(4n\tau) = x(4n\tau). \tag{7}$$

After each new input sample enters register $R_1$, the output, held in register $R_2$, increments four times to make its value equal to the input.

## VI. RESAMPLING

Figure 5 shows the circuit that is used to resample the signal at the desired output rate without causing conflict with the internal clock. Here the output from the linear interpolator is loaded into register $R_3$ in time with the internal clock running at 1 MHz. This loading is inhibited by the presence of the output clock, which after a short delay loads $R_4$ from $R_3$. The frequency response associated with the holding action of this circuit may be expressed as

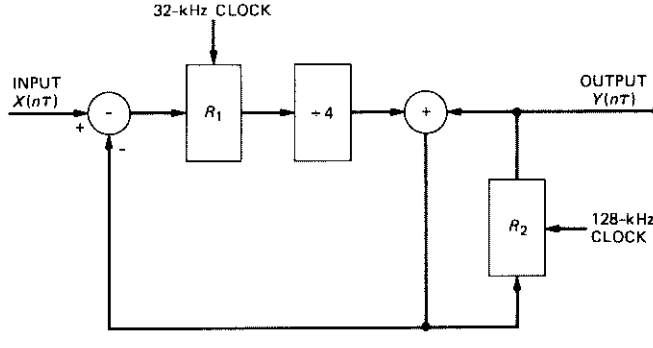



Fig. 4—An outline of the linear interpolator used to raise the sampling rate from 32 to 128 kHz.

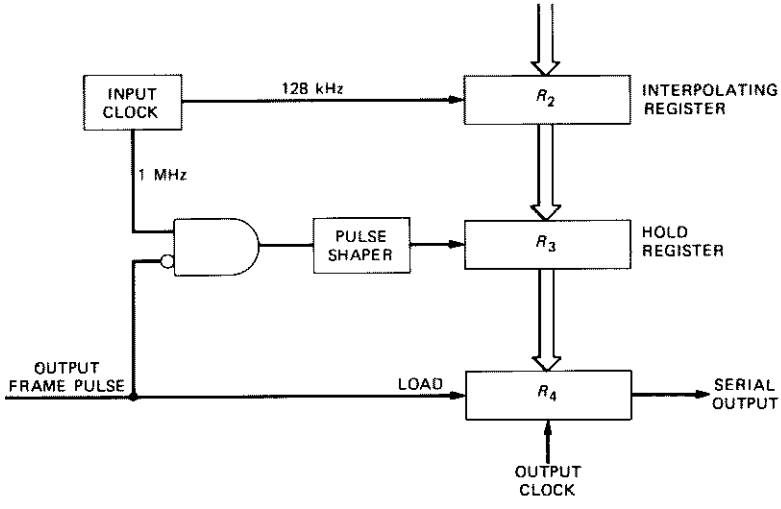



Fig. 5—The circuit used to sample the output. Register $R_2$ is part of the linear interpolator, $R_3$ is loaded from $R_2$ at 1 MHz. An output demand inhibits loading of $R_3$, and then loads $R_4$ from $R_3$.

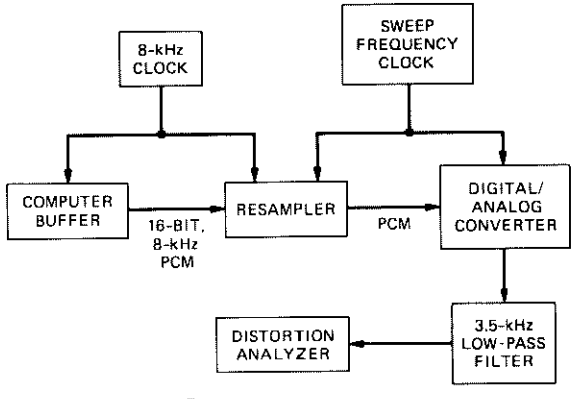



Fig. 6—The test circuit.

$$R(\omega) = \text{sinc}\,(f/128).$$

Images of the signal in the range 124 to 132 kHz are thereby attenuated by more than 30 dB.

## VII. TESTING THE CIRCUIT

A version of the circuit was built of standard digital circuit components and tested in the setup shown in Fig. 6. The net gain of the circuit and its signal-to-noise ratio were measured as the sampling rate was varied continuously from 0 to 256 kHz. For a 1.02-kHz input signal

sampled at 8 kHz, the gain remained constant within ±.05 dB and the signal-to-noise ratio always exceeded 40 dB. Input amplitudes were varied in the range 0 to −60 dB. At lower amplitudes quantization noise inherent in the 16-bit input word was significant.

## VIII. CONCLUDING REMARKS

Experience obtained while designing filters for a version of a codec enables us to estimate that the circuits described here can be implemented on about 5 $mm^2$ of silicon in a standard technology. The performance of the circuit is good enough that imperfections introduced by resampling would be insignificant compared with those normally obtained from $\mu$-255 encoding of the signal.

## IX. ACKNOWLEDGMENT

We thank H. S. McDonald for suggesting this use for our circuits, and J. L. Henry for providing information on the integrated circuit implementation.

## REFERENCES

1. D. J. Goodman and M. J. Carey, "Nine Digital Filters for Decimation and Interpolation," IEEE Trans. Acoust., Speech, Signal Processing, *ASSP-25,* 2(April 1977), pp. 121–6.
2. R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals—A Tutorial Review," 53 references, Proc. IEEE, *69,* pp. 300–31.
3. J. C. Candy, B. A. Wooley, and O. J. Benjamin, "A Voiceband Codec with Digital Filtering," IEEE Trans. Commun., *COM-29* (June 1981), pp. 815–30.

## CONTRIBUTORS TO THIS ISSUE

**John Andrews,** B.S., 1958, Houghton College; Ph.D. (Solid State Physics), 1963, Massachusetts Institute of Technology; Research Associate, Massachusetts Institute of Technology, 1963–1966; Bell Laboratories, 1966—. The subject of Mr. Andrews' doctoral thesis was ultrasonic attenuation in superconductors at radio and microwave frequencies. After coming to Bell Laboratories, he worked on laser mixing in nonlinear optical crystals. Next, he investigated ohmic and rectifying metal-semiconductor contacts formed by metallurgical reaction between transition metals and silicon. From 1970 to 1976 he worked with Western Electric engineers at Allentown on silicon integrated circuit manufacture. Since 1976 he has been associated with the Advanced LSI Development Laboratory at Murray Hill, investigating new materials and special processes for fine-line VLSI technology. Senior Member, IEEE; member APS, ECS, and TMS-AIME; associate editor, *Materials Letters.*

**O'Connell J. Benjamin,** B.S.E.E., 1976, Polytechnic Institute of Brooklyn; graduate studies, 1977–1982, Polytechnic Institute of New York; Bell Laboratories, 1973—. Mr. Benjamin has been involved with research in digital MF receivers, voiceband codecs, and, presently, he is working on computer communications.

**G. Samuel Brockway,** B.S.C.E., 1966, University of Miami; M.S.E.M., 1968, Georgia Institute of Technology; Ph.D. (Applied Mechanics), 1972, California Institute of Technology; Bell Laboratories, 1976–1981. Mr. Brockway has worked extensively in such areas of mechanics as elasticity, viscoelasticity, and fracture mechanics, and in the development of plastics for use in fiber optics. He is presently president of Plastics Engineering Consultants.

**James C. Candy,** B.Sc. and Ph.D. (Engineering), University of Wales, Bangor, in 1951 and 1954, respectively; Bell Laboratories, 1961—. From 1954 to 1956 Mr. Candy was with S. Smith and Sons, Guided Weapons Department, Cheltenham, and for the next four years he worked on nuclear instrumentation at the Atomic Energy Research Establishment, Harwell. In 1959 Mr. Candy went to the United States to take up an appointment as a Research Associate at the University of Minnesota, and a year later he joined Bell Laboratories, Holmdel, NJ, where he has investigated digital circuits and the efficient transmission of video signals. He is now concerned with techniques for converting signals between analog and digital formats.

**Stephen E. Levinson,** B.A. (Engineering Sciences), 1966, Harvard; M.S. and Ph.D. (Electrical Engineering), University of Rhode Island, Kingston, Rhode Island, 1972 and 1974, respectively; General Dynamics, 1966–1969; Yale University, 1974–1976; Bell Laboratories, 1976—. From 1966 to 1969, Mr. Levinson was a design engineer at Electric Boat Division of General Dynamics in Groton, Connecticut. From 1974 to 1976, he held a J. Willard Gibbs Instructorship in Computer Science at Yale University. In 1976, he joined the technical staff at Bell Laboratories, where he is pursuing research in the areas of speech recognition and cybernetics. Member, Association for Computing Machinery, Acoustical Society of America, editorial board of Speech Technology; associate editor, IEEE Transactions on Acoustics, Speech and Signal Processing.

**Lawrence R. Rabiner,** S.B. and S.M., 1964, Ph.D. (Electrical Engineering), The Massachusetts Institute of Technology: Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975) and *Digital Processing of Speech Signals* (Prentice-Hall, 1978). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication of the Acoustical Society, ASSP Technical Committee on Speech Communication; Member, IEEE Proceedings Editorial Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

**Adel A. M. Saleh,** B.Sc. (Electrical Engineering), 1963, University of Alexandria, Alexandria, Egypt; M.S. and Ph.D. (Electrical Engineering) Massachusetts Institute of Technology, Cambridge, MA, 1967 and 1970, respectively; Bell Laboratories, 1970—. From 1963 to 1965 Mr. Saleh worked as an instructor at the University of Alexandria. At Bell Laboratories he is engaged in research on microwave circuits, components, and communication systems. Senior Member, IEEE; member, Sigma Xi.

**Jack Salz,** B.S.E.E., 1955, M.S.E., 1956, and Ph.D., 1961, University of Florida; Bell Laboratories, 1961—. Mr. Salz first worked on remote line concentrators for the electronic switching system. Since 1968 he has supervised a group engaged in theoretical studies in data commu-

nications and is currently a member of the Communications Methods Research Department. During the academic year 1967–68, he was on leave as Professor of Electrical Engineering at the University of Florida. In Spring 1981, he was a visiting lecturer at Stanford University. Member, Sigma Xi.

**Manuel R. Santana,** B.Sc. (Electrical Engineering), 1970, University of Hartford; M.S., (Electrical Engineering), 1971, Georgia Institute of Technology; Bell Laboratories, 1970—. Mr. Santana has worked on the development of multipair cable and in the design, analysis, and testing of lightguide cable. He is presently Supervisor of the Lightguide Cable Group. Member, IEEE, Kappa Mu.

**Man Mohan Sondhi,** B.Sc. (Physics), Honours degree, 1950, Delhi University, Delhi, India, D.I.I.Sc. (Communications Engineering), 1953, Indian Institute of Science, Bangalore, India; M.S., 1955; Ph.D. (Electrical Engineering), 1957, University of Wisconsin, Madison, Wisconsin; Bell Laboratories, 1962—. Before joining Bell Laboratories, Mr. Sondhi worked for a year at the Central Electronics Engineering Research Institute, Pilani, India and taught for a year at the University of Toronto. At Bell Laboratories his research has included work on speech signal processing, echo cancellation, adaptive filtering, modelling of auditory and visual processes, and acoustical inverse problems. From 1971 to 1972 Mr. Sondhi was a guest scientist at the Royal Institute of Technology, Stockholm, Sweden.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Bank R. E., Rose D. J., **Analysis of a Multilevel Iterative Method for Non-Linear Finite-Element Equations.** Math Comput 39(160): 453–465, 1982.

Benevit C. A., Cassard J. M., Dimmler K. J., Dumbri A. C., Mound M. G., Procyk F. J., Rosenzweig W., Yanof A. W., **A 256K Dynamic Random-Access Memory.** IEEE J Soli 17(5):857–862, 1982.

Cleveland W. S., **Correlations Perceived and Measured.** Science 218(4575):919, 1982.

Conway J. H., Sloane N. J. A., **Laminated Lattices.** Ann Math 116(3):593–620, 1982.

Conway J. H., Sloane N. J. A., **Leech Roots and Vinberg Groups.** P Roy Soc A 384(1787):233–258, 1982.

Daniels L. A. et al., **A High-Density Magnetic-Tape Cartridge Storage-System.** IEEE Magnet 18(6):1227–1229, 1982.

Fishburn P. C., **Moment-Preserving Shifts and Stochastic-Dominance.** Math Oper R 7(4):629–634, 1982.

Flajolet P., Odlyzko A., **The Average Height of Binary-Trees and Other Simple Trees.** J Comput Sy 25(2):171–213, 1982.

Flatto L., **Limit-Theorems for Some Random-Variables Associated with Urn Models.** Ann Probab 10(4):927–934, 1982.

Green M. L., Gyorgy E. M., **Computer Design of Induction Coil Cores.** IEEE Magnet 18(6):1556–1558, 1982.

Hawkins D. T., **Inspec on Dialog.** Database 5(4):12–25, 1982.

Jarvis J. F., **Research Direction in Industrial Machine Vision—A Workshop Summary.** Computer 15(12):55–61, 1982.

Lax M., Agrawal G. P., **Evaluation of Fourier Integrals Using B-Splines.** Math Comput 39(160):535–548, 1982.

Stroustrup, B., **An Experiment With the Interchangeability of Processes and Monitors.** Software 12(11):1011–1025, 1982.

Yannakakis M., Papadimi C. H., **Algebraic Dependencies.** J Comput Sy 25(1):2–41, 1982.

## ENGINEERING

Amitay N., Gans M. J., **Narrow Multibeam Satellite Ground Station Antenna Employing a Linear-Array With a Geosynchronous Arc Coverage of 60-Degrees. 1. Theory.** IEEE Antenn 30(6):1063–1067, 1982.

Bjorkholm J. E. et al., **Optical Bistability Based on Self-Focusing—An Approximate Analysis.** IEEE J Q El 18(12):2016–2022, 1982.

Capasso F., **The Channeling Avalanche Photo Diode—A Novel Ultra-Low-Noise Interdigitated p-n-Junction Detector.** IEEE Device 29(9):1388–1395, 1982.

Cohen L. G., Mammel W. L., Jang S. J., **Low-Loss Quadruple-Clad Single-Mode Lightguides With Dispersion Below 2 ps/km nm Over the 1.28 μm–1.65 μm Wavelength Range.** Electr Lett 18(24):1023–1024, 1982.

Forrest S. R., Smith R. G., Kim O. K., **Performance of $In_{0.53}Ga_{0.47}As/InP$ Avalanche Photodiodes.** IEEE J Q El 18(12):2040–2048, 1982.

Jin S., Tiefel T. H., Bennett J. E., Archer W. E., **New Pe-Cr-Mo Glass-Sealable Semihard Magnet Alloys for Remanent Reed Contacts.** IEEE Magnet 18(6):1454–1456, 1982.

Joyce W. B., **Role of the Conductivity of the Confining Layer in DH-Laser Spatial Hole Burning Effects.** IEEE J Q El 18(12):2005–2009, 1982.

Kantor W. M., **Independent Pairs of Good Self-Dual Codes (Letter).** IEEE Info T 28(6):966–967, 1982.

Kaspi A. H., Berger T., **Rate-Distortion for Correlated Sources With Partially Separated Encoders.** IEEE Info T 28(6):828–840, 1982.

Kiehl R. A., Flahive P. G., Wemple S. H., Cox H. M., **Direct-Coupled GaAs Ring Oscillators With Self-Aligned Gates.** Elec Dev L 3(11):325–326, 1982.

Levey L., **Calculation of Ground Fault Currents Using an Equivalent-Circuit and a Simplified Ladder Network.** IEEE Power 101(8):2491–2497, 1982.

Nelson T. J. et al., **Effects of N-$H_2$ Doses in Ion-Implanted Bubble Memory Test Chips.** IEEE Magnet 18(6):1358–1360, 1982.

Rainal A. J., **Optimal Detection of FM Clicks (Letter).** IEEE Info T 28(6):971–973, 1982.

Rana V. V. S., Lecraw R. C., **High Curie-Temperature Vanadium Containing Films for 8-$\mu$m Period Magnetic-Bubble Devices.** IEEE Magnet 18(6):1292–1294, 1982.

Stayt J. W., **A Fail-Safe Automatic Vent Control for Pfeiffer Turbomolecular Pumps.** J Vac Sci T 21(4):1028–1030, 1982.

Trumbore F. A., **Citation Classic-Solid Solubilities of Impurity Elements in Germanium and Silicon.** CC/Eng Tech 1983(1):16, 1983.

## MANAGEMENT/ECONOMICS

Allen S. J., Wilson B. A., Tsui D. C., **Cyclotron-Resonance, the Memory Function, and the Random Potential at the Si-$SiO_2$ Interface.** Phys Rev B 26(10):5590–5595, 1982.

Buchsbaum S. J., **How Much R-and-D is Enough (Editorial).** Laser Focus 18(11):6, 1982.

## PHYSICAL SCIENCES

Bally J., **Energetic Molecular Flows in Star-Forming Cloud Cores.** Ann NY Acad 395(Oct):191–196, 1982.

Banavar J. R. et al., **Dynamic Monte-Carlo Scaling Method for Polymers.** Chem P Lett 93(1):35–37, 1982.

Bishop D. J., Sprenger W. O., Pindak R., Neubert M. E., **Mechanical Measurements of the Melting Transition in Thin Liquid-Crystal Films.** Phys Rev L 49(25):1861–1864, 1982.

Capasso F., Logan P. A., Tsang W. T., Hayes J. R., **Channeling Photodiode—A New Versatile Interdigitated P-N-Junction Photodetector.** Appl Phys L 41(10):944–946, 1982.

Carroll P. J. et al., **The Crystal and Molecular-Structure of Bi(1,2,4,5,6-Pentahydro-1,3-Ditellurapentalen-2-Lidene)(Hexamethylenetetratellurafulvalene)—The Shortest Observed Intermolecular Te-Te Contact.** J Chem S Ch 1982(22):1316–1318, 1982.

Citrin P. H., Hamann D. R., Mattheiss L. F., Rowe J. E., **Geometry and Electronic-Structure of Cl on the Cu(001) Surface.** Phys Rev L 49(23):1712–1715, 1982.

Cohen R. L., Sherwood R. C., West K. W., **Particle-Size Distribution of Ni Microprecipitates in $LaNi_5$ Used for Hydrogen Storage.** Appl Phys L 41(10):999–1001, 1982.

Craighead H. G., Mankiewich P. M., **Ultra-Small Metal-Particle Arrays Produced by High-Resolution Electron-Beam Lithography.** J Appl Phys 53(11):7186–7188, 1982.

Dicenzo S. B., **Relaxed Incommensurate Structure of the Intercalant Layer in Higher-Stage Graphite-Intercalation Compounds.** Phys Rev B 26(10):5878–5881, 1982.

DiCenzo S. B., Wertheim G. K., Buchanan D. N., **XPS Studies of Adatom-Adatom Interactions: I/Ag(111) and I/Cu(111).** Surf Sci 121(3):411–420, 1982.

Donnelly V. M., Flamm D. L., Tu C. W., Ibbotson D. E., **Temperature-Dependence of InP and GaAs Etching in a Chlorine Plasma.** J Elchem So 129(11):2533–2537, 1982.

Douglass D. C. et al., **The Use of NMR Linewidths to Study b-Axis Distributions in Poled and Unpoled PVDF.** J Chem Phys 77(11):5826–5835, 1982.

Draper C. W., Kaufmann E. N., Buene L., **Suppression of Metal-Surface Oxidation During Laser Surface Treatments.** Surf Int An 4(1):8–13, 1982.

Dutta, N. K., **Calculated Threshold Current of GaAs Quantum Well Lasers.** J Appl Phys 53(11):7211–7214, 1982.

Egan W. G., Aspnes D. E., **Finite-Wavelength Effects in Composite Media.** Phys Rev B 26(10):5313–5320, 1982.

Englert T., Tsui D. C., Portal J. C., Beerens J., Gossard A., **Magnetophonon Resonances of the Two-Dimensional Electron-Gas in GaAs/AlGaAs Heterostructures.** Sol St Comm 44(8):1301–1304, 1982.

Etienne B., Shah J., Leheny R. F., Nahory R. E., **Influence of Hot Carriers on the Temperature-Dependence of Threshold in 1.3-$\mu$m InGaAsP Lasers.** Appl Phys L 41(11):1018–1020, 1982.

Everaert J., Rentzepis P. M., **Picosecond Fluorescence Kinetics of N-All-*Trans*-Retinylidene-*N*-Butylamine.** Photochem P 36(5): 543–550, 1982.

Frerking M. A., Langer W. D., Wilson R. W., **The Relationship Between Carbon-Monoxide Abundance and Visual Extinction in Inter-Stellar Clouds.** Astrophys J 262(2):590–605, 1982.

Goldfarb R. B., Fickett F. R., Rao K. V., Chen H. S., **Spin-Freezing Below the Ferromagnetic Transition Determined by the Imaginary Component of AC Magnetic-Susceptibility.** J Appl Phys 53(11):7687–7689, 1982.

Gottscho R. A., Burton R. H., Davis G. P., **Radiative Lifetime and Collisional Quenching of Carbon Monochloride (A 2-Delta) in an Alternating Current Glow-Discharge.** J Chem Phys 77(11):5298–5301, 1982.

Greenside H. S., Ahlers G., Hohenberg P. C., Walden R. W., **A Simple Stochastic-Model for the Onset of Turbulence in Rayleich-Benard Convection.** Physica D 5(2–3):322–334, 1982.

Harrison T. R. et al., **Thin-Film $CaF_2$ Inorganic Electron Resist and Optical-Read Storage Medium.** Appl Phys L 41(11):1102–1104, 1982.

Helfand E., Skolnick J., **Mechanism and Rates of Conformational Transitions in Heterogeneous Polymers.** J Chem Phys 77(11):5714–5724, 1982.

Hwang J. C. M. et al., **Application of Molecular-Beam Epitaxy To III-V-Microwave and High-Speed Device Fabrication.** Sol St Tech 25(10):166–169, 1982.

Inoue A., Okamoto S., Masumoto T., Chen H. S., **The Effect of Cold-rolling on the Superconducting Properties of an Amorphous $Nb_{50}Zr_{35}Si_{15}$ Alloy.** Scrip Metal 16(12):1393–1396, 1982.

Jarvis, J. F., Valides F., Tyson J. A., **Analysis of Synthesized Dim Galaxy Images.** P Soc Photo 331:473–477, 1982.

Joyce W. B., **Carrier Transport in Double-Heterostructure Active Layers.** J Appl Phys 53(11):7235–7239, 1982.

Keramidas, V. G., **Metallurgical Interactions at Metallization Compound Semiconductor Interfaces.** Thin Sol Fi 96(4):347–363, 1982.

Lee P. A., **Density of States and Screening Near the Mobility Edge.** Phys Rev B 26(10):5882–5885, 1982.

Manheimer M. A., Bhagat S. M., Chen H. S., **Disappearance of Ferromagnetism in Amorphous $(Fe_{1-x}Mn_x)_{75}P_{16}B_6Al_3$ Ribbons.** J Appl Phys 53(11):7737–7739, 1982.

Marcuse D., **Influence of Curvature on the Losses of Doubly Clad Fibers.** Appl Optics 21(23):4208–4213, 1982.

McBrierty V. J., Douglass D. C., Wang T. T., **Correlation Between the Piezoelectric Constant $d_{31}$ and *b*-Axis Distribution in Polyvinylidene Fluoride.** Appl Phys L 41(11):1051–1053, 1982.

Moncton D. E., Pindak R., Davey S. C., Brown G. S., **Melting of Variable-Thickness Liquid-Crystal Thin-Films—A Synchrotron X-Ray Study.** Phys Rev L 49(25):1865–1868, 1982.

Miller T. A., Bondybey V. E., **The Laser Spectroscopy of Cool, Cold, and Very Cold Molecular-Ions.** Phi T Roy A 307(1500):617–631, 1982.

Murphy D. W. et al., **Lithium Insertion in Anatase—A New Route to the Spinel $LiTi_2O_4$.** Rev Chim Mi 19(4–5):441–449, 1982.

Pirronello V., et al., **Formaldehyde Formation in a $H_2O/CO_2$ Ice Mixture Under Irradiation by Fast Ions.** Astrophys J 262(2):636–640, 1982.

Presby H. M., Marcuse D., **Immersionless Single-Mode Preform Index Profiling.** Appl Optics 21(23):4253–4259, 1982.

Pressman L. D., Forrest S. R., Bonner W. A., Van Uitert L. G., **Non-Invasive Analysis of InP Surfaces using Hg-InP Schottky-Barrier Diodes.** Appl Phys L 41(10):969–971, 1982.

Rao K. V., Gerber J., Halley J. W., Chen H. S., **High-Field Magnetization Studies on an Amosphous Alloy with Para, Ferro and Spin-Glass Regions.** J Appl Phys 53(11):7731–7733, 1982.

Rentzepis P. M., **Advances in Picosecond Spectroscopy.** Science 218(4578):1183–1189, 1982.

Rossetti R., Brus L., **Electron-Hole Recombination Emission as a Probe of Surface-Chemistry in Aqueous CdS Colloids (Letter).** J Phys Chem 86(23):4470–4472, 1982.

Shank C. V., Yen R., Fork P. L., Orenstein, J., **Picosecond Dynamics of Photo-Excited GaP States in Polyacetylene.** Phys Rev L 49(22):1660–1663, 1982.

Stillinger F. H., **Structure of the Interface Between Coexisting Fluid Phases.** Int J Quant 1982(S16):137–147, 1982.

Tamari N. et al., **High Quantum Efficiency InGaAsP InP Lasers.** Appl Phys L 41(11):1025–1027, 1982.

Temkin H., Dutt B. V., Bonner W. A., Keramidas V. G., **Deep Radiative Levels in InP.** J Appl Phys 53(11):7526–7533, 1982.

Tolk N. H. et al., **The Creation of Excited Hydrogenic States Near Surfaces.** Nucl Instru 202(1-2):247–251, 1982.

Tsang W. T. et al., **1.3-$\mu$m Wavelength GaInAsP InP Double Heterostructure Lasers Grown by Molecular-Beam Epitaxy.** Appl Phys L 41(11):1094–1096, 1982.

Valdes F., **Resolution Classifier.** P Soc Photo 331:465–472, 1982.

Vandenberg J. M. et al., **Structural Study of Alloyed Gold Metallization Contacts on InGaAsP/InP Layers.** J Appl Phys 53(11):7385–7389, 1982.

Vanuitert L. G., **The Effects of Electron Retention and Spin Pair Bonding on Crystal Cohesion in Insulator Oxides and Halides.** J Appl Phys 53(11):7140–7143, 1982.

Walstedt R. E., Walker, L. R., **Numerical-Simulation of Spin-Glass Transition Phenomena.** J Appl Phys 53(11):7985–7990, 1982.

Wilkins C. W., Reichmanis E., Chandross E. A., **Lithographic Evaluation of an Ortho-Nitrobenzyl Ester Based Deep UV Resist System.** J Elchem So 129(11):2552–2555, 1982.

Wilson R. J., Hahn E. L., **Off-Resonance Transient-Response of a Three-Level System.** Phys Rev A 26(6):3404–3407, 1982.

Wood D. L., Nassau K., Chadwick D. L., **Optical-Properties of New Oxide Glasses With Potential for Long-Wavelength Optical Fibers.** Appl Optics 21(23):4276–4279, 1982.

## SOCIAL AND LIFE SCIENCES

Friedman J. M. et al., **Transient Raman Study of Hemoglobin—Structural Dependence of the Iron-Histidine Linkage.** Science 218(4578):1244–1246, 1982.

Pashley R. D., Pedersen R. A., **Special Issue on Logic and Memory—Foreword (Editorial).** IEEE J Soli 17(5):791–792, 1982.

Tartter V. C., Fischer S. D., **Perceiving Minimal Distinctions in ASL Under Normal and Point-Light Display Conditions.** Perc Psych 32(4):327–334, 1982.

## SPEECH AND ACOUSTICS

Broek H. W., **Temporal and Spatial Fluctuations in Single-Path Underwater Acoustic-Wave Fronts. 1. Transmission From the 1st Convergence Zone at 43-nmi Range.** J Acoust So 72(5):1527–1532, 1982.

Broek H. W., **Temporal and Spatial Fluctuations in Single-Path Underwater**

**Acoustic-Wave Fronts. 2. Transmission over 270-nmi Range.** J Acoust So 72(5):1533–1537, 1982.

Broek H. W., **Temporal and Spatial Fluctuations in Single-Path Underwater Acoustic-Wave Fronts. 3. Transmission Over 540-nmi Range.** J Acoust So 72(5):1538–1543, 1982.

Mangano D. T., Broek H. W., **Correlation of Phase and Amplitude Fluctuations at Hydrophones Spaced 1/2 to 2 Wavelengths Apart.** J Acoust So 72(5):1556–1563, 1982.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 May-June 1983 Number 5, Part 1

Copyright © 1983 American Telephone and Telegraph Company. Printed in U.S.A.

# A Long-Wavelength Optical Receiver Using a Short-Channel Si-MOSFET

By K. OGAWA, B. OWEN, and H. J. BOLL

(Manuscript received April 21, 1982)

*Recent improvements in fine-line technology have resulted in silicon metal oxide semiconductor field-effect transistors (MOSFETs) with channel lengths between 0.2 and 0.8 μm. We have measured the low-frequency noise in these transistors and find it to be smaller than that in comparable GaAs-metal Schottky valve field-effect transistors (MESFETs). Theoretical considerations on the FET noise and experimental results at 45 Mb/s indicate that Si-MOSFETs can compete with GaAs-MESFETs in hybrid photoamplifier circuits. As a natural extension, Si-MOSFETs can also be used for the complete monolithic integration of the receiver circuit with the benefits of reliability and improved performance.*

## I. INTRODUCTION

In the absence of high-performance avalanche photodiodes for long-wavelength optical receivers, p-i-n photodiodes with low-noise amplifiers have been used.[1] The amplifiers are designed with ultra-low-noise components to realize high receiver sensitivities. Up until now, GaAs-metal Schottky valve field-effect transistors (MESFETs) were used exclusively as low-noise components at bit rates less than 300 Mb/s.[2,3] We have fabricated a short-channel Si-metal oxide semiconductor field-effect transistor (MOSFET)[4] and used this MOSFET in a hybrid integrated receiver circuit at 45 Mb/s. The receiver's performance is similar to that of a receiver employing a GaAs-MESFET. The use of

Si-MOSFETs creates the opportunity for monolithic integration of the entire front-end amplifier, with the ensuing benefits of circuit reliability and stability.

In the past, Si-FETS were ignored for this application because they were believed to have a lower transconductance than GaAs-FETs of comparable dimensions. This was due to the difference in mobility between the two materials. Recent improvements in fine-line technology have resulted in silicon MOSFETs with short channels from 0.2 to 0.8 $\mu$m. Also, we find that a short-channel Si-FET operating in the saturation region of the electron drift velocity has a transconductance comparable to that of the best GaAs-FET.[5,6] GaAs-FETS exhibit additional noise because of electron scattering in the high electric field of the channel. Since this effect is generally absent in Si-FETs,[7,8] the receiver sensitivity obtained using a Si-MOSFET is now expected to be comparable to or slightly better than that obtained using a GaAs-MESFET.

## II. SILICON SHORT-GATE MOSFET

Table I lists the characteristics of a Si-NMOSFET with a channel length between 0.5 and 0.8 $\mu$m, and for comparison shows the typical characteristics of a GaAs-MESFET with a channel length between 0.5 and 1.0 $\mu$m. The figure of merit, $g_m/C$ of a Si-N-channel metal oxide semiconductor field-effect transistor (NMOSFET) is smaller than that of a GaAs-MESFET when structures with the same dimensions are compared.[6] However, the noise factor, $\Gamma$, of a Si-NMOSFET is smaller than that of the GaAs FET if induced gate noise and its correlation with channel noise are considered.[5,7-9] The mean square of the equivalent input noise current of an FET is given by

$$i_n^2 = 4k\Gamma \frac{\omega^2 (C_{gs} + C_{in})^2}{g_m} \Delta f$$

$$\Gamma = P + \left| \frac{C_{gs}}{C_{in} + C_{gs}} \right|^2 R - 2Q \left| \frac{C_{gs}}{C_{in} + C_{gs}} \right|,$$

where $\Gamma$ is the noise factor, $g_m$ the transconductance, $C_{gs}$ the gate-source capacitance, and $C_{in}$ the input capacitance consisting of the

Table I—Typical Si-NMOSFET characteristics

| | Gate-Source Capacitance (pF) | Transconductance (mS) | Figure of Merit, $g_m/C_{gs}$ (mS/pF) |
|---|---|---|---|
| Si-NMOS | 0.5 to 0.8 | 40 to 50 | 60 to 70 |
| GaAs-MESFET (0.5 to 1.0 $\mu$m) | 0.2 to 0.5 | 25 to 50 | 60 to 140 |

Table II—Noise factor Γ, $P$, $Q$, $R$ for Si-NMOSFET

| | |
|---|---|
| $P$ | 0.763 |
| $Q$ | −0.206 |
| $R$ | 0.245 |
| Typical Γ $C_{gs} = 0.5_{pF} C_{in} = 0.5_{pF}$ | 1.03 |

photodiode capacitance and any parasitic capacitance. $P$ is the noise factor for the channel noise, $R$ is the noise factor for the induced gate noise, and $Q$ is the correlation factor. Table II indicates the values of Γ, $P$, $Q$, and $R$ for a Si-FET with a 0.5-$\mu$m channel length. The noise factor, Γ, for the Si-FET (1.03) is much smaller than the value for a typical GaAs-FET (1.78).

Low-frequency $1/f$ noise in FETs has an important effect on the performance of an optical receiver at low bit rates. We have measured the low-frequency noise of both a Si-NMOSFET and a GaAs FET. The results are shown in Fig. 1. Whereas the low-frequency noise for the GaAs FET does indeed have a $1/f$ dependence, the results show a $f^{-1/2}$ dependence for the Si-NMOSFET. This result has not yet been explained.

The noise measured in Fig. 1 was normalized to the expected channel noise $4kT\Gamma\Delta f/g_m$. The FET transconductance, $g_m$, was 48 mS; the filter bandwidth, $\Delta f$, was 3.1 kHz; and the noise factor, Γ, was 1.03 for

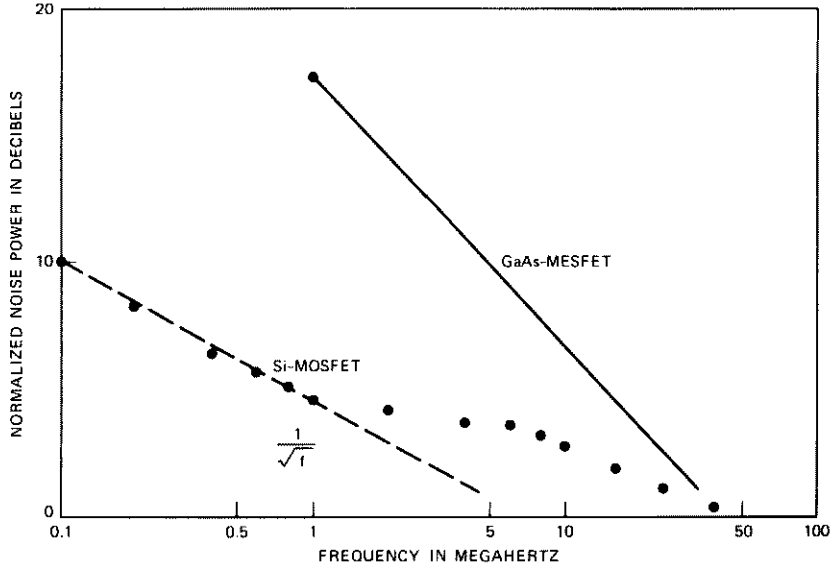



Fig. 1—Low-frequency characteristics of Si-NMOSFET. The dotted points show results measured with a 3.1-kHz filter. The best fit showed by the dashed line has a $f^{-1/2}$ slope. The $1/f$ noise of the GaAs-MESFET is shown (solid line) for comparison.

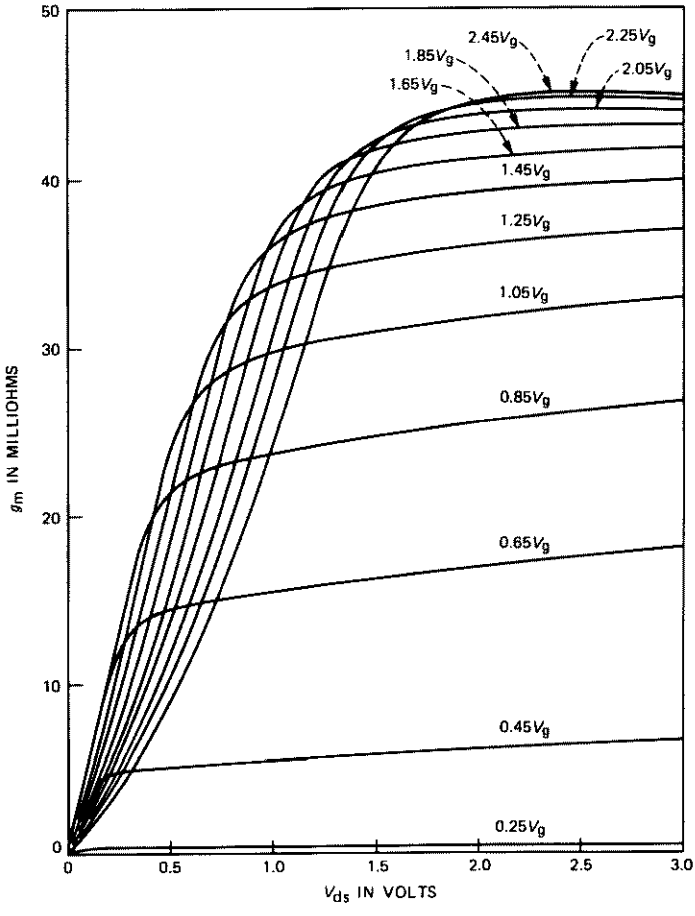



Fig. 2—Transconductance of Si-NMOSFET. Parameters are the gate-source voltage.

the Si-NMOSFET and 1.78 for the GaAs FET. In the Si-NMOSFET, the channel noise exceeded the $4kTT\Delta f/g_m$ value. The excess noise is believed to be thermal noise associated with the large series resistance of the polysilicon gate. This series gate resistance can be reduced by improved fabrication techniques, such as metallizing the gate. From Fig. 1, the noise corner frequency, $f_c$, for the Si-NMOSFET is $\approx$5 MHz. The noise corner frequency for the GaAs FET is $\approx$30 MHz. Therefore, even with the excess noise from the gate resistance, the low-frequency noise contribution of the Si-NMOSFET is clearly smaller than that of the GaAs FET.

Another FET parameter that affects its performance in an optical receiver is gate leakage current. The gate leakage current contributes shot noise at the receiver front end. Again, the Si-NMOSFET is

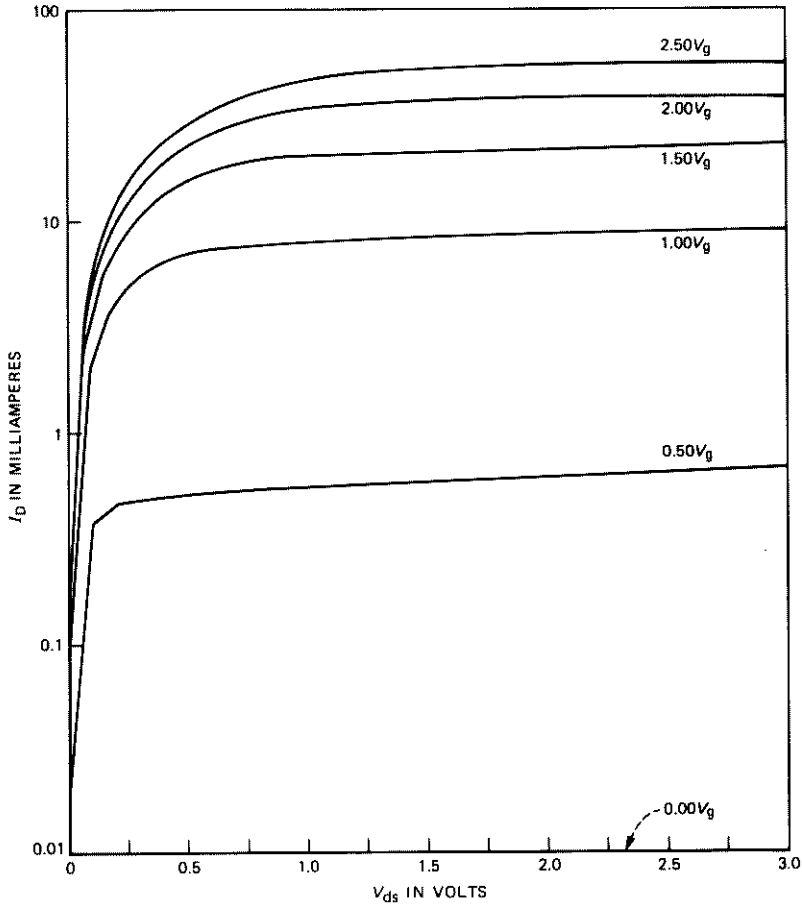



Fig. 3—I-V characteristics of Si-NMOSFET. The parameters are the gate-source voltage.

superior to the GaAs FET. The measured gate leakage current of the GaAs FET is ≈5 nA. The measured gate leakage current of the Si-NMOSFET is ≈10 pA.

Based on these measurements, we have calculated the sensitivity of a 45-Mb/s optical receiver using both a Si-NMOSFET and a GaAs FET. The receiver sensitivity is given by

$$\eta \bar{P} = \left| \frac{h\nu Q}{q} \right| \langle i^2 \rangle_T^{1/2},$$

where the prefactor $(h\nu Q/q)$ is 4.950 W/A at 1.3 μm and at $10^{-7}$ bit error rate; and where the equivalent input noise current, $\langle i^2 \rangle_T$, is given by

$$\langle i^2 \rangle_T = 2qI_L I_2 B$$ Shot noise from the gate leakage current, $I_L$.

$$+ 2qI_D I_2 B$$ Shot noise from the photodiode dark current, $I_D$.

$$+ \frac{4kT}{R_F} I_2 B$$ Thermal noise from the bias resistor, $R_F$.

$$+ \frac{16\pi^2 kTT(C_T^2) I_3 B^3}{g_m}$$ Channel noise in the FET.

$$+ \frac{16\pi^2 kTT(C_T)^2 f_c I_f B^2}{g_m}$$ $1/f$ noise in the FET.

$$+ \langle i^2 \rangle_c$$ Postamplifier noise,

where $B$ is the bit rate, and $I_2$, $I_3$, etc. are Personick integrals associated with the circuit noise. Assuming $R_F$ to be 500 kΩ, and the p-i-n photodiode dark current, $I_D$, at 30 nA, the sensitivity of a 45 Mb/s receiver at $10^{-7}$ bit error rate is −51.3 dBm for a GaAs FET front-end amplifier and −51.8 dBm for a Si-NMOSFET front-end amplifier.

## III. EXPERIMENTS

The Si-NMOSFET used for our experiments was fabricated on a $p$-type substrate (carrier concentration $\approx 2 \times 10^{15}/\text{cm}^3$) with an implanted

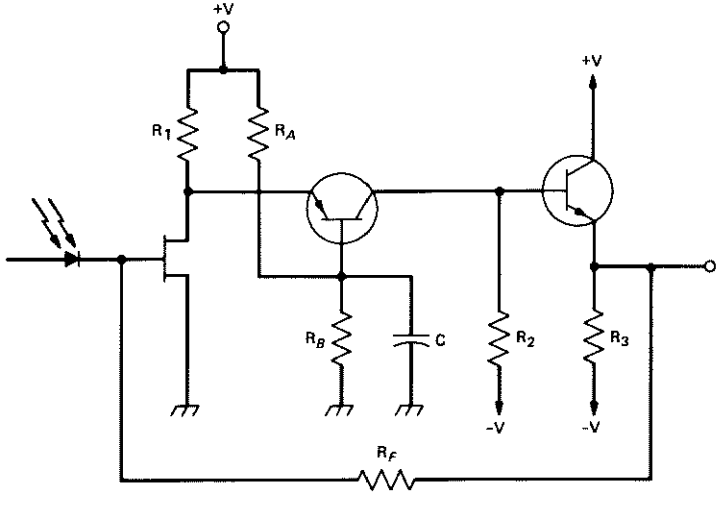



Fig. 4—Front-end amplifier circuit diagram with three active components involved. The first stage with the Si-NMOSFET provides a high input impedance. The second stage p-n-p transistor cascode circuit provides a gain stage and reduces the Miller effect. The last stage is an emitter follower for low output impedance.

$n$-layer (carrier concentration $\approx 5 \times 10^{16}/cm^3$), which had a thickness between 0.5 and 0.6 $\mu$m.[4] The channel width was 500 $\mu$m, and the effective channel length was 0.45 $\mu$m.[5] The gate-source capacitance was 0.5 pF and the transconductance was 45 mS. The gate leakage current was $\approx$10 pA. Figures 2 and 3 show the drain current and transconductance of the Si-NMOSFET versus drain source voltage with different gate voltages.

We have fabricated a transimpedance front-end circuit[3] using an InGaAs p-i-n photodiode with the Si-NMOSFET as the first amplifier stage. The circuit is shown in Fig. 4. The primary gain was achieved in the second stage, which used a p-n-p transistor. The third stage was an emitter-follower circuit using an n-p-n transistor. The feedback

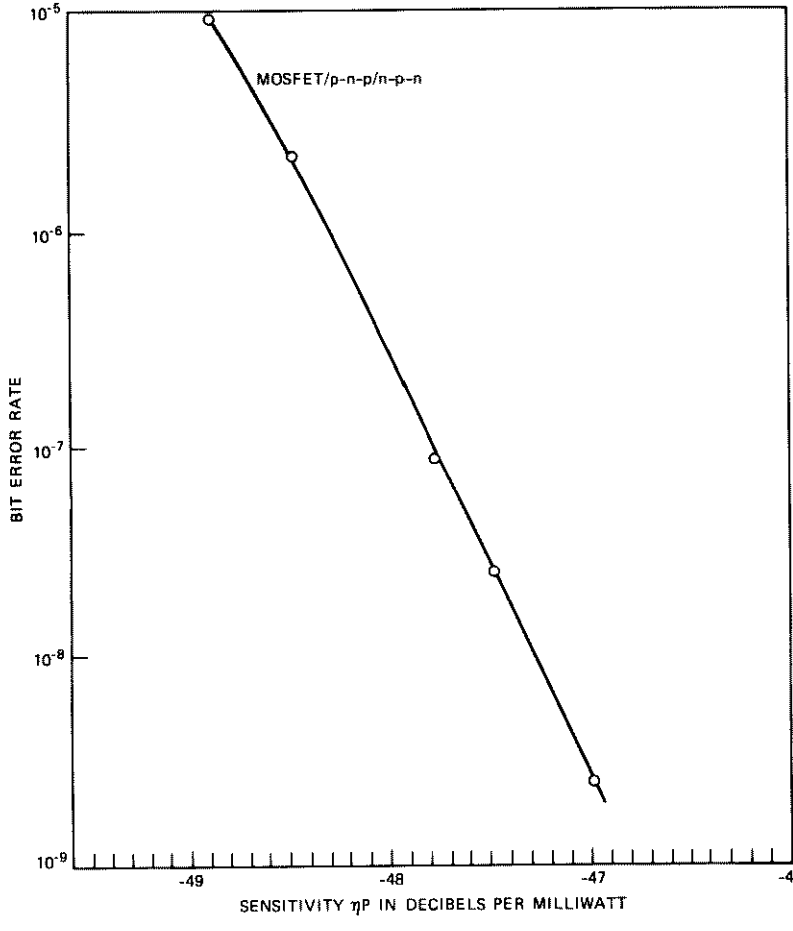



Fig. 5—Error-rate measurement of an InGaAs p-i-n-Si-NMOSFET receiver at 45 Mb/s.

circuit was 500 kΩ. The bit error rate was measured using a 44.7 Mb/s pseudorandom nonreturn to zero (NRZ) optical signal from an InGaAsP LED emitting at 1.3 μm. The receiver circuit was combined with a regenerator circuit and a retiming circuit designed for optical receivers. As shown in Fig. 5, the measured sensitivity at $10^{-7}$ bit error rate was −47.8 dBm (−51.8 dBm theoretical). With a GaAs FET first-amplifier stage, the same receiver circuit had a measured sensitivity of −49.5 dBm (−51.3 dBm theoretical).

## IV. CONCLUSION

Further work is in progress to improve the Si-NMOSFET receiver performance and to integrate the front-end amplifier. The circuit, especially the second stage, is not presently optimized. Also, the Si-NMOSFET has a high series gate resistance because the gate was fabricated with polysilicon. The noise penalty associated with this resistance can be eliminated by metallizing the gate.

In conclusion, theoretical considerations and our first experiments indicate that Si-NMOSFETs can have sensitivity performance comparable to that of GaAs FETs in a 45-Mb/s optical receiver. The natural extension of this result is the complete monolithic integration of the entire receiver circuit using silicon fine-line MOS technology.

## REFERENCES


1. Tingye Li, "Optical Fiber Communication—The State of the Art," IEEE Trans. Commun., *COM*-26 (July 1978), pp. 946–55.
2. D. R. Smith, R. C. Hooper, R. P. Webb, and M. F. Saunders, "PIN Photodiode Hybrid Optical Receivers," Proc. Optical Commun. Conf., Amsterdam, September 17–19, 1979.
3. K. Ogawa and E. L. Chinnock, "GaAs-FET Transimpedance Front-End Design for a Wide Band Optical Receiver," Elec. Lett., *15*, No. 20 (September 1979), pp. 650–3.
4. P. I. Suoiu, E. N. Fuls, and H. J. Boll, "High-Speed NMOS Circuits Made with X-Ray Lithography and Reactive Sputter Etching," IEEE Elec. Device Lett., *EDL*-1, No. 1 (January 1980), pp. 10–11.
5. W. Baechtold, "Si and GaAs 0.5 μm Gate Schottky Barrier Field Effect Transistors," Elec. Lett., *9*, No. 10 (May 1973), p. 232.
6. T. Wada and J. Frey, "Physical Basis of Short Channel MESFET Operation," IEEE Trans. Elec. Dev., *ED*-20, No. 4 (April 1979), pp. 476–89.
7. W. Baechtold, "Noise Behavior of Schottky Barrier Gate Field-Effect Transistors at Microwave Frequencies," IEEE Trans. Elec. Dev., *ED*-18, No. 2 (February 1971), pp. 99–104.
8. W. Baechtold, "Noise Behavior of GaAs Field Effect Transistors with Short Gate Lengths," IEEE Trans. Elec. Dev., *ED*-19, No. 5 (May 1972), pp. 676–80.
9. K. Ogawa, "Noise Caused by GaAs-MESFETs in Optical Receivers," B.S.T.J., *60*, No. 6 (July–August 1981), pp. 923–8.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 5, May–June 1983
*Printed in U.S.A.*

# Modal Analysis of Loss and Mode Mixing in Multimode Parabolic Index Splices

By I. A. WHITE and S. C. METTLER

(Manuscript received November 18, 1982)

*In this paper we present an electromagnetic modal theory for characterizing parabolic-index multimode fiber splices with either intrinsic or extrinsic mismatches. The theory agrees with previously published theoretical results for transverse offset using a uniform power distribution. It also agrees with new experimental measurements made with a long, spliced input fiber using a published, theoretical, steady-state modal power distribution. This modal theory predicts, and experiment confirms, a previously unreported periodic fluctuation in splice loss as a function of wavelength for intrinsic parameter mismatch. The analysis also predicts a large degree of mode mixing for transverse offset but negligible mode mixing for parameter mismatch in typical multimode fiber splices.*

## I. INTRODUCTION

Theoretical predictions of splice loss in multimode fibers have been attempted by many researchers (see Ref. 1 for a list). However, these analyses[1–3] do not adequately predict measured splice loss results, and most do not address mode mixing effects, which are important for the prediction of the bandwidth of concatenated lengths of fiber. The best agreement with splice loss measurement data is an empirical model based on geometric optics.[2] However, such an analysis of splice loss provides only a limited description of the effect of splices. Electromagnetic theory, using the coupling coefficients of individual modes, provides a complete treatment of splice loss and mode mixing. The only published electromagnetic theory[3] appears to be in error, since predictions do not approach the well-known correct geometric optics limit for splice loss of fibers with a large normalized frequency, $V$, and a uniform modal power distribution. This paper presents an electromagnetic analysis for splices, which gives single-term expressions for mode

coupling coefficients for the case of either transverse offset or profile parameter mismatch for parabolic-index multimode fibers. These coupling coefficients are used to calculate loss and degree of mode mixing in splices. The results approach the proper geometric optics limit for a uniform modal power distribution. Furthermore, if we compare theoretical and experimental loss to transverse offset results, we can verify a published theoretical "steady-state" power distribution[4] in multimode parabolic-index fibers. Comparing the results of this theory with a measurement of splice loss as a function of transverse offset gives the modal power distribution in a fiber. A wavelength dependence of splice loss for intrinsic parameter mismatch is predicted by this theory and has been experimentally verified. Because splices change the modal power distribution, the power redistribution can cause additional loss as the system evolves towards the steady state in the receiving fiber. In the past[2] this loss has been considered part of the total splice loss, but the improvement of fiber quality has reduced this effect significantly for typical lengths between splices. Because we have ignored the power redistribution effects (after the splice), these theoretical results are only strictly valid for short lengths of fiber after the splice; however, they should remain valid for typical distances between splices.

## II. THEORY

The modal amplitude coupling coefficient, $C[m_1\alpha_1; m_2\alpha_2]$, for mode $m_1\alpha_1$ of the transmitting fiber and mode $m_2\alpha_2$ of the receiving fiber is obtained directly from the theory of excitation of weakly guiding fibers.[5] (The variable $m$ is the radial mode number and $\alpha$ is the azimuthal mode number.)

$$C[m_1\alpha_1; m_2\alpha_2] = \frac{1}{2}\left(\frac{\epsilon_{co}}{\mu}\right)^{1/2} \int \bar{E}_{m_1\alpha_1} \cdot \bar{E}^*_{m_2\alpha_2} dA, \tag{1}$$

where for parabolic-index fibers (using the infinite profile approximation) the electromagnetic modal field, $E_{m_1\alpha_1}$, propagating in the $z$ direction in the fiber core is:

$$\bar{E}_{m_1\alpha_1} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} A_{m_1\alpha_1} t^{\frac{|\alpha_1|}{2}} L_{m_1}^{|\alpha_1|}(t) e^{-\frac{t}{2}} e^{-i(\alpha_1\phi+\beta z)}, \tag{2}$$

where $\hat{x}$ and $\hat{y}$ are the linearly polarized unit field vectors of the mode. $\epsilon_{co}$ and $\mu$ are the permittivity and permeability of the core, respectively, and $\epsilon_{co} = n_{co}^2\epsilon$. Then

$$A_{m_1\alpha_1} = \left[\left(\frac{\mu}{\epsilon_{co}}\right)^{1/2} \frac{V}{\pi a^2} \frac{m_1!}{(m_1 + |\alpha_1|)!}\right]^{1/2}, \tag{3}$$

and

$$\left.\begin{aligned} t &= V\left(\frac{r}{a}\right)^2 \\ \beta^2 &= \left(\frac{2\pi}{\lambda} n_{co}\right)^2 - 2N\frac{V}{a^2} \\ V &= \frac{2\pi}{\lambda} n_{co} a\sqrt{2\Delta}, \end{aligned}\right\} \tag{4}$$

where the principal mode number is $N(= 2m + \alpha + 1)$, and $L_m^{\alpha}(t)$ is an associated Laguerre polynomial. The variable $a$ is the fiber core radius and $\Delta = (n_{co} - n_{cl})/n_{co}$ is the normalized maximum refractive index difference. For weakly guiding fibers the polarization property of the incoming mode is preserved by the splice, as can be seen from eq. (1).

In general we can write:

$$C(m_1\alpha_1; m_2\alpha_2) = K \cdot I(m_1\alpha_1; m_2\alpha_2), \tag{5}$$

where

$$K = \left[\frac{m_1!m_2!}{(m_1 + |\alpha_1|)!(m_2 + |\alpha_2|)!}\right]^{1/2}. \tag{6}$$

$I(m_1\alpha_1; m_2\alpha_2)$ is a function that depends on the type of splice mismatch. Table I shows the expressions for $I(m_1\alpha_1; m_2\alpha_2)$ for the two cases of intrinsic parameter ($V$) mismatch and transverse offset between identical fibers ($r_o$). The derivation of these equations is given in the appendix.

For the parameter mismatch case, the coupling coefficients are a function of the ratio of the normalized frequencies of the two fibers. The hypergeometric function, ${}_2F_1$, is, in this case, a power series in $y^2$ of order $m_1$. Note that because azimuthal symmetry is preserved in the splice, only modes with the same azimuthal mode number couple. In the identical fiber transverse offset case, the coupling coefficients are simple products of Laguerre-Gaussian polynomials with argument proportional to the normalized offset ($r_o/a$). In this case all azimuthal modes have finite coupling coefficients, but, for small offsets, coupling is much stronger for nearest neighbor azimuthal modes.

Assuming a random phase relationship between the modes of the transmitting fiber, the total power coupled into mode $m_2\alpha_2$, $P_{m_2\alpha_2}$, is:

$$P_{m_2\alpha_2} = \sum_{m_1,\alpha_1}^{T} |C(m_1\alpha_1; m_2\alpha_2)|^2 P_{m_1\alpha_1}, \tag{7}$$

and the total splice loss, $\delta_s$, is then:

$$\delta_s = -10 \log \left( \frac{\sum_{m_2,\alpha_2}^{R} P_{m_2\alpha_2}}{\sum_{m_1,\alpha_1}^{T} P_{m_1\alpha_1}} \right) \tag{8}$$

The mode mixing factor, $\Gamma$ (i.e., the fraction of the power that is redistributed into different modes), is:

$$\Gamma = 1 - \frac{\sum_{m_1=m_2,\alpha_1=\alpha_2}^{T} |C(m_1\alpha_1; m_2\alpha_2)|^2 P_{m_1\alpha_1}}{\sum_{m_2,\alpha_2}^{R} P_{m_2\alpha_2}}. \tag{9}$$

The superscripts $R$ and $T$ refer to sums over the bound mode spectra of the receiving and transmitting fibers, respectively.

The range of allowed radial and azimuthal mode numbers, $m$ and $\alpha$, for a fiber with normalized frequency $V$ is:

$$\begin{aligned} m \leqslant m_{\max} &= \mathrm{INT}(V/4) \\ \alpha \leqslant \alpha_{\max} &= \mathrm{INT}(V/2) - 1 \\ N_{\max} &= \mathrm{INT}(V/2), \end{aligned} \tag{10}$$

Table I—Mode coupling functions

| Splice | Parameter of Fiber: Transmitting | Parameter of Fiber: Receiving | $I(m_1\alpha_1; m_2\alpha_2)$ |
|---|---|---|---|
| Parameter Mismatch | | | $= 0$ for $\alpha_1 \neq \alpha_2$ |
| | $a_T$ | $a_R$ | $= \frac{(m_1+m_2+\alpha)!}{m_1!m_2!} \cdot \frac{(1+\epsilon)^{(\alpha+1)/2}}{\left(1+\frac{\epsilon}{2}\right)^{\alpha+1}} \cdot (-1)^{m_1}$ |
| | $\Delta_T$ | $\Delta_R$ | $\cdot y^{-(m_1+m_2)}\, {}_2F_1[-m_2;-m_1;-(m_1+m_2+\alpha);y^2],$ |
| | $V_T$ | $V_R$ | where $y = (2+\epsilon)/\epsilon$ and $\epsilon = \left(\frac{V_R}{\alpha_R^2} \cdot \frac{\alpha_T^2}{V_T}\right) - 1$ |
| Transverse Offset for Identical Fibers | $a$ | $a$ | $= (R_o)^{p/2} e^{-R_o} \cdot \phi(m_1\alpha_1; m_2\alpha_2)$ for $\alpha_1 > 0$, $\alpha_2 < 0$; |
| | $\Delta$ | $\Delta$ | $\phi = L_{m_1}^{\lvert\alpha_2\rvert-\Delta m}(R_o) L_{m_2}^{\lvert\alpha_1\rvert+\Delta m}(R_o)$ for $\alpha_1 > 0$, $\alpha_2 > 0$; |
| | Transverse displacement of the fiber axes $= r_o$ | | $\phi = \frac{(m_2+\alpha_2)!}{m_1!} \cdot (-R_o)^{\Delta m} L_{m_2+\alpha_2}^{p+\Delta m}(R_o) \cdot L_{m_2}^{\Delta m}(R_o),$ where $R_o = \frac{V}{4}\left(\frac{r_o}{a}\right)^2$; $\Delta m = m_1 - m_2$; $p = \lvert\alpha_1 - \alpha_2\rvert$ |

where INT indicates the integer part. It is essential to realize the discrete nature of the bound mode spectra to understand the splice loss predictions for intrinsic parameter ($V$) mismatch. The explicit wavelength dependence of splice loss and mode mixing is demonstrated by the presence of $V$ in the coupling coefficients. However, an implicit wavelength dependence due to the allowed spectrum of bound modes occurs from eq. (10).

## III. RESULTS

Equations (8) and (9) can be used to calculate splice losses and degree of mode mixing for the two cases of parameter mismatch and transverse offset. To validate this analysis, they must be compared with existing theories wherever possible. Most existing results are geometric optics evaluations of splice loss, and the modal power distribution equivalent to any assumed ray distribution is difficult to evaluate in general. However, the ray distribution equivalent to the uniform modal power distribution has been well documented,[6] and comparison of splice loss predictions for this case demonstrates that this analysis is correct. Calculations and experimental results for the effects of different modal power distributions on the mode mixing and loss at a splice are shown in the sections that follow.

### *3.1 Identical fiber transverse offset*

This section presents the results for mode mixing and splice loss associated with transverse offset of the fiber axes for identical fibers.

#### *3.1.1 Splice loss*

Splice loss predictions for transverse offset with a uniform power distribution for several values of the normalized frequency, $V$, are shown in Fig. 1. It is well known that, as $V$ increases, the electromagnetic analysis should asymptotically approach the geometric optics predictions. This is confirmed in Fig. 1. The only other published electromagnetic splice loss theory[3] does not appear to converge to the correct geometric optics limit, e.g., from Fig. 3 of Ref. 3, the splice loss for $0.2a$ offset (for $2a = 50$ $\mu$m) is $\sim$1.0 dB, whereas geometric optics predict a value of 0.8 dB. The small deviation of our results for small offset ($<0.1a$) is believed to be caused by coupling of power to the bound modes of the receiving fiber through the evanescent cladding fields. This effect cannot be predicted using ray optics.

Uniform power distribution results do not agree with realistic splice loss measurements.[2] Figure 2 demonstrates loss versus transverse offset for several power distributions, the Gaussian splice loss model,[2] and experimental data.[8] The experimental data shown were obtained using 0.82-$\mu$m laser excitation of a 7-km input fiber containing about

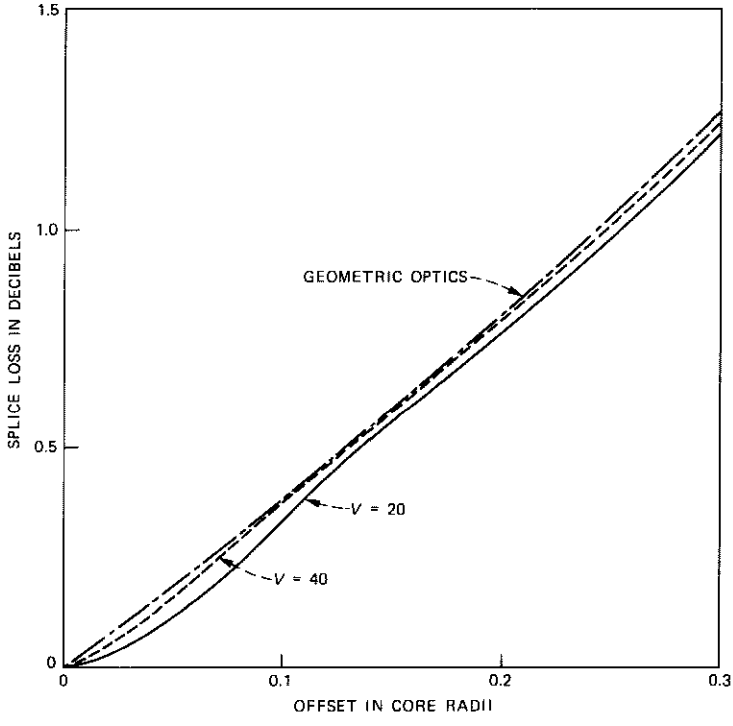



Fig. 1—Splice loss vs. transverse offset with uniform input power.

14 splices into a final splice that was offset in both $x$ and $y$ orthogonal axes. The offset data are in good agreement with previous data and should represent a realistic "steady-state" condition. The theoretical power distribution for steady state due to microbending[4] gives excellent agreement with this data. (A power distribution that has been used to approximate this steady-state distribution is also plotted to show the sensitivity of the loss to the choice of power distribution.) The excellent agreement between the data and this analysis using the steady-state power distribution confirms both the distribution and the splice loss theory. Although this power distribution gives excellent agreement with the data for very long lengths of fiber before the splice, it does not agree as well as the Gaussian model[2] with shorter-length input fiber measurements because the power distribution has not achieved "steady state."

Figure 3 compares experimental data for shorter lengths of input fiber (~1 km) and different excitation conditions[8] with theoretical predictions using both the Gaussian model[2] and various power distributions in this modal analysis. Note that the different excitation conditions significantly affect the splice loss for larger offsets. The

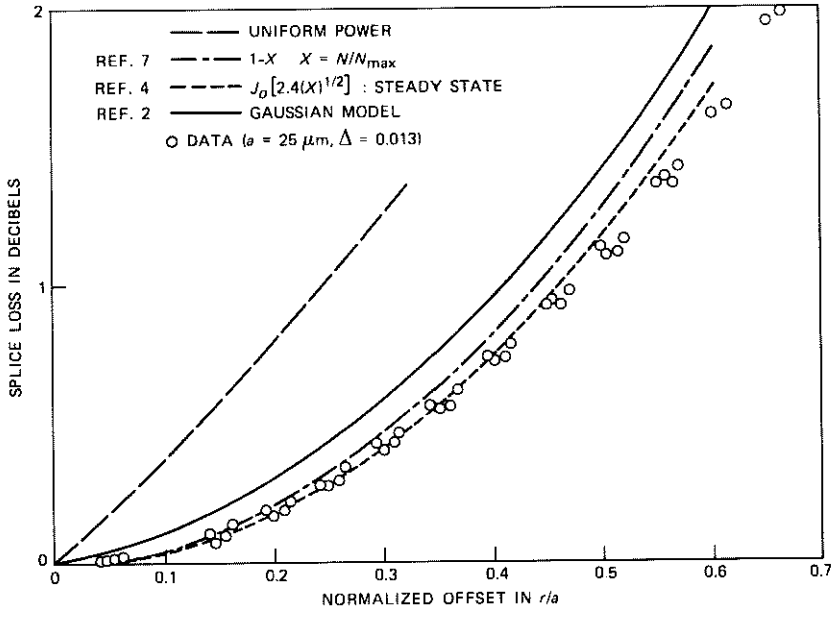



Fig. 2—Loss vs. transverse offset for different modal power distributions.

empirical model of Ref. 2 agrees well with loss measurements for typical offsets expected in practice of 0.1 to 0.2 ($r_o/a$) for both sources and agrees very well over a wider range for the laser source. Modal power distributions for the electromagnetic analysis to describe this situation were empirically chosen so that, for very long fiber lengths, they would degenerate to the steady-state distribution. The simplest choice for a power distribution satisfying this criteria is:

$$P(m_1\alpha_1) = J_o\left[2.405\sqrt{\frac{N}{N_{\max}}}(1 - e^{\eta L})\right]. \tag{11}$$

The choices of $\eta$ shown in Fig. 3 (for $L = 1$ km) demonstrate that excellent agreement between the modal analysis and the splice loss versus transverse offset for the shorter input fiber lengths is possible for both sources. Furthermore, for $\eta = 3$, the results agree with the predictions of the empirical model[2] within 0.03 dB. The theoretically predicted steady-state distribution gives optimistic results for this short length. The existence of a modal power distribution that is in good agreement with the results of Ref. 2 for transverse offset supports the basic assumptions of the empirical model.

### *3.1.2 Mode mixing*

Equation (9) can be used to calculate the amount of power in the

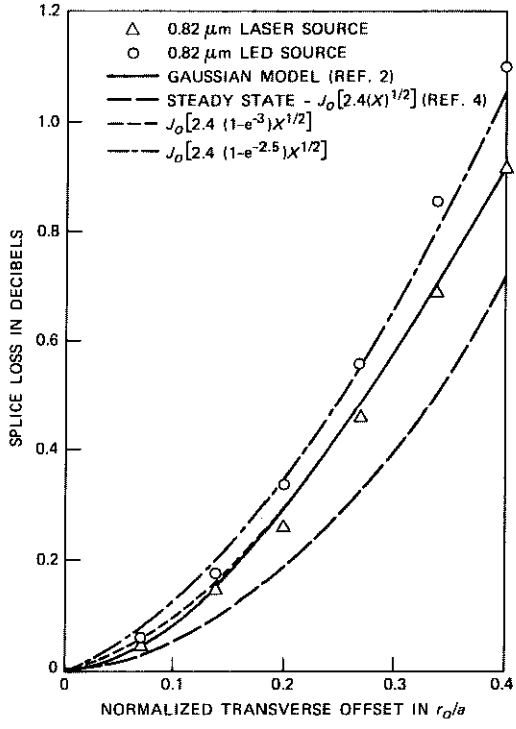



Fig. 3—Splice loss vs. transverse offset for 1-km input fiber.

receiving fiber that has changed propagation mode because of the splice. By changing the summation criteria one can calculate any desired mode mixing factor, such as power coupled within the same principal mode group, nearest neighbor mode groups, etc., where the summation is over all $m_1$, $\alpha_1$ and $m_2$, $\alpha_2$ that differ in principal mode number by $\Delta N$, which is written as:

$$\sum_{2m_2+\alpha_2-2m_1-\alpha_1=\Delta N} \quad \text{for} \quad \Delta N = 0, 1, 2, \cdots . \tag{12}$$

Mode mixing as a function of normalized transverse offset for two different input power distributions is shown in Fig. 4 as the percentage of receiving fiber power that has been redistributed into neighboring mode groups. These differ in principal mode number, $N$, by $\Delta N$ of 0, 1, 2, 3, and 4. ($\Delta N = 0$ represents power coupled within the same degenerate mode group in both transmitting and receiving fibers.) The power distributions used were the uniform and the theoretically predicted power distribution for the steady state for microbending.[4] It is important to note that the percentage of the power-changing propagating mode groups is very high even for small offsets with low splice

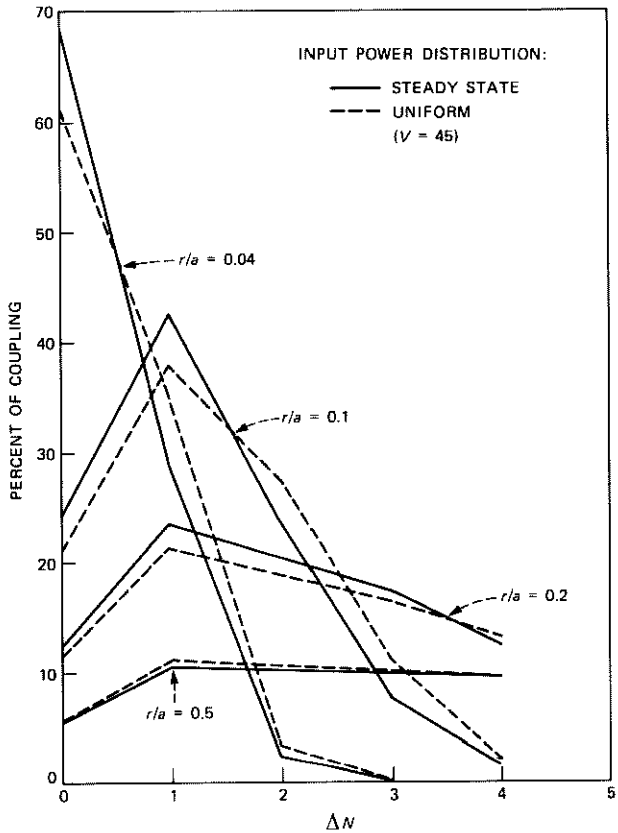



Fig. 4—Power coupling into neighboring mode groups for transverse offset.

loss, i.e., 31 percent for $0.04a$ offset and 0.07 dB splice loss for a $V = 45$ fiber with the steady-state power distribution. Mode mixing is a fairly weak function of the modal power distribution, which indicates that the mode mixing is uniform over all the mode groups as can be seen from the local numerical aperture arguments.[2] As offset increases, the mode mixing changes from being primarily to the nearest neighbor mode groups ($\Delta N = \pm 1$) to being redistributed over a wider range of principal mode groups. The strength of the mode coupling for even small offsets with low splice loss, and the relatively small offsets at which it spreads over many mode groups, are both initially surprising.

### *3.2 Parameter mismatch*

For fibers with no axial offset, the individual mode fields are insensitive to small changes in the $V$ parameter, so that for splices with only slight parameter mismatch, as can be seen in eq. (1), self-coupling

dominates (i.e., mode $m_1\alpha_1$ of the transmitting fiber couples primarily with mode $m_1\alpha_1$ of the receiving fiber). For example, there is less than 3-percent mode mixing for a 5-percent parameter mismatch with uniform modal excitation over the wavelength range of 0.8 to 1.4 $\mu$m. Therefore, splice loss due to parameter mismatch is caused only by this slight field mismatch unless the ranges of the bound mode spectra of the two fibers [eq. (10)] differ, i.e., if the normalized frequencies of the fibers are such that $\Delta N_{\max} \geq 1$. In that case:

$$\Delta N_{\max} = N^T_{\max} - N^R_{\max} \tag{13}$$

or

$$\Delta N_{\max} = \text{INT}\left[\frac{\pi}{\lambda} n_{co}(a_T\sqrt{2\Delta_T})\right] - \text{INT}\left[\frac{\pi}{\lambda} n_{co}(a_r\sqrt{2\Delta_R})\right]. \tag{14}$$

When $\Delta N_{\max}$ is zero, the splice loss is small, as discussed above. However, when $\Delta N_{\max} > 0$, the modes in the highest-order mode group of the transmitting fiber are not bound modes in the receiving fiber. Because self-coupling dominates, most of the power in the highest-order modes couples to leaky/radiation modes in the receiving fiber and is therefore lost. The large degeneracy of the higher-order mode groups [eq. (4)] accentuates this loss. From this discussion we see that parameter mismatch splice loss is caused primarily by the difference between the bound mode volumes of the two fibers, which is a function of the normalized frequencies, $V$. Therefore, both $\Delta$ and radius mismatches can be expressed simply as $V$ mismatch. (An $x$-percent radius mismatch is equivalent to a $2x$-percent $\Delta$ mismatch for small degrees of mismatch.)

Theoretical predictions of splice loss versus normalized parameter mismatch ($\Delta V/V$) for uniform and steay-state input power distributions at 0.82 and 1.3 $\mu$m are shown in Fig. 5. The discontinuities of the splice loss curves are caused by the discreteness of the bound mode spectra, as can be seen from the definition of $\Delta N_{\max}$ in eq. (14). In Fig. 5a this discreteness is most obvious due to the choice of a uniform input power distribution, which accentuates the power in the highest-order mode group. Note that at the longer wavelength the loss is initially higher for a given parameter mismatch because the percentage of the total power that is in the highest-order mode groups is greater than at shorter wavelengths. Each discontinuous increase of splice loss is caused by the stripping of an additional high-order mode group of the input fiber, which does not propagate in the receiving fiber as $\Delta V/V$ increases. The splice loss predicted by geometric optics (which ignores the discreteness of the mode spectra) is also shown.[1] The results for the steady-state power distribution used previously are shown in Fig. 5b displaying similar discontinuous features. However,

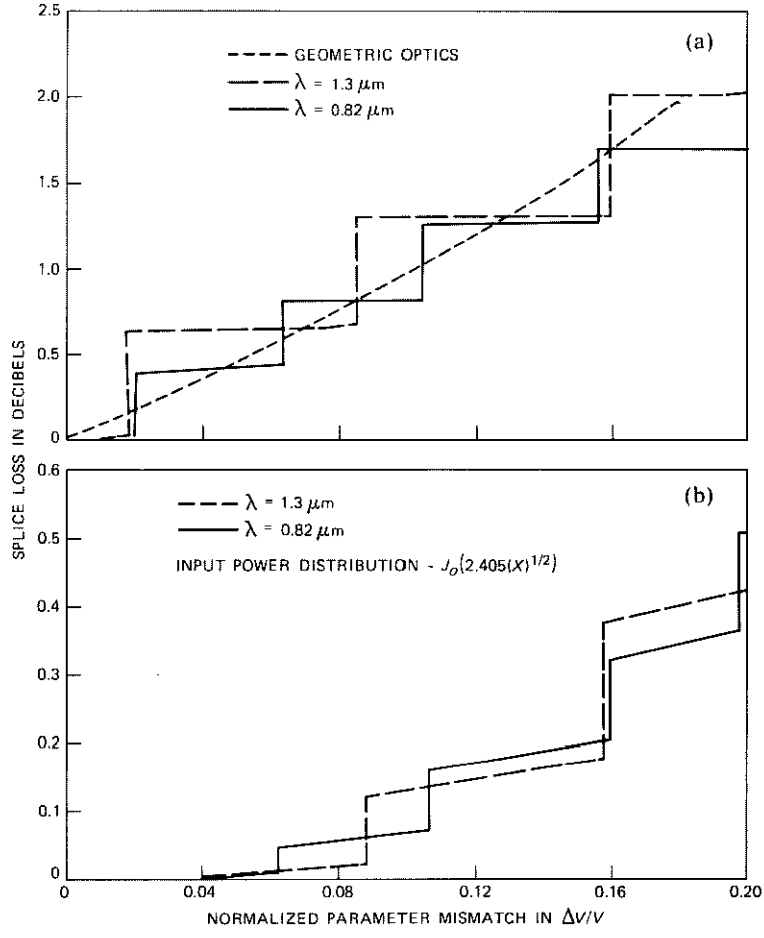



Fig. 5—Splice loss vs. normalized parameter mismatch in $\Delta V/V$ for (a) uniform power distribution and (b) steady-state power distribution.

the decrease in the order of magnitude of the loss allows the observation of the small loss component caused by the slight field mismatch between similar modes of the two fibers. This is reflected in the small slope of the levels. Because the steady-state distribution assumes that there is no power in the highest-order mode group of the transmitting fiber, the first discontinuous change of splice loss of Fig. 5a is negligible in Fig. 5b.

Figure 6 shows the percentage of mode mixing at a splice caused by parameter mismatch for 0.82- and 1.3-μm wavelengths, and the uniform and steady-state power distributions. The substantially higher mode mixing for the uniform power case compared to the steady-state indicates that mode mixing is dominated by the higher-order modes.

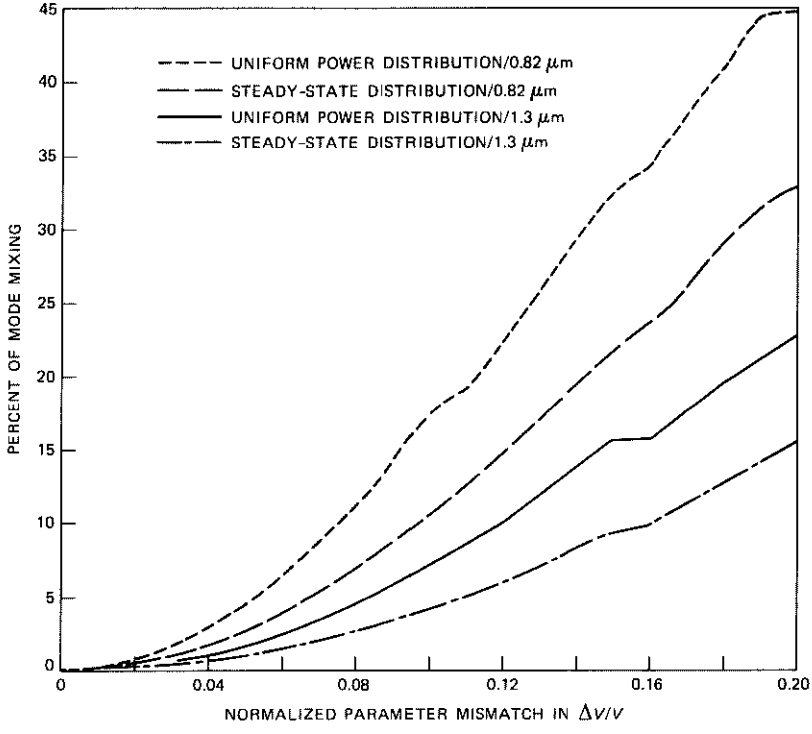



Fig. 6—Percent of mode mixing vs. normalized parameter mismatch.

This contrasts with the transverse offset case shown in Fig. 4, where mode mixing is essentially the same for uniform and steady-state distributions. Reduced mode mixing at longer wavelengths is due to the more diffuse bound mode fields, which result in stronger self coupling between modes. In comparison to the transverse offset case, the degree of mode mixing for the case of parameter mismatch is substantially reduced. Furthermore, a detailed analysis of the mode mixing shows that the coupling is almost completely nearest neighbor coupling ($\Delta N = 1$), e.g., for $\Delta V/V \sim 0.2$, 38 percent of the total 44-percent mode mixed power is coupled to nearest neighbor modes. The deviations from a smoothly increasing function in these curves occur at the transition points of the splice loss curves (see Fig. 5). This is caused by the transition of the highest-order mode group in the receiving fiber from propagating to lossy, removing its contribution to the overall mode mixing.

### *3.3 Wavelength dependence of splice loss for parameter mismatch*

As we can see from eq. (13), for a given degree of parameter mismatch, $\Delta N_{max}$ is also a discontinuous function of wavelength. For

example, if, at some particular wavelength, $V_T$ is 46.1 and $V_R$ is 45.9, $\Delta N_{max}$ is 1 and splice loss is ~0.5 dB. A slight change of wavelength (~1 percent) will cause both $V$'s to simultaneously be greater or less than 46 with $\Delta N_{max} = 0$, and splice loss is ~0.02 dB. Theoretical considerations [eq. (14)] also indicate that the period of the splice loss fluctuations should increase with increasing wavelength because the mode volume is proportional to $1/\lambda$. The magnitude of the fluctuation should also increase with wavelength because the relative amount of power in the highest-order mode group increases as the mode volume decreases. The splice loss wavelength dependence for a parameter mismatch ($\Delta V/V$) of 0.04 and uniform input power distribution is shown in Fig. 7. The magnitude of the splice loss changes considerably over the entire wavelength range. Splice loss calculated from geometric optics for this parameter mismatch case is also shown in Fig. 7 and is approximately the wavelength averaged splice loss as calculated from modal theory. Although the splice loss displays this pathological behavior, the mode mixing remains relatively small and continuous.

Figure 8 gives two further examples of the theoretical splice loss versus wavelength for uniform power excitation and core radius mismatches of 1.0 and 1.5 $\mu$m. For $\Delta a$ of 1.5 $\mu$m, there are ranges of wavelength over which $\Delta N_{max} = 2$, resulting in even larger fluctuations. The choice of a uniform power distribution unrealistically enhances the magnitude of this effect. For example, with the choice of the steady-state distribution, there would be very little power in the highest-order mode group, and therefore this change in loss would be very small, as we can see in Fig. 5b. Splice loss is not expected to vary as abruptly as shown in Figs. 7 and 8, even for a uniform power distribution, because the cladding splits the degeneracy of the highest-order mode group and some modes of this group remain bound even when the group itself, as predicted by the infinite parabolic approxi-

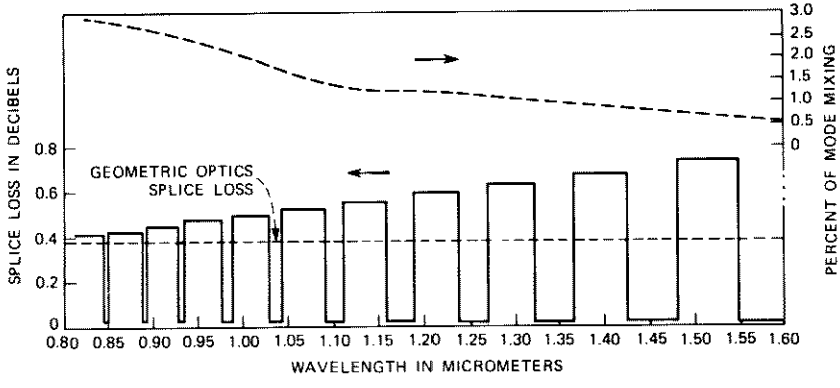



Fig. 7—Splice loss and mode mixing vs. wavelength for $\Delta V/V = 0.04$.

mation, is not bound. This results in a smoothing of the loss versus λ curve, but the periodicity should not be affected. The observation of this effect would further verify this analysis.

### *3.4 Experiment*

The experimental setup shown in Fig. 9 was designed to enhance the power in higher-order mode groups in an attempt to observe the predicted wavelength dependence of splice loss shown in Figs. 7 and 8. A 5 m, 63-μm core, step-index fiber was over-filled and used to excite

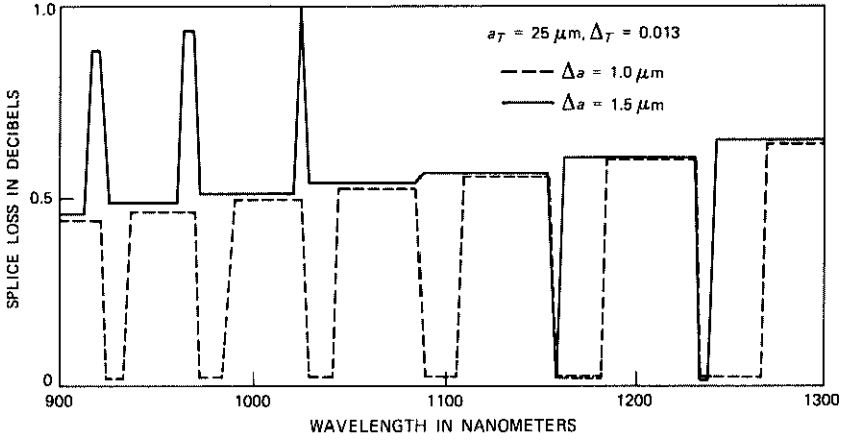



Fig. 8—Theoretical loss vs. wavelength for parameter mismatch.

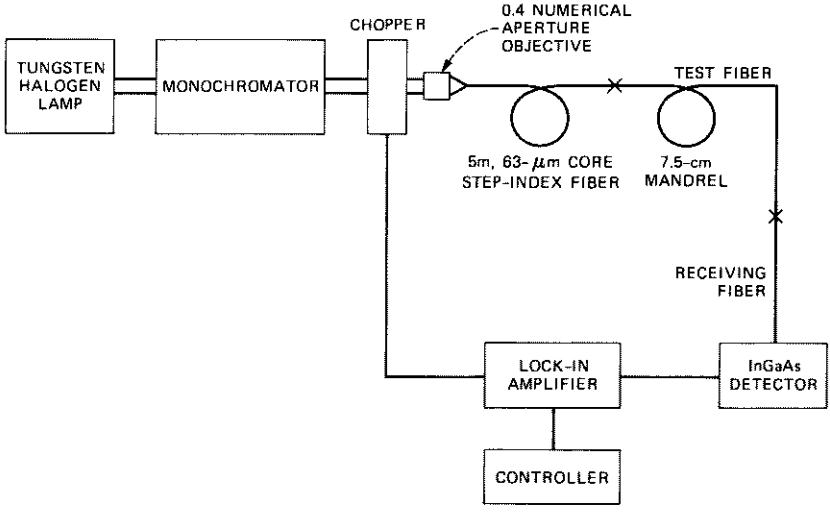



Fig. 9—Splice loss measurement system.

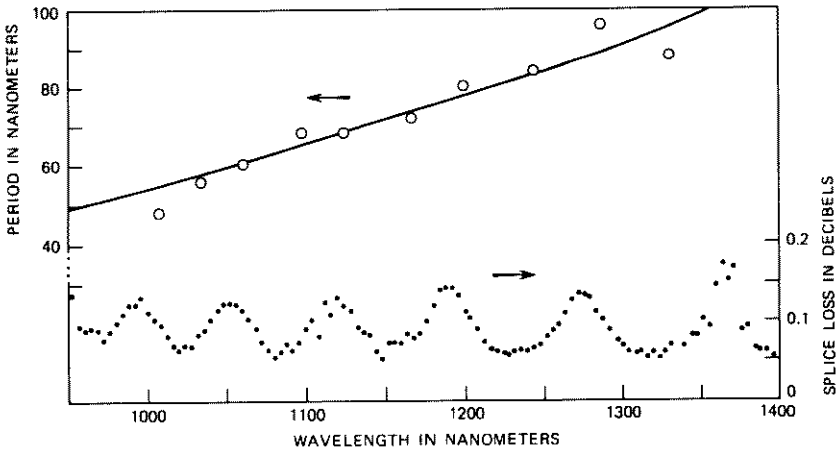



Fig. 10—Measured period and splice loss for parameter mismatch.

the test fiber, which was wound on a 7.5-cm mandrel. A cut-back test determined that approximately 15 m of fiber on this mandrel successfully eliminated leaky modes in the transmitting fiber. This fiber was spliced with micropositioners to a 1 to 2 m length of fiber, which differed in core radius by 1.0 ± 0.2 $\mu$m (but with the same $\Delta$), as determined by refracted near-field measurements. Monochromator resolution was 2 nm and the repeatability of the splice loss measurements was 0.02 dB. As a final check the test fiber was spliced to a length of identical fiber. Maximum splice loss change seen in this case was 0.02 dB, with no discernible wavelength dependence.

The measured wavelength dependence of splice loss and its period are plotted in Fig. 10 for the mismatched fibers. The data show excellent agreement with the period versus wavelength curve obtained from Fig. 8. The increase in maximum amplitude with wavelength seen in Fig. 10 is in qualitative agreement with the theory. Cladding effects (ignored in the infinite parabolic approximation) and only partial success in achieving a uniform power distribution are the probable reasons that the maximum amplitude is less than expected from Fig. 8.

## IV. CONCLUSIONS

In this paper we have presented a modal theory that allows the calculation of mode mixing and loss resulting from parabolic-index multimode fiber splices. The predictions of this model have been confirmed by experiment, including a previously unreported wavelength dependence of splice loss for intrinsic parameter mismatch.

This phenomenon is not expected to be significant for practical "steady-state" system losses but is important for providing insight into the actual loss mechanisms at a splice as well as verifying the validity of the theory. Another effect of this discreteness of the bound mode spectrum has previously been reported for the wavelength dependence of bending loss.[9] The theory also successfully describes splice loss versus transverse offset phenomena and has been used to verify the predicted steady-state power distribution. The analysis provides a basis for understanding mode-mixing effects at splices, which is important for bandwidth studies on concatenated fiber lengths. Splice losses caused by offset or parameter mismatch produce greatly different degrees of intermodal coupling. This implies that the effect of the splice on the bandwidth of a concatenated length is insignificant for the case of small parameter mismatch, but may be significant for transverse offset (due to the strong mode mixing). The sensitivity of the splice loss predictions to the choice of the input power distribution may allow this analysis to provide an evaluation of modal power distributions in fibers from experimental measurements of splice loss versus transverse offset.

The particular case of longitudinal offset has been obtained from a generalization of the parameter mismatch theory but has not been treated here due to its relative unimportance in practical systems. Gloge used geometric optics to show that fiber axis tilt at the splice is equivalent to a transverse offset via the relationship $(r_o/a)^2$ is equal to $\sin^2\theta/2\Delta$, where $\theta$ is the angle between the fiber axes.[10] This equivalence is also valid from the electromagnetic analysis for pure tilt or offset. The modal analysis of combinations of offset (both transverse and longitudinal) with tilt and parameter mismatch (as discussed in Ref. 10) are subjects of continuing work.

## V. ACKNOWLEDGMENTS

We would like to thank D. W. Peckham for providing loss versus transverse offset data; M. J. Saunders for index profile data; and C. M. Miller, P. J. Rich, and R. B. Kummer for helpful discussions.

## REFERENCES

1. S. E. Miller and A. G. Chynoweth, Eds., *Optical Fiber Telecommunications*, New York, NY: Academic Press, 1979, p. 74.
2. C. M. Miller and S. C. Mettler, "A Loss Model for Parabolic Profile Fiber Splices," B.S.T.J., *57*, No. 9 (November 1978), pp. 3167–80.
3. N. Kashima, "Transmission Characteristics of Splices in Graded-Index Multimode Fibers," Appl. Optics, *20*, No. 22 (November 15, 1981), pp. 3859–66.
4. D. Marcuse, "Losses and Impulse Response of a Parabolic Index Fiber With Random Bends," B.S.T.J., *52*, No. 8 (October 1973), pp. 1423–37.
5. A. W. Snyder, "Excitation and Scattering of Modes on a Dielectric or Optical Fiber," IEEE Trans., *MTT*-17 (December 1969), pp. 1138–44.
6. D. Gloge and E. A. J. Marcatili, "Multimode Theory of Graded-Core Fibers," B.S.T.J., *52*, No. 9 (November 1973), pp. 1563–78.

7. R. Olshansky, "Mode Coupling Effects in Graded-Index Optical Fibers," Appl. Optics, *14*, No. 4 (April 1975), pp. 935–45.
8. D. W. Peckham, unpublished work.
9. T. Ohshima et al., "Cutoff Wavelength Measurements of a Multimode Fiber," Proc. of 7th E.C.O.C., Copenhagen, Denmark, 1981, pp. 5-1-5-4.
10. D. Gloge, "Offset and Tilt Loss in Optical Fiber Splices," B.S.T.J., *55*, No. 7 (September 1976), pp. 905–16.
11. S. C. Mettler, "A General Characterization of Splice Loss for Multimode Optical Fibers," B.S.T.J., *58*, No. 10 (December 1979), pp. 2163–82.
11. I. S. Gradshteyn et al., *Tables of Integrals, Series, and Products*, New York, NY: Academic Press, 1980, p. 844, Eqn. 7.414.4.
12. N. N. Lebedev, *Special Functions and Their Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1965, p. 83, Eq. 4.20.3.
13. J. A. Stratton, *Electromagnetic Theory*, New York, NY: McGraw-Hill, 1941, p. 373.
14. A. Erdélyi, Ed., *Higher Transcendental Functions*, The Bateman Manuscript Project, New York, NY: McGraw-Hill, Vol. 2, p. 188.
15. Ref. 11, p. 848, Eqn. 7.422.2.
16. A. Erdélyi, Ed., *Tables of Integral Transforms*, The Bateman Manuscript Project, New York, NY: McGraw-Hill, 1954, Vol. 2, p. 43.

## APPENDIX

### *Derivation of Modal Coupling Coefficients*

The principal steps in the derivation of the modal coupling coefficients (eq. 1) presented in Table I are shown in this appendix.

#### *A.1 Intrinsic parameter ($V$) mismatch*

In this case both fibers have fields as defined in eq. (2), where the argument of the field, $Z$, is

$$Z_T = V_T(r/a_T)^2 \tag{15}$$

and

$$Z_R = Z_T(1 + \epsilon), \tag{16}$$

where

$$V_T = (2\pi/\lambda) \cdot n_{co} \cdot \sqrt{2\Delta_T} \cdot a_T \tag{17}$$

and

$$\epsilon = \left(\frac{V_R}{a_R^2}\right)\left(\frac{a_T^2}{V_T}\right) - 1, \tag{18}$$

where the normalization expressions $A_{m\alpha}$ also contain $V_T$ and $V_R$.

Examination of the azimuthal integral of eq. (1) shows that, because there is no transverse offset of the fiber axes, azimuthal symmetry is preserved and only modes with the same azimuthal symmetry couple.

Changing the radial integral from an integration over $r$ to one over $Z_T$, we find that it is in the form of a well-documented integral of Laguerre Gaussian functions,[11] giving the expression shown in Table I. The hypergeometric function, ${}_2F_1(-m_2, -m_1, -(m_1 + m_2 + \alpha), y^2)$ is a finite power series in $y^2$ whose order is the lesser of $m_1$ or $m_2$, since both $m_1$ and $m_2$ are integers.

### A.2 Transverse offset—identical fibers

The modal fields to be used in eq. (1) can still be expressed by eq. (2) for both fibers. However, the radial and azimuthal variations of the receiving fiber ($r_R$, $\phi_R$) need to be expressed in terms of the transmitting fiber ($r_T$, $\phi$) and the transverse offset, $r_o$, of the axes. Using

$$r_R^2 = r_T^2 + r_o^2 - 2r_T r_o \cos\phi \tag{19}$$

and the integral relation for Laguerre-Gaussian functions[12]

$$e^{-Z/2}Z^{|\alpha|/2}L_m^{|\alpha|}(Z) = \frac{(-1)^m}{2}\int_o^\infty J_\alpha(\sqrt{xZ})e^{-x/2}x^{|\alpha|/2}L_m^{|\alpha|}(x)dx \tag{20}$$

we can then use the addition theorem for circular cylindrical waves[13] to write the receiving fiber field amplitude $E_R(m, \alpha)$ as

$$E_R(m, \alpha) = \frac{(-1)^m}{2}\sum_{p=-\infty}^{\infty} e^{-i(p+\alpha)\phi} \cdot \int_o^\infty J_p(\sqrt{xZ_o})J_{p+\alpha}(\sqrt{xZ_T})e^{-x/2}x^{|\alpha|/2}L_m^{|\alpha|}(x)dx, \tag{21}$$

where

$$Z_o = V(r_o/a)^2$$
$$Z_T = V(r_T/a)^2. \tag{22}$$

The azimuthal integral of eq. (1), for coupling between mode ($m_1\alpha_1$) of the transmitting fiber and mode ($m_2\alpha_2$) of the receiving fiber, reduces the infinite summation over $p$, introduced by eq. (21), to a single term for

$$p + \alpha_2 = \alpha_1. \tag{23}$$

After a change of variable from $r_T$ to $Z_T$ defined above, the radial integral of eq. (1) can be evaluated by again using eq. (20). (In evaluating this integral it is imperative to realize that the radial field variation of the modes is a function only of the magnitude of the azimuthal mode number, $|\alpha|$.) This evaluation produces two cases, which must be treated separately:

(*i*) $\alpha_1$ and $\alpha_2$—both the same sign

(*ii*) $\alpha_1$ and $\alpha_2$—different sign.

For both cases the final integral over the dummy variable $x$, introduced by the Laguerre-Gaussian integral identity [eq. (20)], can be evaluated by using the relationship[14]

$$L_n^{-\delta}(x) = (-1)^\delta x^\delta \frac{(n-\delta)!}{n!}L_{n-\delta}^\delta(x) \tag{24}$$

to write the integral in the form

$$I = \int_o^\infty y^{p+1} e^{-\alpha y^2} L_m^{p-a}(\alpha y^2) L_n^a(\alpha y^2) J_p(x \cdot y) dy, \tag{25}$$

which is a well-documented integral giving

$$I = (-1)^{m+n}(2\alpha)^{-(p+1)} x^p e^{-x^2/4\alpha} L_m^{a-m+n}(x^2/4\alpha) \cdot L_n^{p-a+m-n}(x^2/4\alpha). \tag{26}$$

In both Refs. 15 and 16, the formula is slightly incorrect due to a reversal of the subscripts of the Laguerre polynomials. The formula of Refs. 15 and 16 is seen to be incorrect by noting that eq. (25) above is a Hankel transform of order $p$, so that, by applying the inverse transform to the result, one should expect to obtain the original function, i.e., the argument of the original Hankel transform. The error persists in Ref. 15 because it used Ref. 16 as its source for this result. The use of this relationship then provides the results in Table I.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 5, May–June 1983
*Printed in U.S.A.*

# Experimental Results of 20-Mb/s FSK Digital Transmission on 4-GHz (TD) Radio

By Y. Y. WANG

(Manuscript received June 9, 1982)

*This paper presents the results of tests of 20-Mb/s frequency-shift-keying digital transmission on the 4-GHz TD radio system. On the basis of the test results, the performance of the 20-Mb/s system is projected to be satisfactory to support long-haul digital services. The system employs a 20-Mb/s terminal that multiplexes 12 signals at the first level of digital signal hierarchy (DS-1) (1.5 Mb/s) into a 10-Mbaud, 4-level signal to be transmitted by the Bell System standard 4-GHz (TD) FM microwave radio system. The maximum distance of a digital regeneration span in normal operation is limited by the intermodulation noise to approximately 10 typical hops of TD radio. The 20-Mb/s TD radio system uses the standard frequency-diversity protection switching system, which was designed for analog message service. A fundamental system trade-off is, therefore, the choice of switch threshold: long periods of error-free transmission interrupted by infrequent error bursts due to switch transients versus an occasional low background error rate with less frequent switch transients. We concluded that the protection switch threshold of a 1500-message-circuit channel is suitable for 20-Mb/s TD radio channels.*

## I. INTRODUCTION

The growth of *Dataphone** Digital Service[1] (DDS)† and the needs of new services such as *Picturephone*‡ Meeting Service[2] (PMS) require a substantial increase in digital long-haul transmission capacity. The existing long-haul digital facilities, mainly Data-Under-Voice[3] (DUV), are near exhaustion in many areas. Furthermore, some existing frequency modulation (FM) radios will be replaced by the single-sideband

* Service mark of AT&T.
† Acronyms and abbreviations are defined in the Glossary at the back of this paper.
‡ Service mark of AT&T.

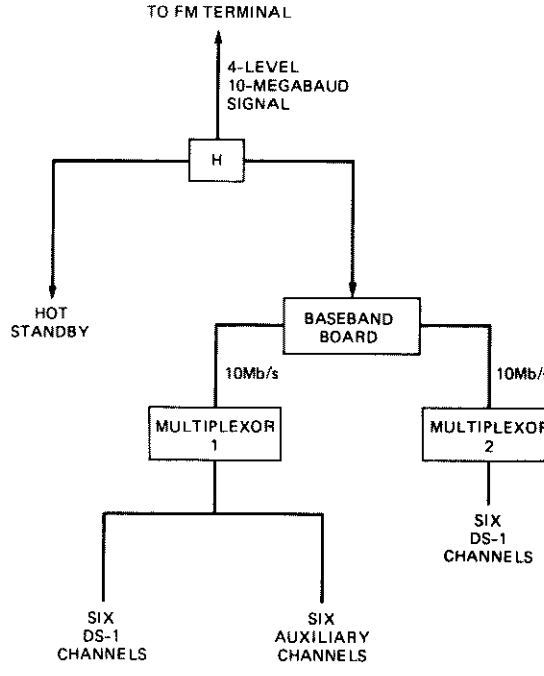



Fig. 1—The 20-Mb/s terminal.

AR-6A[4] radio to increase analog transmission capability. Such replacement reduces the number of DUV channels. New technologies, such as 45-Mb/s transmission on TD radio[5,6] (TD-45A[7,8]) and long-haul digital radio at 4 GHz, will not be available until 1983 and beyond. During the interim, the 20-Mb/s TD system* will be used to provide long-haul digital connectivity.

AT&T Long Lines is deploying an approximately 3000-mile, 20-Mb/s TD network in 1981 and 1982 for DDS application. By the end of 1982, the 20-Mb/s network will have 70 terminals providing 159 digroups. This number of DDS digroups will double the capacity of the 1980 DDS network. The 20-Mb/s terminal has other applications in the Bell System. For example, a portable microwave radio facility can be set up quickly to work with the 20-Mb/s terminals to carry digital services on a temporary basis.

The 20-Mb/s terminal is capable of multiplexing up to 12 asynchronous signals at the first level of digital signal hierarchy (DS-1) into a 9.856-Mbaud, 4-level signal (see Fig. 1). This 4-level signal can be transmitted on the long-haul microwave network using standard FM

* Using the VIDAR DM-12A 20-Mb/s terminal.

terminals and a dedicated 20-MHz bandwidth radio channel suitable for 1500-message-circuit loading. Recommended maximum digital regeneration span is ten TD radio hops (about 250 miles). The existing 100A[9] or 400A[10] protection switch equipment will protect against propagation fading and radio equipment failures.

Following a summary of the test results in Section 2, the test system configuration and the deployment of test equipment are discussed in Section 3. Section 4 records six tests in the absence of multipath fading, e.g., the jitter performance test, fade margin measurements, and the protection switch compatibility test. Section 5 discusses the performance under the multipath fading condition. The projected performance of the 20-Mb/s TD system for DDS and PMS is discussed in Section 6.

Subsequent to these tests, we found that the preferred power level at the input to the Frequency Modulation Transmitter (FMT) could be reduced by 2 dB, from −14.9 dBm to −16.9 dBm. Combined with the use of a narrower band receiver filter, this power reduction improves the adjacent channel interference and the intermodulation noise performance significantly without degrading the performance of the 20-Mb/s channel. The results presented in this report have been adjusted to reflect the reduced drive level and the tighter receiver filters that are being implemented.

## II. SUMMARY

Tests of a 20-Mb/s terminal on a 12-hop, 188-mile TD radio loop were conducted in New Jersey from July to November, 1980. The main results are as follows:

(*i*) It satisfactorily transmitted the digital signal over 12 hops of TD radio without baseband digital regeneration.

(*ii*) The 20-Mb/s terminal showed satisfactory jitter performance.

(*iii*) The system is essentially error-free during normal propagation and operating conditions. System performance as tested in New Jersey is satisfactory for DDS and PMS.

(*iv*) The required cochannel Carrier-to-Interference Ratio (CIR) into a desired 20-Mb/s TD channel is 25 dB at the protection switching point.

(*v*) A switch transient of the 100A frequency-diversity protection switch causes a 6 to 30 millisecond burst of errors. The switch threshold of a 1500-message-circuit channel is suitable for the 20-Mb/s TD channel.

Despite the fact that the protection switch transients will cause transmission errors, the tests confirmed that the use of the 20-Mb/s terminal with the TD radio system can meet long-haul transmission requirements.

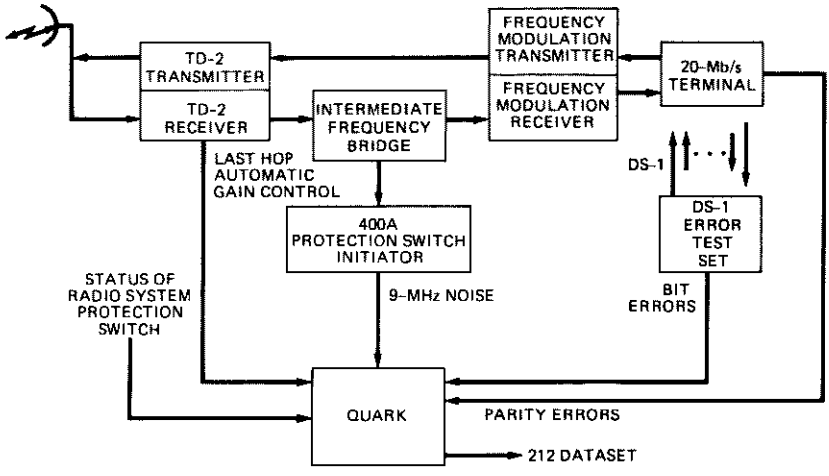



Fig. 2—20-Mb/s TD radio system: test arrangement in Freehold, NJ.

## III. TEST DESCRIPTION

The 20-Mb/s TD radio system under test consists of a 20-Mb/s terminal, a 4A FM terminal, a 12-hop loop of TD-2 radio, and a 100A frequency-diversity protection switch (Section 3.1). Two bit-error-rate test sets (Bowmar), an errored-bit accumulator, a 400A protection switch initiator, and a Quantizer, Analyzer, and Record Keeper (QUARK)[11] were used to monitor and to record the performance statistics (Fig. 2).

### *3.1 System configuration*

The 20-Mb/s terminal multiplexes six even-numbered and six odd-numbered DS-1 (1.544 Mb/s) channels into two 10-Mb/s rails and then encodes the two rails into a 4-level, 10-Mbaud baseband signal. The 4-level, 10-Mbaud baseband signal is connected to an FM terminal that provides Frequency Shift Keying (FSK) modulation and demodulation. The FM terminal is then connected to the TD-2 radio via Intermediate Frequency (IF) cables. The baseband power level at the input to the FMT and the output of the Frequency Modulation Receiver (FMR) were −16.9 dBm and −0.9 dBm, respectively.

The test radio route consisted of two 2-way, 4-GHz radio channels connecting Freehold (FH) and Cherryville (CH), NJ, as shown in Fig. 3. All radio units used in the test were TD-2, retrofitted with solid-state microwave generators. The radio channels are suitable for 1500-message-circuit loading.

### *3.2 Deployment of test equipment*

The switch initiator of a frequency-diversity protection system de-

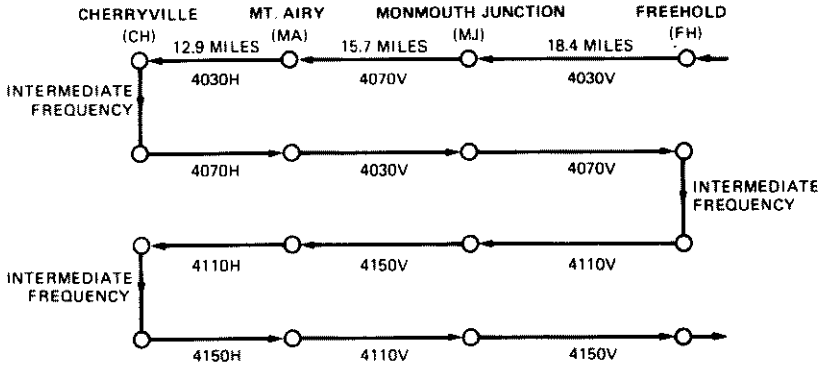



Fig. 3—Frequencies (in megahertz) of TD radio channels for a digital transmission experiment in New Jersey.

tects the noise power in a 1.74-kHz bandwidth centered at 8.9 MHz at the end of a switch section to determine if a working channel needs protection. We therefore installed a 400A protection switch initiator in the receiving IF path to measure the amount of accumulated channel fading over the entire 12-hop route (Fig. 2). For convenience, the power in the 1.74-kHz bandwidth centered at 8.9 MHz will be referred to as the 9-MHz (slot) noise power in the rest of the text.

### *3.3 Data acquisition system*

A QUARK was installed in the Freehold station to record the Automatic Gain Control (AGC) voltage of the last receiving main amplifier and the 9-MHz slot noise of the 12-hop FM radio route, switch activities, errors in two DS-1 channels, and the parity errors in data rail one (9.8 Mb/s, 3672 bits/frame). The statistics of various inputs were accumulated in the QUARK memory. The relationships among DS-1 errors, parity errors, 9-MHz noise, and protection switch activities were studied in terms of statistics of simultaneous events as collected by the QUARK.

## IV. TESTS IN THE ABSENCE OF PROPAGATION FADING

Tests were performed to characterize the system performance under normal operating conditions, as well as under controlled stressing conditions. These tests and their results are described in the following sections.

### *4.1 Baseband spectrum*

Figure 4 shows the received baseband spectra of the 20-Mb/s signal measured at FMR-OUT with and without the 12-hop radio loop. The digital spectrum is down about 15 dB at 6 MHz. The 9-MHz noise

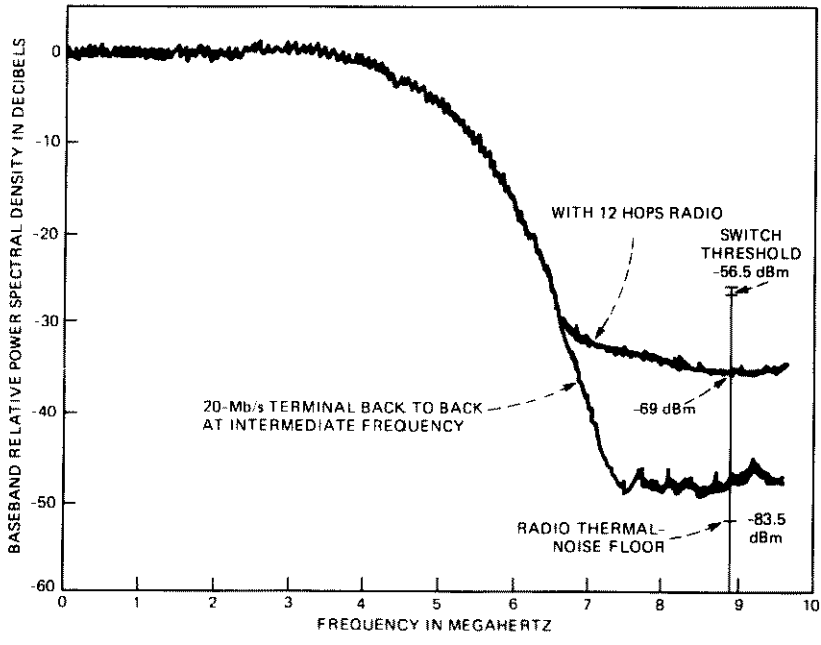



Fig. 4—Baseband spectrum.

power was −69 dBm and −83.5 dBm, with and without the 20-Mb/s modulation, respectively. The 14.5-dB difference represents the 12-hop accumulated intermodulation noise, which dominates the 9-MHz slot noise.

### *4.2 S/N at baseband versus BER at DS-1 level*

Baseband noise was injected before the baseband receiver filter to stress the 20-Mb/s transmission. The main objective was to study the Bit Error Rate (BER) versus baseband signal-to-noise ratio (s/n) relationship.

Figure 5 shows the BER performance of the even- and odd-numbered DS-1 channels of the terminal. The odd-numbered channels perform better than even-numbered channels by 1 dB in s/n for equal BER. This difference is due to the circuit design of the terminal. The 20-Mb/s signal consists of two 10-Mb/s rails. At the terminal receiver, the odd-numbered DS-1 channels are derived from a 10-Mb/s rail, which is decoded from the plus-minus sign decision of the received 10-Mbaud, 4-level signal. The even-numbered DS-1 channels are derived from the other 10-Mb/s rail, which is decoded from the amplitude threshold decision of the received 4-level signal. The plus-minus sign decision is more robust than the amplitude threshold decision by 1 dB

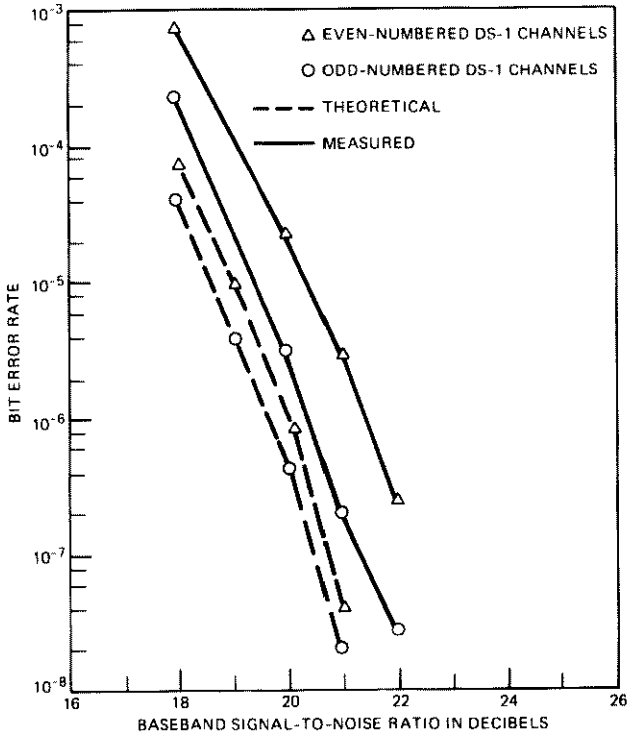



Fig. 5—Bit error rate vs. $s/n_{BB}$ intermediate frequency loopback.

in s/n, as measured, and by 0.5 dB, as theoretically predicted. The theoretical relationship is derived in the appendix. Thus, throughout this study, a discussion of error performance in terms of an even-numbered DS-1 channel implies a conservative (lower) bound on the digital error performance.

Figure 6 shows the measured and calculated BER of the odd-numbered DS-1 channel versus baseband s/n. The 12-hop TD radio degrades the performance of the odd-numbered DS-1 channel by less than 0.5 dB. The effect of 12 hops of TD radio on even-numbered DS-1 channels is practically indiscernible. The maximum regeneration interval (ten TD hops were recommended for the field) is imposed, therefore, by intermodulation noise at the 9-MHz noise slot (Section 4.1), rather than by the digital transmission impairments.

### *4.3 Jitter performance*

Jitter performance of the 20-Mb/s terminal satisfies the requirements of the existing digital network. Under normal operating conditions, the amount of output jitter among DS-1 channels was uniform

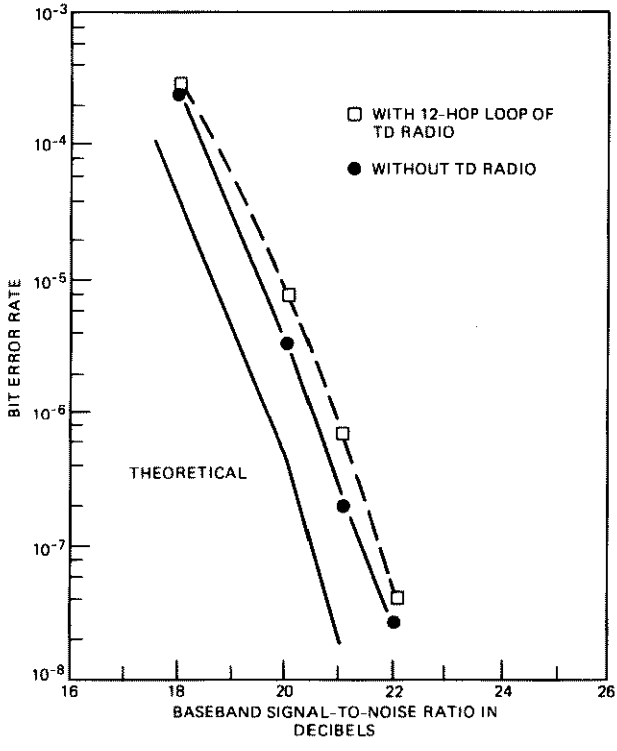



Fig. 6—Bit error rate vs. $s/n_{BB}$ for odd-numbered DS-1 channels.

and comparable to that of the office Quasi-Random-Signal-Source (QRSS), the standard DS-1 signal. In fact, the terminal is effectively a de-jitterizer. The amount of jitter in an output signal was about half that of the input signal, as shown in Table I. Furthermore, digital error performance of two DS-1 channels was compared when the Radio Frequency (RF) signal power was severely attenuated. The channel that took a clean office QRSS as the input consistently made fewer errors than the one that took a jittered source with 11- to 13-percent rms jitter. However, the difference was so small that it was indistinguishable in terms of the fade margin.

#### *4.4 Flat fade margin*

The Bell System microwave radio plant was engineered to have adequate flat fade margins to meet the outage objective of allowing less than 0.01-percent outage for all causes over one-way, 4000-mile transmission. The outage of a digital system comprises the time when the one-second-averaged BER exceeds $1 \times 10^{-3}$. This objective need

Table I—Jitter performance

| | Input to the 20-Mb/s Terminal | | Output of the 20-Mb/s Terminal | |
|---|---|---|---|---|
| | Percent of Jitter (rms) | Percent of Jitter (Peak to Peak) | Percent of Jitter (rms) | Percent of Jitter (Peak to Peak) |
| 1 loop* | 13 to 14 | 60 to 75 | 5 to 6 | 25 to 30 |
| 2 loops | 9 to 11 | 50 to 65 | 4 to 5 | 15 to 25 |
| 3 loops | 14 to 16 | 75 to 120 | 5 to 7 | 25 to 40 |
| 4 loops | 15 to 18 | 90 to 130 | 7 to 9 | 35 to 50 |

* Each loop consists of two complete M13 passes and four hops of digital radio.

Table II—Flat fade margin

| Radio Hop | $F_s$ (dB) | $F_d$ (dB) | $F_d$-$F_s$ |
|---|---|---|---|
| FH-MJ | 43.0 | 43.0 | 0 |
| MJ-MA | 39.3 | 42.7 | 3.4 |
| MA-CH | 44.0 | 46.0 | 2.0 |
| CH-MA | 45.0 | 42.8 | −2.2 |
| MA-MJ | 41.5 | 42.5 | 1.0 |
| MJ-FH | 41.5 | 40.0 | −1.5 |
| FH-MJ | 39.5 | 41.5 | 2.0 |
| MJ-MA | 39.5 | 42.8 | 3.3 |
| MA-CH | 40.5 | 44.0 | 3.5 |
| CH-MA | 40.0 | 42.3 | 2.2 |
| MA-MJ | 37.8 | 39.5 | 1.7 |
| MJ-FH | 38.0 | 40.0 | 2.0 |

$F_s$ = switch point fade margin.
$F_d$ = fade margin to $10^{-6}$ BER.

not be met on a per-hop basis, because of the existence of severe interference conditions at some junction radio stations, as long as the prorated objective on a per switch section basis was satisfied.

This test measured the required amount of RF attenuation at each radio transmitter to reach the protection switch threshold, which is −56.5 dBm noise power at the 9-MHz slot. This test showed that the fade margin against a BER of $1 \times 10^{-3}$ always exceeded the switch point fade margin, indicating that the 20-Mb/s TD system can be engineered to meet the facility outage objective. In fact, even the margin against a BER of $1 \times 10^{-6}$ was generally found greater than the corresponding switch point fade margin. The $10^{-6}$ margin is of interest because the number of seconds at a BER of $1 \times 10^{-6}$ at the DS-1 level is an estimate of the number of errored seconds (ES), which is an often referenced service performance parameter.

Table II summarizes the measured switch point fade margins and the $10^{-6}$ margins of the test system. There were two hops where the BER could exceed $1 \times 10^{-6}$ before the radio channel would request protection. Since this condition is expected at a few junction stations of the network and we wanted to have a "typical" TD route to conduct this test, no effort was invested to identify the cause and eliminate the situation.

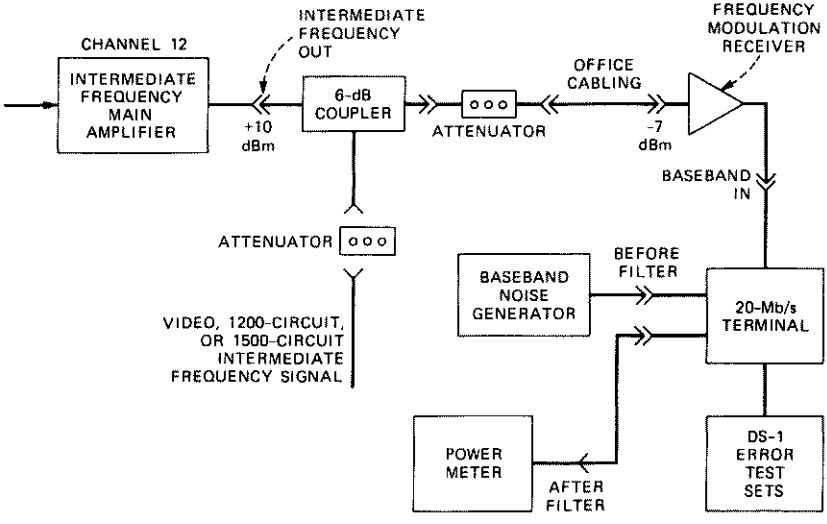



Fig. 7—CIR vs. s/n test setup.

### *4.5 Effect of cochannel interference*

To study the impact of severe cochannel interference on the 20-Mb/s TD channel, an interfering signal was injected at IF. Three kinds of interference signals [i.e., 1200-circuit message, 1500-circuit message, and video (color bar)] were used. The test setup is illustrated in Fig. 7. Figure 8 shows the results for a fixed BER of $1 \times 10^{-6}$. The color-bar video signal was observed to be the most interfering; the 1500-circuit message was the least interfering among the three. This is due to the higher concentration of spectral energy near the carrier in certain signals. The baseband filter does not reduce this type of signal appreciably.

At the protection switch initiation point, the s/n versus CIR relationship was also measured, as shown in Fig. 8. In the region where the faded CIR exceeds 25 dB, as the s/n decreases, a protection switch will be initiated before BER degrades to $1 \times 10^{-6}$. However, if the faded CIR were less than 25 dB, thermal noise in the radio channel could cause BER to exceed $1 \times 10^{-6}$ before the switch point. Therefore, severe cochannel interference reduces the effectiveness of the protection switch and degrades the digital error performance.

### *4.6 Switch system compatibility*

For this portion of the test only, a two-way radio channel from Freehold to Cherryville was included in the 100A switching system between these locations. The connection thus had two one-way switch

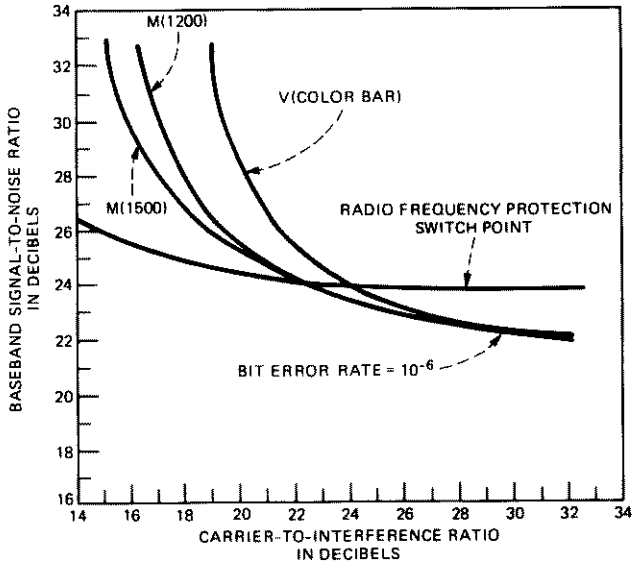



Fig. 8—CIR vs. $s/n_{BB}$ for even-numbered DS-1 channels.

sections, each consisting of three hops of TD radio. Test results include:

(*i*) A switch cycle (to and back from the protection channel) consists of two transients, i.e., two interruptions in data transmission. More than one switch cycle in a second is possible, and the time span from initial loss of signal to the resynchronization of the receiving part of the 20-Mb/s terminal is approximately 30 milliseconds.

(*ii*) No special modification to the 100A switch equipment is necessary. The frequency-diversity protection switch is compatible with the 20-Mb/s TD radio system.

## V. PERFORMANCE UNDER MULTIPATH FADING CONDITION

### *5.1 Amount of multipath fading activity*

During the period from July 10, 1980 to August 13, 1980, the impact of multipath fading over the test route was studied. Figure 9 shows the statistics of the 9-MHz slot noise during this period. The measured distribution of 9-MHz noise displayed the inverse slope of 10 dB per decade of probability (the $L^2$ law[12]). This slope is a well-known characteristic of multipath fading for unprotected radio. The number of events was greater than the number of seconds when the 9-MHz noise exceeded a given abscissa. This observation suggests that multiple switch requests could occur in a second. A switch transient causes errors, and, therefore, frequency-diversity protection is expected to

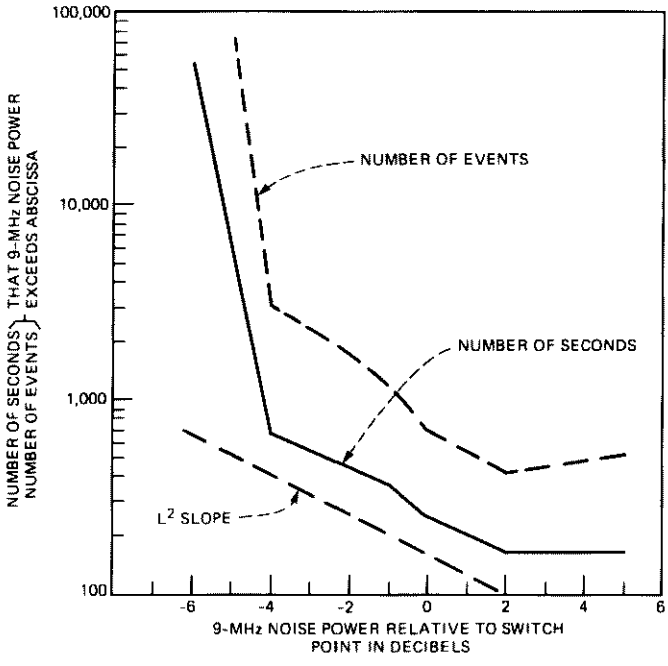



Fig. 9—Distribution of time and events of 9-MHz noise power. The time period monitored was July 10, 1980 through August 13, 1980.

offer limited reduction in the number of errored seconds during multipath fading periods.

The fading statistics of a single hop from Monmouth Junction (MJ) to Freehold are shown in Fig. 10. The solid curve in Fig. 10 also has an inverse slope of 10 dB per decade of probability. The engineering model[12] predicted 166 seconds below 30 dB for this hop during a heavy fading month. We recorded 115 seconds, which is about two thirds of a heavy fading month. Hence, we did experience a fair amount of multipath fading during the test.

### *5.2 Digital error performance*

The measured distribution of BER of an even-numbered DS-1 channel is shown in Fig. 11. There were 432 errored seconds (ES); 263 of them have BER exceeding $10^{-3}$. The BER statistics for simultaneous errored seconds of even-numbered and odd-numbered DS-1 channels were also plotted in Fig. 11. They differ in the number of low-BER ES, as expected (see Section 4.2 and the appendix). The number of ES in even-numbered DS-1 channels is approximately 25 percent more than the ES in odd-numbered DS-1 channels.

The data format of the terminal multiplexor contains one parity bit

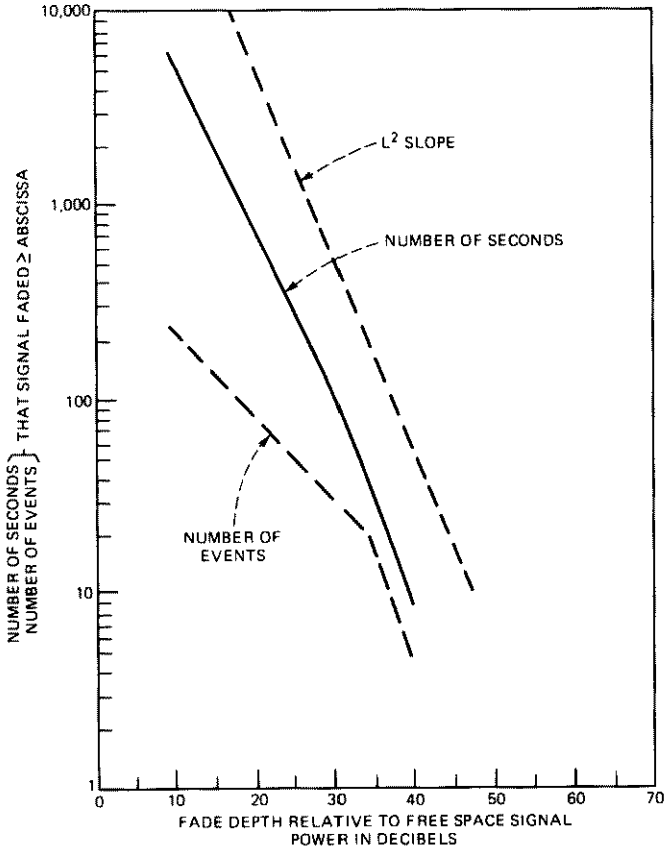



Fig. 10—Distribution of time and events of received signal power. The time period monitored was July 10, 1980 through August 13, 1980.

in a 3672-bit frame. Fault-detection algorithms internal to the terminal and a real-time error-performance monitoring plan to be used by the AT&T Long Lines operations personnel depend on the detection of the parity violations. The distribution of parity violations in a 9.8-Mb/s rail (which feeds six DS-1 channels) can be found in Fig. 12. Those parity violation seconds occurring simultaneously with the ES for the information bits of one even-numbered DS-1 channel are also plotted in Fig. 12. The difference between the two was found to be insignificant. The parity violation seconds offers an effective representation of the real DS-1 ES.

### *5.3 Relationship between digital errors and 9-MHz noise*

Figure 13 shows the distribution of BER of an even-numbered DS-1 channel and the portion of the errored seconds with 9-MHz noise

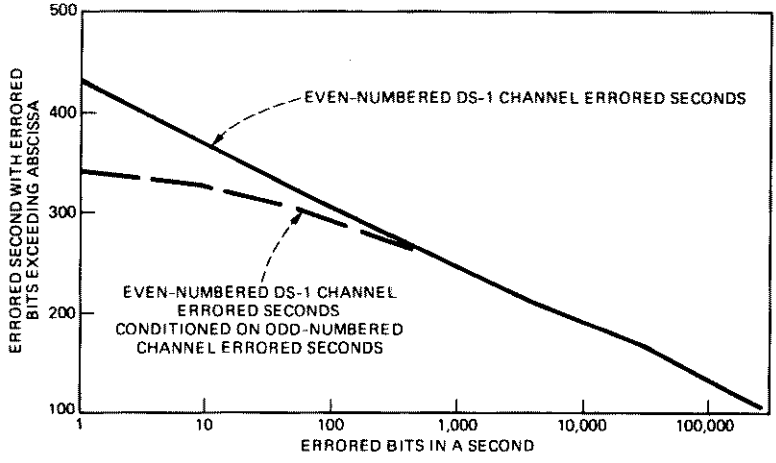



Fig. 11—DS-1 errored-second statistics for the time period July 10, 1980 through August 13, 1980.

exceeding switching point from August 14, 1980 to September 20, 1980. About 10 percent of the errored seconds occurred before the 9-MHz noise reached the switching threshold. All ES with more than 100 errors occurred at the same time the 9-MHz noise exceeded the switch point. Thus, these errored seconds are likely to be accompanied by switch activities in the plant environment.

### *5.4 Impact of frequency-diversity protection*

Based on the assumption that the protection channel is always available for error-free transmission, this section will show that:

(*i*) The switching threshold of a 1500-message-circuit loaded channel is suitable for a 20-Mb/s TD channel, and

(*ii*) During multipath fading periods, the protection switch would have offered little improvement in terms of ES reduction owing to the frequent switching activities.

Under the assumption of perfect frequency-diversity protection, those errored seconds that occurred while the 9-MHz noise exceeded the switching threshold would have been prevented. The exception would be those seconds in which the 9-MHz noise passes through the switching threshold. This is because the actual switch transfer causes an errored second. Based on the observation that a switch cycle could be completed within a second (Sections 4.6 and 5.1), we estimate the number of seconds containing switching activities by the number of seconds in which the maximum value of the measured 9-MHz noise in a second exceeded the switching threshold. Hence, if the test route had been frequency-diversity protected, we would have $X$ ES, where

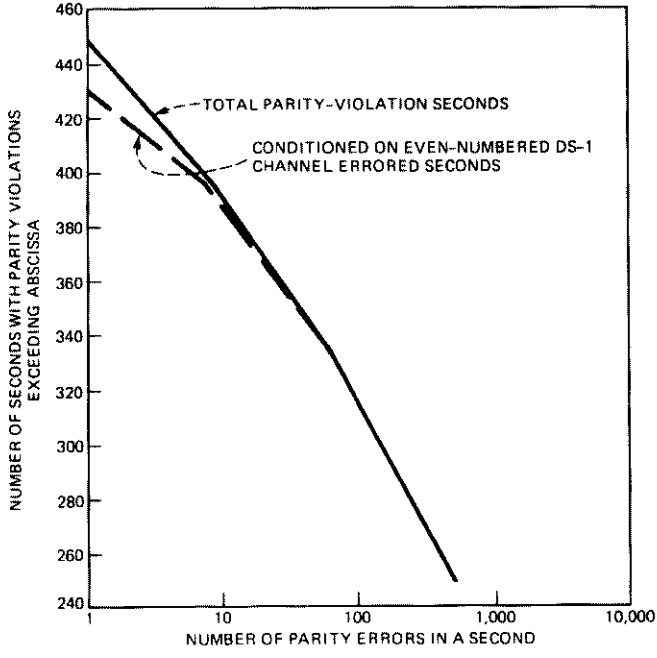



Fig. 12—Parity-violation-second statistics for the time period July 10, 1980 through August 13, 1980.

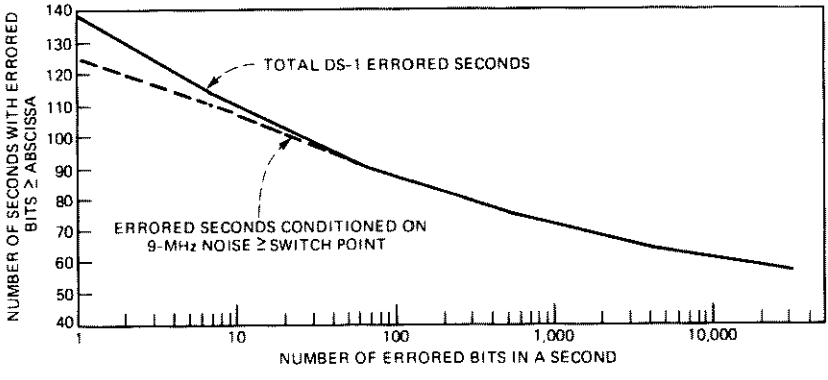



Fig. 13—DS-1 errored-second statistics for the time period August 14, 1980 through September 20, 1980.

$X$(at a given threshold) =

(Total ES) − (ES conditioned on 9-MHz noise power ≥ threshold)

+ (number of seconds with maximum 9-MHz noise power

≥ threshold).

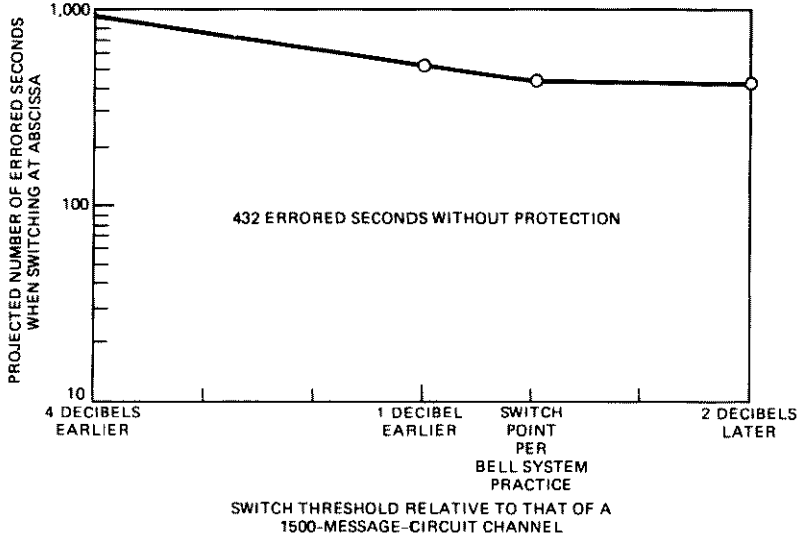



Fig. 14—Sensitivity of switch threshold to errored-second performance projection based on data collected July 10, 1980 through August 13, 1980.

Figure 14 shows the projected number of ES as a function of switching threshold. The projected number of ES decreases as the switch threshold decreases until the switch threshold reaches the level of a 1500-message-circuit loaded channel. Since the 20-Mb/s channel will coexist with 1500-message-circuit loaded channels in the plant, it is desirable from operations and maintenance considerations to use the switch threshold of a 1500-message-circuit loaded channel for a 20-Mb/s TD channel.

## VI. PERFORMANCE PROJECTION FOR DDS AND PMS

A properly engineered 20-Mb/s TD system can meet the long-haul outage and quality objectives for DDS and PMS. The long-haul outage objective[13,14] for DDS or PMS is the same as that for Message Telecommunications Service (MTS). Engineering guidelines for TD radio were developed to meet the MTS outage objective and, therefore, those for DDS and PMS as well.

The following demonstrates the ability of the 20-Mb/s TD system to meet the long-haul quality objectives for DDS and PMS.

### *6.1 Meeting the DDS quality objective*

#### *6.1.1 The DDS quality objective*

The quality objective of the DDS transmission design requires 99.5-percent error-free-seconds at DS-0 (56-kb/s rate) for 4000-mile, one-

way, end-to-end transmission. The test results showed that 90 percent of the errored seconds would be associated with switch transients, which cause errors in all 23 DS-0 channels of a DS-1 signal. We, therefore, assume as a worst case that the DDS quality objective for the 20-Mb/s TD system is the same at DS-0 and DS-1 levels. The objective is thus to have less than 0.5-percent ES for a one-way, 4000-mile transmission.

#### *6.1.2 Generalized performance projection*

The majority (ninety percent in this test) of ES of a properly engineered 20-Mb/s TD radio system are expected to be caused by protection switch transients. Therefore, to project the performance of a 20-Mb/s TD system, the statistics of switch activities in a 4000-mile TD route must be considered.

Every switch system (e.g., 100A, 400A) has a switch register that counts the number of automatic switch completions. Any manually forced protection switches for routine maintenance are not included in these counts. AT&T Long Lines records the number of switches on a weekly basis. These records of switch completions from June 16, 1979 through June 7, 1980 on five TD routes (New York City to Boston, New York City to Philadelphia, Philadelphia to Silver Spring, Pittsburg to Silver Spring, and Chicago to Kalamazoo) suggest that there are 13,500 switch completions per year for an average one-way radio channel for multipath fading protection in a 4000-mile TD radio system. There could be two errored seconds associated with a switch cycle; therefore, 27,000 ES due to multipath fading are expected annually.

The one-year switch register data also revealed that, in addition to the switchings due to multipath fading, there were protection switches, called the background switch activities, which were attributed to craft activities, hardware problems, and other causes. These background switches are of a transient nature. A switch cycle lasts much less than a second. According to the same database, this type of switch amounted to 0.03 switch completion per mile per week per channel on average. Thus, we project 6240 ES due to background switches.

Table III summarizes the projected number of ES in a year due to all causes. The total number of ES is 34,078 which is 0.11 percent of a year. This projection compares favorably with the objective of 0.5 percent and shows that the 20-Mb/s TD system performance is satisfactory for DDS.

### *6.2 Meeting the PMS quality objective*

We have observed essentially error-free performance without fading. Multipath fading seldom occurs during an 8 am to 9 pm business day,[12]

Table III—Projection of errored seconds caused by switching in a one-way, 4000-mile TD route in a year

| Cause | Errored seconds |
|---|---|
| FMT or 20-Mb/s terminal failure | 18 |
| Routine maintenance | 720 |
| Radio failure | 100 |
| Background switch activity | 6240 |
| Multipath fading | 27,000 |
| Total | 34,078 |
| | =0.11 percent/year |

when most PMS calls will take place. Therefore, based on the rate of background switches, as discussed in the previous section, we project that the 20-Mb/s system performance is satisfactory for PMS.

## VII. ACKNOWLEDGMENT

Many colleagues at Bell Laboratories and AT&T Long Lines offered their prompt assistance at every stage of this project. I am particularly grateful to L. A. Dietrich and S. H. Lin for their participation and technical guidance throughout the project. The QUARK was designed by T. G. Szekeres and G. A. Zimmerman.

## REFERENCES

1. Special issue on "Digital Data System," B.S.T.J., *54,* No. 5 (May–June 1975).
2. H. S. London and D. B. Menist, "A Description of the AT&T Video Teleconferencing System," Conf. Rec. Nat. Telecommun. Conf., *3* (November 1981), p. F5.4.
3. R. R. Grady and J. W. Knapp, "1A Radio Terminals Put Data Under Voice," Bell Lab. Rec., *52* (May 1974), pp. 161–6.
4. J. Gammie and J. P. Moffatt, "The AR-6A Single Sideband Microwave Radio System," Conf. Rec. Int. Conf. Commun., 1981, pp. 3.5.1–.5.
5. A. A. Roetken, K. D. Smith, and R. W. Friis, "The TD-2 Microwave Radio System," B.S.T.J., *30,* No. 4 (October 1951), pp. 1041–77.
6. S. D. Hathaway, W. G. Hensel, D. R. Jordan, and R. C. Prime, "TD-3D Microwave Radio Relay System," B.S.T.J., *47,* No. 7 (September 1968), pp. 1143–88.
7. R. T. Cooney and S. A. Harvey, "TD-45A, a System for 45-Mb/s Digital Transmission over the TD Radio Network," GLOBECOM 1982, pp. D3.6.1–4.
8. W. P. Jaeger, P. L. Penny, and S. L. Vonrump, "Measured Performance of 45-Mb/s Transmission over the Bell System 4-GHz Radio," GLOBECOM 1982, pp. D3.7.1–4.
9. H. D. Griffiths and J. Nedelka, "100A Protection Switching System," B.S.T.J., *44,* No. 10 (December 1965), pp. 2295–336.
10. H. D. Griffiths, "The 400A Microwave Radio Protection Switching System," Conf. Rec. Int. Conf. Commun., 1973, pp. 28.12–.16.
11. N. F. Dinn, "Facilitating the Use of Digital Radio," Bell Lab. Rec., *57* (May 1979), pp. 135–41.
12. W. T. Barnett, "Multipath Propagation at 4, 6, and 11 GHz," B.S.T.J., *52,* No. 2 (February 1972), pp. 321–61.
13. A. Vigants and M. V. Pursley, "Transmission Unavailability of Frequency-Diversity Protected Microwave FM Radio Systems Caused by Multipath Fading," B.S.T.J., *58,* No. 8 (October 1979), pp. 1778–96.
14. W. Y. S. Chen, "Estimated Outage in Long-Haul Radio Systems with Protection Switching," B.S.T.J., *50,* No. 4 (April 1971), pp. 1455–85.
15. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., *Principles of Data Communication,* New York, NY: McGraw-Hill, 1968.

## APPENDIX

### *Predicted Error Performance With Ideal Channel Plus Gaussian Noise*

This appendix approximates a set of relationships between a 1-second bit error rate at DS-1 level and the baseband signal-to-noise ratio based on our understanding of the 20-Mb/s terminal and an idealized FM channel with Gaussian noise.

There are two multiplexor units in the main/working portion of a terminal. The number 1 multiplexor unit accepts data from the six odd-numbered DS-1 channels plus the six auxiliary (AUX) channels. The number 2 multiplexor unit accepts data from six even-numbered DS-1 channels. Each multiplexor combines its inputs into a single data-bit stream of 9.856 MHz. These two bit streams form binary groups: the first digit in the binary group is taken from the number 1 multiplexor, the second from the number 2 multiplexor. These binary groups are Gray coded and are then converted into a single 4-level signal. It has been verified in the laboratory that the four levels at the transmitter output are indeed equally spaced.

The decoder at the terminal receiver performs the inverse operation. The probabilities of error for even- and odd-numbered DS-1 channels separately can be derived in a manner similar to the analysis of Lucky[15] and others under the following assumptions:

1. The amplitudes of the four symbols are equally likely to assume any of the four equally spaced values $\pm d$ and $\pm 3d$. Symbols occurring at different times are independent.

2. The bit error performance of the terminal depends on s/n but is insensitive to the spectral shape of noise. (This has been verified in the laboratory.) We assume that the additive noise is Gaussian.

3. The pulse shaping $X(w)$ is raised cosine with 50 percent roll-off:

$$X(w) = \begin{cases} T & 0 \leq w \leq \dfrac{\pi}{2T} \\ \dfrac{T}{2}(1 + \sin wT) & \dfrac{\pi}{2T} \leq w \leq \dfrac{3\pi}{2T}, \end{cases}$$

where $T$ is the baud interval.

The terminal puts all pulse shaping at the transmitter. Therefore, the signal power input to the FM channel is:

$$p_s = \frac{\overline{a^2}}{2\pi T} \int_{-\infty}^{\infty} |X(w)|^2 dw$$

$$= \frac{35}{8} d^2,$$

where $\overline{a^2}$ is the average symbol power, $\overline{a^2} = 5d^2$.

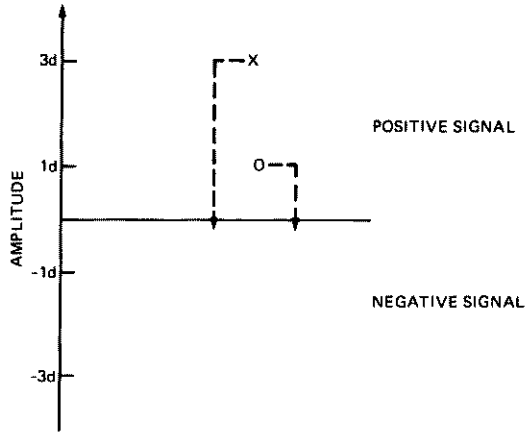



Fig. 15—Error occurrence in an odd-numbered DS-1 channel.

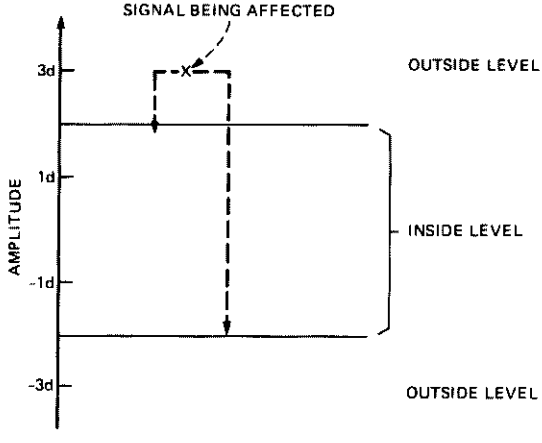



Fig. 16—Error occurrence of an outside-level signal in an even-numbered DS-1 channel.

The receiver detects the signal levels and places slicing levels at 0 and $\pm 2d$. An error occurs when the noise at a sampling time pushes the received signal amplitude (voltage) across the slicing levels. The slicing level drift due to noise is assumed insignificant. The probability of a signal being at one of the outside two levels equals that of being at one of the inside two levels. A signal at the outside level, or inside level, can only cross the zero level in one direction when the noise voltage $|y|$ exceeds $3d$ or $d$, respectively, as illustrated in Fig. 15. Therefore, the probability of error for an odd-numbered channel is:

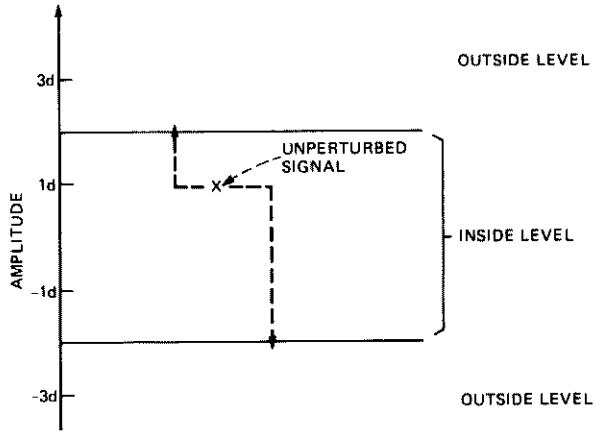



Fig. 17—Error occurrence of an inside-level signal in an even-numbered DS-1 channel.

$$p^{\text{odd}} = \tfrac{1}{2} \cdot \tfrac{1}{2}p(|y| > 3d) + \tfrac{1}{2} \cdot \tfrac{1}{2}p(|y| > d), \tag{1}$$

where $p(|y| > d)$ represents the probability of noise voltage exceeding $d$.

The probability of error for an even-numbered channel, $p^{\text{even}}$, can be derived similarly. The probability of a signal in the outside state is ½. For that signal to be in error, the noise in one direction only with a magnitude $|y|$,

$$5d > |y| > d$$

is required, as illustrated in Fig. 16. The probability of a signal in an inside level crossing a $\pm 2d$ slicing level is $p(|y| > d) - \tfrac{1}{2}p(3d > |y| > d)$, as shown in Fig. 17. Therefore,

$$\begin{aligned} p^{\text{even}} &= \tfrac{1}{4}p(5d > |y| > d) \\ &\quad + \tfrac{1}{2}p(|y| > d) - \tfrac{1}{4}p(3d > |y| > d) \\ &= \tfrac{1}{4}p(|y| > d) - \tfrac{1}{4}p(|y| > 5d) \\ &\quad + \tfrac{1}{2}p(|y| > d) \\ &\quad - \tfrac{1}{4}p(|y| > d) + \tfrac{1}{4}p(|y| > 3d) \\ &= \tfrac{1}{2}p(|y| > d) + \tfrac{1}{4}p(|y| > 3d) \\ &\quad - \tfrac{1}{4}p(|y| > 5d). \end{aligned} \tag{2}$$

This probability is easily computed since the noise at the receiver input is assumed Gaussian. The probability of noise voltage exceeding

$d$ is conveniently expressed in terms of the normal probability integral, $p(|y| > d) = 2Q\left(\frac{d}{\sigma}\right)$, where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2}\, dt$$

and $\sigma$ is the root-mean-square (rms) noise voltage.

The signal to noise power ratio, $p_s/p_N$, can be expressed as

$$\frac{p_s}{p_N} = \frac{35d^2}{8\sigma^2}.$$

Hence, eqs. (1) and (2) become

$$p^{\text{odd}} = \frac{1}{2}\left\{Q\left[3\left(\frac{8X}{35}\right)^{1/2}\right] + Q\left[\left(\frac{8X}{35}\right)^{1/2}\right]\right\}$$

$$p^{\text{even}} = \frac{1}{2} Q\left[\left(\frac{8X}{35}\right)^{1/2}\right]\left\{1 - 2Q\left[5\left(\frac{8X}{35}\right)^{1/2}\right]\right\} + Q\left[\left(\frac{8X}{35}\right)^{1/2}\right] - \frac{1}{2} Q\left[\left(\frac{8X}{35}\right)^{1/2}\right]\left\{1 - 2Q\left[3\left(\frac{8X}{35}\right)^{1/2}\right]\right\},$$

where $X = \frac{p_s}{p_N}$.

A digital computer was used to compute the probabilities of error as a function of the baseband signal-to-noise ratio. Results are discussed in Section 4.2.

## GLOSSARY

| | |
|---|---|
| AGC | Automatic gain control circuitry |
| AM | Amplitude modulation |
| AR-6A | A Western Electric 6-GHz single-sideband AM microwave radio system |
| BER | Bit error rate |
| CIR | Carrier-to-interference power ratio, expressed in dB |
| DDS | *Dataphone*® Digital Service, a synchronous full-duplex digital service at 2.4-, 4.8-, 9.6-, and 56-kb/s rates on point-to-point and multipoint bases |
| DS-0 | Digital signal at the 0th level of the TDM hierarchy, the DS-0 level; a signal at the 64-kb/s rate, the DS-0 rate |

| | |
|---|---|
| DS-1 | Digital signal at the 1st level of the TDM hierarchy, the DS-1 level; a signal at the 1.544-Mb/s rate, the DS-1 rate |
| DUV | Data under voice, a system that provides for the transmission of one DS-1 signal over an FM microwave radio link. (This system is also known as 1A radio digital system.) |
| ES | Errored second; a second that contains at least one errored bit |
| EFS | Error free second |
| FM | Frequency modulation |
| FMT/FMR | Frequency modulation terminal transmitter/receiver |
| FSK | Frequency shift keying |
| IF | Intermediate frequency (70-MHz ± 10-MHz for TD radio) |
| MTS | Message Telecommunications Service |
| PMS | *Picturephone*® Meeting Service, a switched, common-user, interactive visual and audio teleconferencing service offered between two remote conference room locations |
| s/n | Signal-to-noise ratio in dB |
| TD-45A | A system for 45-Mb/s digital transmission over the TD radio network |
| TDM | Time division multiplexing, the process of combining a number of digital signals into a single digital stream by an orderly assignment of time slots |
| TD-2 | A Western Electric point-to-point 4-GHz microwave radio transmission system |

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 5, May–June 1983
*Printed in U.S.A.*

# An Experimental Broadband Imaging Feed

By T. S. CHU and R. W. ENGLAND*

(Manuscript received November 8, 1982)

*Imaging with Fresnel diffraction taken into account is applied to the design of a broadband narrow pattern feed. This feed is not only a basic building block of an imaging beam waveguide, but also essential for an offset reflector antenna of large effective F/D ratio. Furthermore, it can be used as a constant beamwidth radiometer antenna for multifrequency remote sensing. We have built and tested a practical example that consists of an offset ellipsoidal reflector and a corrugated horn. Measured amplitude and phase patterns agree with calculated results, which include truncation effects. Systematic design procedures are obtained for a given feed horn and the required reflector illumination. Necessary and sufficient conditions of the thin lens model are translated into design parameters of an offset ellipsoidal reflector with projected circular aperture. Geometrical relations of the offset ellipsoid and calculations of radiation patterns are described in the appendices.*

## I. INTRODUCTION

The successful performance of an offset dual-reflector antenna often depends upon illumination by a broadband feed with a narrow feed pattern.[1] For example, a broadband corrugated horn with the required pattern could be used, but it would be excessively long for most practical applications. Good illumination can also be achieved by a narrowband offset launcher,[2] which is essentially an offset reflector fed by a relatively short feed horn. Excellent 19/28.5 GHz dual-frequency performance was demonstrated by the Crawford Hill 7-meter antenna using a quasi-optical frequency diplexer to combine two narrowband offset launchers.[1] However, this approach is not usually cost effective, especially for lower frequency systems. In this paper we present design

* Now at AT&T Long Lines.

procedures for *broadband* offset launchers and comparisons between calculated and measured patterns of both amplitude and phase. These feeds represent a considerable simplification of the hardware, and thus improve the prospect for system application of offset dual-reflector antennas.

The following analysis will assume the thin lens approximation for an imaging offset reflector. It will also make use of the frequency insensitivity of the field distribution in the aperture of a corrugated horn.[3,4] The principles of imaging have been discussed previously in connection with frequency-independent far-field beamwidths,[5] and the laws of geometrical optics have been used for the design of imaging reflector arrangements.[6] However, broadband feed illumination for reflector antennas needs not only constant beamwidth, but also constant phase center. The condition for satisfying the latter requirement is obtained in this paper by an interpretation of the additional phase shift[7] due to Fresnel diffraction. The imaging laws of geometrical optics are thus modified to provide a theoretically frequency-independent design of offset-launcher feeds. However, the practical bandwidth will be limited by the corrugated horn and truncation effects. These effects will be examined by numerical calculations and experimental measurements.

The imaging feed discussed here has important potential for application in radio communication and other scientific explorations. Furthermore, the basic properties of single-stage imaging are of vital interest in the design of multistage-imaging beam waveguides.[8] One notes the difficulty of performing pattern measurements of a bulky beam waveguide assembly. Both single- and double-stage imaging are also special cases of a proposed technique[9] for broadband astigmatic compensation in which the image of a corrugated horn through two astigmatic lenses can be designed to produce a specified astigmatic illumination.

Section II will discuss the imaging feed within thin lens approximation. Section III will describe how to translate the design parameters from the thin lens model to a practical offset ellipsoidal reflector with the required projected circular aperture. Section IV will give comparisons between calculated and measured data for both amplitude and phase patterns of an experimental broadband imaging feed. Geometrical relations of an offset ellipsoid and a calculation of radiation patterns will be given in Appendices A and B, respectively.

## II. THIN LENS MODEL

It is well known that the aperture distribution of a corrugated horn with radius $a_0$ is the Bessel function $J_0\ (\alpha\rho/a_0)$.[3,4] The edge field vanishes when $\alpha = 2.405$. This normally occurs at the design frequency

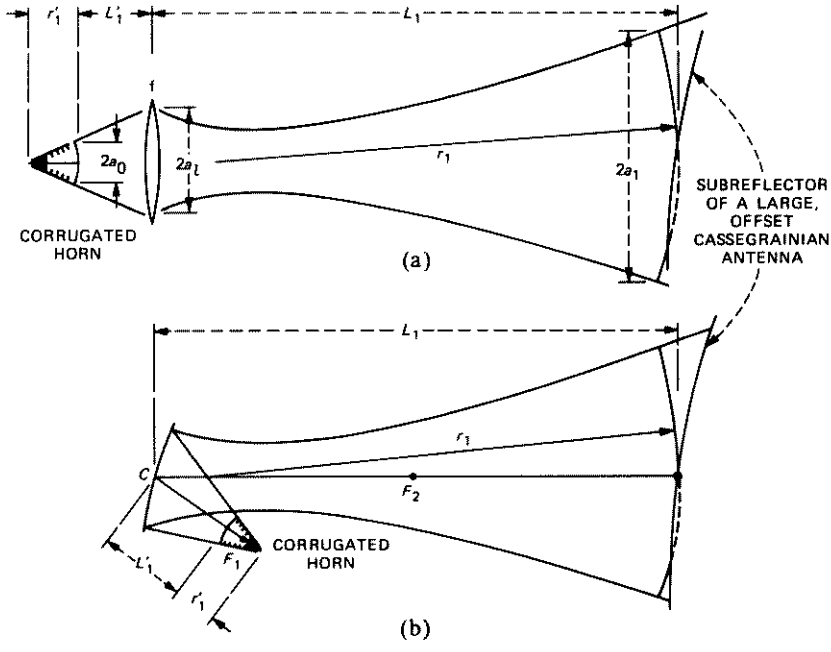



Fig. 1—Sketch of a broadband imaging feed. (a) Thin lens approximation. (b) Offset ellipsoidal reflector.

for resonant corrugations. However, $\alpha$ remains close to the above optimum value over a broad bandwidth if the aperture diameter is about five wavelengths or greater. If a magnified image of the corrugated-horn-aperture distribution is used to illuminate a reflector antenna, the illumination is expected to be a truncated Bessel function. For a specified reflector edge taper of $T$(in dB), the radius $a_r$ of the illumination circle at the reflector aperture should have a value satisfying

$$20 \log_{10} J_o(\alpha a_r / a_1) = -T, \tag{1}$$

where $a_1 = Ma_0$ is the magnified corrugated-horn-aperture radius. The magnification is given by

$$M = \frac{L_1}{L_1'}, \tag{2}$$

where $L_1$ *and* $L_1'$ are the distances of the illuminated reflector and the corrugated horn from the imaging lens, respectively, as shown in Fig. 1.

Now the focal length, $f$, of the imaging lens should obey the thin

lens formula:

$$\frac{1}{f}=\frac{1}{L_1}+\frac{L}{L_1'}. \tag{3}$$

Furthermore, the image-illumination phase-front radius of curvature $r_1$ can be expressed in terms of the feed-horn-aperture phase-front radius of curvature $r_1'$ as

$$\frac{1}{r_1}=\frac{1}{L_1}\left[1+\frac{L_1'}{L_1}\left(1+\frac{L'_1}{r_1'}\right)\right]. \tag{4}$$

Equation (4), which is the same as the relation for Gaussian beam imaging,[8,9] is essentially an alternative form of eq. (7) in Ref. 7. This equation represents the additional phase shift due to Fresnel diffraction.

Equation (4) shows that $r_1$ will be always less than $L_1$ if all parameters on the right side are positive. $r_1$ may become greater than $L_1$ for negative $r_1'$. One notes that $r_1'$ is always positive for any corrugated-feed horn unless modified by another lens or offset reflector. To obtain some feeling about the required physical spacings for a given pair of specified illumination and feed horn, normalized $L_1$ (with respect to $r_1$) has been plotted from eq. (4) versus normalized $L_1'$ (with respect to $r_1'$) for several ratios of $L_1'/L_1$ in Fig. 2. For a given pair of specified illumination and corrugated horn, i.e., $L_1'/L_1$, $r_1$ and $r_1'$ specified, it is also convenient to rearrange eq. (4), as follows:

$$\frac{L_1}{r_1}=\frac{1+\dfrac{L_1'}{L_1}}{1-\left(\dfrac{L_1'}{L_1}\right)^2\dfrac{r_1}{r_1'}}. \tag{5}$$

It is of interest to note a few special cases of eq. (4). The near-field gregorian configuration[6] corresponds to $r_1'=\infty$ in eq. (4). J. A. Arnaud's confocal feed reflector arrangement[10] is also a special case in which $r_1'=-L_1'$ and $r_1=L_1$. However, Arnaud assumed both constant beamwidth and constant phase center for the corrugated horn to obtain a frequency-independent aperture distribution at his first reflector.

If the radiation from a corrugated horn is approximated by a Gaussian beam, the $1/e^*$ beam radius at the horn aperture has been shown[5] to be $w_1'=0.64a_o$, where $a_o$ is the horn-aperture radius. Then

* Here $e=2.71828$, whereas $e$ represents eccentricity of a conic in the next section and appendices.

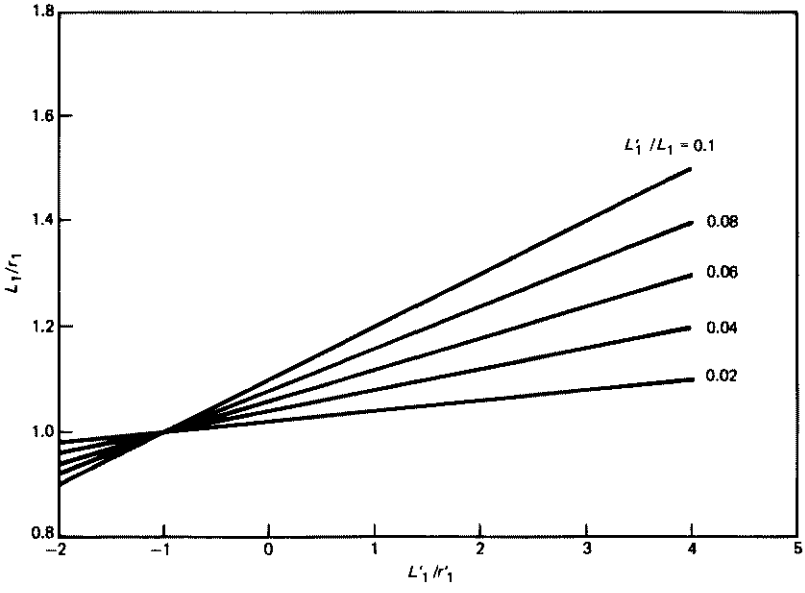



Fig. 2—Relation between image and object phase curvatures for various ratios of $L_1'/L_1$.

the Gaussian beam radius $w_2$ at the imaging lens can be found from[11]

$$w_2 = w_1' \sqrt{\left(\frac{L_1'}{r_1'} + 1\right)^2 + \left(\frac{\lambda L_1'}{\pi w_1'^2}\right)^2}. \tag{6}$$

The lens diameter should be 3.04 times the Gaussian beam radius $w_2$ for a truncation edge taper of $-20$ *dB*.

## III. OFFSET ELLIPSOIDAL REFLECTOR WITH PROJECTED CIRCULAR APERTURE

In practice, the thin lens will be approximated by an offset ellipsoidal reflector, as shown in Fig. 1(b). The distances from the center of the reflector to the two foci, $F_1$ and $F_2$, are, respectively, the incident and reflected phase-front radii of curvature. The incident phase-front radius of curvature can be obtained from the Gaussian beam propagation formula[11]

$$R_1 = L_1 \Bigg/ \left[ 1 - \frac{\dfrac{L_1'}{r_1'} + 1}{\left(\dfrac{L_1'}{r_1'} + 1\right)^2 + \left(\dfrac{\lambda L_1}{\pi w_1'^2}\right)^2} \right]. \tag{7}$$

If the offset ellipsoid is in the far zone of the horn, $R_1$ becomes the distance $CF_1$ from the phase center of the horn to the center of the reflector. The thin lens formula yields $R_2 = CF_2$, the reflected phase-front radius of curvature

$$\frac{1}{R_1} + \frac{1}{R_2} = \frac{1}{f}. \tag{8}$$

The frequency-dependent $R_1$ and $R_2$ are often not the same as $L_1$ and $L_1'$ in eq. (3). The equivalent focal length $f$ is identical to that in eq. (3). It has been shown[8] that the second-order terms of the offset ellipsoidal surface are only functions of $f$ and the angle of incidence, whereas the third-order terms are also dependent upon $R_1$ and $R_2$.

The design of an offset ellipsoidal reflector with oversized aperture of rectangular shape was discussed in Ref. 8. Since corrugated feed horns and required reflector illuminations are often of circular shape, we shall now describe the design of offset ellipsoidal reflectors with projected circular apertures.

The intersection of an ellipsoid and a circular cone subtended at one focus is a plane ellipse subtended by another circular cone at the other focus[1] (see Appendix A for proof and other geometrical relations). However, the two circular cone axes do not intersect the ellipsoid at the same point. In an ideal approximation of the thin lens by an offset ellipsoidal reflector, one would like to have both beam axes of the incident and reflected beams intersect the ellipsoid at the center of the offset reflector. This condition can be approximately realized by locating the intersection of $R_1$ and $R_2$ midway between the intersections 0 and 0′ of two circular cone axes with the ellipsoid, as shown in Fig. 3, i.e.,

$$R_1(\theta_{p1} - \theta_o) = R_2(\theta_o' - \theta_{p2}), \tag{9}$$

where

$$\frac{\sin \theta_{p1}}{\sin 2\theta_i} = \frac{R_2}{\sqrt{R_1^2 + R_2^2 - 2R_1R_2\cos 2\theta_i}} \tag{10}$$

and

$$\frac{\sin \theta_{p2}}{\sin 2\theta_i} = \frac{R_1}{\sqrt{R_1^2 + R_2^2 - 2R_1R_2\cos 2\theta_i}}. \tag{11}$$

$\theta_o$ and $\theta_o'$ are defined in Appendix A. We also need the expressions for the eccentricity, $e$, and the distance, $f_o$, between the vertex and the near focus:

$$f_o = \frac{|R_1 + R_2| - \sqrt{R_1^2 + R_2^2 - 2R_1R_2\cos 2\theta_i}}{2} \tag{12}$$

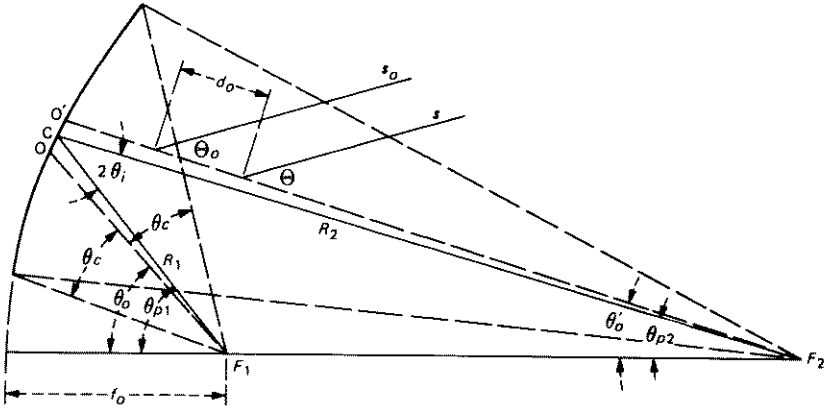



Fig. 3—Schematic diagram of an offset ellipsoidal reflector.

$$e = 1 - \frac{2f_o}{|R_1 + R_2|}. \tag{13}$$

Now the minor radius of the plane ellipse in eq. (23) should be equal to the radius of the projected circular aperture in the beam direction. (i.e., the thin lens aperture radius)

$$\rho_{\text{major}}\sqrt{1 - e^2\sin^2\theta_{po}} = a_l. \tag{14}$$

Substituting eqs. (20) and (22) into eq. (14), we obtain

$$e^2\cos^2\theta_o + 2e\cos\theta_c\cos\theta_o + 1 - \sin^2\theta_c\left[e^2 + \frac{(1+e)^2f^2}{a_l^2}\right] = 0. \tag{15}$$

When we are given an angle of incidence, $\theta_i$, and a pair of radii of curvature, $R_1$ and $R_2$, eqs. (9) and (15) can be solved simultaneously for numerical values of $\theta_o$ *and* $\theta_c$. These parameters completely determine both the shape and size of the ellipsoid.

## IV. NUMERICAL AND MEASURED RESULTS

To demonstrate practical feasibility of the proposed imaging feed design, we have built and tested an experimental broadband offset launcher. This feed was designed to provide the illumination required for the hyperboloidal subreflector of the Crawford Hill 7-meter offset cassegrainian antenna.

Following the method of Section II, we obtain parameters for the thin lens model of this experimental feed, as listed in Table I. Using the procedures described in Section III, we translated the lens parameters into dimensions of an offset ellipsoid, as shown in the schematic diagram of Fig. 3. A design frequency of 22 GHz was used in eqs. (7) and (8) to find $R_1$ and $R_2$, as shown in Table II.

Table I—Design parameters of the thin lens model for a broadband imaging feed (all lengths are in cm.)

| | |
|---|---|
| Required −15 dB illumination circle radius | 47.5 |
| Required phase-front radius of curvature | 535.9 |
| Frequency band (GHz) | 19/28.5 |
| Corrugated horn aperture radius, $a_0$ | 3.8 |
| Horn apex to aperture distance, $r_1'$ | 15.9 |
| Horn aperture to imaging lens distance, $L_1'$ | 47.5 |
| Imaging lens to subreflector distance, $L_1$ | 684.6 |
| Lens focal length, $f$ | 44.5 |
| Lens radius (truncation edge taper $\approx$ 20 $dB$) | 21 |
| Lens to phase-center distance, $\Delta$ | 148.7 |

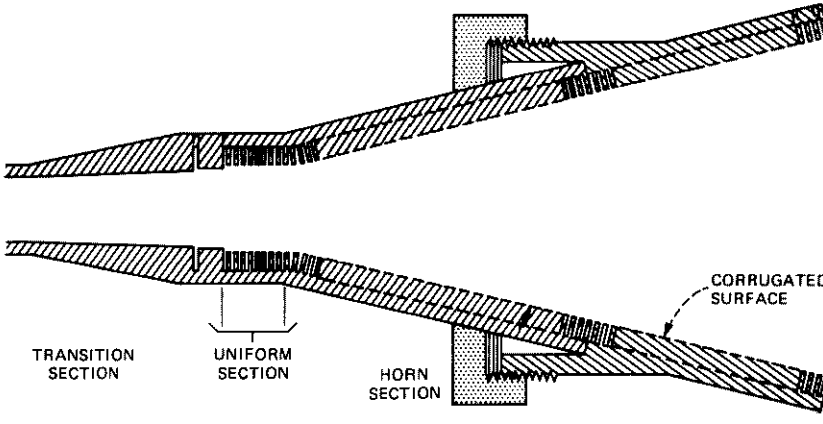



Fig. 4—Corrugated horn geometry.

A sketch of the corrugated horn is shown in Fig. 4. The corrugation depth is about a quarter wavelength at 18 GHz. The steps between smooth and corrugated sections help impedance matching because the desirable $HE_{11}$ mode in the corrugated guide is concentrated toward the center of the cross section. The return loss of the horn is better than 20 dB for frequencies above 16 GHz. The E- and H-plane beamwidths are nearly identical to each other from 16 to 30 GHz.

Figure 5 shows a photograph of the combination of offset ellipsoidal reflector and corrugated horn. Amplitude and phase patterns have been measured in both the offset plane and the transverse plane orthogonal to the offset. Since the phase center is located about 1.5 m from the center of the offset ellipsoid, physical rotation around the phase center would cause problems of mechanical unbalance. Therefore, the center of rotation in the pattern measurements is located midway between the ellipsoid and the phase center. The measured data can be transformed into measured patterns around the phase center by the relations between the angles of rotation and between the path lengths.

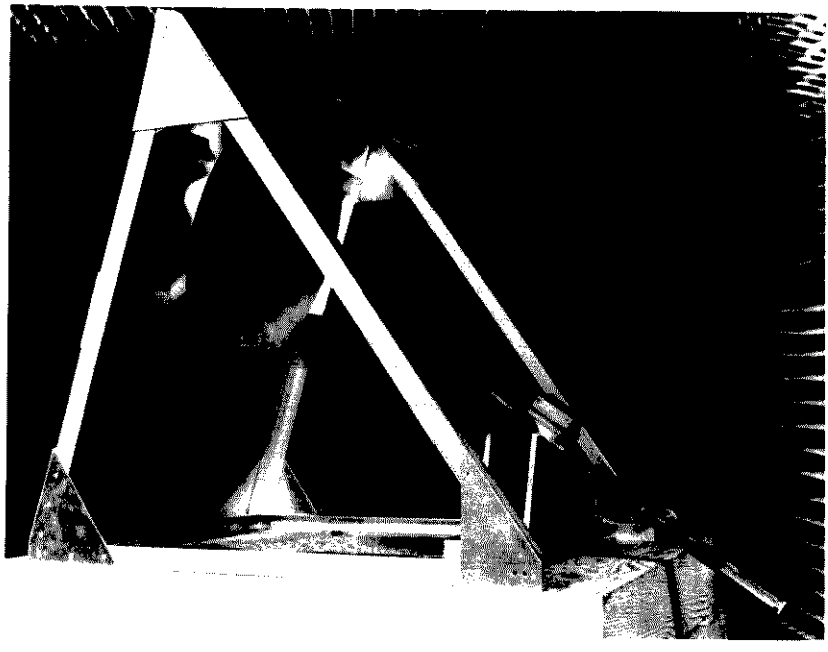


Fig. 5—Experimental broadband imaging feed.

Table II—Design parameters of the offset ellipsoidal reflector (all lengths are in cm.)

| | |
|---|---|
| Incident phase-front radius of curvature $R_1$ | 54.36 |
| Reflected phase-front radius of curvature $R_2$ | 244.22 |
| Angle of incidence, $\theta_i$ | 17° |
| Angle between incident center ray and major ellipsoidal axis, $\theta_{p1}$ | 42.68° |
| Angle between reflected center ray and major ellipsoidal axis, $\theta_{p2}$ | 8.68° |
| Offset angle, $\theta_o$ | 41.87° |
| Half-cone angle, $\theta_c$ | 22.09° |
| Offset angle at the distant focus, $\theta_o'$ | 8.86° |
| Vertex to near focus distance, $f_o$ | 48.56 |
| Eccentricity, $e$ | 0.67475 |
| Semi-major-axis of plane ellipse, $\rho_{major}$ | 21.98 |
| Semi-minor-axis of plane ellipse, $\rho_{minor}$ | 21 |
| Semi-major-axis of ellipsoid, $a$ | 149.29 |
| Semi-minor-axis of ellipsoid, $b$ | 110.18 |

The two equations below

$$\sin \Theta = \frac{s_o}{s} \sin \Theta_o \tag{16}$$

$$s = \sqrt{s_o^2 + d_o^2 - 2s_o d_o \cos \Theta_o} \tag{17}$$

are similar to eqs. (32) and (31). The parameters are explained in Fig. 3. The angular conversion of eq. (16) will be required in both

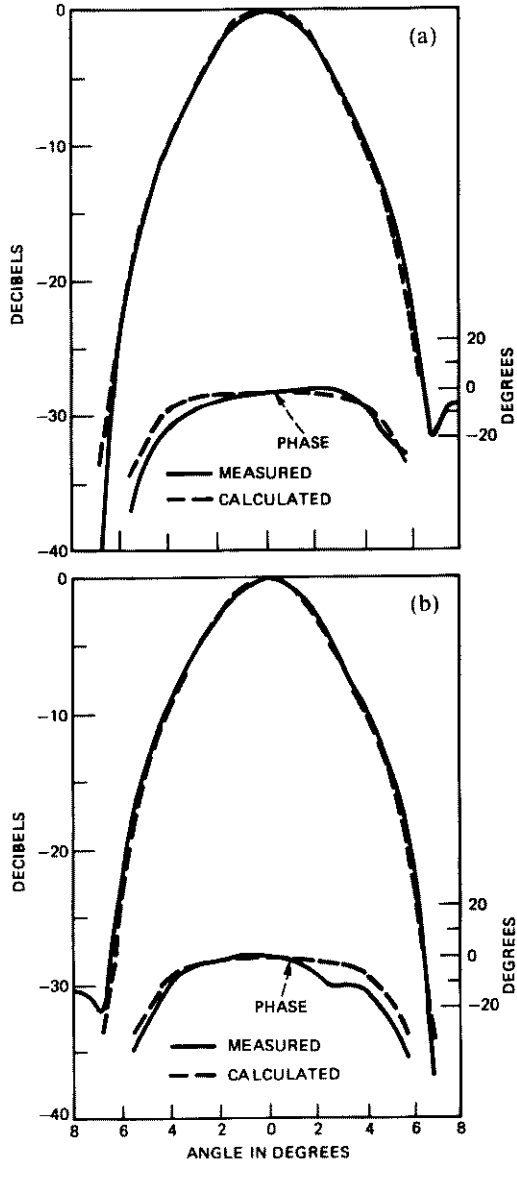



Fig. 6—19-GHz radiation pattern of transverse polarization (transverse perpendicular to offset) in (a) offset plane and (b) transverse plane.

amplitude- and phase-pattern transformations, whereas another factor of $(s + d_o - s_o)$ will be also added to the measured phase pattern.

To ensure reliable phase data, all pattern measurements were made in the azimuthal plane where a rotary joint can be used for avoiding

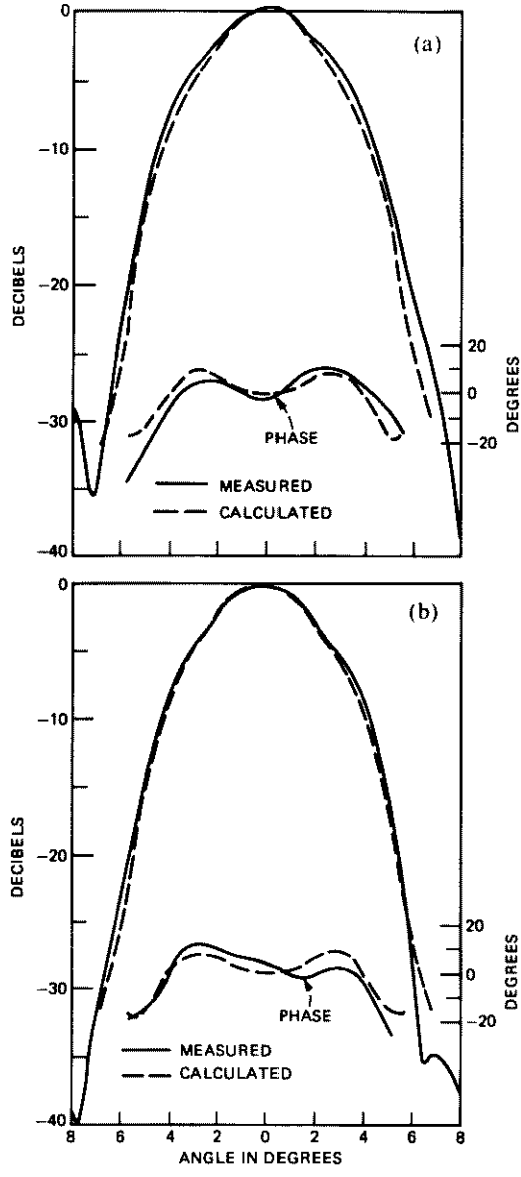



Fig. 7—23-GHz radiation pattern of transverse polarization (transverse perpendicular to offset) in (a) offset plane and (b) transverse plane.

cable motion during rotation of the turntable. Care was taken in keeping the same center of rotation by flipping over the feed assembly between measuring the transverse plane and offset plane cuts.

Calculations of radiation patterns are described in Appendix B. Both

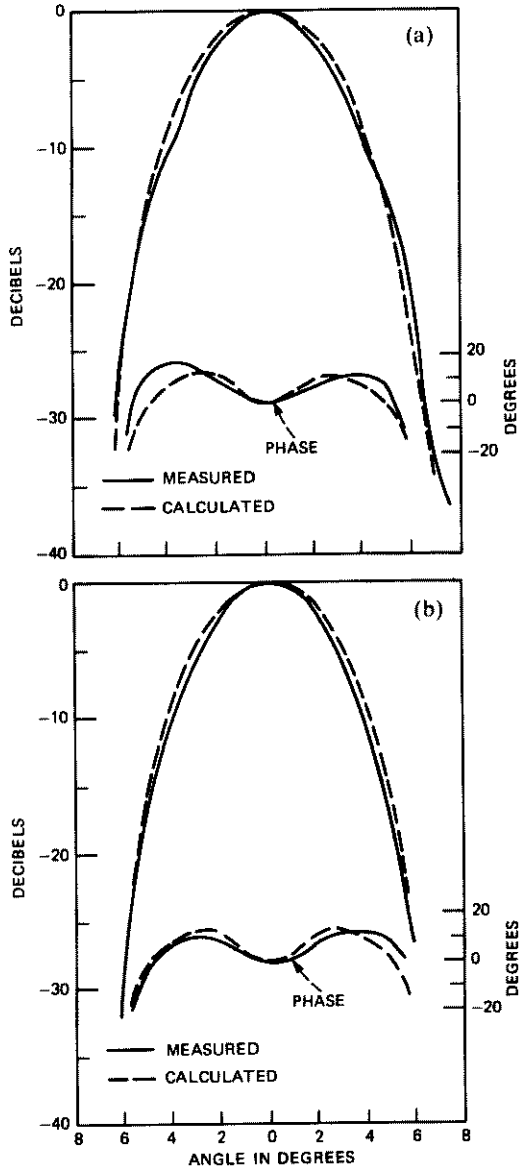



Fig. 8—28-GHz radiation pattern of transverse polarization (transverse perpendicular to offset) in (a) offset plane and (b) transverse plane.

measured and calculated patterns around the phase center are shown in Figs. 6, 7, and 8 for 19, 23, and 28 GHz, respectively. Comparisons between measured and calculated data show generally good agreement for both amplitude and phase patterns. An ideal imaging feed would

have a frequency-independent amplitude pattern of −15 dB taper at 5 degrees and a straight-line phase pattern that represents an exact spherical wave originating from a point source at the phase center. Deviations from these ideal patterns can be attributed to the imperfection of the paraxial approximation and to the truncation effect. Measured patterns also include the effect of deviation of the corrugated-horn-aperture field from the theoretical model. Figures 6 through 8 show that 28-GHz measured pattern widths are slightly narrower than those of 19 GHz. Phase deviations remain less than 20 degrees. These results are very similar to those of a long broadband corrugated horn designed for constant beamwidth feed.

The patterns in Figs. 6 through 8 were measured with the polarization transverse to the offset plane. Measurements with the polarization parallel to the offset plane showed similar patterns. Cross-polarized pattern measurements showed a maximum cross polarization of −26 dB, which is what would be expected from the offset geometry.[12]

## V. DISCUSSIONS

Theoretical and experimental studies have demonstrated the feasibility of a broadband imaging feed using the combination of an offset ellipsoid and a corrugated horn. This feed is also important for serving as a basic building block of the imaging beam waveguide.

Since a broadband feed design will avoid the need of quasi-optical frequency diplexing, a much simpler, cost-effective feed system can be built to achieve performance similar to that of the 19/28.5 GHz feed of the Crawford-Hill 7-meter antenna.

The application of imaging of a corrugated horn by an offset ellipsoidal reflector is certainly not limited to broadband narrow feed patterns for large ground station antennas. For example, Dragone suggests[13] its application to terrestrial microwave repeater antennas. Furthermore, it can be used as a constant beamwidth radiometer antenna for multifrequency remote sensing.

## REFERENCES


1. T. S. Chu et al., "The Crawford Hill 7-Meter Millimeter Wave Antenna," B.S.T.J., *57*, No. 5 (May–June 1978), pp. 1257–88.
2. M. J. Gans and R. A. Semplak, "Some Far-Field Studies of An Offset Launcher," B.S.T.J., *54*, No. 7 (September 1975), pp. 1319–40.
3. T. S. Chu and W. E. Legg, "Gain of Corrugated Conical Horns," IEEE Trans., *AP-30* (July 1982), pp. 698–703.
4. C. Dragone, "Reflection, Transmission, and Mode Conversion in a Corrugated Feed," B.S.T.J., *56*, No. 6 (July–August 1977), pp. 835–67.
5. C. Dragone, "An Improved Antenna for Microwave Radio System Consisting of Two Cylindrical Reflectors and a Corrugated Horn," B.S.T.J., *53*, No. 7 (September 1974), pp. 1351–77.
6. C. Dragone and M. J. Gans, "Imaging Reflector Arrangements to Form a Scanning Beam Using a Small Array," B.S.T.J., *58*, No. 2 (February 1979), pp. 501–15.

7. H. Kogelnik, "Imaging of Optical Modes—Resonators With Internal Lenses," B.S.T.J., *44*, No. 3 (March 1965), pp. 455–94.
8. T. S. Chu, "An Imaging Beam Waveguide Feed," IEEE Trans. AP-S, *AP-31*, No. 4 (July 1983).
9. T. S. Chu, "Broadband Astigmatic Compensation," Digest of 1981 IEEE Antenna and Propagation Symp., p. 131–4.
10. M. J. Gans, "Cross-Polarization in Reflector-Type Beam Waveguides and Antennas," B.S.T.J., *55*, No. 3 (March 1976), pp. 289–316.
11. T. S. Chu, "Geometrical Representation of Gaussian Beam Propagation," B.S.T.J., *45*, No. 2 (February 1966), pp. 287–99.
12. T. S. Chu and R. H. Turrin, "Depolarization of Offset Reflector Antennas," IEEE Trans., *AP*-21 (May 1973), pp. 339–45.
13. C. Dragone, private communication.
14. J. S. Cook, E. M. Elam, and H. Zucker, "The Open Cassegrain Antennas: Part I, Electromagnetic Design and Analysis," B.S.T.J., *44*, No. 7 (September 1965), pp. 1255–1300.
15. V. Jamnejad-Dailami and Y. Rahmat-Samii, "Some Important General Features of Conic-Section-Generated Offset Reflector Antennas," IEEE Trans., *AP*-28 (November 1980), pp. 952–7.

## APPENDIX A

### *Geometry of Offset Ellipsoids*

The geometrical properties of offset ellipsoids can be generalized from those[14] of offset paraboloids. The following formulas are derived by lengthy but straightforward algebra. The intersection of an offset ellipsoid and a circular cone subtended at the focus is a plane ellipse, which is subtended by another circular cone at the other (distant) focus. This property was indicated in Ref. 1, and also observed later in Ref. 15.

Let us define the ellipsoid in Figure 9 by

$$r = \frac{(1 + e) f_o}{1 + e \cos \theta_p}, \tag{18}$$

where the eccentricity, $e$, is less than unity. $f_o$ is the distance between the origin in $X_p Y_p Z_p$ coordinates and the vertex, and $\theta_p$ is the polar angle with respect to the $Z_p$ axis. We first find the intersection between the ellipsoid and the $x_o y_o$ plane, which is perpendicular to $x_p z_p$ plane, located at a distance $r_o$ from the origin, and its normal makes an angle $\theta_{po}$ with the $Z_p$ axis:

$$(1 - e^2 \sin^2 \theta_{po}) \left\{ x_o - \frac{e(\sin \theta_{po})}{1 - e^2 \sin^2 \theta_{po}} [(1 + e) f_o - e r_o \cos \theta_{po}] \right\}^2 + y_o^2 = \frac{[e r_o \cos \theta_{po} - (1 + e) f_o]^2}{1 - e^2 \sin^2 \theta_{po}} - r_o^2. \tag{19}$$

Next, the following expressions are found for $\theta_{po}$ and $r_o$:

$$\theta_{po} = \sin^{-1} \left( \frac{\sin \theta_o}{\sqrt{1 + 2e \cos \theta_o \cos \theta_c + e^2 \cos^2 \theta_c}} \right) \tag{20}$$

$$r_o = \frac{(1 + e) f_o \cos \theta_c}{\sqrt{1 + 2e \cos \theta_o \cos \theta_c + e^2 \cos^2 \theta_c}}, \tag{21}$$

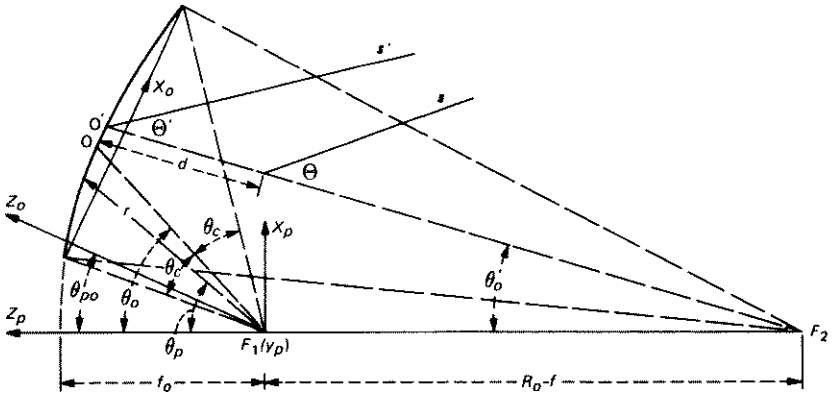



Fig. 9—Geometry of an offset ellipsoid subtended by circular cones at two foci.

where $\theta_o + \theta_c$ and $\theta_o - \theta_c$ are the polar angles of the top and bottom edges of the offset ellipsoid in the $x_p z_p$ plane. Now it can be shown that the intersection of the $x_o y_o$ plane with the circular cone, which has its axis oriented at $\theta_p = \theta_o$ and a half cone angle of $\theta_c$, is the same ellipse as represented by eq. (19).

The major and minor radii of the ellipse are

$$\rho_{\text{major}} = \frac{(1+e) f_o \sin\theta_c \sqrt{1 + 2e\cos\theta_o \cos\theta_c + e^2\cos^2\theta_c}}{1 + 2e\cos\theta_o\cos\theta_c + e^2\cos^2\theta_c - e^2\sin^2\theta_o} \tag{22}$$

$$\rho_{\text{minor}} = \rho_{\text{major}} \sqrt{1 - e^2\sin^2\theta_{po}}. \tag{23}$$

When $e = 1$, the above equations are reduced to those of Appendix A in Ref. 14. One notes an error in eq. (50) for the plane ellipse in Ref. 14, i.e., missing a term $(r_o \tan\theta_{po})$ in the bracket on the left-hand side of that equation.

The axis of the circular cone subtended at the second (distant) focus is oriented with respect to the major axis of the ellipsoid at

$$\theta'_o = \frac{1}{2}\left\{\sin^{-1}\left[\frac{r_2}{R_o + f_o - r_2}\sin(\theta_o + \theta_c)\right] + \sin^{-1}\left[\frac{r_1}{R_o + f_o - r_1}\sin(\theta_o - \theta_c)\right]\right\}, \tag{24}$$

where

$$r_{1,2} = \frac{(1+e) f_o}{1 + e\cos(\theta_o \mp \theta_c)} \tag{25}$$

are the radial distances from the (near) focus to the termini of the major axis of the ellipse represented by eq. (19) and

$$R_o = f_o\left(\frac{1+e}{1-e}\right) \tag{26}$$

is the distance from the vertex to the second focus.

## APPENDIX B

### *Calculation of Radiation Pattern*

In this appendix we shall calculate the radiation pattern of an offset ellipsoid illuminated by a corrugated feed horn. The numerical integration of the diffraction integral accomplishes computer simulation of the hardware.

When an ellipsoid is illuminated by a feed located at the near focus, the geometrical optic rays reflected from the ellipsoid are expected to converge toward the distant focus. Although the diffraction effect will take place before reaching that second focus, the ray approximation should be valid in the vicinity of the ellipsoid. To find the radiation from the ellipsoid, we will employ a spherical wave field in an equivalent plane aperture that is perpendicular to the axis of, and subtended by, the cone of reflected rays, and passes through the point, O′, of intersection of the centric reflected ray with the ellipsoid. The radius of curvature of the spherical wave is the distance, $R'_o$, from the distant focus to the point O′.

$$R'_o = \frac{(1+e)f_o}{1 - e\cos\theta'_o} \tag{27}$$

The transverse cartesian coordinates in the plane aperture are

$$\begin{cases} x = R(\cos\theta_o\sin\theta\cos\phi + \sin\theta_o\cos\theta)\cos\theta'_o \\ \qquad - [R(-\sin\theta_o\sin\theta\cos\phi + \cos\theta_o\cos\theta) + R_o - f]\sin\theta'_o \\ y = R\sin\theta\sin\phi, \end{cases} \tag{28}$$

where

$$R = \frac{(1+e)f_o}{1 + e(\cos\theta_o\cos\theta - \sin\theta\sin\theta_o\cos\phi)} \tag{29}$$

is the position radius from the focus to the ellipsoid. $\theta$ and $\phi$ are the standard spherical coordinates with the polar axis $F_1O$ along the axis of the circular cone subtended at the near focus, as shown in Fig. 9.

Using a small angle scalar approximation, we can determine the radiation of the offset ellipsoid as

$$E = \frac{je^{-jk(s+d)}}{\lambda(s+d)}\int\frac{F}{R}\exp\left\{jk\left[x\sin\Theta'\cos\Phi + y\sin\Theta'\sin\Phi + \frac{x^2+y^2}{2R'_o} - \frac{x^2+y^2}{2(s+d)} + (s+d-s')\right]\right\}dA, \tag{30}$$

where $F$ is the feed pattern, which will be described later. The term $(x^2 + y^2)/2(s + d)$ is a phase correction needed for the Fresnel region in which the Cassegrainian subreflector is located. The factor $(s + d - s')$ is added to give a phase pattern with respect to the phase center, which is located at a distance $d$ in front of the ellipsoid. $s$ is the distance between the phase center and the field point, and $s'$ is the distance between the point O', which is the effective center of the ellipsoid aperture, and the field point

$$s' = \sqrt{s^2 + d^2 + 2sd \cos \Theta}. \tag{31}$$

The angle $\Theta'$, between $s'$ and the cone axis $O'F_2$, is related by the sine law to the angle $\Theta$ between $s$ and the cone axis, as shown below:

$$\sin \Theta' = \frac{s}{s'} \sin \Theta. \tag{32}$$

$\phi$ is the azimuth coordinate of the field point. The surface element $dA$ is

$$dA = \frac{R^2 \sin \theta d\theta d\phi}{\cos \theta_i} \cos \theta_r, \tag{33}$$

where $\theta_i$ is the angle of incidence between the incident ray and the unit vector normal to the ellipsoidal surface, and $\theta_r$ is the angle between this unit normal and the beam axis. Since the ellipsoids under consideration are only slight perturbations of paraboloids, $\cos \theta_i \approx \cos \theta_r$. Noting the symmetry of the aperture field, eq. (30) becomes

$$E = \frac{je^{-jk(s+d)}}{\lambda(s + d)} \int_o^{\pi} \int_o^{\theta_c} F \exp jk \left[ x \sin \Theta' \cos \Phi + \frac{x^2 + y^2}{2R'_o} - \frac{x^2 + y^2}{2(s + d)} + (s + d - s') \right] \cos[ky \sin \Theta' \sin \Phi] R \sin \theta d\theta d\phi. \tag{34}$$

The feed pattern of a corrugated horn is given by

$$F(\Theta') = \int_0^1 J_o(\alpha r) J_o \left( ka \sin \theta' \frac{R}{R'} r \right) e^{-jk\left[ ar \tan \frac{\beta}{2} + \frac{(ar)^2}{2R'} + \ell(1 - \cos \theta') \right]} r dr, \tag{35}$$

where $a$ is the aperture radius, $\beta$ is the half cone angle of the horn, $\alpha = 2.405$ for frequencies close to the resonance of the corrugated depth, $r$ is the normalized radial coordinate of the horn aperture, and $\sin \theta'$ is multiplied by $R/R'$ because $\theta'$ is referred to the ellipsoid focus (the phase center of the horn), which is located at a distance $\ell$ behind the horn aperture. $R'$ is the distance from the center of the horn aperture to a point on the ellipsoid:

$$R' = \sqrt{R^2 + \ell^2 - 2R\ell \cos \theta'}. \tag{36}$$

The first term $kar^2\tan(\beta/2)$ in the exponential bracket of eq. (35) is the phase deviation in the horn aperture, the second term is a phase correction in the Fresnel region, and the third term is added to give a phase pattern with respect to the horn phase center (ellipsoidal focus) instead of the horn aperture.

To compensate for the difference in length between the radii from the focus toward the top and bottom edges of the offset ellipsoid, the axis of the feed horn is offset by an angle $\theta_1$ from the axis of the circular cone subtended by the ellipsoid at the focus. $\theta'$ can be expressed in terms of the angular coordinates $(\theta, \phi)$ of the offset ellipsoid:

$$\cos\theta' = \cos\theta\cos\theta_1 + \sin\theta\sin\theta_1\cos\phi. \tag{37}$$

$\theta_1 = 1.6$ degrees is used in the calculated patterns of Figs. 6, 7, and 8.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 5, May–June 1983
*Printed in U.S.A.*

# An M/G/1 Queue With a Hybrid Discipline

By B. T. DOSHI

(Manuscript received October 27, 1982)

*In this paper we analyze the delay in a single-server queue in which the server, when it becomes free, selects for the next service the oldest customer with current delay smaller than T. If no such customer is present, then it selects the youngest customer with the current delay in excess of T. This service discipline is desirable in applications where the success or failure of a service depends on the delay in providing the service. Telephone call processing and steel rolling are two of these applications. We obtain the delay distribution for this service discipline using a combination of level-crossing arguments and renewal theory, and compare this performance with that of the last-in-first-out discipline with respect to the throughput of successfully served customers.*

## I. INTRODUCTION

The following situation is common in telephone call processing or data-processing systems. When a customer requests service, an entry is made in the queue that is serviced by a processor. The processor serves the queue of entries according to some specified service discipline. When an entry is served, the corresponding customer is notified of the completion of the service. The customer, however, does not wait forever for the completion of the service. At some random time, $R$, after its arrival, the customer will renege if service is not completed. The associated entry remains in the queue, and the server does not know that the customer has reneged until after it completes the service and attempts to notify the customer. Such a service is wasted. The customer may make the situation worse by reattempts at getting the desired service, thereby increasing the load on the server. It is therefore necessary to keep the proportion of reneging customers as small as possible. This can be done by selecting an appropriate discipline.

In this paper we study a specific model of the above situation. In

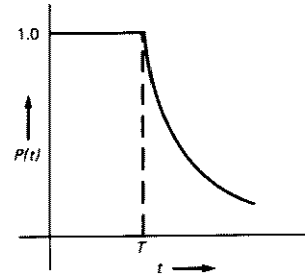



Fig. 1—Customer behavior function 1.

particular, we have a single-server queue with Poisson arrivals at rate $\lambda > 0$.

Let $G$ be the distribution of the service time. Let $\bar{P}(t)$ be the probability that the customer does not renege before time $t$. We may think of $\bar{P}(t)$ as the expected reward obtained by completing the service of a customer at $t$ time units after its arrival. With this general interpretation $\bar{P}(t)$ need not be restricted to be between 0 and 1. Let

$$P(t) = \int_{0^-}^{\infty} \bar{P}(t + y)dG(y) \tag{1}$$

for $0 \leq t < \infty$. Then $P(t)$ is the expected reward from a customer whose *waiting time* (excluding the service time) is $t$. Let $W_\pi$ be the distribution of the waiting time under the specified service discipline, $\pi$. Then the expected reward from an arbitrary customer is

$$V_\pi = \int_{0^-}^{\infty} P(t)dW_\pi(t). \tag{2}$$

We want to select a service discipline that maximizes $V_\pi$. It was shown in Doshi and Lipper[1] that if $P(t)$ is convex (respectively, concave), then the last-in-first-out (LIFO) [respectively, first-in-first-out (FIFO)] discipline is optimal. More realistic functions $P(t)$, however, are of the forms given in Figs. 1 and 2.

For such functions, $P(t)$, an optimal service discipline, is not known. However, our results for concave and convex $P(t)$ indicate that a hybrid discipline may provide better performance than either the FIFO or the LIFO discipline does. In this hybrid discipline the server, when it completes a service, first looks at the customers with the current waiting time less than $T$ and selects the oldest waiting customer for the next service. If no such customer is waiting, then the server looks at the customers with the current waiting time in excess of $T$ and selects the youngest customer. Note that this hybrid discipline includes FIFO ($T = \infty$) and LIFO ($T = 0$) as special cases. Since $P(t)$

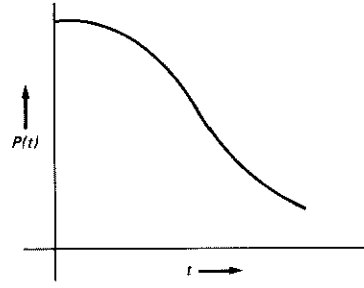



Fig. 2—Customer behavior function 2.

is assumed to be given, we only need $W_\pi(t)$, $0 \leq t < \infty$, for this hybrid discipline $\pi$, to calculate $V_\pi$. In this paper we obtain $W_\pi$ for general service time distribution. We do this in three steps. First we describe another queueing system for which the distribution of the waiting time is the same as for the original system. Moreover, for this equivalent queueing system the distribution of the waiting time can easily be expressed in terms of the distribution, $F(x)$, of the work in a subsystem. We then use level-crossing arguments to derive an integral equation satisfied by $f(x) = F'(x)$. Finally, we use some results from renewal theory to solve this integral equation.

Some comments about the model are in order before we proceed to give an outline of the rest of this paper. Models similar to the one studied here can be useful in a variety of other applications. Some of these are the management of steel-rolling operation and the management of blood bank. Also, in many applications the customers do not necessarily renege. They simply take actions (start to dial, become cold, etc.) which make any subsequent service worthless.

This paper is organized as follows: In Section II we formally define the queueing system under consideration. We describe an equivalent queueing system in Section III. There we also show the relationship between the distributions of the waiting time in the original system and of the work in a subsystem of the equivalent queueing system. In Section IV we derive an integral equation for the steady-state density of the work in the subsystem. We give the solution of this integral equation in Section V. There we also derive the steady-state distribution of the waiting time in the original system. Finally, we give some numerical results in Section VI.

## II. MODEL

The queueing system and the hybrid service discipline discussed in Section I can be formally described as follows: We have a queueing system with a single server and two queues, Q1 and Q2. Customers in

Q1 have a nonpreemptive priority over those in Q2. The order of service is first-in-first-out (FIFO) in Q1 and last-in-first-out (LIFO) in Q2. Customers arrive according to a Poisson process at rate $\lambda > 0$. On arrival the customer is put in Q1. If its service has not started within $T$ seconds after its arrival, then at that time it is transferred to Q2. The service times of the customers are independent and identically distributed with distribution function $G$ with continuous density $g$.

Let $\mu_K$ denote the $K$th moment of the service time and let $\rho = \lambda\mu_1$. Assume that the waiting-time process is in the steady state. Let $W$ denote the distribution function of the waiting time seen by an arbitrary customer. Since $G$ has a continuous density, $W$ is differentiable on $(0, \infty)$. Let

$$w(x) = W'(x) \qquad 0 < x < \infty.$$

We are interested in obtaining an expression for $W(x)$ or, equivalently, for $W(0)$ and $w(x)$, $0 < x < \infty$.

## III. AN EQUIVALENT QUEUEING SYSTEM

We now describe a queueing system that is equivalent to the one described in Section II as far as the waiting times of the customers are concerned. However, the number of customers in Q1 and Q2 at a given time may be different in the two systems.

Consider the subsystem consisting of the server and Q1. Let $X_t$ denote the work, at time $t$, in this subsystem. Thus, $X_t$ is the sum of the remaining service time of the customer, if any, being served and the service times of all the customers in Q1. If a customer arriving at time $t$ finds $X_t \leq T$, then it joins Q1; otherwise it joins Q2. Recall that Q1 has a nonpreemptive priority over Q2 and that Q1 is served FIFO and Q2 is served LIFO. A little reflection shows that the waiting time of a customer is the same in this system as in the one described in Section II.

We now relate the waiting-time distribution, $W$, to the steady-state distribution, $F$, of the work, $X$, in the subsystem consisting of the server and Q1. If an arriving customer sees $X \leq T$, then its waiting time will be $X$ because the service in Q1 is FIFO and because Q1 has a nonpreemptive priority over Q2. Thus,

$$W(x) = F(x) \qquad 0 \leq x \leq T. \tag{3}$$

In particular,

$$W(0) = F(0), \tag{4}$$

and

$$w(x) = f(x) = F'(x) \qquad 0 < x < T. \tag{5}$$

If an arriving customer sees $X > T$, then it joins Q2. Since Q1 has priority over Q2 and since Q2 is served LIFO, this customer has to wait for the duration of a busy period in the M/G/1 queue with initial work, $X$. Let $B_y$ denote the distribution of the busy period in the M/G/1 queue started by initial work, $y$. Since $G$ has a continuous density,

$$b_y(x) = B'_y(x) \tag{6}$$

exists for all $x > y$. Then

$$w(x) = \int_T^x f(y) b_y(x) d(y) \qquad x > T. \tag{7}$$

Thus, it is sufficient to find the distribution of $X$.

## IV. INTEGRAL EQUATION FOR $f(x)$

We now use level-crossing arguments (see Ref. 2) to derive an integral equation for $f$. Figure 3 shows a typical sample function for the process $\{X_t\}$. Assume that at time 0 the queues are empty, the server is idle, and a customer arrives and enters service. If $X_t > 0$, then it decreases at unit rate as in a M/G/1 queue. Symbols o represent arrivals which see $X_t \leq T$ and join the subsystem, thus increasing $X_t$ by a service time. A symbol in the shape of a dot (●) represents arrivals that see $X_t > T$, and join Q2 without affecting $X_t$ on their arrival. When $X_t$ reaches zero, two things can happen: Q2 is empty and the server remains idle until the next arrival, or Q2 is nonempty and a customer from Q2 enters service, thus increasing $X_t$ by its service time. Such arrivals from Q2 into the server are denoted by a symbol in the shape of a square (■).

The stochastic process $\{X_t\}$ is not Markovian because what happens

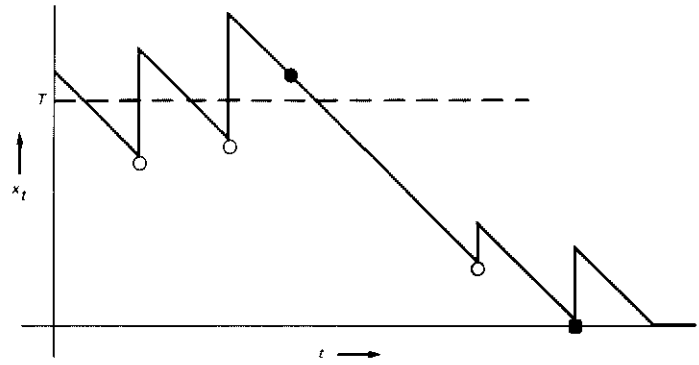



Fig. 3—Typical sample function for the process $\{X_t\}$.

when $X_t$ reaches zero depends on the past. On the other hand, the vector-valued process $\{(X_t, N_t)\}$, where $N_t$ is the number in Q2 at time $t$, is a Markov process. We can derive the steady-state distribution of $(X_t, N_t)$ using the standard results and use that to obtain the marginal steady-state distribution of $X$. We, however, use a simpler approach here. Recall that $\mu_2 < \infty$. Consider the following two cases:

(*i*) $\rho < 1$. In this case the process $\{X_t\}$ is regenerative and with the regeneration points corresponding to the external arrivals that make an idle server busy. Denote this event by E. Then E is a positive recurrent, regenerative event.

(*ii*) $\rho \geq 1$. In this case the queue length in Q2 grows without bound and, in the steady state, can be assumed to be $\infty$. Thus, a customer is removed from Q2 to enter service every time $X_t$ reaches 0. This event, E′, is then positive recurrent.

Standard regenerative arguments now show that $\{X_t\}$ has a steady-state distribution and that

$$X_t \xrightarrow{D} X,$$

where the distribution of $X$ is the steady-state distribution of $\{X_t\}$. Moreover, for any $x$, $0 < x < \infty$, the steady-state rate, $D(x)$, at which $X_t$ crosses $x$ from above, equals the rate $U(x)$, at which $X_t$ jumps from below $x$ to above $x$. We now express $D(x)$ and $U(x)$ in terms of $f(x)$ and get the desired integral equation by equating these expressions.

$X_t$ decreases at unit rate until an arrival occurs or until $X_t = 0$. Thus, during every downcrossing of level $x$, the $\{X_t\}$ process spends $dx$ units of time in the interval $(x - dx, x)$. Hence,

$$D(x) = f(x) \qquad 0 < x < \infty. \tag{8}$$

Before deriving an expression for $U(x)$ we introduce some notation. Let $p$ denote the rate at which $X_t$ jumps from 0 to some positive value. These jumps may be due to either external arrivals coming to an idle system or to customers from Q2 moving to the server. Also, let $\bar{G}$ denote the complementary service time distribution defined by

$$\bar{G}(x) = 1 - G(x) \qquad 0 \leq x < \infty. \tag{9}$$

Assume that $G(0) = 0$, $\bar{G}(0) = 1$. Then, for $x \leq T$

$$U(x) = \lambda \int_0^x f(y)\bar{G}(x - y)dy + p\bar{G}(x). \tag{10}$$

Since an external arrival causes a jump in $X_t$ only when $X_t \leq T$, we have, for $x > T$,

$$U(x) = \lambda \int_0^T f(y)\bar{G}(x - y)dy + p\bar{G}(x). \tag{11}$$

Thus,

$$U(x) = \lambda \int_0^{T \wedge x} f(y)\bar{G}(x-y)dy + p\bar{G}(x) \qquad 0 < x < \infty. \tag{12}$$

Let

$$f(0) = \lim_{x \downarrow 0} f(x). \tag{13}$$

Then $f(0)$ is the rate at which $\{X_t\}$ hits level zero. In the steady state, this must equal the rate at which $\{X_t\}$ jumps from 0 to some positive value. Thus

$$p = f(0). \tag{14}$$

We now have for $0 < \rho < \infty$, $0 < x < \infty$,

$$\begin{aligned} f(x) &= D(x) \\ &= U(x) \\ &= \lambda \int_0^{x \wedge T} f(y)\bar{G}(x-y)dy + f(0)\bar{G}(x). \end{aligned} \tag{15}$$

Let

$$q = \frac{f(0)}{\lambda}.$$

Then

$$f(x) = \lambda \int_0^{x \wedge T} f(y)\bar{G}(x-y)dy + q\lambda\bar{G}(x). \tag{16}$$

This is the desired integral equation for $f$. The additional conditions needed to solve this completely depend on whether $\rho < 1$ or $\rho \geq 1$.

First consider the case where $\rho < 1$. Let $P_2$ be the probability that an arriving customer sees $X_t > T$ and joins Q2. Since, for $\rho < 1$, every arriving customer is eventually served, the rate at which customers enter the server from Q2 is $\lambda P_2$. Also, the rate of arrivals coming to an empty system is $\lambda F(0)$. Thus,

$$q = \frac{f(0)}{\lambda} = \frac{\lambda[F(0) + P_2]}{\lambda} = F(0) + P_2. \tag{17}$$

Also,

$$P_2 = \int_T^{\infty} f(x)dx, \tag{18}$$

and

$$F(0) + \int_0^\infty f(x)dx = 1. \tag{19}$$

For $\rho \geq 1$, Q2 is always nonempty in steady state. Hence $F(0) = 0$,

$$q = \frac{f(0)}{\lambda}, \tag{20}$$

and

$$\int_0^\infty f(x)dx = 1. \tag{21}$$

Equation (16), together with either conditions [eqs. (17) to (19) or 20], characterizes $f$ completely. We solve this equation in the next section.

## V. SOLUTION OF THE INTEGRAL EQUATION

We now solve eq. (15) to obtain an expression for $f(x)$. For $0 < x < \infty$, let

$$h(x) = \lambda \bar{G}(x) \tag{22}$$

and let $m(x)$ be the renewal density function for $h(x)$. Then $m(x)$ satisfies (see Ref. 3):

$$m(x) = h(x) + \int_0^x h(x - y)m(y)dy. \tag{23}$$

Equation (15) can now be rewritten as

$$f(x) = qh(x) + \int_0^x f(x - y)h(y)dy \qquad (0 < x < T), \tag{24}$$

and

$$f(x) = qh(x) + \int_0^T f(y)h(x - y)dy \qquad T \leq x < \infty. \tag{25}$$

Equation (24) is a renewal equation and its solution is given by

$$\begin{aligned} f(x) &= qh(x) + q\int_0^x h(x - y)m(y)dy \\ &= q\left[h(x) + \int_0^x h(x - y)m(y)dy\right] \\ &= qm(x), \end{aligned} \tag{26}$$

where the last equality follows from (23). Note that $0 < h(x) < \infty$, and

$$\int_0^\infty h(x)dx = \lambda\mu_1 = \rho.$$

Thus, $h$ is like a probability density with total mass, $\rho$. The function $m(x)$ is well defined for any finite $x$ irrespective of the value of $\rho$, $0 < \rho < \infty$. To obtain an expression for $f(x)$, $x \geq T$, we note that the right-hand side of (25) involves $f(y)$ for $y$ only in the interval $(0, T)$, which we have obtained in (26). Thus, replacing $f(y)$ on the right-hand side of (25) by $qm(y)$, we get

$$f(x) = \left[ qh(x) + \int_0^T qm(y)h(x-y)dy \right]$$

$$= q\left[ h(x) + \int_0^T h(x-y)m(y)dy \right] \qquad x \geq T. \tag{27}$$

We now use conditions (17) through (19) or (20) through (21) to evaluate $q$ and thus completely characterize $f$. First, consider the case $\rho < 1$. We have

$$q = F(0) + P_2, \tag{28}$$

and

$$F(0) + q\int_0^T m(x)dx + P_2 = 1. \tag{29}$$

Also, equating the rate of customers coming to the system with the rate of customers leaving the system, we get

$$\lambda = \frac{1}{\mu_1}[1 - F(0)],$$

or

$$F(0) = 1 - \rho. \tag{30}$$

We can now solve (28) through (30) for $q$ and $P_2$ to get

$$P_2 = \frac{\rho - (1-\rho)\displaystyle\int_0^T m(x)dx}{1 + \displaystyle\int_0^T m(x)dx},$$

and

$$q = F(0) + P_2 = \frac{1}{1 + \int_0^T m(x)dx}.$$

Thus,

$$F(0) = 1 - \rho \tag{31}$$

$$f(x) = \frac{m(x)}{1 + \int_0^T m(x)dx} \qquad x < T \tag{32}$$

$$= \frac{h(x) + \int_{y=0}^{T} h(x-y)m(y)dy}{1 + \int_0^T m(x)dx} \qquad x \geq T. \tag{33}$$

Next, consider the case where $\rho \geq 1$. Here,

$$q = \frac{f(0)}{\lambda},$$

and

$$\int_0^T f(x)dx + \int_T^\infty f(x)dx = 1.$$

Thus,

$$f(0)\left\{\int_0^T m(x)dx + \rho\left[1 + \int_0^T m(x)dx\right] - \int_0^T m(x)dx\right\} = 1,$$

or

$$f(0) = \frac{1}{\rho\left[1 + \int_0^T m(x)dx\right]}, \tag{34}$$

$$f(x) = \frac{m(x)}{\rho\left[1 + \int_0^T m(x)dx\right]} \qquad x < T, \tag{35}$$

and

$$f(x) = \frac{h(x) + \int_0^T h(x-y)m(y)dy}{\rho\left[1 + \int_0^T m(x)dx\right]} \qquad x \geq T. \tag{36}$$

We now consider a special case where $\bar{G}(x) = e^{-\mu x}$, $\mu = 1/\mu_1$. Then,

$$h(x) = \lambda e^{-\mu x} \qquad 0 < x < \infty,$$

$$m(x) = \lambda e^{(\lambda-\mu)x} \qquad 0 < x < \infty,$$

$$1 + \int_0^T m(x)dx = \frac{1}{(1-\rho)}[1 - \rho e^{-\mu(1-\rho)T}],$$

and, for $x \geq T$,

$$\begin{aligned} h(x) + \int_0^T h(x-y)m(y)dy &= \lambda e^{-\mu x} + \lambda^2 \int_0^T e^{-\mu(x-y)} e^{-(\mu-\lambda)y} dy \\ &= \lambda e^{-\mu x} + \lambda^2 e^{-\mu x} \int_0^T e^{+\lambda y} dy \\ &= \lambda e^{-\mu x + \lambda T}. \end{aligned}$$

Thus, for $\rho < 1$

$$F(0) = 1 - \rho$$

$$f(x) = \frac{1-\rho}{1 - \rho e^{-\mu(1-\rho)T}} e^{-\mu(1-\rho)x} \qquad x < T,$$

and

$$f(x) = \frac{1-\rho}{1 - \rho e^{-\mu(1-\rho)T}} e^{-\mu x + \lambda T} \qquad x \geq T.$$

For $\rho \geq 1$

$$F(0) = 0,$$

$$f(x) = \frac{1-\rho}{\rho[1 - \rho e^{\mu(\rho-1)T}]} e^{\mu(\rho-1)x} \qquad x < T,$$

and

$$f(x) = \frac{1-\rho}{\rho[1-\rho e^{\mu(\rho-1)T}]} e^{-\mu x+\lambda T} \qquad x \geq T.$$

## VI. WAITING-TIME DISTRIBUTION

We now use the results of Section V to obtain an expression for the waiting-time distribution and its Laplace Stieltjes Transform. From eqs. (4), (5) and (6) we get

$$W(0) = F(0) = \begin{cases} 1-\rho & \rho < 1 \\ 0 & \rho \geq 1, \end{cases} \tag{37}$$

$$w(x) = f(x) = \begin{cases} \dfrac{m(x)}{\left[1 + \displaystyle\int_0^T m(x)dx\right]} & \rho < 1 \\ \dfrac{m(x)}{\rho\left[1 + \displaystyle\int_0^T m(x)dx\right]} & \rho \geq 1, \end{cases} \tag{38}$$

and

$$w(x) = \begin{cases} \dfrac{\displaystyle\int_T^x \left[h(y) + \int_0^T h(y-z)m(z)dz\right] b_y(x)dy}{1 + \displaystyle\int_0^T m(x)dx} & \rho < 1 \\ \dfrac{\displaystyle\int_T^x \left[h(y) + \int_0^y h(y-z)m(z)dz\right] b_y(x)dy}{\rho\left[1 + \displaystyle\int_0^T m(x)dx\right]} & \rho \geq 1. \end{cases} \tag{39}$$

Let

$$W^*(\theta) = E[e^{-\theta W}] = W(0) + \int_0^\infty e^{-\theta x} w(x)dx \qquad \text{Re } \theta > 0. \tag{40}$$

Also, for Re $\theta > 0$ let

$$g^*(\theta) = \int_0^\infty e^{-\theta x} g(x)dx \tag{41}$$

$$h^*(\theta) = \int_0^\infty e^{-\theta x} h(x)dx$$

$$= \frac{\lambda[1 - g^*(\theta)]}{\theta}, \tag{42}$$

and

$$m^*(\theta) = \int_0^\infty e^{-\theta x} m(x)dx$$

$$= \frac{h^*(\theta)}{1 - h^*(\theta)}$$

$$= \frac{\lambda[1 - g^*(\theta)]}{\theta - \lambda[1 - g^*(\theta)]}. \tag{43}$$

Let $B^*(\theta)$ be the Laplace Stieltjes Transform of the ordinary busy period in an M/G/1 queue with the arrival rate $\lambda > 0$ and the service distribution, $G$. Then $B^*$ satisfies

$$B^*(\theta) = g^*\{\theta + \lambda[1 - B^*(\theta)]\}. \tag{44}$$

Also, let $B_y^*(\theta)$ be the Laplace Stieltjes Transform of $B_y$. Then

$$B_y^*(\theta) = e^{-y\{\theta + \lambda[1 - B^*(\theta)]\}}. \tag{45}$$

We now express $W^*(\theta)$ in terms of $m$, $m^*$ and $B^*$. First, consider the case $\rho < 1$, where

$$W^*(\theta) = 1 - \rho + \frac{1}{1 + \int_0^T m(x)dx} \left\{ \int_0^T e^{-\theta x} m(x)dx \right.$$

$$\left. + \int_T^\infty e^{-\theta x} \int_{y=T}^x \left[ h(y) + \int_0^T h(y - z)m(z)dz \right] b_y(x)dydx \right\}$$

$$= 1 - \rho + \frac{1}{1 + \int_0^T m(x)dx} \left( \int_0^T e^{-\theta x} m(x)dx \right.$$

$$\left. + \frac{\theta}{\theta + \lambda[1 - B^*(\theta)]} \int_T^\infty e^{-\{\theta + \lambda[1 - B^*(\theta)]\}x} m(x)dx \right). \tag{46}$$

For $\rho > 1$,

$$W^*(\theta) = \frac{1}{\rho\left[1 + \int_0^T m(x)dx\right]}\left\{\int_0^T e^{-\theta x}m(x)dx + \frac{\theta}{\theta + \lambda[1 - B^*(\theta)]}\int_T^\infty e^{-[\theta+\lambda(1-B^*(\theta))]x}m(x)dx\right\}. \quad (47)$$

### 6.1 Mean value of the waiting time

Let $\bar{W}$ denote the mean value of the waiting time. For $\rho < 1$, every customer is eventually served. Hence, $\bar{W}$ is the average over all the customers. For $\rho > 1$, some of the arriving customers do not get served and, in this case, $\bar{W}$ is the average waiting time of those who do get served. In the first case,

$$\bar{W} = -W^{*\prime}(0^+),$$

and for the second case,

$$\bar{W} = \frac{-W^{*\prime}(0^+)}{W^*(0^+)}.$$

For $\rho < 1$, all customers are served and the order of service does not affect the mean waiting time. Thus, in this case the mean waiting time is the same as that for an M/G/1 queue with the FIFO discipline. That is,

$$\bar{W} = \frac{\lambda\mu_2}{2(1-\rho)}. \quad (48)$$

For $\rho > 1$, the busy period distribution is defective. Let

$$b_o = P\{\text{Busy period} < \infty\}.$$

Then $b_o$ is the unique solution in (0, 1) of

$$B^*(0^+) = b_o = g^*[\lambda(1 - b_o)]. \quad (49)$$

Also,

$$B^{*\prime}(0^+) = \frac{g^{*\prime}[\lambda(1-b_o)]}{1 + \lambda g^{*\prime}[\lambda(1-b_o)]} \quad (50)$$

and

$$B^{*\prime\prime}(0^+) = \frac{g^{*\prime\prime}[\lambda(1-b_o)]}{\{1 + \lambda g^{*\prime}[\lambda(1-b_o)]\}^3}. \quad (51)$$

Now, from eq. (47), we get

$$W^*(0^+) = \frac{1}{\rho\left[1 + \int_0^T m(x)dx\right]}$$

$$\cdot\left[\int_0^T m(x)dx + \frac{1}{\lambda(1-b_o)} \lim_{\theta\to 0^+} \int_T^\infty \theta e^{-\{\theta+\lambda[1-B^*(\theta)]\}x} m(x)dx\right]$$

$$= \frac{1}{\rho\left[1 + \int_0^T m(x)dx\right]}\left[\int_0^T m(x)dx + 1\right]$$

$$= \frac{1}{\rho} \tag{52}$$

and

$$-W^{*\prime}(0^+) = \frac{1}{\rho\left[1 + \int_0^T m(x)dx\right]}$$

$$\cdot\left\{\int_0^T xm(x)dx + \frac{1}{\lambda(1-b_o)}\left[1 + \int_0^T m(x)dx\right]\right\}$$

$$= \frac{1}{\rho}\left[\frac{\int_0^T xm(x)dx}{1 + \int_0^T m(x)dx} + \frac{1}{\lambda(1-b_o)}\right]. \tag{53}$$

From (52) and (53) we have

$$\bar{W} = \frac{-W^{*\prime}(0^+)}{W^*(0^+)} = \frac{\int_0^T xm(x)dx}{1 + \int_0^T m(x)dx} + \frac{1}{\lambda(1-b_o)}. \tag{54}$$

Equation (54) shows that the mean waiting time of the customers who get served is minimized by the LIFO discipline ($T = 0$).

## VII. NUMERICAL RESULTS

In this section we present some numerical results. Instead of calcu-

lating the waiting-time distribution, we calculate the quantity of interest, namely

$$V = \int_{0^-}^{\infty} P(t)dW(t)$$

for a specific $P$. Let

$$P(t) = \begin{cases} 1 & t \le T \\ e^{-\sigma(t-T)} & t > T. \end{cases} \tag{55}$$

We now evaluate $V$ for the hybrid discipline with parameter $T$ and also for the LIFO discipline. For the hybrid discipline we have

$$\begin{aligned} V_T = q \cdot \Bigg[ & \int_0^T m(x)dx + e^{\sigma T} m^*\{\sigma + \lambda[1 - B^*(\sigma)]\} \\ & \cdot (1 + h^*\{\sigma + \lambda[1 - B^*(\sigma)]\}) \\ & - e^{\sigma T}(1 + h^*\{\sigma + \lambda[1 - B^*(\sigma)]\}) \\ & \cdot \int_0^T m(y) e^{-y(\sigma+\lambda)[1-B^*(\sigma)]} dy \Bigg], \end{aligned} \tag{56}$$

where

$$q = \begin{cases} \dfrac{1}{1 + \displaystyle\int_0^T m(x)dx} & \rho < 1 \qquad (57) \\[2ex] \dfrac{1}{\rho\left[1 + \displaystyle\int_0^T m(x)dx\right]} & \rho > 1. \qquad (58) \end{cases}$$

We need some more notation before writing an expression for $V_L$. Let $b_{\mathrm{FR}}(\cdot)$ denote the density function for the busy period started by the forward recurrence time of the service time. Thus,

$$b_{\mathrm{FR}}(x) = \int_0^x b_y(x) \frac{1 - G(y)}{\mu_1} dy. \tag{59}$$

Then, for $\rho < 1$,

$$\begin{aligned} V_L &= 1 - \rho + \rho \int_0^T b_{\mathrm{FR}}(x) + \rho e^{\sigma T} \int_T^{\infty} e^{-\sigma x} b_{\mathrm{FR}}(x)dx \\ &= 1 - \rho + \rho \left[ \int_0^T b_{\mathrm{FR}}(x)dx - e^{\sigma T} \int_0^T e^{-\sigma x} b_{\mathrm{FR}}(x)dx \right] \\ &\quad + e^{\sigma T} \frac{\lambda[1 - B^*(\sigma)]}{\sigma + \lambda[1 - B^*(\sigma)]}, \end{aligned} \tag{60}$$

and, for $\rho > 1$,

$$V_L = \int_0^T b_{FR}(x)dx - e^{\sigma T}\int_0^T e^{-\sigma x}b_{FR}(x)dx + \frac{1 - B^*(\sigma)}{\mu_1\{\sigma + \lambda[1 - B^*(\sigma)]\}}. \tag{61}$$

For numerical calculations we considered all integrals in eqs. (56) through (58) and (60) through (61) as functions of $T$, obtained their Laplace transforms, and inverted the transform at the specified value of $T$ using the inversion method of D. Jagerman.[3] Thus, let

$$R_1(t) = \int_0^t m(x)dx,$$

$$R_2(t) = \int_0^t m(x)e^{-x(\sigma+\lambda[1-B^*(\sigma)])}dx,$$

$$R_3(t) = \int_0^t b_{FR}(x)dx,$$

and

$$R_4(t) = \int_0^t e^{-\sigma x}b_{FR}(x)dx.$$

Also, for $i = 1, 2, 3, 4$, and $\theta$ in the appropriate domain, let

$$R_i^*(\theta) = \int_0^\infty e^{-\theta t}R_i(t)dt.$$

Then

$$R_1^*(\theta) = \frac{m^*(\theta)}{\theta},$$

$$R_2^*(\theta) = \frac{m^*\{\theta + \sigma + \lambda[1 - B^*(\sigma)]\}}{\theta},$$

$$R_3^*(\theta) = \frac{1 - B^*(\theta)}{\theta\{\theta + \lambda[1 - B^*(\theta)]\}},$$

and

$$R_4^*(\theta) = \frac{1 - B^*(\theta + \sigma)}{\sigma + \theta + \lambda[1 - B^*(\theta + \sigma)]}.$$

For numerical examples we had the service time distribution gamma with mean 1 and variance $1/K$. We used two different values of $K$, $K = 1$ (exponential distribution), and $K = 10$. We used two different

values of $T$, 1 and 3. Finally, we used two values of $\sigma$, 2.0 and 0.15. These give us eight parameter sets. The values of $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ are given in Figs. 4 through 11. From these figures we observe:

(*i*) For both service disciplines, the throughput of good calls is larger for larger $T$, smaller $\sigma$, and larger $K$. The behavior of the throughput with respect to $T$ and $\sigma$ is obvious. Larger $K$ implies smaller variability in the service time, thus reducing the probability of a customer getting served after a long wait. This, in turn, results in a higher throughput.

(*ii*) For the assumed customer behavior, the hybrid discipline always provides higher throughput than the LIFO discipline does. The difference is larger for larger $T$, larger $\sigma$, and larger $K$.

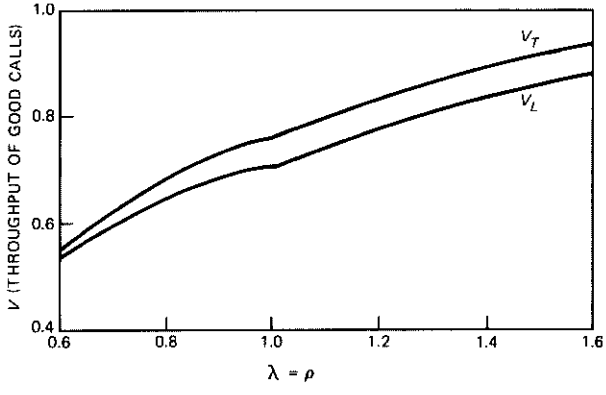



Fig. 4—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 3.0$, $\sigma = 2.0$, and $K = 1$.

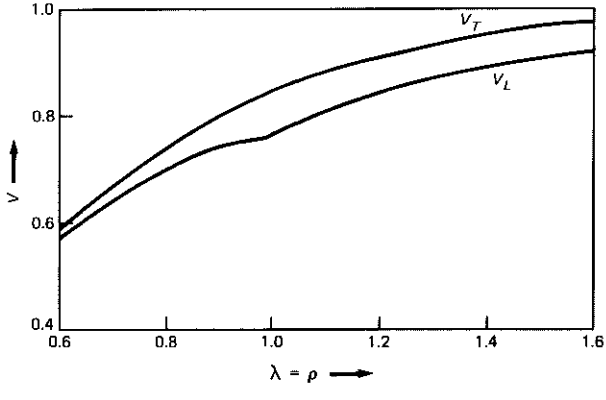



Fig. 5—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 3.0$, $\sigma = 2.0$, and $K = 10$.

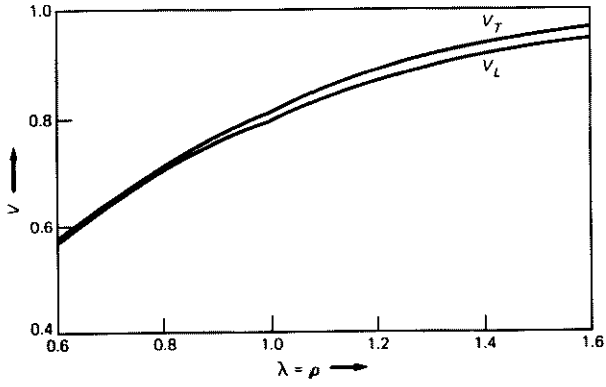



Fig. 6—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 3.0$, $\sigma = 0.15$, and $K = 1$.

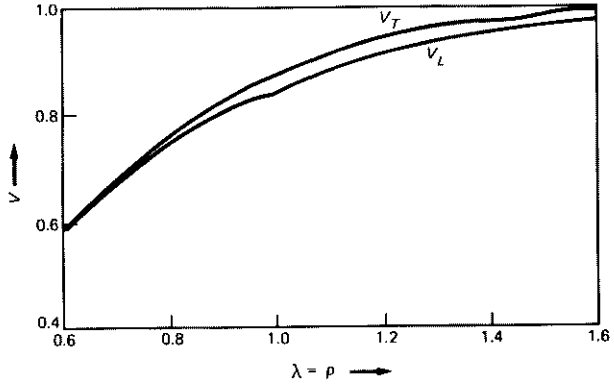



Fig. 7—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 3.0$, $\sigma = 0.15$, and $K = 10$.

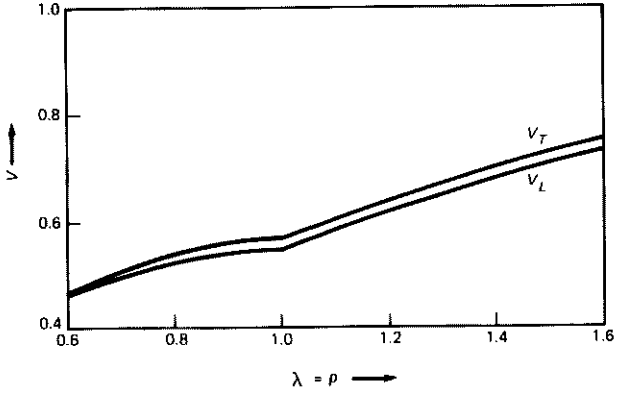



Fig. 8—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 1.0$, $\sigma = 2.0$, and $K = 1$.

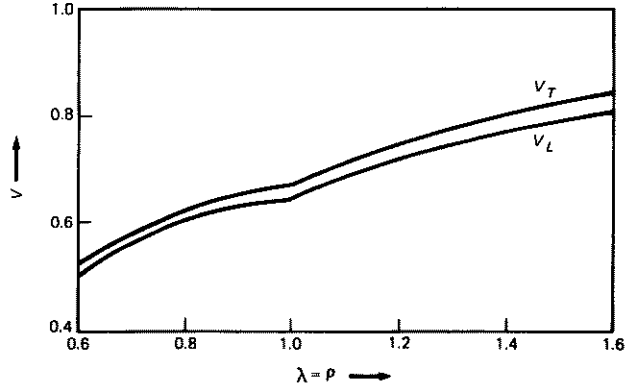



Fig. 9—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 1.0$, $\sigma = 2.0$, and $K = 10$.

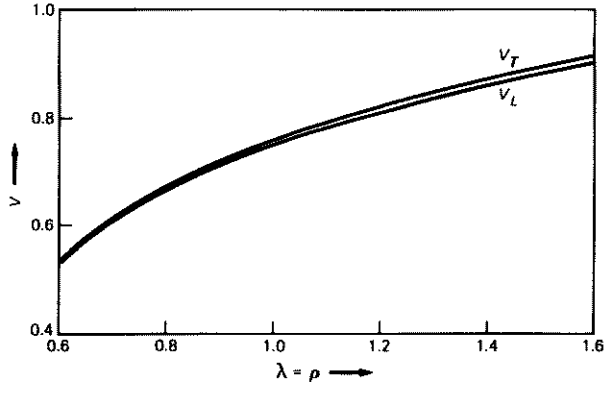



Fig. 10—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 1.0$, $\sigma = 0.15$, and $K = 1$.

Of course, our knowledge of the customer behavior may be more or less accurate, depending on the application. An issue of interest then is the sensitivity of the throughput to the assumed customer behavior. This was studied for a special case ($K = 1$) in Ref. 1. The analysis in this paper can be used to answer such issues for more general service time distributions. Qualitatively, however, the conclusions will remain the same: the last-in-first-out (LIFO) discipline is robust with respect to the knowledge of customer behavior. The hybrid discipline, on the other hand, is very sensitive to the customer behavior and should be used only when the customer behavior is adequately known and does not change in time, or when the parameters of the customer behavior can be estimated and used to change the control parameters dynamically.

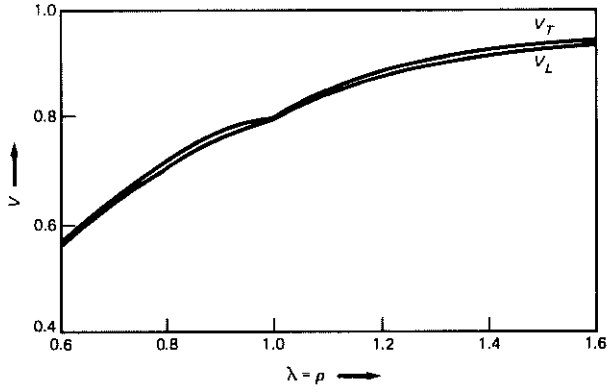



Fig. 11—The values $V_T$ and $V_L$ as functions of the load $\lambda = \rho$ for $T = 1.0$, $\sigma = 0.15$, and $K = 10$.

## REFERENCES


1. B. T. Doshi and E. H. Lipper, "Comparisons of Service Disciplines in a Queueing System with Delay Dependent Customer Behavior," Proceedings of "Applied Probability, Computer Science, The Interface" Conf., *II,* Birkhauser, 1982, pp. 269–301.
2. P. H. Brill and M. J. M. Posner, "Level Crossings in Point Processes Applied to Queues: Single Server Case," Opns. Res., *25* (July 1977), pp. 662–74.
3. S. Karlin and H. Taylor, *A First Course in Stochastic Processes*, New York, NY: Academic Press, 1975, p. 183.
4. D. Jagerman, "An Inversion Technique for the Laplace Transform with Applications to Approximation," B.S.T.J., *61* (October 1982), pp. 1995–2002.

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 5, May–June 1983
*Printed in U.S.A.*

# Off-Line Quality Control in Integrated Circuit Fabrication Using Experimental Design

By M. S. PHADKE, R. N. KACKAR, D. V. SPEENEY, and M. J. GRIECO

(Manuscript received March 25, 1982)

*In this paper we describe the off-line quality control method and its application in optimizing the process for forming contact windows in 3.5-μm complementary metal-oxide semiconductor circuits. The off-line quality control method is a systematic method of optimizing production processes and product designs. It is widely used in Japan to produce high-quality products at low cost. The key steps of off-line quality control are: (i) Identify important process factors that can be manipulated and their potential working levels; (ii) perform fractional factorial experiments on the process using orthogonal array designs; (iii) analyze the resulting data to determine the optimum operating levels of the factors (both the process mean and the process variance are considered in this analysis; (iv) conduct an additional experiment to verify that the new factor levels indeed improve the quality control.*

## I. INTRODUCTION AND SUMMARY

This paper describes and illustrates the off-line quality control method, which is a systematic method of optimizing a production process. It also documents our efforts to optimize the process for forming contact windows in 3.5-μm technology complementary metal-oxide semiconductor (CMOS) circuits fabricated in the Murray Hill Integrated Circuit Design Capability Laboratory (MH ICDCL). Here, by optimization we mean minimizing the process variance while keeping the process mean on target.

A typical very large scale integrated circuit (IC) chip has thousands of contact windows (e.g., a *BELLMAC**-32 microprocessor chip has

* Trademark of Bell Laboratories.

250,000 windows on an approximately 1.5-cm$^2$ area), most of which are not redundant. It is critically important to produce windows of size very near the target dimension. (In this paper windows mean contact windows.) Windows that are not open or are too small result in loss of contact to the devices, while excessively large windows lead to shorted device features. The application of the off-line quality control method has reduced the variance of the window size by a factor of four. Also, it has substantially reduced the processing time required for the window-forming step.

This study was inspired by Professor Genichi Taguchi's visit to the Quality Theory and Systems Group in the Quality Assurance Center at Bell Laboratories during the months of August, September, and October, 1980. Professor Taguchi, director of the Japanese Academy of Quality and a recipient of the Deming award, has developed the method of off-line quality control during the last three decades. It is used routinely by many leading Japanese industries to produce high-quality products at low cost. An overview of Professor Taguchi's off-line and on-line quality control methods is given in Taguchi,[1] and Kackar and Phadke.[2] This paper documents the results of the first application of Professor Taguchi's off-line quality control method in Bell Laboratories.

The distinctive features of the off-line quality control method are experimental design using orthogonal arrays and the analysis of signal-to-noise ratios (s/n). The orthogonal array designs provide an economical way of simultaneously studying the effects of many production factors on the process mean and variance. Orthogonal array designs are fractional factorial designs with the orthogonality property defined in Section IV. The s/n is a measure of the process variability. According to Professor Taguchi,[3] by optimizing the process with respect to the s/n, we ensure that the resulting optimum process conditions are robust or stable, meaning that they have the minimum process variation.

The outline of this paper is as follows: Section II gives a brief description of the window-forming process, which is a critical step in IC fabrication. The window-forming process is generally considered to be one of the most difficult steps in terms of reproducing and obtaining uniform-size windows. Nine key process factors were identified and their potential operating levels were determined. A description of the factors and their levels is given in Section III. The total number of possible factor-level combinations is about six thousand.

The aim of the off-line quality control method is to determine a factor-level combination that gives the least variance for the window size while keeping the mean on target. To determine such a factor-level combination we performed eighteen experiments using the $L_{18}$

orthogonal array. The experimental setup is given in Section IV. These eighteen experiments correspond to eighteen factor-level combinations among the possible six thousand combinations. For each experiment, measurements were taken on the line width and the window-size control features. The resulting data were analyzed to determine the optimum factor-level combination. The measurements and the data analysis are presented in Sections V through IX.

The optimum factor levels, inferred from the data analysis, were subsequently used in fabricating the *BELLMAC*-32 microprocessor, the *BELLMAC*-4 microcomputer, and some other chips in the Murray Hill ICDCL. The experience of using these conditions is discussed in Section X.

The experiment was designed and preliminary analysis of the experimental data was performed under Professor Taguchi's guidance and collaboration.

## II. THE WINDOW-FORMING PROCESS

Fabrication of integrated circuits is a complex, lengthy process.[4] Window forming is one of the more critical steps in fabricating state of the art CMOS integrated circuits. It comes after field and gate oxides are grown; polysilicon lines have been formed; and the gate, source, and drain areas are defined by the process of doping. Figure 1 shows the windows in a cross section of a wafer. A window is a hole of about 3.5 $\mu$m diameter etched through an oxide layer of about 2 $\mu$m thickness. The purpose of the windows is to facilitate the interconnections between the gates, sources, and drains. For this reason these windows are called contact windows.

The process of forming windows through the oxide layers involves photolithography. First the *P*-glass surface is prepared by depositing

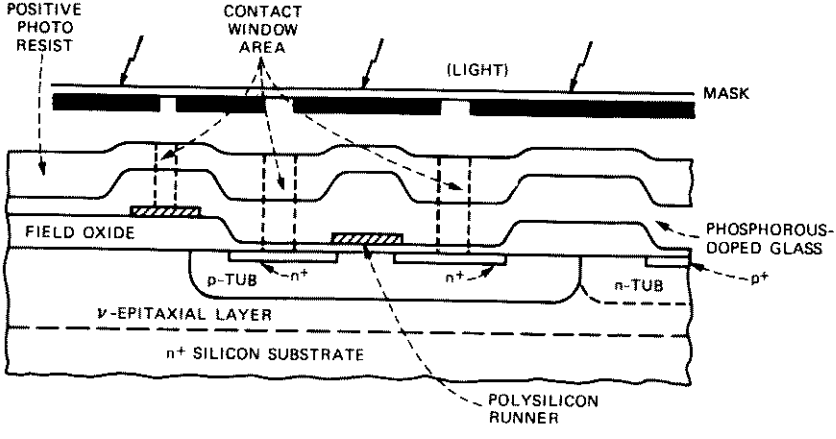



Fig. 1—Cross section of a wafer.

undoped oxide on it and prebaking it. The window-forming process is described below.

(*i*) Apply Photoresist: A wetting agent is sprayed on the wafer to promote adhesion of photoresist to the oxide surface. Then an appropriate photoresist is applied on the wafer and the wafer is rotated at high speed so that the photoresist spreads uniformly.

(*ii*) Bake: The wafer is baked to dry the photoresist layer. The thickness of the photoresist layer at this stage is about 1.3 to 1.4 $\mu$m.

(*iii*) Expose: The photoresist-coated wafer is exposed to ultraviolet radiation through a mask. The windows to be printed appear as clear areas on the mask. In addition to the windows, which are parts of the desired circuits, the mask has some test patterns. Light passes through these areas and causes the photoresist in the window areas and the test pattern areas to become soluble in an appropriate solvent (developer). The areas of the photoresist where light does not strike remain insoluble.

(*iv*) Develop: The exposed wafer is dipped in the developer, which dissolves only the exposed areas. In properly printed windows, the exposed photoresist is removed completely and the oxide surface is revealed.

(*v*) Plasma Etch: The wafers are placed in a high-vacuum chamber wherein a plasma is established. The plasma etches the exposed oxide areas faster than it etches the photoresist. So at the places where the windows are printed, windows are cut through the oxide layers down to the silicon surface.

(*vi*) Remove Photoresist: The remaining photoresist is now removed with the help of oxygen plasma and wet chemicals.

In the formation of the final contact windows there are additional steps: (*vii*) removal of cap-oxide, (*viii*) oxidation of the contact area to prevent diffusion of phosphorus in the subsequent step, (*ix*) reflow of the *P*-glass to round the window corners, (*x*) hydrogen annealing, and (*xi*) pre-metal wet-etching to remove any remaining oxides from the contact window areas.

At the time we started this study, the target window size at step 6 was considered to be 3.0 $\mu$m. The final target window size (after step *xi*) was 3.5 $\mu$m.

## III. SELECTION OF FACTORS AND FACTOR LEVELS

For the present study only the steps numbered (*i*) through (*v*) were chosen for optimization. Discussions with process engineers led to the selection of the following nine factors for controlling the window size. The factors are shown next to the appropriate fabrication steps.

(*i*) Apply Photoresist: Photoresist viscosity ($B$) and spin speed ($C$).

(*ii*) Bake: Bake temperature (*D*) and bake time (*E*).

(*iii*) Expose: Mask dimension (*A*), aperture (*F*), and exposure time (*G*).

(*iv*) Develop: Developing time (*H*).

(*v*) Plasma etch: Etch time (*I*).

No factor was chosen corresponding to the photoresist removal step because it does not affect the window size.

The standard operating levels of the nine factors are given in Table I. Under these conditions, which prevailed in September 1980, the contact windows varied substantially in size and on many occasions even failed to print and open. Figure 2 shows a typical photograph of the programmed logic array (PLA) area of a microcomputer chip. The wide variation in window size and the presence of unopened windows is obvious from the figure.

The principle of off-line quality control is to systematically investigate various possible levels for these factors with an aim of obtaining uniform-size windows.

In the window-forming experiment a number of alternate levels were considered for each of the nine factors. These levels are also listed in Table I. Six of these factors have three levels each. Three of the factors have only two levels.

Table I—Test levels

| Label | Factors Name | Levels | | |
|---|---|---|---|---|
| | | | Standard Levels | |
| A | Mask Dimension (μm) | | 2 | 2.5 |
| B | Viscosity | | 204 | 206 |
| C | Spin Speed (rpm) | Low | Normal | High |
| D | Bake Temperature (°C) | 90 | 105 | |
| E | Bake Time (min) | 20 | 30 | 40 |
| F | Aperture | 1 | 2 | 3 |
| G | Exposure Time | 20% Over | Normal | 20% Under |
| H | Developing Time (s) | 30 | 45 | 60 |
| I | Plasma Etch Time (min) | 14.5 | 13.2 | 15.8 |

Dependence of spin speed on viscosity

| | | Spin Speed (rpm) | | |
|---|---|---|---|---|
| | | Low | Normal | High |
| Viscosity | 204 | 2000 | 3000 | 4000 |
| | 206 | 3000 | 4000 | 5000 |

Dependence of exposure on aperture

| | | Exposure (PEP-Setting) | | |
|---|---|---|---|---|
| | | 20% Over | Normal | 20% Under |
| Aperture | 1 | 96 | 120 | 144 |
| | 2 | 72 | 90 | 108 |
| | 3 | 40 | 50 | 60 |

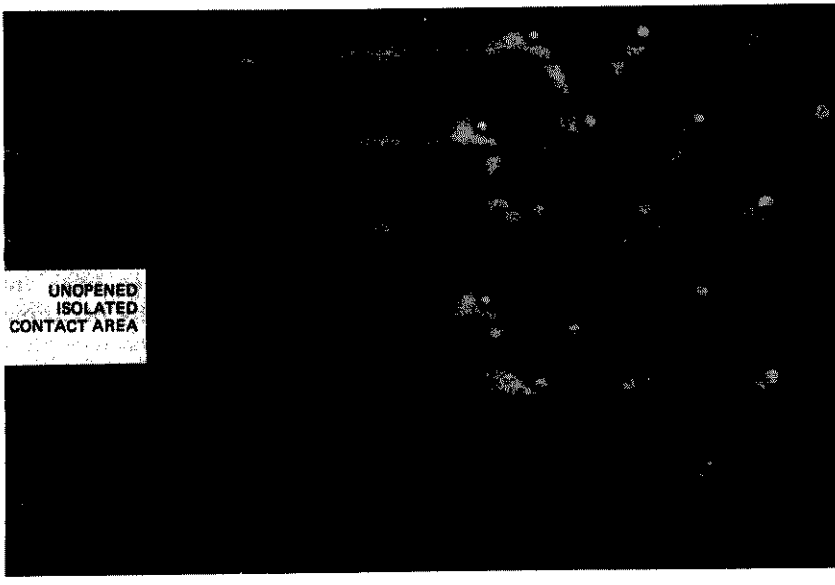



Fig. 2—Example of nonuniform contact window sizes and an isolated, unopened contact window. Both are typical results obtained in August 1980, for the PLA area of microprocessor and microcomputer chips. The contact windows are round shaped.

The levels of spin speed are tied to the levels of viscosity. For the 204 photoresist viscosity the low, normal, and high spin speeds mean 2000 rpm, 3000 rpm, and 4000 rpm, respectively. For the 206 photoresist viscosity the spin speed levels are 3000 rpm, 4000 rpm, and 5000 rpm. Likewise, the exposure setting depends on the aperture. These relationships are also shown in Table I.

## IV. THE ORTHOGONAL ARRAY EXPERIMENT

The full factorial experiment to explore all possible factor-level combinations would require $3^6 \times 2^3 = 5832$ experiments. Considering the cost of material, the time, and the availability of facilities, the full factorial experiment is prohibitively large. Also from statistical considerations it is unnecessary to perform the full factorial experiment because processes can usually be adequately characterized by a relatively few parameters.

The fractional factorial design used for this study is given in Table II. It is the $L_{18}$ orthogonal array design consisting of 18 experiments taken from Taguchi and Wu.[3] The rows of the array represent runs while the columns represent the factors. Here we treat $BD$ as a joint factor with the levels 1, 2, and 3 representing the combinations $B_1D_1$, $B_2D_1$, and $B_1D_2$, respectively. This is done so that we can study all the

Table II—The $L_{18}$ orthogonal array

| Experiment Number | Column Number & Factor | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 A | 2 BD | 3 C | 4 E | 5 F | 6 G | 7 H | 8 I |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 1 | 2 | 1 | 1 | 2 | 2 | 3 | 3 |
| 5 | 1 | 2 | 2 | 2 | 3 | 3 | 1 | 1 |
| 6 | 1 | 2 | 3 | 3 | 1 | 1 | 2 | 2 |
| 7 | 1 | 3 | 1 | 2 | 1 | 3 | 2 | 3 |
| 8 | 1 | 3 | 2 | 3 | 2 | 1 | 3 | 1 |
| 9 | 1 | 3 | 3 | 1 | 3 | 2 | 1 | 2 |
| 10 | 2 | 1 | 1 | 3 | 3 | 2 | 2 | 1 |
| 11 | 2 | 1 | 2 | 1 | 1 | 3 | 3 | 2 |
| 12 | 2 | 1 | 3 | 2 | 2 | 1 | 1 | 3 |
| 13 | 2 | 2 | 1 | 2 | 3 | 1 | 3 | 2 |
| 14 | 2 | 2 | 2 | 3 | 1 | 2 | 1 | 3 |
| 15 | 2 | 2 | 3 | 1 | 2 | 3 | 2 | 1 |
| 16 | 2 | 3 | 1 | 3 | 2 | 3 | 1 | 2 |
| 17 | 2 | 3 | 2 | 1 | 3 | 1 | 2 | 3 |
| 18 | 2 | 3 | 3 | 2 | 1 | 2 | 3 | 1 |

nine factors with the $L_{18}$ orthogonal array. Thus, experiment 2 would be run under level 1 of factors *A*, *B*, and *D*, and level 2 of the remaining factors. In terms of the actual settings, these conditions are: 2-$\mu$m mask dimension, 204 viscosity, 90°C bake temperature, 3000-rpm spin speed, bake time of 30 minutes, aperture 2, exposure PEP setting 90, 45-second developing time, and 13.2 minutes of plasma etch. The other rows are interpreted similarly.

Here are some of the properties and considerations of this design:

(*i*) This is a main-effects-only design; i.e., the response is approximated by a separable function. A function of many independent variables is called separable if it can be written as a sum of functions where each component function is a function of only one independent variable.

(*ii*) For estimating the main effects there are two degrees of freedom associated with each three-level factor, one degree of freedom for each two-level factor, and one degree of freedom with the overall mean. We need at least one experiment for every degree of freedom. Thus, the minimum number of experiments needed is $2 \times 6 + 1 \times 3 + 1 = 16$. Our design has 18 experiments. A single-factor-by-single-factor experiment would need only 16 experiments, two fewer than 18. But such an experiment would yield far less precise information compared with the orthogonal array experiment.[3,5]

(*iii*) The columns of the array are pairwise orthogonal. That is, in every pair of columns, all combinations of levels occur and they occur an equal number of times.

(*iv*) Consequently, the estimates of the main effects of all factors

as shown in Table II and their associated sums of squares are independent under the assumption of normality and equality of error variance. So the significance tests for these factors are independent. Though $BD$ is treated as a joint factor, the main effects and sums of squares of $B$ and $D$ can be estimated separately under the assumption of no interaction. In general, these estimates would be correlated with each other. However, these estimates are not correlated with those for any of the other seven factors.

(*v*) The estimates of the main effects can be used to predict the response for any combination of the parameter levels. A desirable feature of this design is that the variance of the prediction error is the same for all parameter-level combinations covered by the full factorial design.

(*vi*) It is known that the main-effect-only models are liable to give misleading conclusions in the presence of interactions. However, in the beginning stages of this study the interactions are assumed to be negligible. If we wished to study all two-factor interactions, with no more than 18 experiments we would have enough degrees of freedom for studying only two three-level factors, or five two-level factors! That would mean in the present study we would have to eliminate half of the process factors without any experimental evidence. Alternately, if we wished to study all the nine process factors and their two-factor interactions, we would need at least 109 experiments! Orthogonal array designs can, of course, be used to study interactions.[3]

(*vii*) Optimum conditions obtained from such an experiment have to be verified with an additional experiment. This is done to safeguard us against the potential adverse effects of ignoring the interactions among the manipulatable factors.

In conducting experiments of this kind, it is common for some wafers to get damaged or broken. Also, the wafer-to-wafer variability of window sizes is typically large. So we decided to run each experiment with two wafers.

#### *4.1 Analysis of variance*

Data collected from such experiments are analyzed by a method called analysis of variance (ANOVA).[6] The purpose of ANOVA is to separate the total variability of the data, which is measured by the sum of the squared deviations from the mean value, into contributions by each of the factors and the error. This is analogous to the use of Parseval's theorem to separate the signal strength into contributions by the various harmonics.[3] To see which of the factors have a significant effect, $F$-tests are performed. In performing the standard $F$-test we assume that the errors are normally distributed with equal variance and are independent. The results of the $F$-test are indicated by the significance level. When we say that a factor is significant at 5-percent

level we mean that there is 5 percent or less chance that, if we change the level of the factor, the response will remain the same. If the $F$-test indicates that a factor is not significant at the 5-percent level it means that, if we change the level of that factor, there is more than a 5-percent chance that the response will remain the same.

The levels of factors which are identified as significant are then set to obtain the best response. The levels of the other factors can be set at any levels within the experimental range. We choose to leave them at the starting levels.

If the assumptions of the $F$-test are not completely satisfied, the quoted significances are not accurate. However, the standard $F$-test is relatively insensitive to deviations from the assumptions used in its derivation. Thus, for making engineering decisions about which factor levels to change, the accuracy of the significance level is an adequate guide. In this paper we will use the standard $F$-test even though some of the assumptions are not strictly satisfied.

## V. QUALITY MEASURES

The window size is the relevant quality measure for this experiment. The existing equipment does not give reproducible measurements of the sizes of windows in the functional circuits on a chip. This is because of the small size of these windows and their close proximity to one another. Therefore, test patterns—a line-width pattern and a window pattern—are provided in the upper left-hand corner of each chip. The following measurements were made on these test patterns to indicate the quality.

(*i*) Line width after step (*iv*), called the pre-etch line width or photo-line width.

(*ii*) Line width after step (*vi*), called the post-etch line width.

(*iii*) Size of the window test pattern after step (*vi*), called the post-etch window size.

Five chips were selected from each wafer for making the above measurements. These chips correspond to specific locations on a wafer—top, bottom, left, right, and center.

All three quality measures are considered to be good indicators of the size of the functional windows. However, between the geometries of the window-size pattern and the line-width pattern, the geometry of the window-size pattern is closer to the geometry of the functional windows. So, among the three quality measures, the post-etch window size may be expected to be better correlated with the size of the functional windows.

## VI. EXPERIMENTAL DATA

Only thirty-four wafers were available for experimentation. So experiments 15 and 18 were arbitrarily assigned only one wafer each. One

Table III—Experimental data

| Experiment No. | Line-Width Control Feature Photoresist—Nanoline Tool (Micrometers) | | | | | Comments |
|---|---|---|---|---|---|---|
| | Top | Center | Bottom | Left | Right | |
| 1 | 2.43 | 2.52 | 2.63 | 2.52 | 2.5 | |
| 1 | 2.36 | 2.5 | 2.62 | 2.43 | 2.49 | |
| 2 | 2.76 | 2.66 | 2.74 | 2.6 | 2.53 | |
| 2 | 2.66 | 2.73 | 2.95 | 2.57 | 2.64 | |
| 3 | 2.82 | 2.71 | 2.78 | 2.55 | 2.36 | |
| 3 | 2.76 | 2.67 | 2.9 | 2.62 | 2.43 | |
| 4 | 2.02 | 2.06 | 2.21 | 1.98 | 2.13 | |
| 4 | 1.85 | 1.66 | 2.07 | 1.81 | 1.83 | |
| 5 | — | — | — | — | — | Wafer Broke |
| 5 | 1.87 | 1.78 | 2.07 | 1.8 | 1.83 | |
| 6 | 2.51 | 2.56 | 2.55 | 2.45 | 2.53 | |
| 6 | 2.68 | 2.6 | 2.85 | 2.55 | 2.56 | |
| 7 | 1.99 | 1.99 | 2.11 | 1.99 | 2.0 | |
| 7 | 1.96 | 2.2 | 2.04 | 2.01 | 2.03 | |
| 8 | 3.15 | 3.44 | 3.67 | 3.09 | 3.06 | |
| 8 | 3.27 | 3.29 | 3.49 | 3.02 | 3.19 | |
| 9 | 3.0 | 2.91 | 3.07 | 2.66 | 2.74 | |
| 9 | 2.73 | 2.79 | 3.0 | 2.69 | 2.7 | |
| 10 | 2.69 | 2.5 | 2.51 | 2.46 | 2.4 | |
| 10 | 2.75 | 2.73 | 2.75 | 2.78 | 3.03 | |
| 11 | 3.2 | 3.19 | 3.32 | 3.2 | 3.15 | |
| 11 | 3.07 | 3.14 | 3.14 | 3.13 | 3.12 | |
| 12 | 3.21 | 3.32 | 3.33 | 3.23 | 3.10 | |
| 12 | 3.48 | 3.44 | 3.49 | 3.25 | 3.38 | |
| 13 | 2.6 | 2.56 | 2.62 | 2.55 | 2.56 | |
| 13 | 2.53 | 2.49 | 2.79 | 2.5 | 2.56 | |
| 14 | 2.18 | 2.2 | 2.45 | 2.22 | 2.32 | |
| 14 | 2.33 | 2.2 | 2.41 | 2.37 | 2.38 | |
| 15 | 2.45 | 2.50 | 2.51 | 2.43 | 2.43 | |
| 15 | — | — | — | — | — | No wafer |
| 16 | 2.67 | 2.53 | 2.72 | 2.7 | 2.6 | |
| 16 | 2.76 | 2.67 | 2.73 | 2.69 | 2.6 | |
| 17 | 3.31 | 3.3 | 3.44 | 3.12 | 3.14 | |
| 17 | 3.12 | 2.97 | 3.18 | 3.03 | 2.95 | |
| 18 | 3.46 | 3.49 | 3.5 | 3.45 | 3.57 | |
| 18 | — | — | — | — | — | No wafer |

of the wafers assigned to experiment 5 broke in handling. So experiments 5, 15, and 18 have only one wafer.

The experimental data are shown in Table III.

The data arising from such experiments can be classified as two types—continuous data and categorical data. Here, the pre-etch and the post-etch line-width data are of the continuous type. The post-etch window size data are mixed categorical-continuous type, because some windows are open while some are not. The two types of data are analyzed somewhat differently, as we explain the following two sections.

## VII. ANALYSIS OF THE LINE-WIDTH DATA

Both the pre-etch and the post-etch line widths are continuous variables. For each of these variables the statistics of interest are the

Table III—Experimental data (Continued)

| Experiment No. | Line-Width Control Feature Etched—Nanoline Tool (Micrometers) | | | | | Comments |
|---|---|---|---|---|---|---|
| | Top | Center | Bottom | Left | Right | |
| 1 | 2.95 | 2.74 | 2.85 | 2.76 | 2.7 | |
| 1 | 3.03 | 2.95 | 2.75 | 2.82 | 2.85 | |
| 2 | 3.05 | 3.18 | 3.2 | 3.16 | 3.06 | |
| 2 | 3.25 | 3.15 | 3.09 | 3.11 | 3.16 | |
| 3 | 3.69 | 3.57 | 3.78 | 3.55 | 3.40 | |
| 3 | 3.92 | 3.62 | 3.71 | 3.71 | 3.53 | |
| 4 | 2.68 | 2.62 | 2.9 | 2.45 | 2.7 | |
| 4 | 2.29 | 2.31 | 2.77 | 2.46 | 2.49 | |
| 5 | — | — | — | — | — | Wafer Broke |
| 5 | 1.75 | 1.15 | 2.07 | 2.12 | 1.53 | |
| 6 | 3.42 | 2.98 | 3.22 | 3.13 | 3.17 | |
| 6 | 3.34 | 3.21 | 3.23 | 3.25 | 3.28 | |
| 7 | 2.62 | 2.49 | 2.53 | 2.41 | 2.51 | |
| 7 | 2.76 | 2.94 | 2.68 | 2.62 | 2.51 | |
| 8 | 4.13 | 4.38 | 4.41 | 4.03 | 4.03 | |
| 8 | 4.0 | 4.02 | 4.18 | 3.92 | 3.91 | |
| 9 | 3.94 | 3.82 | 3.84 | 3.57 | 3.71 | |
| 9 | 3.44 | 3.30 | 3.41 | 3.28 | 3.20 | |
| 10 | 3.17 | 2.85 | 2.84 | 3.06 | 2.94 | |
| 10 | 3.70 | 3.34 | 3.45 | 3.41 | 3.29 | |
| 11 | 4.01 | 3.91 | 3.92 | 3.80 | 3.90 | |
| 11 | 3.67 | 3.31 | 2.86 | 3.41 | 3.23 | |
| 12 | 4.04 | 3.80 | 4.08 | 3.81 | 3.94 | |
| 12 | 4.51 | 4.37 | 4.45 | 4.24 | 4.48 | |
| 13 | 3.40 | 3.12 | 3.11 | 3.25 | 3.06 | |
| 13 | 3.22 | 3.03 | 2.89 | 2.92 | 2.98 | |
| 14 | 3.18 | 3.03 | 3.4 | 3.17 | 3.32 | |
| 14 | 3.18 | 2.83 | 3.17 | 3.07 | 3.02 | |
| 15 | 2.86 | 2.46 | 2.3 | 2.6 | 2.55 | |
| 15 | — | — | — | — | — | No wafer |
| 16 | 2.85 | 2.14 | 1.22 | 2.8 | 3.03 | |
| 16 | 3.4 | 2.97 | 2.96 | 2.87 | 2.88 | |
| 17 | 4.06 | 3.87 | 3.90 | 3.94 | 3.87 | |
| 17 | 4.02 | 3.49 | 3.51 | 3.69 | 3.47 | |
| 18 | 4.49 | 4.28 | 4.34 | 4.39 | 4.25 | |
| 18 | — | — | — | — | — | No wafer |

mean and the standard deviation. The objective of our data analysis is to determine the factor-level combination such that the standard deviation is minimum while keeping the mean on target. We will call this the optimum factor-level combination. Professor Taguchi's method for obtaining the optimum combination is given next.

### *7.1 Single response variable*

Let us first consider the case where there is only one response variable. Instead of working with the mean and the standard deviation, it is preferable to work with the transformed variables—the mean and the signal-to-noise ratio (s/n). The s/n is defined as

$$\text{s/n} = \log_{10}\left(\frac{\text{Mean}}{\text{Standard Deviation}}\right)$$

$$= -\log_{10}(\text{coefficient of variation}).$$

Table III—Experimental data (Continued)

| Experiment No. | Window-Control Feature Etched—Vickers Tool (Micrometers) | | | | | Comments |
|---|---|---|---|---|---|---|
| | Top | Center | Bottom | Left | Right | |
| 1 | WNO* | WNO | WNO | WNO | WNO | |
| 1 | WNO | WNO | WNO | WNO | WNO | |
| 2 | 2.32 | 2.23 | 2.30 | 2.56 | 2.51 | |
| 2 | 2.22 | 2.33 | 2.34 | 2.15 | 2.35 | |
| 3 | 2.98 | 3.14 | 3.02 | 2.89 | 3.16 | |
| 3 | 3.15 | 3.08 | 2.78 | WNO | 2.86 | |
| 4 | WNO | WNO | WNO | WNO | WNO | |
| 4 | WNO | WNO | WNO | WNO | WNO | |
| 5 | — | — | — | — | — | Wafer Broke |
| 5 | WNO | WNO | WNO | WNO | WNO | |
| 6 | 2.45 | 2.19 | 2.14 | 2.32 | 2.12 | |
| 6 | WNO | WNO | WNO | WNO | WNO | |
| 7 | WNO | WNO | WNO | WNO | WNO | |
| 7 | WNO | WNO | WNO | WNO | WNO | |
| 8 | WNO | WNO | WNO | WNO | WNO | |
| 8 | 2.89 | 2.97 | 3.13 | 3.25 | 3.19 | |
| 9 | 3.16 | 2.91 | 3.12 | 3.18 | 3.11 | |
| 9 | 2.43 | 2.35 | 2.14 | 2.40 | 2.28 | |
| 10 | 2.0 | 1.75 | 1.97 | 1.91 | 1.72 | |
| 10 | WNO | 2.7 | WNO | 2.61 | 2.73 | |
| 11 | 2.76 | 3.09 | 3.22 | 3.05 | 3.04 | |
| 11 | 3.12 | 3.21 | WNO | 2.71 | 2.27 | |
| 12 | 3.24 | 3.08 | WNO | 2.89 | 2.72 | |
| 12 | 3.5 | 3.71 | 3.52 | 3.53 | 3.71 | |
| 13 | 2.54 | 2.63 | 2.88 | 2.31 | 2.71 | |
| 13 | WNO | WNO | WNO | WNO | WNO | |
| 14 | WNO | 1.74 | 2.24 | 2.07 | 2.38 | |
| 14 | WNO | WNO | WNO | WNO | WNO | |
| 15 | WNO | WNO | WNO | WNO | WNO | |
| 15 | — | — | — | — | — | No wafer |
| 16 | WNO | WNO | WNO | WNO | WNO | |
| 16 | WNO | WNO | WNO | WNO | WNO | |
| 17 | 3.09 | 2.91 | 3.06 | 3.09 | 3.29 | |
| 17 | 3.39 | 2.5 | 2.57 | 2.62 | 2.35 | |
| 18 | 3.39 | 3.34 | 3.45 | 3.44 | 3.33 | |
| 18 | — | — | — | — | — | No wafer |

* WNO—Window not open.

In terms of the transformed variables, the optimization problem is to determine the optimum factor levels such that the s/n is maximum while keeping the mean on target. This problem can be solved in two stages:

(*i*) Determine which factors have a significant effect on the s/n. This is done through the analysis of variance (ANOVA) of the s/n. These factors are called the *control factors*, implying that they control the process variability. For each control factor we choose the level with the highest s/n as the optimum level. Thus the overall s/n is maximized.

(*ii*) Select a factor that has the smallest effect on the s/n among all factors that have a significant effect on the mean. Such a factor is called a *signal factor*. Ideally, the signal factor should have no effect

on the s/n. Choose the levels of the remaining factors (factors that are neither control factors nor signal factors) to be the nominal levels prior to the optimization experiment. Then set the level of the signal factor so that the mean response is on target.

In practice, the following two aspects should also be considered in selecting the signal factor: (*i*) If possible, the relationship between the mean response and the levels of the signal factor should be linear, and (*ii*) It should be convenient to change the signal factor during production. These aspects are important from the on-line quality control considerations. The signal factor can be used during manufacturing to adjust the mean response.[1-3]

Why do we work in terms of the s/n ratio rather than the standard deviation? Frequently, as the mean decreases, the standard deviation also decreases and vice versa. In such cases, if we work in terms of the standard deviation, the optimization cannot be done in two steps; i.e., we cannot minimize the standard deviation first and then bring the mean on target.

Through many applications, Professor Taguchi has empirically found that the two-stage optimization procedure involving the s/n indeed gives the parameter-level combination where the standard deviation is minimum, while keeping the mean on target. This implies that the engineering systems behave in such a way that the manipulatable production factors can be divided into three categories:

(*i*) Control factors, which affect process variability as measured by the s/n

(*ii*) Signal factors, which do not influence (or have negligible effect on) the s/n but have a significant effect on the mean

(*iii*) Factors that do not affect the s/n or the process mean.

The two-stage procedure also has an advantage over a procedure that directly minimizes the mean square error from the target mean value. In practice, the target mean value may change during the process development. The advantage of the two-stage procedure is that for any target mean value (of course, within reasonable bounds) the new optimum factor-level combination is obtained by suitably adjusting the level of only the signal factor. This is so because in step (*i*) of the algorithm the coefficient of variation is minimized for every mean target value.

### *7.2 Multiple response variables*

Now let us consider the case where there are two or more response variables. In such cases, engineering judgment may have to be used to resolve the conflict if different response variables suggest different levels for any one factor. The modified two-stage procedure is as follows:

(*i*) Separately determine control factors and their optimum levels corresponding to each response variable. If there is a conflict between the optimum levels suggested by the different response variables, use engineering judgment to resolve the conflict.

(*ii*) Select a factor that has the smallest effect (preferably no effect) on the signal-to-noise ratios for all the response variables but has a significant effect on the mean levels. This is the signal factor. Set the levels of the remaining factors, which affect neither the mean nor the s/n, at the nominal levels prior to the optimization experiment. Then set the level of the signal factor so that the mean responses are on target. Once again engineering judgment may have to be used to resolve any conflicts that arise.

The selection of the control factors, signal factor, and their optimum levels for the present application will be discussed in Section IX. The remaining portions of Sections VII and VIII contain the data analysis that forms the basis for selecting the optimum factor levels.

### *7.3 Pre-etch line width*

Mean, standard deviation, and s/n were calculated for each of the eighteen experiments. For those experiments with two wafers, ten data points were used in these calculations. When there was only one wafer, five data points were used. These results are shown in Table IV. The presence of unequal sample sizes has been ignored in the subsequent analysis. Let $\bar{x}_i$ and $\eta_i$ denote the mean and the s/n for the $i$th experiment.

Table IV—Pre-etch line-width data

| Experiment Number | Mean Line Width, $\bar{x}$ ($\mu$m) | Standard Deviation of Line Width, $s$ ($\mu$m) | s/n $\eta = \log(\bar{x}/s)$ |
|---|---|---|---|
| 1 | 2.500 | 0.0827 | 1.4803 |
| 2 | 2.684 | 0.1196 | 1.3512 |
| 3 | 2.660 | 0.1722 | 1.1889 |
| 4 | 1.962 | 0.1696 | 1.0632 |
| 5 | 1.870 | 0.1168 | 1.2043 |
| 6 | 2.584 | 0.1106 | 1.3686 |
| 7 | 2.032 | 0.0718 | 1.4520 |
| 8 | 3.267 | 0.2101 | 1.1917 |
| 9 | 2.829 | 0.1516 | 1.2709 |
| 10 | 2.660 | 0.1912 | 1.1434 |
| 11 | 3.166 | 0.0674 | 1.6721 |
| 12 | 3.323 | 0.1274 | 1.4165 |
| 13 | 2.576 | 0.0850 | 1.4815 |
| 14 | 2.308 | 0.0964 | 1.3788 |
| 15 | 2.464 | 0.0385 | 1.8065 |
| 16 | 2.667 | 0.0706 | 1.5775 |
| 17 | 3.156 | 0.1569 | 1.3036 |
| 18 | 3.494 | 0.0473 | 1.8692 |

By computing a single mean $\bar{x}_i$ and a single variance $s_i^2$ (needed for computing $\eta_i$) from the two wafers of each experiment $i$, we pool together the between wafer and the within wafer variance. That is,

$$E(s_i^2) = (\text{Between wafer variance for experiment } i) \times \tfrac{5}{6} + (\text{Within wafer variance for experiment } i).$$

Thus, when we maximize $\eta$, we minimize the sum of the between-wafer and the within-wafer variances of the line width, which is the response of interest to us. There can be situations when one wants to separately estimate the effects of the factor levels on the between-wafer and within-wafer variances. In those cases, one would compute the s/n and the mean line width for each individual wafer.

In the analysis of the pre-etch and the post-etch line widths, we compute the $\bar{x}_i$ and the $s_i^2$ for each experiment by pooling the data from both wafers used in that experiment. A relative measure of the between-wafer and within-wafer variance is obtained in Section VIII, while the post-etch window-size data is being analyzed.

#### *7.3.1 Analysis of s/n*

The estimates of the average s/n for all factor levels are given in Table V. The average for the first level of factor A is the average of the nine experiments (experiments 1 through 9), which were conducted with level 1 of the factor A. Likewise, the average for the second level of factor A is the mean of experiments 10 through 18, which were conducted with level 2 of the factor A. Let us denote these average effects of $A_1$ and $A_2$ by $m_{A_1}$ and $m_{A_2}$, respectively. Here $m_{A_1} = 1.2857$ and $m_{A_2} = 1.5166$. The other entries of Table V were calculated similarly.

The average signal-to-noise ratios for every level of the eight factors are graphically shown in Fig. 3. Qualitatively speaking, the mask dimension and the aperture cause a large variation in the s/n. The

Table V—Pre-etch line width for average s/n

| | Factor | Average s/n Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| A | Mask Dimension | 1.2857 | 1.5166 | |
| BD | Viscosity Bake Temperature | ($B_1D_1$) 1.3754 | ($B_2D_1$) 1.3838 | ($B_1D_2$) 1.4442 |
| B | Viscosity | 1.4098 | 1.3838 | |
| D | Bake Temperature | 1.3796 | 1.4442 | |
| C | Spin Speed | 1.3663 | 1.3503 | 1.4868 |
| E | Bake Time | 1.4328 | 1.4625 | 1.3082 |
| F | Aperture | 1.5368 | 1.4011 | 1.2654 |
| G | Exposure Time | 1.3737 | 1.3461 | 1.4836 |
| H | Developing Time | 1.3881 | 1.4042 | 1.4111 |

Overall average s/n = 1.4011.

developing time and the viscosity cause a small change in the s/n. The effect of the other factors is in between.

For three-level factors, Fig. 3 can also be used to judge the linearity of the effect of the factors. If the difference between levels 1 and 2, and levels 2 and 3 is equal and these levels appear in proper order (1, 2, 3, or 3, 2, 1), then the effect of that factor is linear. If either the differences are unequal or the order is mixed up, then the effect is not linear. For example, the aperture has approximately linear response while the bake time has a nonlinear response.

We shall perform a formal analysis of variance (ANOVA) to identify statistically significant factors. The analysis of variance of general linear models is widely known in literature, see e.g., Searle[7] and Hicks.[6] Simple ANOVA methods for orthogonal array experiments are described in Taguchi and Wu.[3] The linear model used in analyzing this data is:

$$y_i = \mu + x_i + e_i, \tag{1}$$

where

$i = 1, \cdots, 18$ is the experiment number.

$\mu$ is the overall mean.

$x_i$ is the fixed effect of the factor-level combination used in experiment $i$. Here we consider only the main effect for each of the factors. Thus it represents the sum of the effects of the eight factors.

$e_i$ is the random error for experiment $i$.

$y_i$ is the s/n for experiment $i$.

To clarify the meaning of the term $x_i$, let us consider experiment 1, which was run at level 1 of each of the eight factors A through H. Note that the factor I is irrelevant for studying the pre-etch line width. So $x_1$ is the sum of the main effects associated with the first level of each of the factors A through H.

The sum of squares and the mean squares for the eight factors are tabulated in Table VIa. The computations are illustrated in Appendix A.

The expected mean squares are also shown in Table VIa. See Refs. 6 and 7 for the computation of expected mean squares, which are used in forming appropriate F-tests. The error variance, i.e., variance of $e_i$, is denoted by $\sigma^2$. The variability due to the factors A through H is denoted by $\phi$ with an appropriate subscript.

In Table VIa we see that the mean sum of squares for factors BD, C, G, and H are smaller than the mean error sum of squares. So a new ANOVA table, Table VIb, was formed by pooling the sum of squares of these factors with the error sum of squares. The linear model underlying the ANOVA Table VIb is the same as Eq. (1), except that

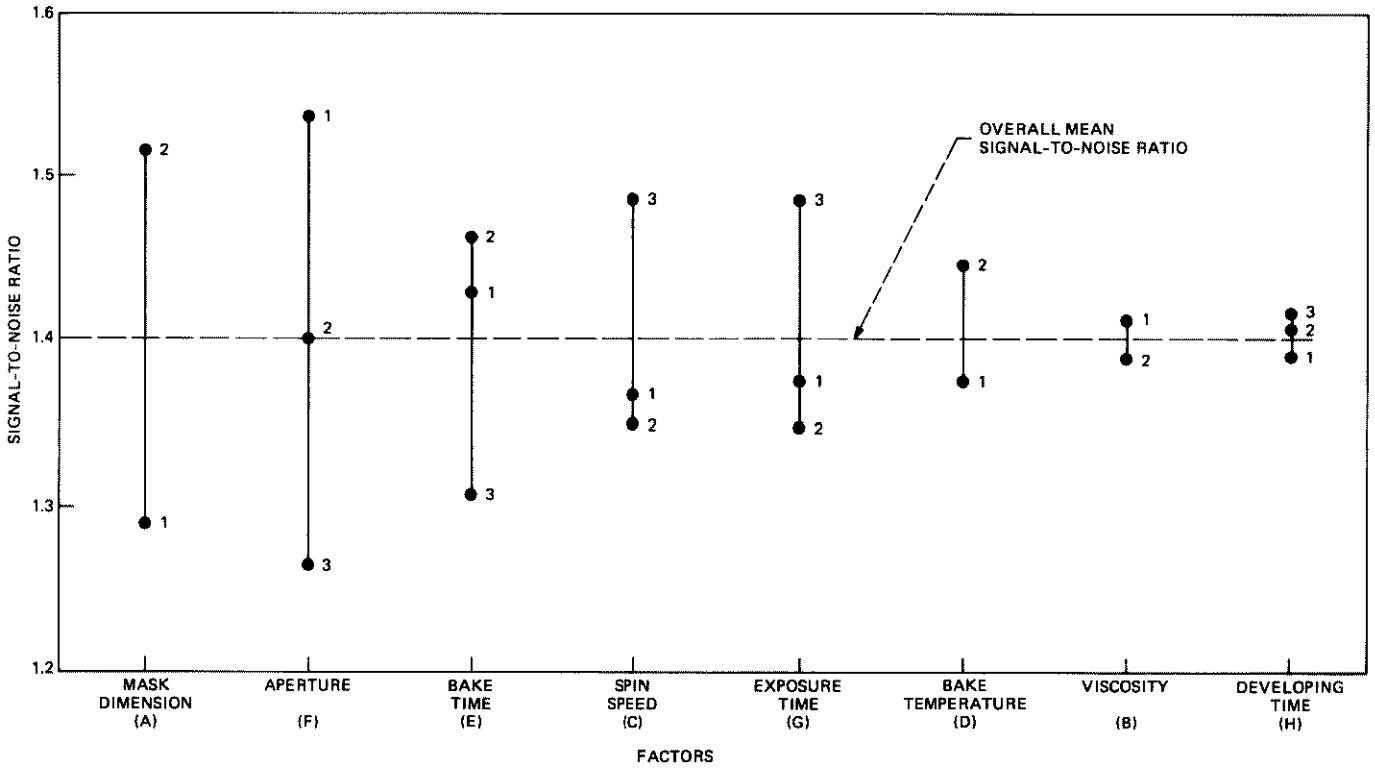



Fig. 3—Signal-to-noise ratios for pre-etch line width. The average s/n for each factor level is indicated by a dot. The number next to the dot indicates the factor level.

now $x_i$ stands for the sum of the main effects of only A, E, and F. The F ratios, computed by dividing the factor mean square by the error mean square, are also shown in Table VIb. Factors A and F are significant using F-table values for the 5-percent significance level. So the mask dimension and the aperture are the control factors.

In performing the analysis of variance, we have tacitly assumed that the response for each experiment, here the s/n, has a normal distribution with constant variance. We are presently investigating the distributional properties of the s/n and their impact on the analysis of variance. In this paper we treat the significance levels as approximate.

The engineering significance of a statistically significant factor can be measured in terms of the percent contribution, a measure introduced by Taguchi.[3] The percent contribution is equal to the percent of the total sum of squares explained by that factor after an appropriate estimate of the error sum of squares has been removed from it. The larger the percent contribution, the more can be expected to be achieved by changing the level of that factor. Computation of the percent contribution is illustrated in Appendix B, and the results are shown in Table VIb.

From Table VIb we see that both the factors A (mask dimension) and F (aperture) contribute in excess of 20 percent each to the total

Table VI—Pre-etch line width

(a) ANOVA for s/n

| | Source | Degrees of Freedom | Sum of Squares | Mean Square | Expected Mean Square |
|---|---|---|---|---|---|
| A | Mask Dimension | 1 | 0.2399 | 0.2399 | $\sigma^2 + \phi_A$ |
| BD | Viscosity Bake Temperature | 2 | 0.0169 | 0.0085 | $\sigma^2 + \phi_{BD}$ |
| C | Spin Speed | 2 | 0.0668 | 0.0334 | $\sigma^2 + \phi_C$ |
| E | Bake Time | 2 | 0.0804 | 0.0402 | $\sigma^2 + \phi_E$ |
| F | Aperture | 2 | 0.2210 | 0.1105 | $\sigma^2 + \phi_F$ |
| G | Exposure Time | 2 | 0.0634 | 0.0317 | $\sigma^2 + \phi_G$ |
| H | Developing Time | 2 | 0.0017 | 0.0009 | $\sigma^2 + \phi_H$ |
| Error | | 4 | 0.1522 | 0.0381 | $\sigma^2$ |
| Total | | 17 | 0.8423 | | |

(b) Pooled ANOVA for s/n

| | Source | Degrees of Freedom | Sum of Squares | Mean Square | F | Percent Contribution |
|---|---|---|---|---|---|---|
| A | Mask Dimension | 1 | 0.2399 | 0.2399 | 9.56* | 25.5 |
| E | Bake Time | 2 | 0.0804 | 0.0402 | 1.60 | 3.6 |
| F | Aperture | 2 | 0.2210 | 0.1105 | 4.40* | 20.3 |
| Error | | 12 | 0.3010 | 0.0251 | | 50.6 |
| Total | | 17 | 0.8423 | | | 100.00 |

$F_{1,12}(0.95) = 4.75$.
$F_{2,12}(0.95) = 3.89$.
* Factors significant at 95-percent confidence level.

sum of squares. So the factors A and F are not only statistically significant, they have a sizable influence on the s/n. These results are consistent with Fig. 3. They will be used in Section IX for selecting the control factors.

#### *7.3.2 Analysis of the means*

Now we analyze the mean pre-etch line widths, $\bar{x}_i$ values, to find a signal factor.

The estimates of the mean line widths for all factor levels are given in Table VII. These estimates are graphically shown in Fig. 4. It is apparent that the levels of viscosity, mask dimension, and spin speed cause a relatively large change in the mean line width. Developing time and aperture have a small effect on the line width. The remaining two factors have an intermediate effect.

The linear model used to analyze this data is the same as eq. (1), except that now $y_i$ stands for the mean pre-etch line width rather than the s/n.

The original and the pooled ANOVA tables for the mean pre-etch line width are given in Tables VIIIa and b, respectively. Because the design is not orthogonal with respect to the factors B and D, we need a special method, described in Appendix C, to separate $S_{BD}$ into $S_B$ and $S_D$.

It is clear from Table VIIIb that the mask dimension (A), viscosity (B), and spin speed (C) have a statistically significant effect on the mean pre-etch line width. Also, these factors together contribute more than 70 percent to the total sum of squares. These results will be used in Section IX for selecting the signal factor.

### *7.4 Post-etch line width*

The analysis of the post-etch line-width data is similar to the analysis of the pre-etch line-width data. The mean, the standard deviation, and the s/n for each experiment are shown in Table IX.

Table VII—Pre-etch line width for the mean line width

| | Factor | Mean Line Width (μm) Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| A | Mask Dimension | 2.39 | 2.87 | |
| BD | Viscosity Bake Temperature | ($B_1D_1$) 2.83 | ($B_2D_1$) 2.31 | ($B_1D_2$) 2.74 |
| B | Viscosity | 2.79 | 2.31 | |
| D | Bake Temperature | 2.57 | 2.74 | |
| C | Spin Speed | 2.40 | 2.59 | 2.89 |
| E | Bake Time | 2.68 | 2.68 | 2.53 |
| F | Aperture | 2.68 | 2.56 | 2.64 |
| G | Exposure Time | 2.74 | 2.66 | 2.49 |
| H | Developing Time | 2.60 | 2.60 | 2.69 |

Overall mean line width = 2.63 μm.

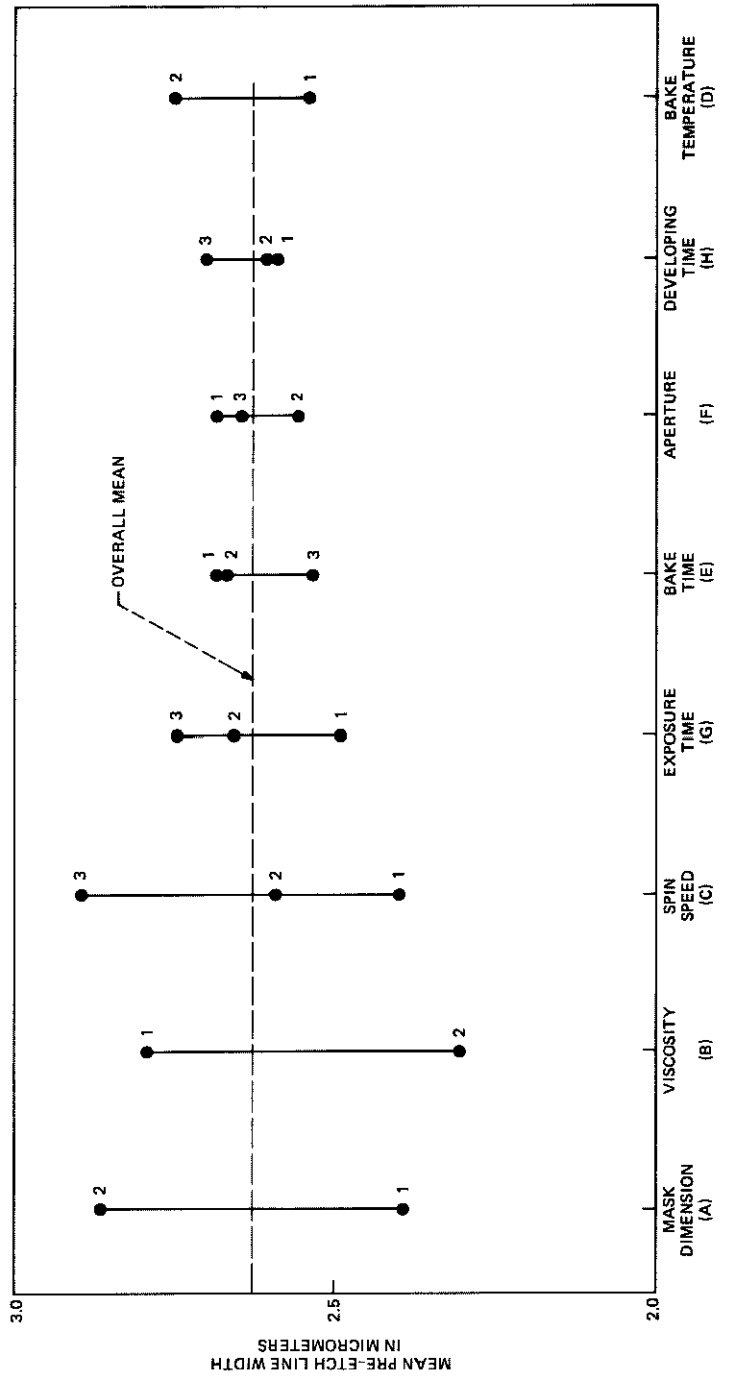



Fig. 4—Mean pre-etch line width. The mean line width for each factor level is indicated by a dot. The number next to each dot indicates the factor level.

Table VIII—Pre-etch line width

(a) ANOVA for mean line width

| | Source | Degrees of Freedom | Sum of Squares | Mean Square | Expected Mean Square |
|---|---|---|---|---|---|
| A | Mask Dimension | 1 | 1.05 | 1.050 | $\sigma^2 + \phi_A$ |
| BD | Viscosity Bake Temperature | 2 | 0.95 | 0.475 | $\sigma^2 + \phi_{BD}$ |
| C | Spin Speed | 2 | 0.73 | 0.365 | $\sigma^2 + \phi_C$ |
| E | Bake Time | 2 | 0.10 | 0.050 | $\sigma^2 + \phi_E$ |
| F | Aperture | 2 | 0.05 | 0.025 | $\sigma^2 + \phi_F$ |
| G | Exposure Time | 2 | 0.19 | 0.095 | $\sigma^2 + \phi_G$ |
| H | Developing Time | 2 | 0.04 | 0.020 | $\sigma^2 + \phi_H$ |
| Error | | 4 | 0.26 | 0.065 | $\sigma^2$ |
| Total | | 17 | 3.37 | | |

(b) Pooled ANOVA for mean line width

| | Source | Degrees of Freedom | Sum of Squares | Mean Square | F | Contribution Percent |
|---|---|---|---|---|---|---|
| A | Mask Dimension | 1 | 1.05 | 1.050 | 19.81* | 29.6 |
| B | Viscosity | 1 | 0.83 | 0.834 | 15.74* | 22.6 |
| C | Spin Speed | 2 | 0.73 | 0.365 | 6.89* | 18.5 |
| G | Exposure Time | 2 | 0.19 | 0.095 | 1.79 | 2.5 |
| Error | | 11 | 0.58 | 0.053 | | 26.8 |
| Total | | 17 | 3.37 | | | 100.0 |

$F_{1,11}(0.95) = 4.84$.
$F_{2,11}(0.95) = 3.98$.
* Factors significant at 95-percent confidence level.

Table IX—Post-etch line-width data

| Experiment Number | Mean Line Width, $\bar{x}$ ($\mu$m) | Standard Deviation of Line Width $s$ ($\mu$m) | s/n $\eta = \log(\bar{x}/s)$ |
|---|---|---|---|
| 1 | 2.84 | 0.11 | 1.42 |
| 2 | 3.14 | 0.063 | 1.70 |
| 3 | 3.65 | 0.15 | 1.40 |
| 4 | 2.57 | 0.20 | 1.11 |
| 5 | 1.72 | 0.40 | 0.63 |
| 6 | 3.12 | 0.27 | 1.07 |
| 7 | 2.62 | 0.19 | 1.14 |
| 8 | 4.10 | 0.18 | 1.37 |
| 9 | 3.55 | 0.26 | 1.13 |
| 10 | 3.31 | 0.35 | 0.98 |
| 11 | 3.60 | 0.38 | 0.98 |
| 12 | 4.17 | 0.27 | 1.18 |
| 13 | 3.10 | 0.16 | 1.29 |
| 14 | 3.14 | 0.16 | 1.29 |
| 15 | 2.55 | 0.21 | 1.09 |
| 16 | 2.81 | 0.37 | 0.88 |
| 17 | 3.78 | 0.22 | 1.23 |
| 18 | 4.34 | 0.078 | 1.75 |

The average s/n and the mean line width for each factor level are shown in Tables Xa and b, respectively.

The linear model (1) was again used to analyze the post-etch line-width data. The ANOVA for the signal-to-noise ratios, Table XIa, indicates that none of the nine process factors has a significant effect (approximately 5-percent level) on the s/n for the post-etch line width. The pooled ANOVA for the mean post-etch line widths is shown in Table XIb. It is obvious from the table that the viscosity, exposure, spin speed, mask dimension, and developing time have significant effects (5-percent level) on the mean line width. The contribution of these factors to the total sum of squares exceeds 90 percent. The mean line width for each factor level is shown graphically in Fig. 5.

## VIII. ANALYSIS OF POST-ETCH WINDOW-SIZE DATA

Some windows are printed and open while the others are not. Thus the window-size data are mixed categorical-continuous in nature. Analysis of such data is done by converting all the data to the categorical type and then using the 'accumulation analysis' method, which is

Table X—Post-etch line width

(a) Average signal-to-noise ratios

| | Factor | Average s/n Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| A | Mask Dimension | 1.22 | 1.19 | |
| BD | Viscosity Bake Temperature | ($B_1D_1$) 1.28 | ($B_2D_1$) 1.08 | ($B_1D_2$) 1.25 |
| B | Viscosity | 1.27 | 1.08 | |
| D | Bake Temperature | 1.18 | 1.25 | |
| C | Spin Speed | 1.14 | 1.20 | 1.27 |
| E | Bake Time | 1.16 | 1.28 | 1.17 |
| F | Aperture | 1.28 | 1.22 | 1.11 |
| G | Exposure Time | 1.26 | 1.33 | 1.02 |
| H | Developing Time | 1.09 | 1.20 | 1.32 |
| I | Etch Time | 1.21 | 1.18 | 1.23 |

Overall average s/n = 1.205

(b) Mean line width

| | Factor | Mean Line Width (μm) Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| A | Mask Dimension | 3.03 | 3.42 | |
| BD | Viscosity Bake Temperature | ($B_1D_1$) 3.45 | ($B_2D_1$) 2.70 | ($B_1D_2$) 3.53 |
| B | Viscosity | 3.49 | 2.70 | |
| D | Bake Temperature | 3.08 | 3.53 | |
| C | Spin Speed | 2.88 | 3.25 | 3.56 |
| E | Bake Time | 3.15 | 3.18 | 3.35 |
| F | Aperture | 3.28 | 3.22 | 3.18 |
| G | Exposure Time | 3.52 | 3.34 | 2.83 |
| H | Developing Time | 3.04 | 3.09 | 3.56 |
| I | Etch Time | 3.14 | 3.22 | 3.32 |

Overall mean line width = 3.23 μm.

described by Taguchi in Refs. 3 and 8. Factors that are found significant in this analysis are control factors.

The window sizes were divided into the following five categories:

| Category | Description (micrometers) |
|---|---|
| I | Window not open or not printed |
| II | (0, 2.25) |
| III | [2.25, 2.75) |
| IV | [2.75, 3.25] |
| V | (3.25, ∞) |

Note that these categories are ordered with respect to window size. The target window size at the end of step (*vi*) was 3 μm. Thus category IV is the most desired category, while category I is the least desired category. Table XII summarizes the data for each of the experiments by categories. To simplify our analysis, we shall presume that a missing wafer has the same readings as the observed wafer for that experiment. This is reflected in Table XII, where we show the combined readings for the two wafers of each experiment.

Table XI—Post-etch line width

a) ANOVA for s/n

| | Source | Degrees of Freedom | Sum of Squares | Mean Square | F |
|---|---|---|---|---|---|
| A | Mask Dimension | 1 | 0.005 | 0.005 | 0.02 |
| B | Viscosity | 1 | 0.134 | 0.134 | 0.60 |
| D | Bake Temperature | 1 | 0.003 | 0.003 | 0.01 |
| C | Spin Speed | 2 | 0.053 | 0.027 | 0.12 |
| E | Bake Time | 2 | 0.057 | 0.028 | 0.13 |
| F | Aperture | 2 | 0.085 | 0.043 | 0.19 |
| G | Exposure Time | 2 | 0.312 | 0.156 | 0.70 |
| H | Developing Time | 2 | 0.156 | 0.078 | 0.35 |
| I | Etch Time | 2 | 0.008 | 0.004 | 0.02 |
| Error | | 2 | 0.444 | 0.222 | |
| Total | | 17 | 1.257 | | |

b) Pooled ANOVA for mean line width

| | Source | Degrees of Freedom | Sum of Squares | Mean Square | F | Percent Contribution |
|---|---|---|---|---|---|---|
| A | Mask Dimension | 1 | 0.677 | 0.677 | 16.92* | 8.5 |
| B | Viscosity | 1 | 2.512 | 2.512 | 63.51* | 32.9 |
| C | Spin Speed | 2 | 1.424 | 0.712 | 17.80* | 17.9 |
| G | Exposure Time | 2 | 1.558 | 0.779 | 19.48* | 19.6 |
| H | Developing Time | 2 | 0.997 | 0.499 | 12.48* | 12.2 |
| Error | | 9 | 0.356 | 0.040 | | 8.9 |
| Total | | 17 | 7.524 | | | 100.0 |

$F_{1,9}(0.95) = 5.12$.
$F_{2,9}(0.95) = 4.26$.
* Factors significant at 95-percent confidence level.

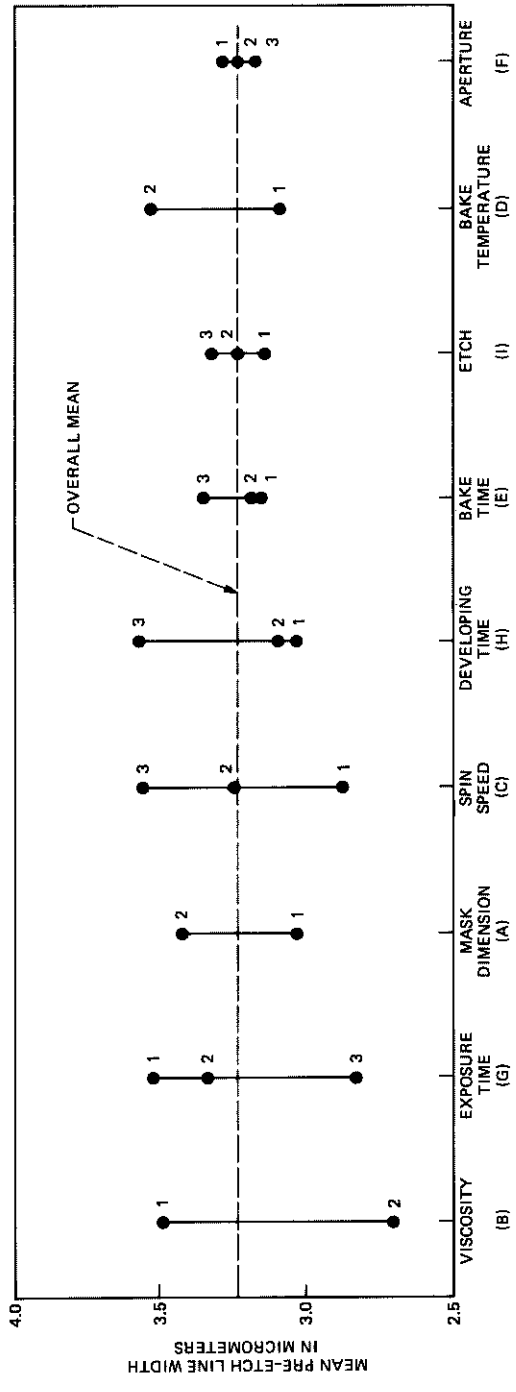



Fig. 5—Mean post-etch line width. The mean line width for each factor level is indicated by a dot. The number next to a dot indicates the factor level.

Table XII—Post-etch window-size data—frequencies by experiment

| Experiment No. | Frequency Distribution for Wafer 1 | | | | | Frequency Distribution for Wafer 2 | | | | | Combined Frequency for the Two Wafers | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | I | II | III | IV | V | I | II | III | IV | V | I | II | III | IV | V |
| 1 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 2 | 2 | 0 | 2 | 3 | 0 | 0 | 0 | 3 | 3 | 2 | 2 |
| 3 | 0 | 0 | 0 | 4 | 0 | 1 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 9 | 0 |
| 4 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| 5 | * | * | * | * | * | 5 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| 6 | 0 | 3 | 2 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 5 | 3 | 2 | 0 | 0 |
| 7 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| 8 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 5 | 0 | 0 | 5 | 0 |
| 9 | 0 | 0 | 0 | 5 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 1 | 4 | 5 | 0 |
| 10 | 0 | 5 | 0 | 0 | 0 | 2 | 0 | 3 | 0 | 0 | 2 | 5 | 3 | 0 | 0 |
| 11 | 0 | 0 | 0 | 5 | 0 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | 2 | 6 | 0 |
| 12 | 1 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 4 | 1 | 0 | 1 | 3 | 5 |
| 13 | 0 | 0 | 3 | 2 | 0 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 3 | 2 | 0 |
| 14 | 1 | 3 | 1 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 6 | 3 | 1 | 0 | 0 |
| 15 | 5 | 0 | 0 | 0 | 0 | * | * | * | * | * | 10 | 0 | 0 | 0 | 0 |
| 16 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 4 | 0 | 1 | 0 | 0 | 4 | 3 | 3 |
| 18 | 0 | 0 | 0 | 0 | 5 | * | * | * | * | * | 0 | 0 | 0 | 0 | 10 |

* Implies data missing.

Table XIII gives the frequency distribution corresponding to each level of each factor. To obtain the frequency distribution for a specific level of a specific factor, we summed the frequencies of all the experiments that were conducted with that particular level of that particular factor. For example, the frequency distribution for the first level of factor C (low spin speed) was obtained by summing the frequency distributions of experiments with serial numbers 1, 4, 7, 10, 13, and 16. These six experiments were conducted with level 1 of factor C.

The frequency distributions of Table XIII are graphically displayed by star plots in Fig. 6. From this figure and the table it is apparent that a change in the level of viscosity, spin speed, or mask dimension causes a noticeable change in the frequency distribution. A change in the level of etch time, bake time, or bake temperature seems to have only a small effect on the frequency distribution. The effects of the other factors are intermediate.

We now determine which factors have a significant effect on the frequency distribution of the window sizes. The standard chi-square test for multinomial distributions is not appropriate here because the categories are ordered. The accumulation analysis method has an intuitive appeal and has been empirically found by Professor Taguchi to be effective in analyzing ordered categorical data. The method consists of the following three steps:

(*i*) Compute the cumulative frequencies. Table XIII shows the cumulative frequencies for all factor levels. The cumulative categories are denoted with parentheses. Thus (III) means sum of categories I,

Table XIII—Post-etch window-size data—frequencies by factor level

| | Frequencies | | | | | Cumulative Frequencies | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Factor Levels | I | II | III | IV | V | (I) | (II) | (III) | (IV) | (V) |
| Mask Dimension | | | | | | | | | | |
| $A_1$ | 51 | 7 | 9 | 21 | 2 | 51 | 58 | 67 | 88 | 90 |
| $A_2$ | 35 | 9 | 14 | 14 | 18 | 35 | 44 | 58 | 72 | 90 |
| Viscosity, Bake Temperature | | | | | | | | | | |
| $B_1D_1$ | 15 | 9 | 9 | 20 | 7 | 15 | 24 | 33 | 53 | 60 |
| $B_2D_1$ | 46 | 6 | 6 | 2 | 0 | 46 | 52 | 58 | 60 | 60 |
| $B_1D_2$ | 25 | 1 | 8 | 13 | 13 | 25 | 26 | 34 | 47 | 60 |
| Spin Speed | | | | | | | | | | |
| $C_1$ | 47 | 5 | 6 | 2 | 0 | 47 | 52 | 58 | 60 | 60 |
| $C_2$ | 22 | 7 | 10 | 16 | 5 | 22 | 29 | 39 | 55 | 60 |
| $C_3$ | 17 | 4 | 7 | 17 | 15 | 17 | 21 | 28 | 45 | 60 |
| Bake Time | | | | | | | | | | |
| $E_1$ | 31 | 2 | 10 | 14 | 3 | 31 | 33 | 43 | 57 | 60 |
| $E_2$ | 26 | 3 | 7 | 7 | 17 | 26 | 29 | 36 | 43 | 60 |
| $E_3$ | 29 | 11 | 6 | 14 | 0 | 29 | 40 | 46 | 60 | 60 |
| Aperture | | | | | | | | | | |
| $F_1$ | 32 | 7 | 5 | 6 | 10 | 32 | 39 | 44 | 50 | 60 |
| $F_2$ | 36 | 3 | 4 | 10 | 7 | 36 | 39 | 43 | 53 | 60 |
| $F_3$ | 18 | 6 | 14 | 19 | 3 | 18 | 24 | 38 | 57 | 60 |
| Exposure Time | | | | | | | | | | |
| $G_1$ | 26 | 3 | 10 | 13 | 8 | 26 | 29 | 39 | 52 | 60 |
| $G_2$ | 18 | 12 | 11 | 7 | 12 | 18 | 30 | 41 | 48 | 60 |
| $G_3$ | 42 | 1 | 2 | 15 | 0 | 42 | 43 | 45 | 60 | 60 |
| Developing Time | | | | | | | | | | |
| $H_1$ | 37 | 4 | 6 | 8 | 5 | 37 | 41 | 47 | 55 | 60 |
| $H_2$ | 27 | 11 | 12 | 5 | 5 | 27 | 38 | 50 | 55 | 60 |
| $H_3$ | 22 | 1 | 5 | 22 | 10 | 22 | 23 | 28 | 50 | 60 |
| Etch Time | | | | | | | | | | |
| $I_1$ | 37 | 5 | 3 | 5 | 10 | 37 | 42 | 45 | 50 | 60 |
| $I_2$ | 21 | 8 | 14 | 15 | 2 | 21 | 29 | 43 | 58 | 60 |
| $I_3$ | 28 | 3 | 6 | 15 | 8 | 28 | 31 | 37 | 52 | 60 |
| Totals | 86 | 16 | 23 | 35 | 20 | 86 | 102 | 125 | 160 | 180 |

II, and III. Note that the cumulative category (V) is the same as the total number of window readings for the particular factor level.

(*ii*) Perform "binary data" ANOVA[7] on each cumulative category except the last category, viz. (V). Note that a certain approximation is involved in the significance levels suggested by this ANOVA because the observations are not normally distributed.

(*iii*) Assign weights to each cumulative category. These weights are inversely proportional to the Bernoulli trial variance. Let $cum_c$ be the total number of windows in the cumulative category, $c$, as given in the bottom row of Table XIII. Then the weight for that category is:

$$W_c = \frac{1}{\frac{cum_c}{180} \times \left(1 - \frac{cum_c}{180}\right)} = \frac{180^2}{cum_c(180 - cum_c)}.$$

These weights are shown in Appendix D for each category.

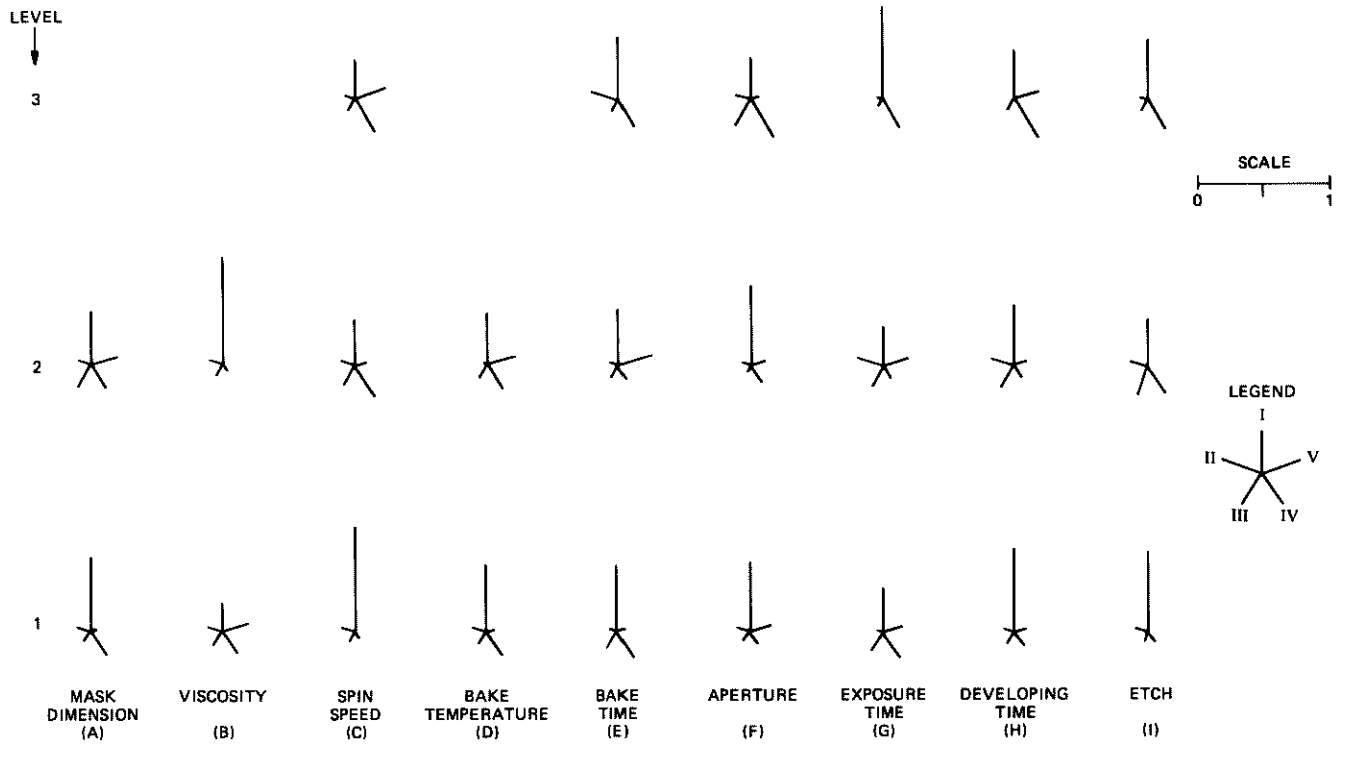



Fig. 6—Post-etch window size. The star gives the frequency distribution for each factor level.

Then for each factor and for each error term the accumulated sum of squares is taken to be equal to the weighted sum of the sum of squares for all cumulative categories.

The intuitive appeal for accumulation analysis is that by taking cumulative frequencies we preserve the order of the categories. By giving weights inversely proportional to the sampling errors in each cumulative category, we make the procedure more sensitive to a change in the variance. The difficulty is that the frequencies of the cumulative categories are correlated. So the true level of significance of the F-test may be somewhat different from that indicated by the F table. More work is needed to understand the statistical properties of the accumulation analysis.

Table XIV gives the final ANOVA with accumulated sum of squares. The computations are illustrated in Appendix D. For each cumulative category, the following nested, mixed linear model was used in performing the ANOVA:

$$y_{ijk} = \mu + x_i + e_{1ij} + e_{2ijk}, \tag{2}$$

where

$i$ = 1, ⋯, 18 stands for the experiment.
$j$ = 1, 2 stands for wafer within the experiment.
$k$ = 1, ⋯, 5 stands for replicate or position within wafer within the experiment.
$\mu$ is the overall mean.
$x_i$ is the fixed effect of the factor-level combination used in experiment $i$. Here we consider only the main effect for each of the factors. See the discussion of model (1) in Section 7.3 for more details of the interpretation of $x_i$.
$e_{1ij}$ is the random effect for wafer $j$ within experiment $i$.
$e_{2ijk}$ is the random error for replicate $k$ within wafer $j$ within experiment $i$.
$y_{ijk}$ is the observation for replicate $k$ in wafer $j$ in experiment $i$. $y_{ijk}$ takes a value 1 if the window size belongs to the particular category. Otherwise, the value is zero.

The expected mean squares for this ANOVA model are also shown in Table XIV. The variances of $e_1$ and $e_2$ are denoted by $\sigma_1^2$ and $\sigma_2^2$, respectively. The effects of the factors A through I are denoted by $\phi$ with an appropriate subscript. The effect of lack of fit is denoted by $\phi_\ell$. We assume that the random variables $e_{1ij}$ and $e_{2ijk}$ are independent for all values of $i, j$ and $k$. The degrees of freedom shown in Table XIV have been adjusted for the fact that three experiments have only one wafer each.

For testing the significance of the effect of error between wafers within experiments, the relevant denominator sum of squares is the

Table XIV—Post-etch window size

ANOVA for accumulation analysis

| Source | Degrees of Freedom | Sum of Squares | Mean Square | F | (Expected Mean Square) + $\bar{W}$ |
|---|---|---|---|---|---|
| A Mask Dimension | 4 | 26.64 | 6.66 | 2.67* | $\sigma_2^2 + 5\sigma_1^2 + \phi_A$ |
| BD Viscosity-Bake Temperature | 8 | 112.31 | 14.04 | 5.64* | $\sigma_2^2 + 5\sigma_1^2 + \phi_{BD}$ |
| C Spin Speed | 8 | 125.52 | 15.69 | 6.30* | $\sigma_2^2 + 5\sigma_1^2 + \phi_C$ |
| E Bake Time | 8 | 36.96 | 4.62 | 1.86 | $\sigma_2^2 + 5\sigma_1^2 + \phi_E$ |
| F Aperture | 8 | 27.88 | 3.49 | 1.40 | $\sigma_2^2 + 5\sigma_1^2 + \phi_F$ |
| G Exposure Time | 8 | 42.28 | 5.29 | 2.12* | $\sigma_2^2 + 5\sigma_1^2 + \phi_G$ |
| H Developing Time | 8 | 45.57 | 5.70 | 2.29* | $\sigma_2^2 + 5\sigma_1^2 + \phi_H$ |
| I Etch Time | 8 | 23.80 | 2.98 | 1.20 | $\sigma_2^2 + 5\sigma_1^2 + \phi_I$ |
| Lack of Fit | 8 | 17.25 | 2.16 | 0.87 | $\sigma_2^2 + 5\sigma_1^2 + \phi_I$ |
| Error Between Wafers Within Experiment | 60 | 149.33 | 2.49 | 11.69* | $\sigma_2^2 + 5\sigma_1^2$ |
| Error Between Replicates Within Wafers Within Experiment | 528 | 112.45 | 0.21 | | $\sigma_2^2$ |
| Total | 656 | 720.00 | | | |

$\bar{W} = (W_{(I)} + W_{(II)} + W_{(III)} + W_{(IV)})/4$.
$F_{4,60}(0.95) = 2.53$, $F_{8,60}(0.95) = 2.10$, $F_{60,528}(0.95) = 1.32$.

(b) Separation of $S_{BD}$

| Source | Degrees of Freedom | Sum of Squares | Mean Square | F |
|---|---|---|---|---|
| B Viscosity | 4 | 87.38 | 21.85 | 8.78* |
| D Bake Temperature | 4 | 6.55 | 1.64 | 0.66 |

* Factors significant at 95-percent confidence level.

estimate of $\sigma_2^2$. The corresponding F value is 11.69, which is significant far beyond the nominal 5-percent level. To test for the lack of fit of the main-effects-only model, the appropriate denominator is the estimate $\sigma_2^2 + 5\sigma_1^2$. The corresponding F ratio is 0.87. This indicates that the main-effects-only model adequately describes the observed data relative to the random errors between wafers. For testing the significance of the process factors, the denominator mean square is again the estimate of $\sigma_2^2 + 5\sigma_1^2$. We see that the mask dimension, viscosity, spin speed, exposure time, and developing time have a significant effect (approximately 5-percent level) on the window size. The effects of the other factors are not significant.

## IX. SELECTION OF OPTIMUM FACTOR LEVELS

The following table summarizes the significant results of the analyses performed in Sections VII and VIII. In each category, the factors are arranged in descending order according to the F value.

Significant effect on s/n:

Pre-etch line width: A, F
Post-etch line width: None
Significant effect on mean:
Pre-etch line width: A, B, C
Post-etch line width: B, G, C, A, H
Significant factors identified by accumulation analysis:
Post-etch window size: B, C, A, H, G

Factors that have a significant effect on the s/n and the factors identified to be significant by the accumulation analysis are all control factors. Setting their levels equal to optimum levels minimizes the process variability. Here the control factors are A, F, B, C, H, and G.

To keep the process mean on target we use a signal factor. Ideally, the signal factor should have a significant effect on the mean, but should have no effect on the s/n. Then changing the level of the signal factor would affect only the mean. In practice, a small effect on the s/n may have to be tolerated.

Among the factors (A, B, C, G, and H) that have a significant effect on the mean, factors A, B, and C are relatively strong control factors as measured by the F statistics for the accumulation analysis and the ANOVA for pre-etch line-width s/n. Also, these factors are relatively difficult to change during production. So A, B, and C are not suitable as signal factor. Between the remaining two factors, G and H, G has greater effect on the mean and also shows as a less significant factor in accumulation analysis. So exposure time was assigned to be the signal factor.

The optimum levels for the control factors were selected as follows. The mask dimension (A) and the aperture (F) have a significant effect on the s/n for pre-etch line width. From Table V we see that the 2.5-$\mu$m mask (level 2) has a higher s/n than the 2.0-$\mu$m mask. Hence 2.5 $\mu$m was chosen to be the optimum mask dimension. Also, aperture 1 (level 1) has the highest s/n among the three apertures studied. However, because of the past experience, aperture 2 was chosen to be the preferred level.

The accumulation analysis of the post-etch window-size data indicated that the viscosity, spin speed, mask dimension, developing time, and exposure have statistically significant effects on the frequency distribution. The optimum levels of these factors can be determined from Table XIII and Fig. 6 to be those that have the smallest fraction of windows not open (category I) and the largest fraction of windows in the range $3.0 \pm 0.25$ $\mu$m (category IV). Because it is more critical to have all the windows open, when there was a conflict we took the requirement on category I to be the dominant requirement. The optimum levels are: 2.5-$\mu$m mask dimension, viscosity 204, 4000-rpm spin speed, 60-second developing time, and normal exposure.

Table XV—Optimum factor levels

| Label | Factors Name | Standard Levels | Optimum Levels |
|---|---|---|---|
| A | Mask Dimension (μm) | 2.0 | 2.5 |
| B | Viscosity | 204 | 204 |
| C | Spin Speed (rpm) | 3000 | 4000 |
| D | Bake Temperature (°C) | 105 | 105 |
| E | Bake Time (min) | 30 | 30 |
| F | Aperture | 2 | 2 |
| G | Exposure (PEP setting) | Normal | Normal |
| H | Developing Time (s) | 45 | 60 |
| I | Plasma Etch Time (min) | 13.2 | 13.2 |

Table XV shows side by side the optimum factor levels and the standard levels as of September 1980. Note that our experiment has indicated that the mask dimension be changed from 2.0 μm to 2.5 μm, spin speed from 3000 rpm to 4000 rpm, and developing time from 45 seconds to 60 seconds. The exposure time is to be adjusted to get the correct mean value of the line width and the window size. The levels of the other factors, which remain unchanged, have been confirmed to be optimum to start with.

In deriving the optimum conditions we have conducted a highly fractionated factorial experiment and have considered only the main effects of the factors. The interactions between the factors have been ignored. If the interactions are strong compared to the main effects, then there is a possibility that the optimum conditions thus derived would not improve the process. So experiments have to be conducted to verify the optimum conditions. The verification was done in conjunction with the implementation, which is described next.

## X. IMPLEMENTATION AND THE BENEFITS OF THE OPTIMUM LEVELS

We started to use the optimum process conditions given in Table XV in the Integrated Circuits Design Capability Laboratory in January 1981. In the beginning the exposure was set at 90, which is the normal setting given in Table I. We observed that the final window at the end of step ($xi$) was much larger than the target size of 3.5 μm. Through successive experiments, we reduced the exposure time until the mean final window size came to about 3.5 μm. The corresponding exposure setting is 140. Since then the process has been run at these conditions. The benefits of running the process at these conditions are:

($i$) The pre-etch line width is routinely used as a process quality indicator. Before September 1980 the standard deviation of this indicator was 0.29 μm on a base line chip (DSO chip). With the optimum process parameters, that standard deviation has come down to 0.14 μm. This is a two-fold reduction in standard deviation, or a four-fold reduction in variance. This is evidenced by Fig. 7, which shows a

typical photograph of the PLA area of a *BELLMAC*-32 microprocessor chip fabricated by the new process. Note that windows in Fig. 7 are much more uniform in size compared to the windows in Fig. 2. Also, all windows are printed and open in Fig. 7.

(*ii*) After the final step of window forming, i.e., after step (*xi*), the windows are visually examined on a routine basis. Analysis of the quality control data on the DSO chip, which has an area of approximately 0.19 $cm^2$, showed that prior to September 1980 about 0.12 window per chip was either not open or not printed (i.e., approximately one incidence of window not open or not printed was found in eight chips). With the new process only 0.04 window per chip is not open or printed (i.e., approximately one incidence of window not open or printed is found in twenty-five chips). This is a three-fold reduction in defect density due to unopened windows.

(*iii*) Observing these improvements over several weeks, the process engineers gained a confidence in the stability and robustness of the new process parameters. So they eliminated a number of in-process checks. As a result the overall time spent by the wafers in window photolithography has been reduced by a factor of two.

The optimum parameter levels were first used in the Integrated Circuit Device Capability Laboratory with only a few codes of ICs. Subsequently, these parameter levels were used with all codes of 3.5-$\mu$m technology chips, including *BELLMAC*-4 microcomputer and *BELLMAC*-32 microprocessor chips. The mask dimension change from 2.0 to 2.5 $\mu$m is now a standard for 3.5-$\mu$m CMOS technology.

## XI. DISCUSSION AND FUTURE WORK

The off-line quality control method is an efficient method of improving the quality and the yield of a production process. The method has a great deal of similarity with the response surface method[9] and the evolutionary operations method,[10] which are commonly known in statistical literature in this country. Both the response surface and the evolutionary operations methods are used to maximize the yield of a production process and they both make use of the experimental design techniques. The main difference is that in the off-line quality control method the process variability that has a great impact on the product quality is the objective function. In the response surface and evolutionary operations methods, the process variability is generally not considered. Thus, intuitively, the optimum levels derived by using the off-line quality control method can be expected to be more robust, stable, and dependable.

In the response surface method one typically uses a relatively large fraction of the factorial experiment. However, in off-line quality control usually a very small fraction is chosen. Another difference is that in

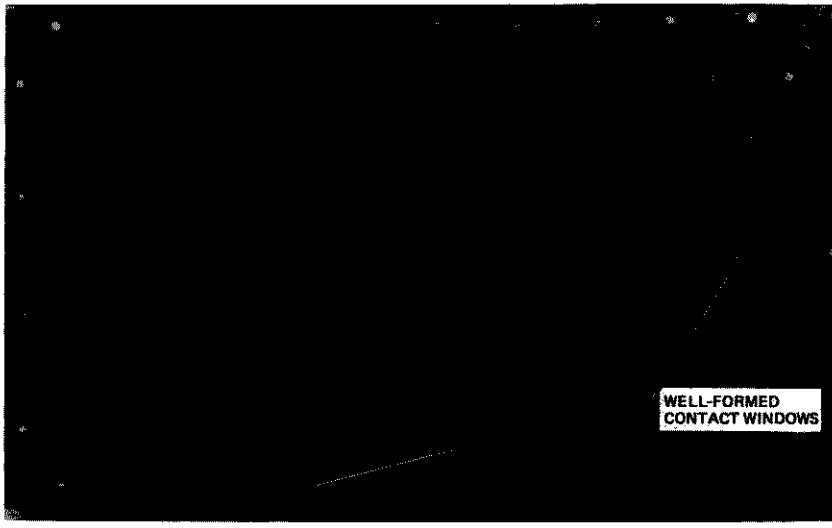



Fig. 7—Typical results for contact windows in PLA area (January through March, 1981) using parameters derived from this experiment. The contact windows are round shaped.

the response surface method the objective function is considered to be a continuous function approximated by a low-order polynomial. In off-line quality control, we can simultaneously study both the continuous and discrete factors.

Our application of the off-line quality control method to the window-cutting process in the Murray Hill 3.5-$\mu$m CMOS technology, as seen from the earlier sections, has resulted in improved control of window size, lower incidence of unopened windows, and reduced time for window photolithography. Presently, we have undertaken to optimize two more steps in IC fabrication. Those steps are polysilicon patterning and aluminum patterning. Both these processes, like the window-cutting process, involve photolithography and are among the more critical processes of IC fabrication. We think that the method has a great potential and would like to see applications in various parts of Bell Laboratories and Western Electric Company.

## XII. ACKNOWLEDGMENTS

The authors acknowledge with gratitude the support and the encouragement given by R. L. Chaddha, R. Edwards, J. V. Dalton, A. B. Hoadley, and G. R. Weber. R. G. Brandes, K. J. Orlowski, and members of the IC Process Operations Group provided valuable help in conducting the experiments. We also thank Ramon Leon for helpful discussions on the graphical representation of the data and I. J. Terpenning for helpful comments on an earlier draft of this paper. The

original manuscript was carefully prepared by Jill Cooper on the *UNIX** operating system.

## REFERENCES


1. G. Taguchi, "Off-line and On-line Quality Control Systems," International Conference on Quality Control, Tokyo, Japan, 1978, pp. B4-1–5.
2. R. N. Kackar and M. S. Phadke, unpublished work.
3. G. Taguchi and Y. Wu, *Introduction to Off-Line Quality Control*, Central Japan Quality Control Association, Nagoya, Japan, 1980.
4. A. B. Glaser and G. A. Subak-Sharp, *Integrated Circuits Engineering*, Reading, MA: Addison-Wesley, 1977.
5. W. Q. Meeker, G. H. Hahn, and P. I. Feder, "A Computer Program for Evaluating and Comparing Experimental Designs and Some Applications," American Statistician, 29, No. 1 (February 1975), pp. 60–4.
6. C. R. Hicks, *Fundamental Concepts in the Design of Experiments*, New York, NY: Holt, Rinehart and Winston, 1973.
7. S. R. Searle, *Linear Models*, New York, NY: John Wiley, 1971.
8. G. Taguchi, "A New Statistical Analysis Method for Clinical Data, the Accumulation Analysis, in Contrast with the Chi-square Test," Shinjuku Shobo, Tokyo, Japan, 1975.
9. R. H. Myers, *Response Surface Methodology*, Newton, MA: Allyn and Bacon, 1971.
10. G. E. P. Box and N. R. Draper, *Evolutionary Operations: A Statistical Method for Process Improvement*, New York, NY: John Wiley, 1969.
11. G. W. Snedecor and W. G. Cochran, *Statistical Methods*, Ames, Iowa: Iowa State University Press, 1980.


## APPENDIX A

### *Computation of the Sum of Squares—Analysis of the Signal-to-Noise Ratio for Pre-Etch*

The computations of the sum of squares tabulated in Table VIa are illustrated below.

$$S_m = \text{Correction Factor}$$

$$= \frac{\left(\sum_{i=1}^{18} \eta_i\right)^2}{18} = \frac{(25.2202)^2}{18} = 35.3366$$

$$S_A = \text{Sum of squares for factor A}$$

$$= \frac{(9m_{A_1})^2 + (9m_{A_2})^2}{9} - S_m$$

$$= \frac{(11.5711)^2 + (13.6491)^2}{9} - 35.3366$$

$$= 0.2399 \qquad (d.f. = 1)$$

$$S_C = \frac{(6m_{C_1})^2 + (6m_{C_2})^2 + (6m_{C_3})^2}{6} - S_m$$

* Trademark of Bell Laboratories.

$$= \frac{(8.1979)^2 + (8.1017)^2 + (8.9206)^2}{6} - S_m$$

$$= 0.0668 \qquad (d.f. = 2).$$

Sums of squares for the factors E, F, G, and H were calculated similarly. The combined sum of squares due to B and D is given by

$$S_{BD} = \text{Sum of squares for the column BD}$$

$$= \frac{(6m_{B_1D_1})^2 + (6m_{B_1D_2})^2 + (6m_{B_2D_1})^2}{6} - S_m$$

$$= \frac{(8.2524)^2 + (8.6649)^2 + (8.3029)^2}{6} - 35.3366$$

$$= 0.0169 \qquad (d.f. = 2).$$

The total sum of squares is

$$S_T = \sum_{i=1}^{18} \eta_i^2 - S_m = 0.8423. \qquad (d.f. = 17).$$

The error sum of squares is calculated by subtraction.

$$S_e = S_T - (S_A + S_{BD} + S_C + S_E + S_F + S_G + S_H)$$

$$= 0.1522 \qquad (d.f. = 4).$$

Here we do not compute the sum of squares due to factor I (etch time), because it has no influence on the pre-etch line width.

## APPENDIX B

### *Computation of the Percent Contribution—Analysis of the Signal-to-Noise Ratio for Pre-Etch Line Width*

The computation of the percent contribution is explained below. The contribution of factor A to the total sum of squares

$$= S_A - (d.f. \text{ of A})(\text{error mean square}).$$

Hence, the percent contribution for factor A

$$= \frac{S_A - (\text{d.f. of A})(\text{error mean square})}{\text{total sum of squares}} \times 100$$

$$= \frac{0.2399 - 0.0251}{0.8423} \times 100 = 25.5\%.$$

The percent contributions of E and F are determined similarly. Now consider, the contribution of error to the total sum of squares:

$$= S_e + (\text{total } d.f. \text{ for factors})(\text{error mean square}).$$

Hence, the percent contribution for error

$$= \frac{S_e + (\text{total } d.f. \text{ for factors})(\text{error mean square})}{\text{total sum of squares}} \times 100$$

$$= \frac{0.3010 + 5 \times 0.0251}{0.8423} \times 100 = 50.6\%.$$

## APPENDIX C

### *Separation of $S_{BD}$ into $S_B$ and $S_D$—Analysis of the Mean Pre-Etch Line Width*

The sum of squares, $S_{BD}$, can be decomposed in the following two ways[11] to obtain the contributions of the factors B and D:

$$S_{BD} = S'_{B(D)} + S'_D$$

and

$$S_{BD} = S'_{D(B)} + S'_B.$$

Here $S'_{B(D)}$ is the sum of squares due to B, assuming D has no effect; $S'_D$ is the sum of squares due to D after eliminating the effect of B. The terms $S'_{D(B)}$ and $S'_B$ are interpreted similarly. We have

$$S'_{B(D)} = \frac{(6m_{B_1D_1} + 6m_{B_1D_2} - 12m_{B_2D_1})^2}{(1^2 + 1^2 + 2^2) \times 6} = 0.903 \qquad (d.f. = 1)$$

$$S'_D = S_{BD} - S'_{B(D)} = 0.047 \qquad (d.f. = 1).$$

Similarly,

$$S'_{D(B)} = \frac{(6m_{B_1D_1} + 6m_{B_2D_1} - 12m_{B_1D_2})^2}{(1^1 + 1^2 + 2^2) \times 6} = 0.116 \qquad (d.f. = 1)$$

$$S'_B = S_{BD} - S'_{D(B)} = 0.834 \qquad (d.f. = 1).$$

For testing the significance of the factors B and D we use $S'_B$ and $S'_D$, respectively. Note that $S'_B$ and $S'_D$ do not add up to $S_{BD}$, which is to be expected because the design is not orthogonal with respect to the factors B and D.

## APPENDIX D

### *Computation of the Sum of Squares for Accumulation Analysis—Analysis of Post-Etch Window Size*

The weights for the cumulative categories (I), (II), (III), and (IV) are given below. The frequencies of the bottom line of Table XIII are used in computing these weights. Therefore:

$$W_{(I)} = \frac{1}{\frac{86}{180} \times \frac{180 - 86}{180}} = 4.008$$

$$W_{(II)} = \frac{180^2}{102 \times (180 - 102)} = 4.072$$

$$W_{(III)} = \frac{180^2}{125 \times (180 - 125)} = 4.713$$

$$W_{(IV)} = \frac{180^2}{160 \times (180 - 160)} = 10.125.$$

Computation of the sum of squares tabulated in Table XIV are illustrated below:

$$S_A = W_{(I)} \times \left(\frac{51^2 + 35^2}{90} - \frac{86^2}{180}\right) + W_{(II)} \times \left(\frac{58^2 + 44^2}{90} - \frac{102^2}{180}\right)$$
$$+ W_{(III)} \times \left(\frac{67^2 + 58^2}{90} - \frac{125^2}{180}\right) + W_{(IV)} \times \left(\frac{88^2 + 72^2}{90} - \frac{160^2}{180}\right)$$
$$= 26.64,$$

and

$$S_C = W_{(I)} \times \left(\frac{47^2 + 22^2 + 17^2}{60} - \frac{86^2}{180}\right)$$
$$+ W_{(II)} \times \left(\frac{52^2 + 29^2 + 21^2}{60} - \frac{102^2}{180}\right)$$
$$+ W_{(III)} \times \left(\frac{58^2 + 39^2 + 28^2}{60} - \frac{125^2}{180}\right)$$
$$+ W_{(IV)} \times \left(\frac{60^2 + 55^2 + 45^2}{60} - \frac{160^2}{180}\right)$$
$$= 125.52.$$

Copyright © 1983 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 5, May–June 1983
*Printed in U.S.A.*

# The Effects of Selected Signal Processing Techniques on the Performance of a Filter-Bank-Based Isolated Word Recognizer

By B. A. DAUTRICH, L. R. RABINER, and T. B. MARTIN

(Manuscript received December 3, 1982)

*To implement an isolated word recognizer based on filter bank techniques, decisions must be made as to how to condition the speech signal prior to the filter bank analysis (preprocessing), how to condition the feature vector at the output of the filter bank analysis (postprocessing), and how to perform the time alignment in the pattern comparison between an unknown test pattern and previously stored reference patterns (registration and distance computation). In the past most designers of such word recognition systems made arbitrary choices about how the various signal processing operations were to be carried out. This paper presents results of a systematic study of the effects of selected signal processing techniques on the performance of a filter bank isolated word recognizer using telephone-quality speech. In particular, the filter bank analyzer was a 13-channel, critical-band-spaced filter bank with excellent time resolution (impulse response durations of from 3 to 30 ms) and poor frequency selectivity (highly overlapping filters with ratios of center frequency to 3-dB bandwidth of about 8 for each band). Among the signal processing techniques studied were: preemphasis of the speech signal; time and frequency smoothing of the filter bank outputs; thresholding, quantization, and normalization of the feature vector; principal components analysis of the feature vector; local and global distance computations for use in the time alignment procedure; and noise analysis in both training and testing. Each of the signal processing techniques was studied individually; hence no tests were run in which several of the techniques were used together. Results showed that some fairly simple signal processing operations provided the best overall performance in the noise-free case; in noisy conditions performance degraded significantly for signal-to-noise ratios less than about 24 dB.*

## I. INTRODUCTION

To implement an isolated word recognizer based on a filter analysis, decisions must be made as to how to preprocess the speech signal prior to the filter bank analysis, how to postprocess the feature vectors obtained at the output of the filter bank analysis, and how to perform the time alignment and distance computation in the pattern comparison between an unknown test pattern and previously stored reference patterns. Often such decisions are made arbitrarily based on experience, heuristic procedures, or sometimes a few brief tests with the system. To our knowledge no one has attempted to systematically examine the effects of various signal processing techniques on the performance (as measured in word error rate) of a filter-bank-isolated word recognizer. This paper provides such a comparison by examining several of the most popular signal processing techniques and showing how they affect the performance of a particular filter bank word recognizer using telephone-quality speech.[1]

There are two inherent problems with any study that attempts to find the best signal processing techniques for a system via experimental means. The first is that the results presented are highly dependent on the signal processing techniques that were studied. Hence, the "optimal" way of processing the signal may not even have been investigated (due to lack of knowledge, etc). With our limited knowledge we know of no way to avoid this difficulty. The second problem is that, of necessity, each of the various signal processing techniques is studied independently of any other (thereby tacitly assuming independence of the various methods). Hence, any interactions between the techniques studied will go unnoticed. Again, we know of no practical way of studying the interactions between signal processing operations; the processing, assuming independence of operations, took about four full months on a modern minicomputer system!

The results to be presented in this paper are an extension of a previous study[1] that examined different filter bank structures and compared their performance to that of a conventional linear predictive coefficient (LPC) word recognizer.[2,3] The key results of this earlier work were:

(*i*) The best performance in word recognition tests was achieved by both a 13-channel, critical-band-spaced filter bank, and a 15-channel, uniformly spaced filter bank. Both filter banks had composite frequency responses without gaps at the band edges. The 13-channel filter bank had highly overlapping filters; the 15-channel filter bank had filters with almost no overlap.

(*ii*) There were significant performance differences between talkers (especially female as opposed to male talkers).

(*iii*) Performance of the LPC and the best filter bank recognizers were comparable for a simple vocabulary of the 10 digits using telephone-quality speech (with no extra noise degradation) over a dialed-up telephone line using a local private branch exchange (PBX).

(*iv*) Performance of the LPC recognizer was superior to that of the best filter-bank recognizers for a complex vocabulary of the alphabet, digits, and three command words, again using telephone-quality speech.

A key question arising from these results was whether any of the proposed signal-processing techniques for the filter bank system could bring up the performance to that of the LPC system for the complex alpha-digits vocabulary. Unfortunately, we will see that none of the proposed methods was able to significantly improve filter-bank performance. (However, some were able to keep performance the same while reducing required storage.)

Two other implementational aspects of word recognizers were studied. The first involves the use of the normalize-and-warp procedure proposed by Myers et al.[4] In this procedure a fixed-length pattern is created for both test and reference patterns prior to time alignment. In this manner the largest warping area is obtained, and the computational aspects of implementing the time-warping procedure are greatly simplified. Instead of considering just the word average length for warping, we studied the effects (for both the filter bank and LPC systems) of warping to prespecified lengths of various amounts. It was found that a large amount of compression could be made before system performance degraded by a significant amount.

The second implementational aspect studied was the effects of additive noise on the performance of both the LPC and filter bank recognizers. We considered cases in which both the training and testing occurred in the noisy background, and when only the testing occurred in the noisy background. It was found that far superior performance was obtained, at all signal-to-noise ratios, when both training and testing occurred in the noisy background. Furthermore, performance of both types of recognizers degraded for signal-to-noise ratios less than or equal to 24 dB. Also, for signal-to-noise ratios greater than 6 dB, the LPC recognizer outperformed the filter bank recognizer.

An overview of the work presented in this paper is as follows. In Section II we review the general implementation of the filter bank isolated word recognizer. In Section III we discuss the signal processing methods that were studied in conjunction with the filter bank. In Section IV we discuss the noise studies. In Section V we describe the experiments performed and give word error rates for the various tests. Finally, a discussion of the results is given in Section VI.

## II. THE FILTER-BANK-ISOLATED WORD RECOGNIZER

Figure 1 shows a block diagram of the overall filter bank word recognizer. The input speech signal is recorded off a dialed-up telephone line, band-limited to 3200 Hz, and digitized at a 6.67-kHz rate. The digitized speech signal, $s(n)$, is first sent to a preprocessor to condition the signal for the filter bank analyzer. Preprocessing is basically a spectra-shaping operation (e.g., linear filtering) for increased immunity to finite word-length processing in the remainder of the system.[5] The preprocessed signal, $\hat{s}(n)$, is then sent to a filter bank analyzer whose structure is shown in Fig. 2. The filter bank contains a set of $Q$ parallel bandpass filters that cover the speech band of

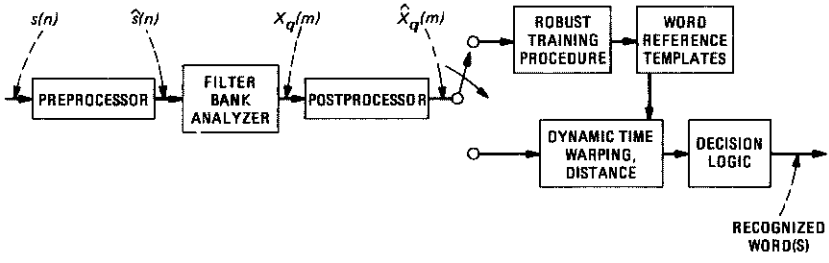



Fig. 1—Filter bank word recognizer with both preprocessing and postprocessing operations.

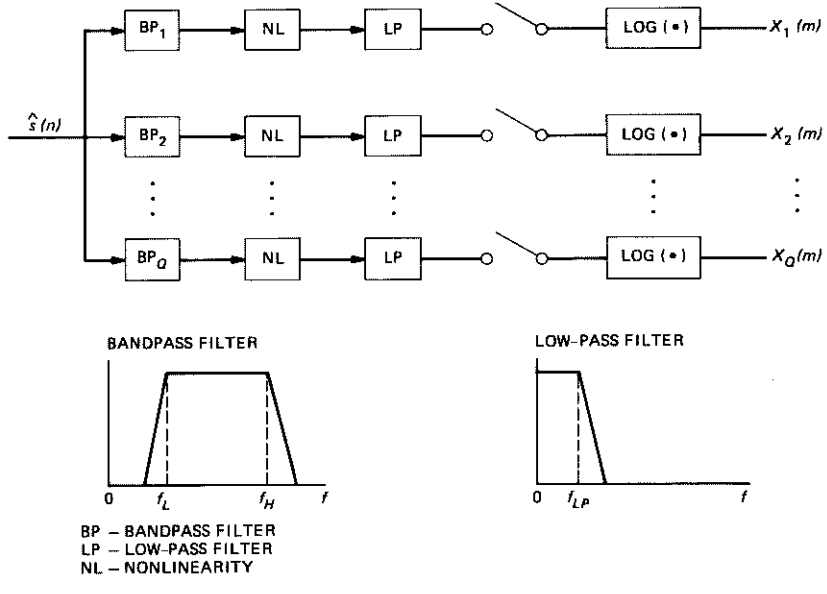



Fig. 2—Structure of filter bank analyzer.

interest (100 to 3200 Hz for telephone speech). Each bandpass filter is followed by a nonlinearity (NL), a low-pass (LP) filter, a sampler, and a logarithmic compressor. The output of the filter bank at time $m$ is a vector

$$X(m) = [X_1(m), X_2(m), \cdots, X_Q(m)], \tag{1}$$

whose components $X_i(m)$ represent the energy in the speech signal in channel $i$ at time $m$.

In our previous work[1] we studied the effects of different types of filter banks on recognizer performance and found that the highest accuracy was obtained with two types of filter banks, namely:

(*i*) A 13-channel, critical-band-spaced filter bank with higher overlapping channels. This filter bank had excellent time resolution (on the order of 10 ms) but poor frequency resolution.

(*ii*) A 15-channel, uniformly spaced filter bank with essentially no overlap between channels. This filter had poor time resolution but excellent frequency resolution. The composite spectrum of this filter bank was flat to within fractions of a dB.

Both of these filter banks used a magnitude nonlinearity and a 3-pole, Bessel, low-pass filter with a 30-Hz cutoff frequency. The sampling rate of the output feature vector was 67 Hz—i.e., adjacent feature estimates were spaced 15 ms in time.

The output of the filter bank was sent to a postprocessor, which performed one or more of the following operations:

(*i*) Time smoothing of feature vectors
(*ii*) Frequency smoothing of channels within a feature vector
(*iii*) Normalization of the feature vector
(*iv*) Thresholding and/or quantization of the feature vector
(*v*) Principal components analysis of the feature vector.

The output of the postprocessor was the input pattern to either the training mode (a robust training procedure[6]), or to the testing mode. In the training mode a set of word reference templates were created based on consistent matches of tokens of a word to previously analyzed tokens of the same word. In the testing mode the test pattern, $T$, consisting of the sequence of feature vectors

$$\mathbf{T} = \{\hat{X}(1), \hat{X}(2), \cdots, \hat{X}(M)\} \tag{2}$$

was compared to the reference pattern, $R^i$, for the $i$th vocabulary word using a dynamic time-warping alignment procedure.[2] For the $i$th reference pattern, a total average distance, $D_i$, between it and the test pattern was computed, and simple decision logic was used to make the word choice for recognition.

To implement the word recognizer of Fig. 1, one has to choose the types of processing to go into the preprocessor, the filter bank analyzer,

the postprocessor, and the dynamic time-warping algorithm. Based on the earlier study[1] we limited the filter bank to the 13-channel critical-band filter bank. However, for each of the remaining signal-processing blocks we tried to choose one or more possibilities and then did experiments to evaluate its usefulness to the overall word recognizer. We discuss our choices in detail in Section III. In addition we chose to study the effects of both noise addition and length quantization of both reference and test patterns on overall performance. These experiments are described in Section IV.

## III. SIGNAL PROCESSING CHOICES IN THE RECOGNIZER

### 3.1 Preprocessor

The function of the preprocessor is to spectrally shape the speech signal to achieve some desired gross spectral shape. The most common form of preprocessing is simple preemphasis, which is used to compensate the inherent 6-dB per octave falloff in the speech spectrum. In such a case a simple first-order network of the form

$$H(z) = 1 - \alpha z^{-1} \tag{3}$$

has been found adequate[3] for recognition purposes. Thus, the difference equation relating $\hat{s}(n)$ to $s(n)$ is of the form

$$\hat{s}(n) = s(n) - \alpha s(n-1), \tag{4}$$

where a value of $\alpha = 0.95$ has been used previously.

### 3.2 Postprocessing

We denote the output of the $q$th channel of the filter bank at frame $m$ as $X_q(m)$, $m = 1, 2, \cdots, M$, $q = 1, 2, \cdots, Q$. All of the postprocessing operations can be expressed in terms of signal processing on $X_q(m)$ to give the signal $\hat{X}_q(m)$. We have considered the operations in Sections 3.2.1 through 3.2.5.

#### 3.2.1 Thresholding and energy normalization

The purpose of channel thresholding is to clamp low-level noise signals from channels at times when essentially no speech signal is present. This is done by applying a threshold so that channel signals below threshold are clamped at the threshold value. In this way much less sensitivity to background noise is achieved. In particular, this is achieved by determining $X_q^{\text{MAX}}$, the peak signal level for the $q$th channel, for each word as:

$$X_q^{\text{MAX}} = \max_{1 \le m \le M} [X_q(m)]. \tag{5}$$

Then the threshold for the $q$th channel is set at

$$T_q^* = X_q^{\text{MAX}} - T^*, \tag{6}$$

where $T^*$ is a parameter of the recognition system. A typical range of $T^*$ is from 30 to 50 (dB).[1] The thresholded channel signal is then given as

$$\hat{X}_q(m) = \max[X_q(m), T_q^*] \tag{7}$$

for all $q$ and $m$.

The purpose of frame energy normalization is to compensate for variations in speech level from utterance to utterance. We have considered two distinct normalization methods, which we call average and peak normalization. For average normalization we calculate the frame average, $\bar{X}(m)$, as

$$\bar{X}(m) = \frac{1}{Q} \sum_{q=1}^{Q} X_q(m), \tag{8}$$

and for peak normalization we calculate the peak as

$$\bar{X}(m) = \max_{1 \le q \le Q} [X_q(m)]. \tag{9}$$

The energy-normalized feature vector is then given as†

$$\hat{X}_q(m) = X_q(m) - \hat{X}(m). \tag{10}$$

It can readily be shown that both peak and average normalization have the property that if a feature set $T$ is derived from the speech signal $s(n)$, then the feature set $T'$ derived from

$$s'(n) = \gamma s(n) \tag{11}$$

will be identical to $T$ after the normalization of eq. (10) is carried out. Hence, gain variations are normalized out of the processing as desired.

#### *3.2.2 Time smoothing of feature vectors*

The purpose of time smoothing of feature vectors is to reduce the variability in channel outputs by averaging adjacent time frames. The cost of such smoothing is a decrease in time resolution achieved by the recognizer. If we assume that $M^*$ adjacent frames are to be overlapped and smoothed, then time smoothing can be expressed as

$$\hat{X}_q(m) = \frac{1}{M^*} \sum_{\hat{m}=m-M^*+1}^{m} X_q(\hat{m}), \qquad m = M^*, \cdots, M. \tag{12}$$

† The reader is reminded that the channel signals are logarithmically encoded. Hence, normalization takes the form of subtraction.

It should be noted that the first $(M^* - 1)$ frames are eliminated and are used as initial conditions for the smoothing.

#### 3.2.3 Frequency smoothing of channel outputs

As in time smoothing, the purpose of frequency smoothing is to reduce the variability in channel outputs by averaging adjacent channels for a given time frame. Again, the cost of this smoothing is a loss in frequency resolution. If we assume that $Q^*$ adjacent channels are to be overlapped and smoothed, then frequency smoothing can be expressed as

$$\hat{X}_q(m) = \frac{1}{Q^*} \sum_{\hat{q}=q-Q^*+1}^{q} X_{\hat{q}}(m), \qquad q = Q^*, \cdots, Q. \tag{13}$$

It should again be noted that the first $(Q^* - 1)$ channels are eliminated and are used as initial conditions for the smoothing.

#### 3.2.4 Quantization of channel outputs

The purpose of quantizing the channel outputs is to reduce the storage requirements of the recognizer both for reference patterns and for the test pattern. If we use a $B$-bit quantizer, and we assume the channel signals are in the range $[0, -T^*]$ because of the thresholding and energy normalization operations, then with a uniform quantizer we have a quantization width of

$$\Delta E = \frac{T^*}{2^B} \tag{14}$$

and we can express the quantized output signal as

$$\hat{X}_q(m) = \lfloor X_q(m)/\Delta E \rfloor \cdot \Delta E, \tag{15}$$

where $\lfloor x \rfloor$ is the greatest integer less than $x$, and it is assumed that $\hat{X}_q(m)$ is already thresholded and energy normalized.

#### 3.2.5 Principal components analysis

The last form of postprocessing that we considered was a principal components analysis[7] in which the $Q$-dimensional filter bank feature vector is transformed into a new $P$-dimensional feature vector, where $P < Q$, such that all the important information in the original vector is retained. The purpose of reducing the feature dimensionality is to reduce the required storage for reference and test patterns. This method has also been used by Pols[7] to recognize vowels with good success.

The way in which the principal components analysis was performed was as follows. The first step is to collect a large number of filter bank

feature vectors and to compute the covariance matrix, $\Lambda$, between dimensions as

$$\Lambda_{ij} = \frac{\sum_{\cdot} [X_i(\cdot) - \bar{X}_i(\cdot)][X_j(\cdot) - \bar{X}_j(\cdot)]}{\left\{\sum_{\cdot} [X_i(\cdot) - \bar{X}_i(\cdot)]^2 \sum_{\cdot} [X_j(\cdot) - \bar{X}_j(\cdot)]^2\right\}^{1/2}}, \tag{16}$$

where the summation is over the training set of feature vectors. The principal components analysis then determines a new dimension, which is a linear combination of the original $Q$ dimensions, that contains as much of the total variance as possible. Then a second new dimension is determined such that it is orthogonal to the first new dimension and contains as much of the remaining variance as possible. This new dimension is again a linear combination of the $Q$ old dimensions. This process is continued until we have $P$ new orthogonal dimensions, all of which are linear combinations of the original $Q$ dimensions. Hence, if we denote the transformed set as $\hat{X}_{\hat{q}}(m)$, the transformation to the new dimensions is of the form

$$\hat{X}_{\hat{q}}(m) = \sum_{q=1}^{Q} \beta_{\hat{q}}(q) X_q(m), \qquad \hat{q} = 1, 2, \cdots, P, \tag{17}$$

where $\beta_{\hat{q}}(q)$ is the coefficient vector for dimension $\hat{q}$.

The resulting $P$-dimensional space of the principal components analysis contains as much of the total variance of the original space as is possible in $P$ dimensions. The new space is obtained formally by doing an eigenvector analysis of the original covariance matrix, $\Lambda$. The resulting eigenvectors are the coefficient vectors for the transformation of eq. (17).

### 3.3 *Dynamic time-warping considerations*

Once the feature vectors have been obtained, the recognizer must compare the unknown test pattern, $T$, to each word reference pattern, $R_i$, $i = 1, 2, \cdots, V$, for a $V$-word vocabulary. For this comparison the technique of dynamic time warping (DTW) is used.[2,8,9] If we denote the test pattern, $T$, as

$$T = \{T(1), T(2), \cdots, T(M)\} \tag{18}$$

and the $i$th reference, $R_i$, as

$$R^i = \{R^i(1), R^i(2), \cdots, R^i(N_i)\}, \tag{19}$$

then the DTW algorithm determines a warping path

$$n = w(m) \tag{20}$$

such that the total distance, $D(T, R^i)$, defined as

$$D(T, R^i) = \frac{1}{M} \sum_{m=1}^{M} d\{T(m), R^i[w(m)]\} \tag{21}$$

is minimized, where $d(T, R^i)$ is the local distance between test and reference frames.

We have considered several variations on the conventional DTW algorithm. First we have modified the global distance of eq. (21) to include a time weighting of the form

$$\hat{D}(T, R^i) = \frac{\sum_{m=1}^{M} W^T(m) d\{T(m), R^i[w(m)]\}}{\sum_{m=1}^{M} W^T(m)}, \tag{22}$$

where $W^T(m)$ is the weight applied to the local distance at frame $m$. It should be noted that in eq. (22) the weight is a function of only the test pattern, $T$.

We have also considered a variety of types of local distance calculations of the form

$$d(T, R) = \frac{\left\{ \sum_{q=1}^{Q} (W_q^T)^p [|\, T(q) - R(q) \,|^p] \right\}^{\frac{1}{p}}}{\left[ \sum_{q=1}^{Q} (W_q^T)^p \right]^{\frac{1}{p}}}, \tag{23}$$

where $W_q^T$ is a frequency weighting curve dependent only on the test pattern, $T$, and $p$ is the distance power for emphasizing the frequency variations. Typical values for $p$ are 1 (magnitude distance), 2 (squared distance), and 1/2 (square root distance). Again, it should be noted that the frequency weight of eq. (22) is only a function of the test frame.

An alternative form of distance weighting was suggested by Silverman and Dixon[10] and is of the form

$$d(T, R) = \frac{1}{Q} \sum_{q=1}^{Q} |\, T(q) - R(q) - f(|\, \bar{T} - \bar{R} \,|) \,|, \tag{24}$$

where

$$f(y) = 1 - \frac{y}{y^{\mathrm{MAX}}} \tag{25}$$

and $y^{\mathrm{MAX}}$ is the largest value that $y$ can attain. The form of eq. (24) is similar to that of the average normalization [eq. (8)] discussed earlier in that the means of $T$ and $R$ (over channels) are essentially subtracted

from each $T(q)$ and $R(q)$ component. However, this is only the case when $|\bar{T} - \bar{R}| \approx 0$, in which case $f(|\bar{T} - \bar{R}|) \approx 1$. For cases in which $|\bar{T} - \bar{R}|$ is large (i.e., close to the maximum difference of $T^*$), then $f(|\bar{T} - \bar{R}|) \approx 0$ and no mean correction is used. Thus, the weighting of eq. (24) places extra emphasis on regions of high average energy difference, and less emphasis on regions of low average energy difference. We denote the distance measure of eq. (24) as the Silverman-Dixon (SD) distance measure.

The last variation on the conventional DTW algorithm that we have investigated is the relaxation of the endpoint constraints on the warping path. Normally we use the simple constraints that

$$w(1) = 1 \qquad \text{Initial Point} \tag{26a}$$

$$w(M) = N_i \qquad \text{Final Point,} \tag{26b}$$

i.e., the first test frame is mapped to the first reference frame and the last test frame is mapped to the last reference frame. We have considered relaxation of both endpoint constraints of eq. (26) to the form

$$1 \le w(M_B) \le \delta_{\text{BEG}}, \qquad 1 \le M_B \le \delta_{\text{BEG}} \tag{27a}$$

$$N_i - \delta_{\text{END}} \le w(M_E) \le N_i, \qquad M - \delta_{\text{END}} \le M_E \le M. \tag{27b}$$

The new endpoint constraints say that the warping path can begin anywhere within a square of size $\delta_{\text{BEG}} \times \delta_{\text{BEG}}$ at the origin of the test-reference plane, and end anywhere with a square of size $\delta_{\text{END}} \times \delta_{\text{END}}$ at the upper right-hand corner of the test-reference plane. This situation is depicted in Fig. 3. By using local path constraints, which keep the slope of the warping path greater than 1/2 and less than 2, the warping path becomes constrained to lie within the shaded area of the test-reference plane.

### *3.4 The normalize-and-warp procedure*

The conventional DTW algorithm works quite well for most cases of interest. However, in cases when the length of the test pattern, $M$, is significantly different from the length of a reference pattern, $N$, then the region in the test-reference plane in which the warping path can lie often becomes very small. To handle such cases the normalize-and-warp procedure was devised,[4] and it basically consists of linearly prenormalizing both the test and reference patterns to a fixed length, $\hat{N}$, and then performing the DTW on these equal length patterns. In this manner the area in the test-reference plane in which the warping path can lie is maximized; hence we have the best chance of finding a good time-alignment path.

The normalize-and-warp procedure has been successfully used in a number of tests with an LPC recognizer[4,11-13] with very good results.

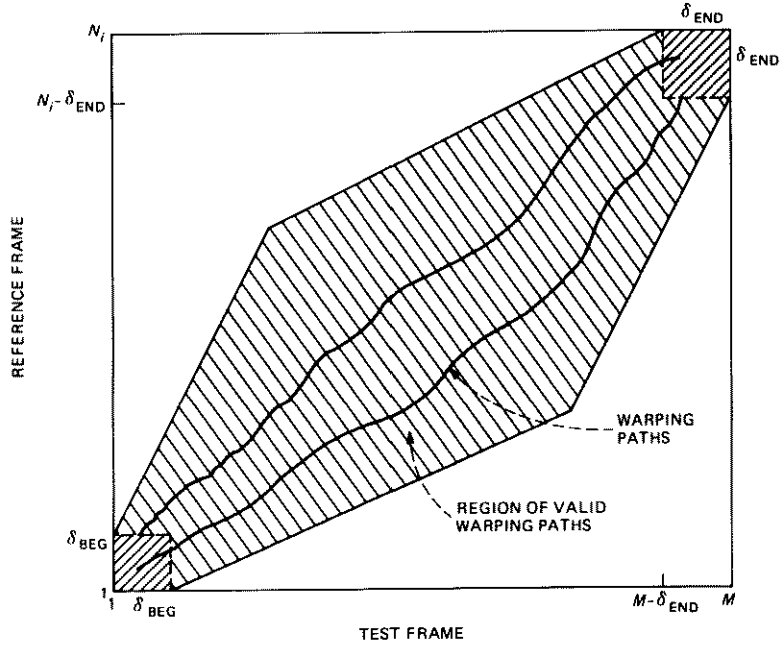



Fig. 3—The allowable warping region in the "test-reference" plane when the warping path has a beginning region of size $\delta_{BEG} \times \delta_{BEG}$ and an ending region of size $\delta_{END} \times \delta_{END}$.

For all these systems the fixed length to which all patterns were warped was the average duration of all words in the vocabulary. In this study we consider use of the normalize-and-warp procedure with the fixed length parameter a free variable.

### *3.5 Summary of signal processing choices*

In this section we have enumerated a number of ways of implementing the signal processing of a filter bank isolated word recognizer. These factors include

(*i*) Spectral preemphasis
(*ii*) Thresholding of channel signals
(*iii*) Energy normalization of channel signals
(*iv*) Time smoothing of feature vectors
(*v*) Frequency smoothing of channel outputs
(*vi*) Quantization of channel signals
(*vii*) Principal components analysis
(*viii*) Time weighting of local DTW distances
(*ix*) Frequency weighting of channel signals in DTW computation
(*x*) Local distance metric for DTW computation
(*xi*) Loosened endpoint constraints in DTW computation

(*xii*) Use of the normalize-and-warp procedure for DTW computation.

In Section V we give the actual choices that were studied for each of the above factors. First, however, we describe the tests of the noise immunity of both filter bank and LPC word recognizers.

## IV. NOISE STUDIES WITH THE ISOLATED WORD RECOGNIZER

Almost all tests of isolated word recognizers are made in a laboratory environment with a high signal-to-noise ratio on the recordings (e.g., greater than 35 dB is typical). There are a wide variety of applications (namely those of the military) in which the word recognizer is required to operate in noisy environments [e.g., signal-to-noise ratios (s/n) around 0 to 20 dB]. Thus, an important consideration in the evaluation of an isolated word recognizer is how the performance degrades as the background goes from laboratory conditions to highly noisy conditions.

When a word recognizer must operate in high-background-noise environments, an important issue arises, namely, whether it is better to train the system in a noise-free environment (and test in the noisy background), or to train and test in the noisy background. We have attempted to study these questions by artificially adding uncorrelated, zero mean, white noise to the speech signals at a specified signal-to-noise ratio, and then performing the required word recognition tests on both the filter bank and LPC word recognizers. A discussion of the test conditions and the results is given in the next section.

When one is concerned with using a word recognizer in an environment with a poor signal-to-noise ratio, another important consideration is whether one would "cancel" any of the noise by using a noise spectral estimation technique and subtracting the noise spectrum out. These techniques have been investigated in the context of voice coding[14-16] and have achieved various degrees of success. For sufficiently stationary noise backgrounds it seems reasonable to expect that a high degree of noise cancellation could be obtained. For such cases it would be of interest to understand how such noise cancellation algorithms work in the context of word recognition.

## V. EXPERIMENTAL RESULTS ON ISOLATED WORD RECOGNITION

To evaluate the effects on performance (word error rate) of each of the recognition system factors of Sections III and IV, a series of tests were run with the following specifications:

| | | |
|---|---|---|
| Vocabulary | — | 39 word alpha-digits |
| Number of talkers | — | 2 male, 2 female |
| Training | — | 7 replications for each word |
| Testing | — | 10 replications for each word. |

Table I—Word error rates for filter bank and LPC word recognizers

| Talker | Candidate Position 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| (a) Rates for baseline filter bank recognizer as a function of talker and candidate position | | | | | |
| 1 (Male) | 9.0 | 4.1 | 0.5 | 0.5 | 0.0 |
| 2 (Male) | 5.4 | 2.3 | 1.0 | 0.5 | 0.3 |
| 3 (Female) | 13.1 | 2.8 | 0.5 | 0.3 | 0.3 |
| 4 (Female) | 18.7 | 8.5 | 4.4 | 2.1 | 1.3 |
| Average | 11.6 | 4.4 | 1.6 | 0.9 | 0.5 |
| (b) Rates for LPC recognizer as a function of talker and candidate position | | | | | |
| 1 (Male) | 5.1 | 0.5 | 0.0 | 0.0 | 0.0 |
| 2 (Male) | 4.1 | 2.3 | 0.8 | 0.3 | 0.3 |
| 3 (Female) | 10.3 | 2.3 | 1.3 | 1.0 | 0.5 |
| 4 (Female) | 11.8 | 6.7 | 3.3 | 1.3 | 0.8 |
| Average | 7.8 | 3.0 | 1.4 | 0.7 | 0.4 |

All recordings were made over dialed-up telephone lines, and the test and training replications were obtained in different recording sessions. The speaker-dependent training used the robust training method[6] to give a single reference pattern for each vocabulary word.

The filter bank used was the 13-channel, critical-band spacing system that gave essentially the best performance in earlier tests.[1] A "baseline" filter bank recognizer was defined that had the following signal processing options:

(*i*) No preemphasis—$\alpha = 0$.

(*ii*) Channel thresholding at $T^* = 50$ dB below the peak in each channel.

(*iii*) Average energy normalization.

(*iv*) No time smoothing—$M^* = 1$.

(*v*) No frequency smoothing—$Q^* = 1$.

(*vi*) No quantization of channel signals (i.e., floating point accuracy)—$B = \infty$.

(*vii*) No principal components analysis.

(*viii*) Uniform time weighting of local distances—$W^T(m) = 1$, all $m$.

(*ix*) Uniform frequency weighting of local distances—$W_q^T = 1$, all $q$.

(*x*) Magnitude local distance—$p = 1$.

(*xi*) No opening up of DTW endpoint regions—$\delta_{\text{BEG}} = \delta_{\text{END}} = 1$.

(*xii*) No length prenormalization prior to the DTW.

(*xiii*) No additive noise—s/n(*Test*) = s/n(*Train*) = $\infty$.

The performance results of this baseline system are given in Table Ia and are shown plotted in Fig. 4a. Both the table and the figure show the word error rate as a function of candidate position for all four

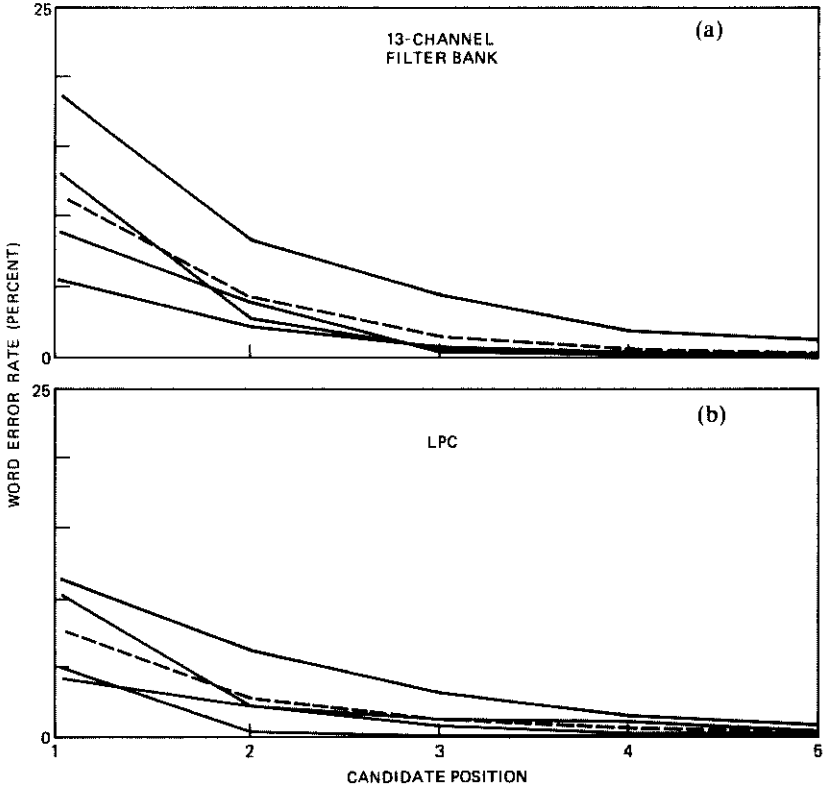



Fig. 4—Plots of word error rate versus candidate position for each of the four talkers and the average for (a) the baseline 13-channel filter bank system, and (b) the LPC word recognizer.

talkers (the solid curves in Fig. 4a) and the average (the dashed curve). These results show an average error rate of 11.6 percent in the top candidate position, with a high degree of variability in error rate across talkers. For comparison purposes, Table Ib and Fig. 4b show similar results on the LPC word recognizer. The average error rate for the top candidate position is about 4 percent lower for the LPC recognizer than for the 13-channel filter bank recognizer. Again we see a fair degree of variability in error rate scores across talkers for the LPC recognizer.

In the following sections we present results of tests designed to measure changes in performance of the filter bank recognizer as the factors noted above are varied. As discussed earlier we have been forced to use the expedient of only varying one parameter at a time; hence all information about interactions between two or more parameters is unavailable.

### 5.1 Effects of simple preemphasis

Only one value of the preemphasis constant, $\alpha$, was studied, namely, $\alpha = 0.95$. This is the value used in previous work on the LPC recognizer.[3] The results of recognition with the use of the preemphasis network are given in Table IIa (only results for top candidate position are included). It can be seen by comparing these results to those of the baseline system that a small improvement in average accuracy was obtained. This improvement was not statistically significant at the 0.9 confidence level.

### 5.2 Effect of clipping threshold

The values studied for the clipping threshold were $T^* = 40$ and $T^* = 30$ (dB). The resulting recognition scores are given in Table IIb for the top candidate position. The results for $T^* = 40$ are comparable to those for $T^* = 50$ in the baseline system, whereas for $T^* = 30$ a significant loss (2.4 percent) in word accuracy is obtained. Hence, a 30-dB range is deleterious to the channel signals in that useful recognition information is lost by clamping the signals at too high a level.

### 5.3 Effects of peak energy normalization

The results of using peak (rather than average) energy normalization of the channel signals are given in Table IIc.† The results show a large increase in word error rate for all talkers, thereby indicating a lack of stability of the peak in each frame and therefore its inappropriateness to be used as an energy normalization aid.

### 5.4 Effects of time smoothing

The value used for the smoothing duration, $M^*$, were 2 and 3 (frames). The recognition results for this condition are given in Table IId for the top candidate position. For $M^* = 2$ an insignificant increase in average error rate occurred, while for $M^* = 3$ there was a very significant increase in error rate. The results show that time smoothing produced far worse scores for female talkers (3 and 4) than for male talkers (1 and 2). This was undoubtedly due to the high variability in channel signals for the females (due to the high pitch frequency), which often led to smearing a "good" frame with a "bad" adjacent frame. The results indicate that time smoothing should not be done.

### 5.5 Effects of frequency smoothing

The results of smoothing across $Q^* = 2$ adjacent frequency channels are given in Table IIe. It can be seen that a uniform increase of about

† Recall that average normalization is one of the standard options used in the recognizer.

Table II—Word error rates for several signal processing techniques in the filter bank recognizer

| | Talker | | | | |
|---|---|---|---|---|---|
| (a) Rate with preemphasis ($\alpha = 0.95$) | | | | | |
| $\alpha$ | 1 | 2 | 3 | 4 | Average |
| 0.95 | 7.7 | 7.7 | 11.8 | 17.4 | 11.2 |
| (b) Rates as a function of threshold parameter, $T^*$ (dB) | | | | | |
| $T^*$ | 1 | 2 | 3 | 4 | Average |
| 40 | 8.7 | 6.9 | 13.3 | 17.4 | 11.6 |
| 30 | 9.5 | 8.5 | 17.2 | 21.0 | 14.0 |
| (c) Rates for peak energy normalization | | | | | |
| Method | 1 | 2 | 3 | 4 | Average |
| Peak | 23.1 | 19.5 | 14.4 | 40.5 | 24.4 |
| (d) Rates as a function of number of frames over which smoothing occurred, $M^*$ | | | | | |
| $M^*$ | 1 | 2 | 3 | 4 | Average |
| 2 | 7.9 | 5.6 | 13.1 | 21.3 | 12.0 |
| 3 | 7.7 | 5.6 | 26.9 | 29.7 | 17.5 |
| (e) Rates as a function of number of frequency channels over which smoothing occurred, $Q^*$ | | | | | |
| $Q^*$ | 1 | 2 | 3 | 4 | Average |
| 2 | 12.6 | 7.4 | 15.4 | 20.5 | 14.0 |
| (f) Rates as a function of $B$, the number of bits used to quantize the channel signals | | | | | |
| $B$ | 1 | 2 | 3 | 4 | Average |
| 6 | 6.7 | 5.6 | 14.4 | 17.4 | 11.0 |
| 4 | 11.3 | 7.2 | 16.9 | 18.7 | 13.5 |
| (g) Rates as a function of $P$, the dimensionality of the principal components analysis | | | | | |
| $P$ | 1 | 2 | 3 | 4 | Average |
| 12 | 11.3 | 9.9 | 25.1 | 29.5 | 18.7 |
| 6 | 11.3 | | | | |
| 4 | 11.5 | | | | |
| 2 | 18.7 | | | | |
| (h) Rates using a nonuniform time weighting in the DTW algorithm | | | | | |
| | 1 | 2 | 3 | 4 | Average |
| | 9.2 | 6.5 | 14.5 | 19.5 | 12.4 |
| (i) Rates using a nonuniform frequency weighting in the DTW algorithm | | | | | |
| Weight | 1 | 2 | 3 | 4 | Average |
| Fig. 5 | 8.7 | 8.7 | 16.2 | 21.3 | 13.7 |
| SD | 8.2 | 7.4 | 16.4 | 17.4 | 12.4 |
| (j) Rates as a function of $p$, the power in the local distance computation | | | | | |
| $p$ | 1 | 2 | 3 | 4 | Average |
| 2 | 12.3 | 7.1 | 21.4 | 23.1 | 16.0 |
| ½ | 10.0 | 6.2 | 13.1 | 20.8 | 12.5 |

(k) Rates as a function of the opening region parameters $\delta_{BEG}$ and $\delta_{END}$ of the DTW algorithm

| $\delta_{BEG}$ | $\delta_{END}$ | Region | 1 | 2 | 3 | 4 | Average |
|---|---|---|---|---|---|---|---|
| 2 | 0 | Square | 8.7 | 5.6 | 13.1 | 22.3 | 12.4 |
| 0 | 4 | Square | 7.9 | 5.4 | 13.8 | 21.5 | 12.2 |
| 2 | 4 | Square | 8.5 | 5.6 | 14.4 | 22.8 | 12.8 |
| 2 | 4 | Line | 8.9 | 6.2 | 24.6 | 22.1 | 15.5 |

2.5 percent in word error rate is obtained for all four talkers. Thus we conclude that smoothing across channels leads to a loss in information for recognition and therefore should not be used.

### 5.6 Quantization of channel signals

The results on channel signal quantization are presented in Table IIf. It can be seen that quantization of the channel signals to 6 bits actually decreases the average error rate by 0.6 percent; however, a further reduction to 4 bits leads to a 1.9-percent increase in error rate over the baseline system. Hence the results indicate that 6-bit quantization is adequate for the channel signals.

### 5.7 Results using principal components analysis

The results obtained using the principal components analysis for a single talker are presented in Table IIg. It can be seen that for $P = 12$ a 7.1-percent increase in average error rate is obtained; however, small increases in error rate were attained for reductions in $P$ down to 4. In fact, for talker 1 the word error rate increased by 0.2 percent in going from 12 to 4 principal components.

An explanation of why the $P = 12$ principal components analysis led to such large increases in error rate is as follows. The transformation of the feature vector used in the principal components analysis has the property that it is invariant to a quadratic distance measure. The distance measure used in the baseline system was an absolute value distance; hence a significant decrease in accuracy resulted. We will show in Section 5.10 that using a quadratic distance measure gave much worse recognition accuracy than the absolute distance. Thus it would appear that the principal components analysis is not a useful tool, at least for this particular filter bank word recognizer.

### 5.8 Results using time weighting in the global distance for DTW

The results using nonuniform time weighting in the DTW global distance calculation are given in Table IIh. The actual weighting function, $W^T$, was a function only of the energy in the test pattern, of the type shown in Fig. 5. The test energy, $E^T$, was estimated as the sum of the individual channel energies. A high correlation ($\approx$0.94) was measured between this estimate of test energy, and the actual test energy (as computed from the raw speech samples). For frames in which the test energy was within 20 dB of the peak energy (suitably normalized to 0 dB), the frame weight was 1.0; for frames with $E_T$ less than 40 dB below the peak, the weight was set to 0.01; a linear interpolation of the weight was used for frames with $-40 \leq E_T \leq -20$ dB.

The results in Table IIh show a small increase in word error rate for

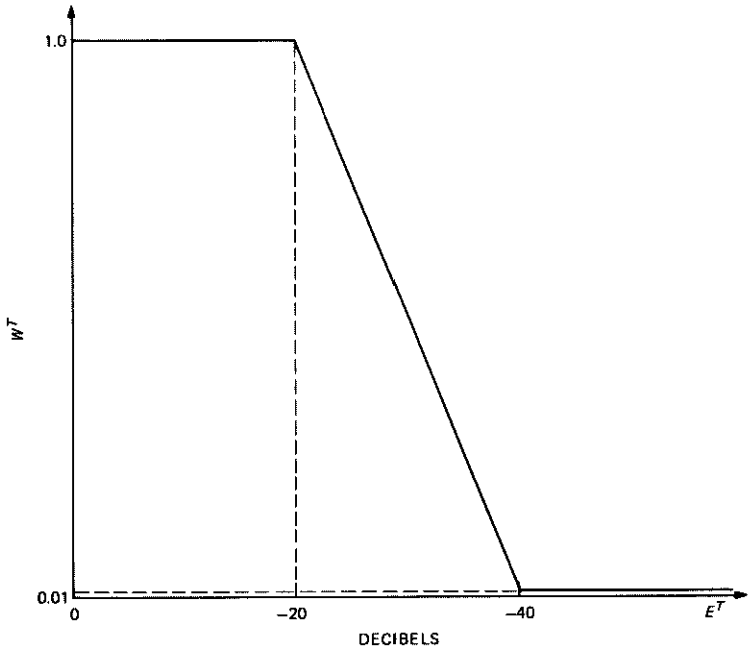



Fig. 5—The nonlinear weighting function, $W^T$, on the local distance as determined from the test energy, $E^T$, estimated from the sum of channel outputs.

*each* talker. Hence we conclude that the addition of time weighting (of the form of Fig. 5) in the DTW distance calculation is unnecessary.

### *5.9 Results using frequency weighting in the local distance for DTW*

The results of using a nonuniform frequency weighting for local distances in the DTW algorithm are shown in Table IIi. The frequency-weighting characteristic was identical to the time-weighting characteristic of Fig. 5, except that the abscissa was the individual channel energy (relative to the peak channel energy for the word) and the ordinate was frequency weight $W_q$. It can be seen from Table IIi that a 2.1-percent increase in average word error rate is obtained using the nonuniform frequency weighting. Hence we again conclude that such weighting should not be used for this particular filter bank recognizer.

The results of using the SD weighting proposed by Silverman and Dixon (based on both reference and test frame energies) are also shown in Table IIi. Although the average word error rate is lower than for the nonuniform weighting of Fig. 5, it is still about 0.8 percent higher than obtained using simple uniform weights.

### 5.10 Effects of different local distance computations

The results of using different local distance computations in the DTW algorithm are given in Table IIj. Values for $p$ of 2 (squared distance) and 1/2 (square root distance) were considered. It can be seen that for $p = 2$ a 4.4-percent increase in average word error rate is obtained; for $p = 1/2$ the increase in average word error rate is 0.9 percent. These results indicate that the magnitude distance ($p = 1$) is the best compromise between giving extra weight to very different channel energies ($p = 2$) and giving a small weight to very different channel energies ($p = 1/2$).

### 5.11 Results of opening up the DTW starting and ending regions

The results of opening up the beginning and/or ending region of the DTW speech regions are given in Table IIk. Results are given for an initial or final square search region, as well as for an initial or final line search region (i.e., the path had to begin or end at the first or last frame of *either* the test or reference; it could not begin or end at a noninitial or nonfinal frame of *both*). It can be seen that all the cases studied led to a small (for square regions) or a large (for the line region) increase in average word error rate. This result is anticipated from previous results, which have shown that opening up the DTW search region consistently aids false matches (reference and test different) as much or more than true matches (reference and test the same).[3,4,17]

### 5.12 Results using length normalization prior to DTW

The results of using fixed-length word normalization prior to the DTW (the normalize-and-warp procedure) are given in Table III and Fig. 6. Table IIIa and Fig. 6a show results for the 13-channel filter bank and Table IIIb and Fig. 6b show results for the LPC-based recognizer. The results show that for a broad range of warping lengths (from 20 to 45 frames) the average word accuracy does not change significantly. Significant degradation in performance is obtained only for the shortest warping lengths considered (i.e., 10 and 15 frames). Hence the results indicate that the normalize-and-warp procedure is suitable for a sizeable range of warping lengths so long as the length used does not become too small.

### 5.13 Results on noise studies

The results of the noise studies are given in Table IV and plotted in Fig. 7. Results are given for three cases:

(*i*) Signal-to-noise ratio (s/n)(Test) = s/n(Train), where s/n varied from $\infty$ down to 0 dB (Table IVa, Fig. 7a).

Table III—Word error rates for the normalize-and-warp procedure applied to both the filter bank and LPC word recognizers

| | | Length of Reference and Test | | | | | |
|---|---|---|---|---|---|---|---|
| (a) Rates as a function of warping length of reference and test prior to DTW for the 13-channel recognizer | | | | | | | |
| Talker | Variable | 40 | 30 | 25 | 20 | 15 | 10 |
| 1 | 9.0 | 8.7 | 9.5 | 8.7 | 8.7 | 10.3 | 12.6 |
| 2 | 5.4 | 5.6 | 5.4 | 5.9 | 5.6 | 6.9 | 11.5 |
| 3 | 13.1 | 13.1 | 14.6 | 14.1 | 14.6 | 15.4 | 19.5 |
| 4 | 18.7 | 21.8 | 20.8 | 21.8 | 20.8 | 22.8 | 23.3 |
| Average | 11.6 | 13.1 | 12.6 | 12.6 | 12.7 | 13.9 | 16.7 |

| (b) Rates as a function of warping length of reference and test prior to DTW for the LPC-based recognizer | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Talker | Variable | 45 | 40 | 35 | 30 | 25 | 20 | 15 | 10 |
| 1 | 5.1 | 4.9 | 6.2 | 4.6 | 4.6 | 5.1 | 4.9 | 8.5 | 11.8 |
| 2 | 4.1 | 4.6 | 3.8 | 4.9 | 5.1 | 4.9 | 5.6 | 6.9 | 7.4 |
| 3 | 10.3 | 8.2 | 9.0 | 7.9 | 8.5 | 10.0 | 10.5 | 12.6 | 12.8 |
| 4 | 11.8 | 11.3 | 12.1 | 12.6 | 11.5 | 12.6 | 12.8 | 14.1 | 20.0 |
| Average | 7.8 | 7.3 | 7.8 | 7.5 | 7.4 | 8.2 | 8.5 | 10.5 | 13.0 |

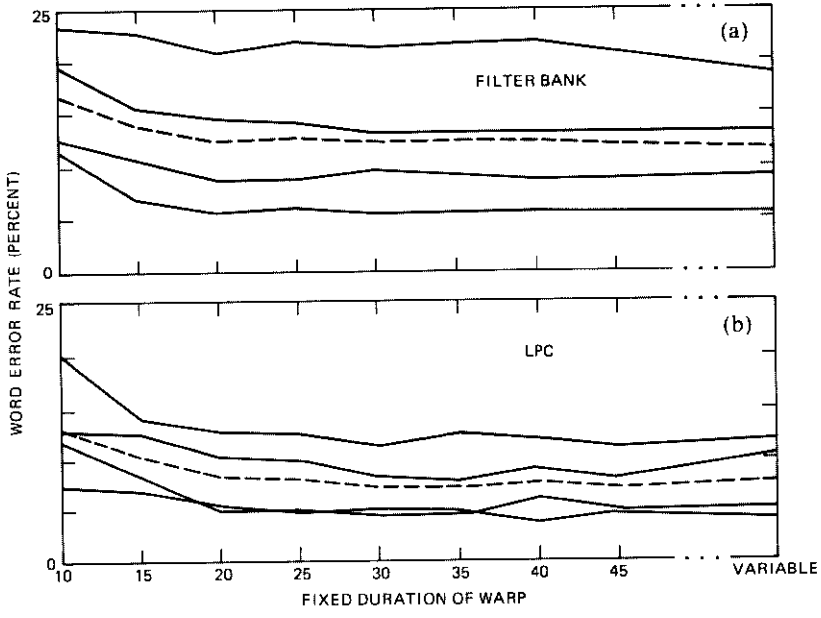



Fig. 6—Plots of word error rate versus fixed frame duration for linear prewarp prior to DTW alignment for (a) the filter bank system, and (b) the LPC system.

(*ii*) s/n(Train) = ∞, s/n(Test) variable from ∞ down to 0 dB (Table IVb, Fig. 7b).

(*iii*) s/n(Test) = 18 dB, s/n(Train) variable from ∞ down to 0 dB (Table IVc, Fig. 7c).

The first case represents the situation when both training and testing

Table IV—Word error rates for noise studies

| | s/n (dB) | | | | | | |
|---|---|---|---|---|---|---|---|
| (a) Rates for filter bank (FB) and LPC word recognizers as a function of s/n for case when noise added to both test and reference signals | | | | | | | |
| System | ∞ | 30 | 24 | 18 | 12 | 6 | 0 |
| FB | 9.0 | 9.5 | 11.0 | 13.1 | 13.6 | 16.7 | 21.0 |
| LPC | 5.1 | 3.8 | 6.2 | 7.4 | 11.3 | 16.7 | 23.6 |
| (b) Rates for filter bank (FB) and LPC word recognizers as a function of s/n for the case when noise added to test only—i.e., s/n of reference for training was ∞ | | | | | | | |
| System | ∞ | 30 | 24 | 18 | 12 | 6 | 0 |
| FB | 9.0 | 14.4 | 17.3 | 32.6 | 61.0 | 82.3 | 92.3 |
| LPC | 5.1 | 10.0 | 17.4 | 37.2 | 65.1 | 76.9 | 90.8 |
| (c) Rates for filter bank (FB) and LPC word recognizers as a function of s/n for case when noise added to reference at variable s/n, and with test s/n set to 18 dB | | | | | | | |
| System | ∞ | 30 | 24 | 18 | 12 | 6 | 0 |
| FB | 32.6 | 12.3 | 11.8 | 13.1 | 14.1 | 18.5 | 89.5 |
| LPC | 37.2 | 10.3 | 7.2 | 7.4 | 10.0 | 17.2 | 31.5 |
| (d) Rates for filter bank (FB) and filter bank with noise removal (FB/NR) for case when noise added to reference at variable s/n, and with test s/n set to 18 dB | | | | | | | |
| System | ∞ | 30 | 24 | 18 | 12 | 6 | 0 |
| FB | 32.6 | 12.3 | 11.8 | 13.1 | 14.1 | 18.5 | 89.5 |
| FB/NR | 26.2 | 12.6 | 11.8 | 13.1 | 14.4 | 18.7 | 28.2 |

of the word recognizer are done in the same noisy background; case (*ii*) represents the situation when there is "clean" training (no noise) but the test words are spoken in the noisy background; case (*iii*) represents the situation when there is noise in both training and testing; however there may be a mismatch in s/n.

The results in Table IV and Fig. 7 show that:

(*i*) For case (*i*), the LPC system performs as well as or better than the filter bank (FB) system for s/n $\geq$ 6 dB. The filter bank (FB) system outperformed the LPC system only at a 0 dB s/n.

(*ii*) For case (*i*), there was little degradation in performance down to s/n's of close to 24 dB for either the FB or LPC recognizer.

(*iii*) For case (*ii*) the performance of both the FB and LPC recognizers was significantly worse at all s/n's than for case (*i*). Hence we see that using clean training data with noisy test data leads to badly degraded system performance for s/n $\leq$ 30 dB.

(*iv*) For case (*iii*) the results indicate that when s/n(Test) and s/n(Train) differ by as little as 6 dB (or more) degraded performance results.

The results of Table IV and Fig. 7 indicate that it is mandatory that both the training (reference) and testing data be obtained in the same background noise conditions for best word recognition performance.

A test was also conducted on the filter bank recognizer to determine whether the effects of additive (background) noise could be lessened by subtracting out an (estimated) average noise spectrum prior to recognition. A one-second average was calculated for each channel

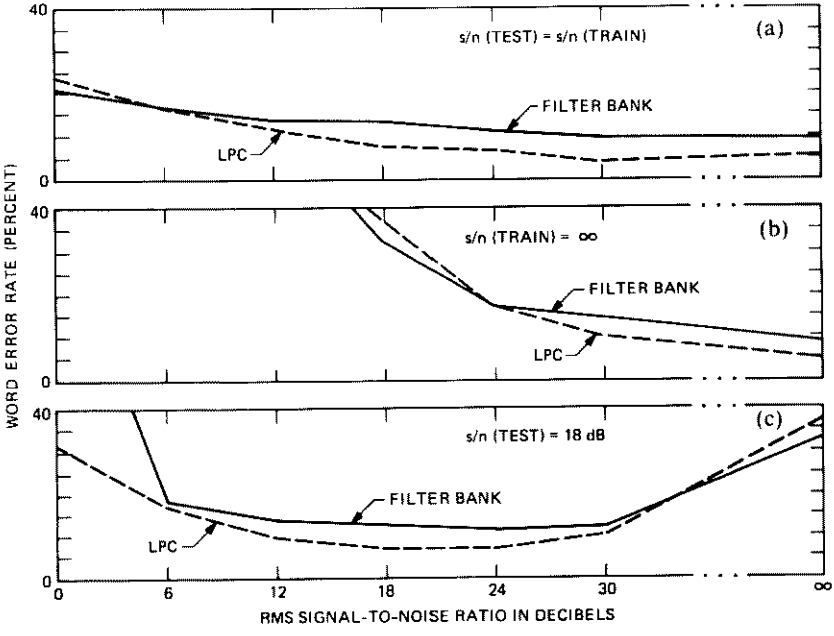



Fig. 7—Plots of word error rate (for a single talker) versus signal-to-noise ratio for both the LPC and filter bank recognizers for, (a) s/n(Test) = s/n(Train); (b) s/n(Train) = ∞, s/n(Test) variable; and (c) s/n(Test) = 18 dB, s/n(Train) variable.

signal of the filter bank when only the additive white noise was present at the input. Each of these 13-channel average noise values was then subtracted from the corresponding channel signal to form a new channel signal, which was used in the recognition processing. For these tests the signal-to-noise ratio for the test data was held constant at 18 dB, and the noise level in the reference data was varied to give signal-to-noise ratios between 0 dB and ∞.

The results of this experiment are given in Table IVd. These results show that the effects of this simple noise-cancelling arrangement are to broaden the range of signal-to-noise ratios over which the filter bank recognizer can operate. It can be seen that recognizer performance is not changed significantly between 30-dB and 6-dB signal-to-noise ratios from that obtained without the noise cancellation. However, for signal-to-noise ratios of ∞ and 0 dB, a considerable reduction of error rate is obtained with the noise cancellation method. The conclusion from this test is that noise cancelling is useful for reducing the effects of variations in noise level between reference and test data.

## VI. DISCUSSION AND CONCLUSIONS

The results presented in Section V lead to the following conclusions:

(*i*) Essentially none of the proposed signal processing techniques

for use in the filter bank word recognizer led to an improvement in performance of the system (i.e., reduced word error rate). At best any single technique led to a small (insignificant) increase or decrease in word error rate; at worst it led to a significant increase in word error rate.

(*ii*) The filter bank coefficients (for telephone inputs) needed only about 6 uniform bits for a representation with no increase in word error rate. Hence, the storage requirements on the $Q = 13$ channel recognizer were about 78 bits per frame using this 6-bit coding scheme.

(*iii*) The normalize-and-warp procedure was an effective method for reducing storage and processing requirements in the DTW computation in that fixed duration linear prewarps of size as small as 20 frames per word did not increase word error rate significantly for either the LPC or FB recognizers.

(*iv*) The best strategy for using a word recognizer in a noisy background was to both train and test the recognizer in the same noise background.

(*v*) The LPC word recognizer gave error rates the same or lower than the FB word recognizer for s/n $\geq$ 6 dB.

Our initial goal was to find signal processing techniques to enhance the performance of the FB word recognizer so as to come closer to that of an existing LPC word recognizer. Our results indicate that we have not succeeded in attaining this goal. Hence our main question is whether we failed because we tried the wrong things, or because there is no way of doing consistently better with the FB limitations. There is no simple answer to this question. Perhaps our best response is that we tried a wide range of techniques that encompassed those methods previously proposed and studied in other FB recognizers. The lack of any significant improvement in performance for any of the proposed techniques indicates to us that perhaps the only way to improve accuracy is by some heuristic based on linguistic knowledge of the vocabulary words. We have meticulously avoided such techniques as they change the nature of the recognizer from a vocabulary-independent system to one that depends on the specific words to be recognized.

Another possible objection to the conclusions as drawn from the results given in Section V is that we studied each proposed signal processing technique independently. As such we avoided interactions between techniques that could have led to improved accuracy. Again we iterate our speculation that since no individual technique led to a real performance improvement, we are skeptical that combinations of techniques would lead to real improvements. Of course we have no concrete evidence that this is indeed the case.

Our noise analysis results dispel the common notion that LPC recognizers "fall apart" in noisy backgrounds while FB recognizers

degrade gracefully. Our results show that with proper training the LPC system outperforms the FB system at all reasonable signal-to-noise ratios.

Finally, the noise results show that training and testing should *always* be done in the same acoustic backgrounds. If there are gross differences in acoustic backgrounds, significant degradation in performance results.

## VII. SUMMARY

We have presented results of a study to measure the effects of selected signal processing techniques on the performance of a filter bank word recognizer. We have shown that a fairly simple set of signal processing techniques led to the best overall performance of the word recognizer in the noise-free case. In noisy conditions the performance of the recognizer degraded significantly for signal-to-noise less than about 24 dB.

## REFERENCES


1. B. A. Dautrich, L. R. Rabiner, and T. B. Martin, "On the Effects of Varying Filter Bank Parameters on Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-31*, No. 4 (August 1983).
2. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-23* (February 1976), pp. 67–72.
3. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," IEEE Trans. on Communication, *COM-29* (May 1981), pp. 621–59.
4. C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance Trade-offs in Dynamic Time Warping Algorithms for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-28* (December 1980), pp. 623–35.
5. J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, New York, NY: Springer-Verlag, 1976, Chapter 9, pp. 212–26.
6. L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," J. Acoust. Soc. Amer., *68* (November 1980), pp. 1271–6.
7. L. C. W. Pols, "Real-Time Recognition of Spoken Words," IEEE Trans. Comput., *C-20* (September 1971), pp. 972–8.
8. H. Sakoe and S. Chiba, "Dynamic Programming Optimization for Spoken Word Recognition," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-26* (February 1978), pp. 43–9.
9. L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-26*, No. 6 (December 1978), pp. 575–82.
10. H. F. Silverman and N. R. Dixon, "A Comparison of Several Speech-Spectra Classification Methods," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-24*, No. 4 (August 1976), pp. 289–95.
11. J. G. Wilpon, L. R. Rabiner, and A. Bergh, "Speaker Independent Isolated Word Recognition Using a 129 Word Airline Vocabulary," J. Acoust. Soc. Am., *72* (August 1982), pp. 390–6.
12. M. K. Brown and L. R. Rabiner, "An Adaptive Ordered Graph Search Technique for Dynamic Time Warping for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-30*, No. 4 (August 1982), pp. 535–44.
13. M. K. Brown and L. R. Rabiner, "On the Use of Energy in LPC-Based Recognition of Isolated Words," B.S.T.J., *61*, No. 10, Part 1 (December 1982), pp. 2971–87.
14. S. F. Boll, "Suppression of Acoustic Noise in Speech Using Spectral Subtraction,"

IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-29,* No. 2 (April 1979), pp. 113–20.
15. J. S. Lim and A. V. Oppenheim, "Enhancement and Bandwidth Compression of Noisy Speech," Proc. IEEE, *67,* No. 12 (December 1979), pp. 1586–604.
16. M. M. Sondhi, C. E. Schmidt, and L. R. Rabiner, "Improving the Quality of a Noisy Speech Signal," B.S.T.J., *60,* No. 8 (October 1981), pp. 1847–59.
17. L. R. Rabiner, "Note on Some Factors Affecting Performance of Dynamic Time Warping Algorithms for Isolated Word Recognition," B.S.T.J., *61,* No. 3 (March 1982), pp. 363–73.

## CONTRIBUTORS TO THIS ISSUE

**Harry J. Boll,** B.S., M.S., Ph.D. (Electrical Engineering), University of Minnesota, Minneapolis, in 1956, 1958, and 1962, respectively; Bell Laboratories, 1962—. After completing graduate work in secondary electron emission, he joined Bell Laboratories, Murray Hill, NJ, where he has worked in the general area of semiconductor devices and integrated circuits. He is presently a Group Supervisor in the Advanced Large-Scale Integration Development Laboratory, where his duties include very large-scale integrated logic design. Fellow, IEEE; member, Eta Kappa Nu, Tau Beta Pi. He received a Bell Laboratories Fellowship award for his work in custom integrated circuits.

**Ta-Shing Chu,** B.S., 1955, National Taiwan University; M.S., 1957, and Ph.D., 1960, Ohio State University; Research Associate, Courant Institute of Mathematical Sciences, New York University, 1961–1963; Bell Laboratories, 1963—. Mr. Chu has been engaged in research on microwave antennas and wave propagation for satellite communication and terrestrial radio systems. His work included surface wave diffraction, precision gain standard, propagation through precipitation, dual-polarization radio transmission, and the Crawford Hill 7-meter offset millimeter wave antenna. Fellow, IEEE; Member, International Scientific Radio Union, Sigma Xi, Pi Mu Epsilon.

**Bruce A. Dautrich,** B.S. (Electrical Engineering), 1980, Pennsylvania State University; M.S. (Electrical Engineering), 1981, Purdue University; Bell Laboratories, 1980—. Mr. Dautrich has been engaged in speech communication research and development and is presently concentrating on problems of speech recognition. Member Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi.

**Bharat T. Doshi,** B. Tech (Mechanical Engineering), 1970, I.I.T., Bombay; Ph.D. (Operations Research), 1974, Cornell University; Bell Laboratories, 1979—. Before joining Bell Laboratories Mr. Doshi was an Assistant Professor at Rutgers University. At Bell Laboratories his technical work has included modeling and analysis of processor schedules and overload controls. His research interests include queueing theory applied to performance analysis of computer and communication systems. Member, IEEE, ORSA; Associate Editor, OR Letters.

**Robert W. England,** B.S., 1973, Capitol Institute of Technology; Bell Laboratories, 1973–1981; AT&T Long Lines, 1981—. At Bell Laboratories, Mr. England worked on microwave antennas for satellite

communication. He was involved in the measurement of the Crawford Hill 7-meter antenna and the development of an earth station receiver for the ATS-6 satellite. Since coming to Long Lines, Mr. England has been Senior Engineer on Long Lines' new coaxial cable transmission system.

**Michael J. Grieco,** B.S. (Chemistry), 1966, Rutgers University, New Brunswick, NJ; Bell Laboratories, 1961–68, 1972—. Mr. Grieco has worked in many aspects of semiconductor processing such as photolitography, etching, C.V.D., and wafer cleaning. He has been involved in photolithographic process engineering for the *BELLMAC**-8, -4, and most recently the *BELLMAC*-32 microprocessor. This process engineering work is being continued in 1-$\mu$m MOS device technology.

**Raghu N. Kackar,** B.Sc. (Mathematics, Physics, Chemistry), 1970, University of Delhi; M. Stat., 1972, Agra University; M.S. (Statistics), 1974, University of Guelph; Ph.D. (Statistics), 1979, Iowa State University; Bell Laboratories, 1980—. Mr. Kackar is applying new methods for total quality control to enhance quality and productivity of several production lines. At present his primary emphasis is on development and application of the technique of designed experiments. Member, ASA, ASQC.

**Thomas B. Martin,** B.S.E.E., 1957, University of Notre Dame; M.S.E.E., Ph.D., University of Pennsylvania, 1960 and 1970, respectively. In 1970 he co-founded Threshold Technology Inc., which developed the first practical automatic speech recognition systems to be used in industry. In March 1982, he joined Bell Laboratories and has continued to work in speech recognition.

**Stephen C. Mettler,** B.S., 1962, U.S.A.F. Academy; M.S. (Physics), 1972, and Ph.D. (M.E.), 1976, Purdue University; Bell Laboratories, 1976—. Mr. Mettler is presently engaged in research on optical fiber components and splicing. Member, OSA.

**Kinichiro Ogawa,** B.S.E.E., 1966, M.S.E.E., 1968, University of Tokyo, D.Sc. (Electrical Engineering), Washington University; Nippon Telegraph & Telephone Public Corporation, 1968–1976; Bell Laboratories, 1976—. At N.T.T., Mr. Ogawa worked on long-haul analog coaxial cable systems and on video transmission. He also worked on

* Trademark of Bell Laboratories

the development of pulse-code modulated/frequency division multiplex digital systems. At Bell Laboratories, he has been involved in research on optical data links and digital lightwave systems. Member, IEEE, Optical Society of America, IECE of Japan.

**Brian Owen,** B.Sc., 1963; Ph.D., 1967 (Electrical Engineering), Birmingham University, England; Bell Laboratories, 1968—. Mr. Owen is a Member of the Technical Staff in the Lightwave Subsystems Department at Allentown. From 1968 to 1977 he was involved in the design of microwave and millimeter-wave components. Since 1977 he has been involved in photonics working primarily on optical receivers.

**Madhav Phadke,** B. Tech, 1969 (Mechanical Engineering), Indian Institute of Technology, Bombay; M.S., 1972 (Statistics), and Ph.D., 1973 (Mechanical Engineering), University of Wisconsin, Madison; Bell Laboratories, 1977—. Prior to joining Bell Laboratories, Mr. Phadke worked as a Research Associate in the Statistics Department of the University of Wisconsin and as a Visiting Scientist at the IBM Watson Research Center. He has published articles in professional journals covering topics in quality assurance, design of experiments, process control, system identification, time series analysis, mechanical design, and air pollution data analysis. His current interests are in the applications of Japanese methods of quality control and productivity improvement in design, development, and manufacturing of products. Member, Phi Kappa Phi, Sigma Xi, American Statistical Association, the American Society for Quality Control.

**Lawrence R. Rabiner,** S.B. and S.M., 1964, Ph.D. (Electrical Engineering), The Massachusetts Institute of Technology; Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice Hall, 1983). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication of the Acoustical Society, ASSP Technical Committee on Speech Communication; Member, IEEE Proceedings Editorial Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

**Donald V. Speeney,** Assoc. in Technology (Electronics), 1960, Temple University; Bell Laboratories, 1960—. Since joining Bell Laboratories, Mr. Speeney has worked in the area of semiconductor device technology development and device physics. This includes work on the discrete components of germanium, silicon, and gallium arsenide, as well as CMOS microcomputer and microprocessor integrated circuits. More recently, he has been working on processes and technology for producing silicon integrated circuits.

**York Y. Wang,** B.S. (Physics), 1970, Chung-Yuan Christian College, Taiwan; M.S. (Physics), 1972, and Ph.D. (Physics), 1976, University of Missouri at Columbia; Jet Propulsion Laboratory, 1975–1977, Bell Laboratories, 1977—. At Jet Propulsion Laboratory, Mr. Wang worked on Viking and Voyager Projects as a mission planner. At Bell Laboratories, he has been involved in microwave radio propagation studies, space diversity signal combining techniques, digital microwave radio systems performance and applications, digitization of analog microwave radio systems, and advanced network services planning. Member, IEEE, Sigma Xi.

**Ian A. White,** B.Sc. (Hon.) (Physics), 1974, University of Queensland, Australia; Ph.D. (Applied Mathematics), 1977, Australian National University; Bell Laboratories, 1979—. From 1977 to 1979 Mr. White did post-doctoral research in the theory of fiber and integrated optics at the Optical Sciences Center, University of Arizona at Tucson. Since 1979 he has been with Bell Laboratories, Norcross, Georgia, where he is a member of the Exploratory Components and Media Studies Group.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Abramovici M., Breuer M. A., **Fault-Diagnosis in Synchronous Sequential-Circuits Based on an Effect Cause Analysis.** IEEE Comput 31(12):1165–1172, 1982.

Brown D. J., Baker B. S., Katseff H. P., **Lower Bounds for Online Two-Dimensional Packing Algorithms.** ACT Inform 18(2):207–225, 1982.

Ding Z. D., Hwang F. K., **Minimizing a Combinatorial Function.** Siam J Alg 3(4):523–528, 1982.

Finkel R. A., Fishburn J. P., **Improved Speedup Bounds for Parallel Alpha-Beta Search (Letter).** IEEE Patt. A 5(1):89–92, 1983.

Flatto L., **A Geometric Problem in Approximation-Theory.** J Approx Th 36(4):321–327, 1982.

Golumbic M. D., Rotem D., Urrutia J., **Comparability-Graphs and Intersection Graphs.** Discr Math 43(1):37–46, 1983.

Irani K. B., Chen K. W., **Minimization of Interprocessor Communication for Parallel Computation.** IEEE Comput 31(11):1067–1075, 1982.

Kasami T., Lin S., Wei V. K., Yamamura S., **Graph Theoretic Approaches to the Code Construction for the 2-User Multiple-Access Binary Adder Channel.** IEEE Info T 29(1):114–130, 1983.

Lagarias J. C., **Best Simultaneous Diophantine Approximations. 2. Behavior of Consecutive Best Approximations.** Pac J Math 102(1):61–88, 1982.

Melici J. A., **The BX-25 Certification Facility.** Comput Netw 6(5):319–329, 1982.

Nair V. N., **Q-Q Plots with Confidence Bands for Comparing Several Populations.** Sc J Stat 9(4):193–200, 1982.

## ENGINEERING

Anthony P. J., **Alteration of Diffusion Profiles in Semiconductors Due to p-n-Junctions.** Sol ST Elec 25(12):1171–1177, 1982.

Bohrer M. P., **Diffusional Boundary-Layer Resistance for Membrane-Transport.** Ind Eng F 22(1):72–78, 1983.

Chen P. C., Pavlidis T., **Segmentation by Texture Using Correlation (Letter).** IEEE Patt A 5(1):64–69, 1983.

Cheng J., Boyd G. D., Westerwick E. H., **Reduction of Threshold Voltage for Bistable Liquid-Crystal Switching Using a Priming Voltage.** IEEE Device 29(12):1853–1856, 1982.

Damen T. C., Duguay M. A., Miller B. I., Lin C., **Single-Mode Semiconductor-Laser Optically Pumped by an Injection-Laser.** Electr Lett 19(1):3–4, 1983.

Emkey W. L., Bogert G. A., **Multiport Polarization Monitor.** Optics Lett 8(2):94–95, 1983.

Engelmaier W., Frisch D. C., **Injection-Molding Shapes New Dimensions for Boards.** Electronics 55(25):155–158, 1982.

Falcone R. W., Bokor J., **Dichroic Beam Splitter for Extreme-Ultraviolet and Visible Radiation.** Optics Lett 8(1):21–23, 1983.

Fichtner W., Cooper J. A., Tretola A. R., Kahng D., **A Novel Buried-Drain DMOSFET Structure.** IEEE Device 29(11):1785–1791, 1982.

Fork R. L., Shank C. V., Hirlimann C., Yen R., Tomlinson W. J., **Femtosecond White-Light Continuum Pulses.** Optics Lett 8(1):1–3, 1983.

Forrest S. R., **Photo-Diodes for Long-Wavelength Communication-Systems.** Laser Focus 18(12):81+, 1982.

Gilbert J. A., et al., **The Application of Fiber Optics to Remote Speckle Metrology Using Incoherent-Light.** Opt Laser E 3(3):183–195, 1982.

Glass A. M., Liao P. F., Olson D. H., Humphrey L. M., **Optical Metal-Oxide Tunnel Detectors with Microstructured Electrodes.** Optics Lett 7(12):575–577, 1982.

Gloge D. C., **Citation Classic—Multimode Theory of Graded-Core Fibers.** CC/Eng Tech 1983(4):22, 1983.

Herbst D., Bosch M. A., Lemons R. A., Tewksbury S. K., Harrison T. R., **PMOS Transistors Fabricated in Large-Area Laser-Crystallized Si on Silica.** Electr Lett 19(1):12–13, 1983.

Hu E. L., Tennant D. M., Howard R. E., Jackel L. D., Grabbe P., **Vertical Silicon Membrane Arrays Patterned With Tri-Level E-Beam Resist.** J Elec Mat 11(5):883–888, 1982.

Iannuzzi M., Kozakiewicz R. P., **Bias Humidity Performance and Failure Mechanisms of Nonhermetic Aluminum SICs in an Environment Contaminated with $SO_2$.** IEEE Compon 5(4):345–354, 1982.

Kaminow I. P., Eisenstein G., Stulz L. W., Dentai A. G., **Lateral Confinement InGaAsP Superluminescent Diode at 1.3 μm.** IEEE J Q El 19(1):78–82, 1983.

Kavehrad M., **Optimum QPSK Transmission Over Bandlimited Channels With Cascaded Nonlinearities.** Proc. Thirteenth Ann. Pittsburg Conf. 13, Pt 1:153–161, 1982.

Kavehrad M., **Phase-Noise Effects on QPSK Carriers in Burst Transmission.** IEE J Comm Radars Sig Proc 129 Pt F No 5:366–372, 1982.

Keller H. N., **Significant Features of Solder Connections to Gold-Plated Thin-Films.** IEEE Compon 5(4):408–419, 1982.

Kuo J. M., Thiel F. A., **Study of Deep Level in Bulk P-InP By Admittance Spectroscopy.** Electr Lett 19(2):41–43, 1983.

Macklin J. J., Wood O. R., Silfvast W. T., **New Recombination Lasers in Li, Al, Ca, and Cu in a Segmented Plasma-Device Employing Foil Electrodes.** IEEE J Q El 18(11):1832–1835, 1982.

Marcuse D., **Computer-Model of an Injection-Laser Amplifier.** IEEE J Q El 19(1):63–73, 1983.

McFee J. H., Swartz R. G., Archer V. D., Finegan S. N., Feldman L. C., **A Quantitative Study of the Relationship Between Interfacial Carbon and Line Dislocation Density in Silicon Molecular-Beam Epitaxy.** J Elchem So 130(1):214–216, 1983.

Pearsall T. P., Hendel R., Oconnor P., Alavi K., Cho A. Y., **Selectively-Doped $Al_{0.48}In_{0.52}As/Ga_{0.47}In_{0.53}As$ Heterostructure Field-Effect Transistor.** Elec Dev L 4(1):5–8, 1983.

Peterson G. E., Paek U. C., **Electrical Transmission-Lines as Models For Soliton Propagation in Infrared Fibers.** P Soc Photo 320:62–68, 1982.

Sanders J. V., **A Three-Dimensional Dynamic Analysis of a Towed System.** Ocean Eng 9(5):483–499, 1982.

Sargent M., et al., **Quantum-Theory of Laser and Optical-Bistability Instabilities.** Optics Lett 8(2):76–78, 1983.

Schwarz S. A., Thornber K. K., **Mobility Degradation due to the Gate Field in the Inversion Layer of MOSFETS—Comment.** Elec Dev L 4(1):11–12, 1983.

Sessler G. M., et al., **Non-Destructive Laser Method for Measuring Charge Profiles in Irradiated Polymer-Films.** IEEE Nucl S 29(6):1644–1649, 1982.

Wiesenfeld J. M., Chraplyvy A. R., Stone J., Burrus C. A., **Measurement of Very High-Speed Photodetectors with Picosecond InGaAsP Film Lasers.** Electr Lett 19(1):22–24, 1983.

## MANAGEMENT/ECONOMICS

Fishburn P. C., **Time Preference.** Int Econ R 23(3):677–694, 1982.

Hakansson N. H., Kunkel J. G., Ohlson J. A., **Sufficient and Necessary Conditions for Information to have Social Value in Pure Exchange.** J Finance 37(5):1169–1181, 1982.

Ordover J. A., Panzar J. C., **On the Non-Linear Pricing of Inputs.** Int Econ R 23(3):659–675, 1982.

Rassenti S. J., Smith V. L., Bulfin R. L., **A Combinatorial Auction Mechanism for Airport Time Slot Allocation.** Bell J Econ 13(2):402–417, 1982.

Vinod H. D., **Maximum-Entropy Measurement Error-Estimates of Singular Covariance Matrices in Undersized Samples.** J Economet 20(2):163–174, 1982.

## PHYSICAL SCIENCES

Allen S. J., Allyn C. L., Cox H. M., Derosa F., Mahoney G. E., **Dispersion of the Saturated Current in GaAs from DC to 1200 GHZ.** Appl Phys L 42(1):96–98, 1983.

Baumgart H., Leamy H. J., Celler G. K., Trimble L. E., **Grain-Boundary Diffusion in Polycrystalline Silicon Films on SiO2.** J Physique 43(NC1):363–368, 1982.

Becker R. S., Golovchenko, J. A., Patel J. R., **X-Ray Evanescent-Wave Absorption and Emission.** Phys Rev L 50(3):153–156, 1983.

Behringer R. P., et al., **Heat-Transport and Temporal Evolution of Fluid-Flow Near the Rayleigh-Benard Instability in Cylindrical Containers.** J Fluid Mec 125(DEC):219–258, 1982.

Bondybey V. E., Schwartz G. P., English J. H., **Laser Vaporization of Alloys—Laser-Induced Fluorescence of Heteronuclear Metal Clusters.** J Chem Phys 78(1):11–15, 1983.

Boolchand P., Grothaus J., Phillips J. C., **Broken Chemical Order and Phase-Separation in $Ge_xSe_{1-x}$ Glasses.** Sol St Comm 45(2):183–185, 1983.

Botineau J., Stolen R. H., **Effect of Polarization on Spectral Broadening in Optical Fibers.** J Opt Soc 72(12):1592–1596, 1982.

Brand H., Cross M. C., **Explanation of Flow Dissipation in $^3$He-B (Letter).** Phys Rev L 49(26):1959, 1982.

Bridges T. J., Bergman J. G., Chraplyvy, A. R., **Broad-Band Infrared Generation by Stimulated Raman-Scattering in Liquid Filled Fibers.** P Soc Photo 320:60–61, 1982.

Bruck M. A., et al., **Extended X-Ray Absorption Fine-Structure (EXAFS) Spectroscopic Analysis of 6-Mercaptopurine Riboside Complexes of Platinum(II) and Palladium(II) (Review or Bibliog.).** Acs Symp S 1983(209):245–262, 1983.

Capasso F., Tsang W. T., Bethea C. G., Hutchinson A. L., Levine B. F. **New Graded Band-Gap Picosecond Phototransistor.** Appl Phys L 42(1):93–95, 1983.

Capasso F., Tsang W. T., Williams G. F., **New Very Low-Noise Multilayer and Graded-Gap Avalanche Photodiodes for the 0.8 to 1.8 μm Wavelength Region.** P Soc Photo 340:50–55, 1982.

Celler G. K., Trimble L. E., **Microelectronics Applications of Deposited Si Films Recrystallized From the Melt.** J Physique 43(NC1):353–362, 1982.

Celler G. K., Robinson M., Lischner D. J., **Seeded Recrystallization of Thick Polysilicon Films on Oxidized 3-in Wafers.** Appl Phys L 42(1):99–101, 1983.

Cheng J., Lemons R. A., **Topography Induced Preferential Crystallization of Thermally Grown Silicon Dioxide Films.** J Elec Mat 12(1):61–69, 1983.

Cowens J. W., Stevie F. A., Alderfer J. L., Hansen G. E., Pendyala L., Creaven P. J., **Synthesis and Identification of Derivatives of a Platinum Containing Complex.** Int J Mass 48(FEB):177–180, 1983.

Chu S. N. G., **Elastic Interaction Between a Screw Dislocation and Surface Crack.** J Appl Phys 53(12):8678–8685, 1982.

Craighead H. G., Howard R. E., Jackel L. D., Mankiewich P. M., **10-nm Linewidth Electron-Beam Lithography on GaAs.** Appl Phys L 42(1):38–40, 1983.

Cross M. C., **Phase Dynamics of Convective Rolls.** Phys Rev A 27(1):490–498, 1983.

Dahlberg S. C., **Photo-Voltage Studies of Organic Semiconductor Interfaces.** Appl Surf S 14(1):47–55, 1982.

Ebeling K. J., Coldren L. A., Miller B. I., Rentschler J. A., **Single-Mode Operation of Coupled-Cavity GaInAsP-InP Semiconductor-Lasers.** Appl Phys L 42(1):6–8, 1983.

English A. T., Kinsbron E., **Electromigration-Induced Failure by Edge Displacement in Fine-Line Aluminum-0.5-Percent Copper Thin-Film Conductors.** J Appl Phys 54(1):268–274, 1983.

English A. T., Kinsbron E., **Electromigration Transport Mobility Associated with Pulsed Direct-Current in Fine-Grained Evaporated Al-0.5-Percent-Cu Thin-Films.** J Appl Phys 54(1):275–280, 1983.

Flamm D. L., Wang D. N. K., Maydan D., **Multiple-Etchant Loading Effect and Silicon Etching in $ClF_3$ and Related Mixtures.** J Elchem So 129(12):2755–2760, 1982.

Franey J. P., **A Novel System for Atmospheric Corrosion Experiments.** Corros Sci 23(1):1–8, 1983.

Freeman R. R., Jopson R. M., Bokor J., Bucksbaum, P., **Laser Generation of Light in the Extreme Ultraviolet and Soft-X-Ray Regime.** AIP Conf Pr1982(94): 491–494, 1982.

Gibbs H. M. et al., **Room Temperature Optical Bistability and Self-Defocusing in Semiconductor Etalons.** Appl Phys B 29(3):171–172, 1982.

Guélin M., Cernicharo J., Linke R. A., **Detection of $HC^{17}O^+$ in Sagittarius 82.** Astrophys J 263(2):L89–L93, 1982.

Hebard A. F., Nakahara S., **Structural Phase-Transitions of Indium Indium Oxide Thin-Film Composites.** Appl Phys L 41(12):1130–1132, 1982.

Heffner W. R., Berreman D. W., **Switching Characteristics of a Bistable Cholesteric Twist Cell.** J Appl Phys 53(12):8599–8606, 1982.

Heller A., Aharon-Shalom E., Bonner W. A., Miller B., **Hydrogen-Evolving Semiconductor Photo-Cathodes—Nature of the Junction and Function of the Platinum Group Metal Catalyst.** J. Am Chem S 104(25):6942–6948, 1982.

Hellman M. Y., Bowmer T., Taylor G. N., **Determination of Gel Content and Percent Gel in Radiation-Cured Polyvinyl Chloride—Cross-linking Monomer Coatings by Combined GPC-LC Techniques.** Macromolec 16(1):34–38, 1983.

Hutton R. S., Roth H. D., **Spin-Density Distribution and Geometric Isomerism in Triplet Carbenes.** J Am Chem S 104(26):7395–7399, 1982.

Hwang J. C. M., Temkin H., Brennan T. M., Frahm R. E., **Growth of High-Purity GaAs-Layers by Molecular-Beam Epitaxy.** Appl Phys L 42(1):66–68, 1983.

Jayaraman A., **Diamond Anvil Cell and High-Pressure Physical Investigations (Review or Bibliog.)** Rev M Phys 55(1):65–108, 1983.

Jelinski L. W., Dumais J. J., Watnick P. I., Bass S. V., Shepherd L., **C-13-NMR of ABS in the Solid-State—Characterization and Relaxation Studies.** J Pol Sc PC 20(12):3285–3296, 1982.

Johari G. P., **Glass-Transition and Molecular Motions in the Glassy State of a Nematic Liquid.** Phil Mag B 46(6):549–564, 1982.

Kazarinov R. F., Nordland W. A., Wagner W. R., Temkin H., Manchon D. D., **Near-Equilibrium LPE Growth of $In_{1-x}Ga_xAS_yP_{1-y}$ Lattice Matched to InP.** J Cryst Gr 60(2):235–238, 1982.

Kelso S. M., Aspnes D. E., Pollack M. A., Nahory R. E., **Optical-Properties of $In_{1-X}Ga_xAs_YP_{1-Y}$ From 1.5 to 6.0 eV Determined by Spectroscopic Ellipsometry.** Phys Rev B 26(12):6669–6681, 1982.

Kolodner P., Tyson J. A., **Remote Thermal Imaging with 0.7-μm Spatial-Resolution Using Temperature-Dependent Fluorescent Thin-Films.** Appl Phys L 42(1):117–119, 1983.

Kramer L., Benjacob E., Brand H., Cross M. C., **Wavelength Selection in Systems Far From Equilibrium.** Phys Rev L 49(26):1891–1894, 1982.

Kuo C. Y., Vieira M. M. F., Kerl R. J., Patel C. K. N., **Observation of Delta-V-4 Vibrational Overtone Transitions in Solid HCP Para-Hydrogen.** Phys Rev L 50(4):256–259, 1983.

Lake G., **The Rotation of Elliptical Galaxies—An Application of the Theory of Tidal Torques.** Astrophys J 264(2):408–412, 1983.

Lamola A. A., **Fluorescence Studies of Protoporphyrin Transport and Clearance.** Act der-Ven1982(S100):57–66, 1982.

Lee T. P., **Recent Development in Light-Emitting-Diodes (LEDs) for Optical Fiber Communications-Systems.** P Soc Photo 340:22–31, 1982.

Leung S. Y., Schumaker N. E., **Simulation of Slider-Induced Convection in Horizontal LPE Slider System.** J Cryst Gr 60(2):421–433, 1982.

Lin C., Lee T. P., Burrus C. A., **Picosecond Frequency Chirping and Dynamic Line Broadening in InGaAsP Injection-Lasers Under Fast Excitation.** Appl Phys L 42(2):141–143, 1983.

Lovinger A. J., **Microstructure and Unit-Cell Orientation in Alpha-Polypropylene.** J Pol Sc PP 21(1):97–110, 1983.

Mayer T. M., Fisanick G. J., Eichelberger, T. S., **Deposition of Chromium Films by Multi-Photon Dissociation of Chromium Hexacarbonyl.** J Appl Phys 53(12): 8462–8469, 1982.

Meiboom S., Sammon M., Brinkman W. F., **Lattice of Disclinations—The Structure of the Blue Phases of Cholesteric Liquid-Crystals.** Phys Rev A 27(1):438–454, 1983.

Mills A. P., **Line-Shape Effects in the Measurement of the Positronium Hyperfine Interval.** Phys Rev A 27(1):262–267, 1983.

Myers P. C., Linke R. A., Benson P. J., **Dense Cores in Dark Clouds. 1. Co Observations and Column Densities of High-Extinction Regions.** Astrophys J 264(2):517–537, 1983.

Nassau K., **Material Dispersion Considerations For Infrared Fibers.** P Soc Photo 320:43–48, 1982.

Obryan H. M., Yan M. F., **Second-Phase Development in Ba-Duped Rutile.** J Am Ceram 65(12):615–619, 1982.

Odagaki T., Lax M., **Some Exact Properties of Hopping Conduction.** Phys Rev B 26(12):6480–6489, 1982.

Oron M., Tamari N., **High-Power Single-Mode InGaAsP Lasers Fabricated by Single Step Liquid-Phase Epitaxy.** Appl Phys L 42(2):139–141, 1983.

Patterson G. D., **Photon-Correlation Spectroscopy of Bulk Polymers (Review or Bibliog.).** Adv Polym S 48:125–159, 1983.

Phillips J. C., **Reversible and Sensitizable Photostructural Transformations in Chalcogenide Alloy Glasses.** J Non-Cryst 51(3):301–304, 1982.

Phillips J. C., **Spectroscopic and Morphological Structure of Tetrahedral Oxide Glasses** (Review or Bibliog.). Sol St Phys 37:93–171, 1982.

Pleiner H., Brand H., **Light-Scattering in Nematic Liquid-Crystals in the Presence of Shear-Flow.** J Phys Lett 44(1):L23–L31, 1983.

Rana V. V. S., Nelson T. J., Lecraw R. C., Blank S. L., **Preparation and Properties of V-Substituted Garnet-Films for Ion-Implanted 1.0-Micron Bubble-Devices with Improved High-Temperature Propagation.** J Appl Phys 53(12):9093–9097, 1982.

Reagor B. T., Kelley D. F., Huchital D. H. Rentzepis P. M., **Kinetics of Intramolecular Electron-Transfer in Binuclear $(NC)_5Fe^{II}$-CN-$Co^{III}$(chelate) Complexes by Pico-second Absorption-Spectroscopy.** J Am Chem S 104(26):7400–7403, 1982.

Reynolds C. L., Jr. and Tamargo M. C., **Liquid Phase Epitaxial Growth of (Al, Ga)As Double-Heterostructure Laser Material in a Sapphire Boat.** J Appl Phys 53(12):9217–9219, 1982.

Roth H. D., Schilling M. L., Hutton R. S., Truesdale E. A., **Photoinduced Electron-Transfer Reactions—The Radical Cations of Methano-Bridged Paracyclo-phanes.** J Am Chem S 105(2):153–157, 1983.

Raine G. P., Schaefer H. F., Haddon R. C., **The Dimers of Carbon-Monoxide and Carbon Monosulfide—Chemically Bound Triplet Electronic Ground-States.** J Am Chem S 105(2):194–198, 1983.

Schmidt P. F., **Contamination-Free High-Temperature Treatment of Silicon or Other Materials.** J Elchem So 130(1):196–199, 1983.

Schubert R., Tompkins H. G., **Effects of Exposure to Vapors of Saturated Salt Aqueous-Solutions.** IEEE Compon 5(4):425–430, 1982.

Shah J., Pinczuk A., Stormer H. L., Gossard A. C., Wiegmann W., **Electric-Field Induced Heating of High Mobility Electrons in Modulation-Doped GaAs-Al-GaAs Heterostructures.** Appl Phys L 42(1):55–57, 1983.

Shank C. V., Yen R., Hirlimann C., **Time-Resolved Reflectivity Measurements of Femtosecond-Optical-Pulse Induced Phase-Transitions in Silicon.** Phys Rev L 50(6):454–457, 1983.

Snyder L. C. et al., **Computations to Model 3-Center Bonds in Hydrogenated Amorphous-Silicon.** Phys Rev B 26(12):6727–6733, 1982.

Stavola M., Patel J. R., Kimerling, L. C., Freeland P. G., **Diffusivity of Oxygen in Silicon at the Donor Formation Temperature.** Appl Phys L 42(1):73–75, 1983.

Stocker H. J., Aspnes D. E., **Surface Chemical-Reactions on $In_{0.53}Ga_{0.47}As$.** Appl Phys L 42(1):85–87, 1983.

Stone J., Whalen M. S., **Index of Refraction Dispersion of Normal-Type and Para-Type InP Between 0.95 and 2.0 eV.** Appl Phys L 41(12):1140–1142, 1982.

Temkin H., Hwang J. C. M., **Undoped, Semi-Insulating GaAs-Layers Grown by Molecular-Beam Epitaxy.** Appl Phys L 42(2):178–180, 1983.

Tumpkins H. G., Sharma S. P., **The Interaction of Imidazole, Senzimidazole and Related Azoles with a Copper Surface.** Surf Int An 4(6):261–266, 1982.

Tsang W. T., Dixon M., Dean B. A., **The Characterization and Functional Reliability of 45 Mbit/s Optical Transmitters Containing MBE-Grown Lasers.** IEEE J Q El 19(1):59–62, 1983.

Tsui D. C., Gossard A. C., Dolan G. J., **Radiation Effects on Modulation-Doped GaAs-$Al_XGa_{1-X}$ As Heterostructures.** Appl Phys L 42(2):180–182, 1983.

Tung R. T., Gibson J. M., Poate J. M., **Formation of Ultrathin Single-Crystal Silicide Films on Si—Surface and Interfacial Stabilization of Si-NiSi2 Epitaxial Structures.** Phys Rev L 50(6):429–432, 1983.

Vandenberg J. M., Hamm R. A., **An Insitu X-Ray Study of Phase Formation in Cu-Al Thin-Film Couples.** Thin Sol Fi 97(4):313–323, 1982.

Varma G. M., Dynes R. C., Banavar J. R., **Thermally Created Tunnelling States in Glasses (Letter).** J Phys C 15(35):1221–1225, 1982.

Walker L. R., **Interpretation of the $Fe_{57}$ Isomer-Shift** (Citation Classics). CC/Phy Chem1983(7):20, 1983.

Wertheim G. K., Phillips J. C., Bridenbaugh P. M., **A Limit on the Site Inequivalence in G-$As_2Se_3$.** Sol St Comm 45(2):179–181, 1983.

White J. C., Henderson D., **Tuning and Saturation Behavior of the Anti-Stokes Raman Laser.** Optics Lett 8(1):15–17, 1983.

Wilson T., Sheppard C. J., **Imaging with Finite Values of Fresnel Number.** J Opt Soc 72(12):1639–1641, 1982.

Wolfe A., **ULF Geomagnetic Power Near L = 4. 7. A Conjugate Area Study of Power Controlled by Solar-Wind Quantities.** J Geo R-S P 87(NA12):425–431, 1982.

Wrobel J., Takahashi K., Lannoy G., Cook J. M., Bertz S. H., Honkan V., **Stereocontrolled Synthesis of (+/−)-Modhephene via the Weiss Reaction (Letter).** J Org Chem 48(1):139–141, 1983.

Yan M. F., Rhodes W. W., **Effects of Cation Contaminants in Conductive $TiO_2$ Ceramics.** J Appl Phys 53(12):8809–8818, 1982.

Zussman A., Eger D., Oron M., Szapiro S., Shachna A., Hans B., **Lung-Wavelength PbSnTe-PbTeSe Lattice-Matched Single-Heterostructure Lasers Grown by LPE.** IEE Proc-I 129(6):203–208, 1982.

## SOCIAL AND LIFE SCIENCES

Clark W. C., Carroll J. D., Yang J. C., Pruzansky S., **Perceptual Dimensions of Thermal Pain—A Multidimensional-Scaling Approach (Review or Bibliog.).** Adv Pain R 5:871–876, 1982.

Desarbo W. S., Hausman R. E., Lin S., Thompson W., **Constrained Canonical Correlation.** Psychometri 47(4):489–516, 1982.

Fishburn P. C., **Foundations of Risk Measurement. 2. Effects of Gains on Risk.** J Math Psyc 25(3):226–242, 1982.

Goddard C. T., **Debunking the Learning-Curve.** IEEE Compon 5(4):328–335, 1982.

Julesz B., Kropfl W., **Binocular Neurons and Cyclopean Visually Evoked-Potentials in Monkey and Man.** Ann Ny Acad 388(Jun):37–44, 1982.

Rosenbaum D. A., Kornblum S., **A Priming Method for Investigation the Selection of Motor-Responses.** Act Psychol 51(3):223–243, 1982.

Rosenthal N. E., Lewy A. J., Wehr T. A., Kern H. E., Goodwin F. K., **Seasonal Cycling in a Bipolar Patient.** Psychiat R 6(1):25–31, 1983.

Starr S. J., Thompson C. R., Shute S. J., **Effects of Video Display Terminals on Telephone Operators.** Human Fact 24(6):699–711, 1982.

## SPEECH/ACOUSTICS

Hurd H. L., **Clipping Degradation in a System Used to Detect Time-Periodic Variance Fluctuations.** J Acoust So 72(6):1827–1830, 1982.

Karim M. R., **An Investigation of Delta Modulation of Speech for Mobile Radio.** IEEE Veh T 31(4):158–165, 1982.

Umeda N., Kahn D., **Frequency of Occurrence of Two-Word and Three-Word Sequences in English (Letter).** J Acoust So 72(6):2031–2033, 1982.

THE BELL SYSTEM
TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 July–August 1983 Number 6, Part 1

Copyright © 1983 American Telephone and Telegraph Company. Printed in U.S.A.

# On the Start-Up Problem in Digital Echo Cancelers

By J. SALZ*

(Manuscript received October 5, 1982)

Digital echo cancellation techniques make it possible to realize efficient full-duplex data transmission over a single loop. The purpose of this paper is to elucidate the solution to the start-up problem in these devices and to present a new, fast, and simple tap-adjustment procedure. The theory indicates that a modified stochastic gradient tap-adjustment algorithm, using pseudorandom input data sequences for the initial training period, converges in $N$ steps, where $N$ is the total number of canceler taps, and that this is the fastest possible convergence time.

## I. INTRODUCTION

Two-way voice communication over a single loop is made possible by the use of a hybrid bridge. However, the suppression of echoes by fixed hybrids is insufficient to support full-duplex data transmission, and therefore makes adaptive data echo cancelers necessary.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

In two-way data communications, transmitted data "echoes" back to the near-end receiver after being reflected and dispersed through an unknown return path. So, if one assumes that the echo path is linear, the estimation of its overall impulse response is sufficient to allow the synthesis of one's own echo signal. This synthesized version is subtracted from the received signal, which then makes it possible for the receiver to extract the data intended for it.

The impulse response of the echo channel (local transmitter output to local receiver input) can be measured in several ways. An obvious way to do this is to transmit a single impulse and measure the echo. However, among other defects of this procedure, the average power would be very low and the resulting signal-to-noise ratio (s/n) would be inadequate. If a pseudorandom sequence (+1 or −1) is transmitted instead, the average power would be much greater and would be essentially constant on the line, more nearly representing a true data signal. The latter is the preferable approach.

Digital data echo cancelers operate in two modes. In the acquisition mode, or start-up, the impulse response of the echo path is measured. This is best accomplished, as we shall see, with the use of fixed data sequences. Since, during this period, no information is conveyed to the far-end, the time allotted for this purpose should be as short as possible. Although it is conceptually possible to start an echo canceler either blind or with random data, the convergence of the taps, or the reliable measurements of the impulse response, are known to require a long time. In the subsequent mode, or during actual data transmission, a tracking algorithm is initiated whose function is to update the measurements when slight changes occur in the impulse response. We focus on the more critical start-up algorithm.

From an operational point of view, it is desirable to implement these algorithms in a recursive fashion, or in a closed-loop manner. This means that the canceler tap coefficients, which represent the sampled impulse response, are updated in response to a measured error between the actual impulse response and the one estimated at any particular instant. Conventional gradient adjustment algorithms, even with fixed data sequences, are known to converge very slowly.

This research was motivated by the need for a theory capable of explaining the behavior of tap-adjustment algorithms. During the course of this investigation a modified stochastic gradient algorithm that is simple to implement and converges in the theoretically smallest number of steps was discovered.

## II. PROBLEM FORMULATION

Figure 1 shows a full-duplex data modem employing a digital echo canceler. We are concerned with the sampled signal values

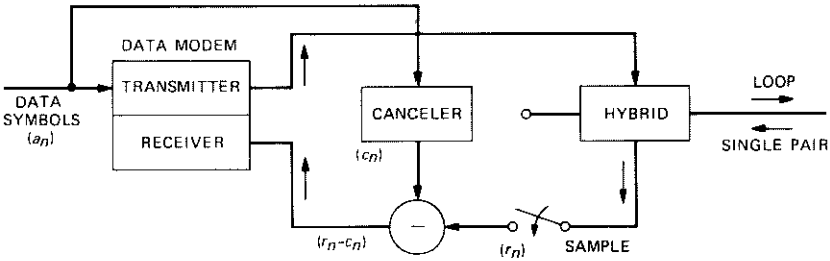



Fig. 1—Digital data canceler.

$$r_n = \sum_{k=-\infty}^{\infty} h_k a_{n-k} + \nu_n, \tag{1}$$

where the $a_n$'s are the transmitted data symbols that echo back to the receiver, the $h_n$'s are the overall impulse response values of the echo path, and the $\nu_n$'s are the desired received signal values plus noise. The object of the canceler is to synthesize signal values, $c_n$, which are estimates of $\sum_k h_k a_{n-k}$ and subtract them from the $r_n$'s. The receiver then proceeds to process the difference signal values, $r_n - c_n$, to extract the data intended for it.

The fundamental problem is to devise procedures for estimating the $h_n$'s from observations of the $r_n$'s, while treating the $\nu_n$'s as undesirable noise. Practically, it must be assumed that only a finite number, $N$, of the $h_n$'s can be estimated, and so we express (1) compactly as

$$r_n = H^t A_n + \nu_n, \tag{2}$$

where $H$ and $A_n$ are finite dimensional column vectors,*

$$H = \begin{pmatrix} h_N \\ \vdots \\ h_1 \end{pmatrix}, \quad \text{and} \quad A_n = \begin{pmatrix} a_{n-N} \\ \vdots \\ a_{n-1} \end{pmatrix},$$

and where $(\,)^t$ indicates the transpose of a matrix.

In the absence of precise statistical knowledge of the $\nu_n$'s and $H$, a natural procedure for choosing an estimator of $H$ is to minimize the sum of squared errors from time 1 to time $l$

$$\epsilon_l = \sum_{n=1}^{l} (r_n - \hat{H}^t A_n)^2.$$

This is a standard problem and the solution is immediate. It involves the solution of a set of linear equations

$$Z_l \hat{H}_l = U_l \tag{3}$$

* We deal with a baseband model for notational convenience. By using complex numbers throughout, the treatment generalizes to passband models.

for the best estimator at time $l$, $\hat{H}_l$.

In (3)
$$U_l = \sum_{n=1}^{l} A_n r_n$$

and
$$Z_l = \sum_{n=1}^{l} A_n A_n^t.$$

In applications it is usually desirable to solve these equations recursively with a minimum computational effort and to assure rapid convergence to the actual $H$. Our attention in the next sections is directed toward these aims. However, before dealing with our main subject we wish first to examine some asymptotic behaviors of the standard solution.

### *2.1 Random input data*

In some applications, echo cancelers must start blind, i.e., from random data. This is the case in echo cancelers used to suppress speech. However, in full-duplex data communications, a preamble word, or words, can be sent first to assure rapid convergence. To emphasize these differences we examine the behavior of the estimator with random data first, where one has no choice in the selection of the starting sequences. So, consider random data such that the $a_n$'s assume $\pm 1$ independently with equal probability and examine the limit as $l \to \infty$. It is found that

$$\begin{aligned} U_l &\to lE\{A_n r_n\} \\ &= lE\{A_n A_n^t\} H, \\ Z_l &\to lE\{A_n A_n^t\} = lI, \end{aligned}$$

and, consequently,

$$Z_l^{-1} U_l \to H. \tag{4}$$

In the above we made use of the fact that the $\nu_n$'s and $A_n$'s are naturally independent. We assumed that these sequences are ergodic and so replaced time averages with mathematical expectations, $E\{\cdot\}$. This then demonstrates that if one is willing to wait forever, it is conceptually possible to determine $H$ exactly—not a terribly startling result.

While the asymptotic behavior is easy to deduce, the statistical behavior of the estimator for finite $l$ is difficult to glean. One immediately encounters an unsolved mathematical problem that involves the conditions on the random sequences that would guarantee the existence of the inverse matrix $Z_l^{-1}$. Clearly, $l$ has to be greater than $N$ for the inverse to even have a chance to exist, and for those

Likewise, the vector

$$U_l = \sum_{n=1}^{l} A_n r_n = U_{l-1} + A_l r_l. \tag{13}$$

Now, applying the matrix inversion lemma[2] to (12), and assuming that the inverses exist, we find that

$$Z_l^{-1} = Z_{l-1}^{-1} - \frac{Z_{l-1}^{-1} A_l A_l^t Z_{l-1}^{-1}}{1 + A_l^t Z_{l-1}^{-1} A_l}. \tag{14}$$

This is the key equation and, in conjunction with (13), makes it possible to claim (11). We note that the algorithm expressed in (13) is computationally complicated since it requires the calculation of a matrix recursion, (14), and multiplication of matrices by vectors at each iteration. For large $N$ this becomes infeasible and simpler procedures are therefore sought. However, before proposing a simpler algorithm we need to review some properties of pseudorandom sequences.

The pseudorandom data sequences that are the inputs to the canceler during the start-up period derive from the binary sequences

$$X = x_1 x_2 x_3 x_4 \cdots$$

($x_i = 0$, or 1). The $n$th digit is computed from certain of the earlier digits by means of the recurrence relation

$$x_n = x_{n-c} + x_{n-b} \bmod 2,$$

where $c$ and $b$ are integers, $0 < c < b$. The actual data sequences $\{a_n\}$ that are applied to the canceler are the $x_n$'s with "0" replaced by "−1".

Returning now to the sequence $X$, we remark that in spite of the fact that $x_n$ is completely determined by the digits that precede it, the sequence $X$ resembles in some respects a completely random sequence. The calculation of the sequence $X$ is carried out in a shift register working in a closed loop and a mod 2 adder. It turned out that for special choices of $c$ and $b$ the sequence $X$ is periodic with period $2^b - 1$.[2]

In our application, we will make use of the following known properties of the sequences

$$1)\ A_n^t A_m = \sum_{i=1}^{N} a_{n-i} a_{m-i} = \begin{cases} N, & n = m \\ -1, & n \neq m \end{cases}$$

$$2)\ Q^t A_n = 1, \quad \text{for} \quad n = 1, \cdots, N$$

$$Q = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}_1^N,$$

the all "1" vector

$$3)\ A_{n+N} = A_n, \text{ periodicity.}$$

Data sequences possessing properties 1 through 3 are also referred to as pseudorandom sequences. As a consequence of these properties, it is easy to verify the following:
Property (a):

$$Z_N = \sum_{n=1}^{N} A_n A_n^t$$

$$= (N+1)I - QQ^t, \ N \text{ x } N \text{ matrix,}$$

where $I$ is the identity matrix. This decomposition is possible since the $ij$th element of the matrix $Z_N$, $(Z_N)_{ij} = N$ for $i = j$ and $-1$ on the off-diagonals. This can readily be seen from property 1.
Property (b): The inverse matrix

$$Z_N^{-1} = \frac{1}{N+1}(I + QQ^t).$$

This can be derived from the matrix inversion lemma or verified by actually computing $Z_N Z_N^{-1} = I$.
Property (c): The set of vectors

$$B_n = Z_N^{-1} A_n, \ n = 1 \cdots N$$

are orthonormal to the vectors $A_n$, $n = 1, \cdots N$. This is a crucial property to what follows, so we prove that

$$A_m^t Z_N^{-1} A_n = \frac{A_m^t}{N+1}(I + QQ^t)A_n$$

$$= \frac{A_m^t A_n + (A_m^t Q)(Q^t A_n)}{N+1} = \begin{cases} 0, & n \neq m \\ 1, & n = m. \end{cases}$$

These properties suggest an approach to a simple and rapidly converging tap-adjustment algorithm.

The key to the simplification of the algorithm, (11), is the recognition that $Z_l^{-1}$ for $l < N$ does not exist, and so it is not possible to start the algorithm at $l = 1$. The basic idea is to replace $Z_l$ by $Z_N$ and thus obtain the simpler algorithm

$$\hat{H}_{n+1} = \hat{H}_n - Z_N^{-1} A_n e_n, \tag{15}$$

where the error at time $n$ is again

$$e_n = A_n^t \hat{H}_n - r_n$$

$$= A_n^t(\hat{H}_n - H) - v_n.$$

Note that this is a measurable quantity at each iteration and $A_n e_n$ is just the gradient of the instantaneous squared error.

The algorithm expressed in (15) is remarkably simple since

$$Z_N^{-1} A_n = \frac{1}{N+1} (I + QQ^t) A_n$$

$$= \frac{1}{N+1} (A_n + Q),$$

which follows from the definition of $Z_N$ and property (b). Inserting this into (15) we obtain

$$\hat{H}_{n+1} = \hat{H}_n - \frac{1}{N+1} (A_n + Q) e_n. \tag{16}$$

This form is immediately recognized as a slightly modified stochastic gradient algorithm with step size equal to $1/N + 1$ and the gradient vector, $A_n$, replaced by $A_n + Q$. It is nothing more than the original vector, $A_n$, in which $a_n$'s equal to $-1$ are replaced by zero and $a_n = 1$ is replaced by $a_n = 2$. We wish to acknowledge that during the course of the development of this theory C. W. Farrow anticipated the form of this algorithm.

It now remains to demonstrate that (16) indeed converges "fast", by which we mean that it converges in $N$ steps. Toward this end, define the error vector

$$\epsilon_n = \hat{H}_n - H,$$

and rewrite (16) in the form,

$$\begin{aligned} \epsilon_{n+1} &= \epsilon_n - Z_N^{-1} A_n (A_n^t \epsilon_n - \nu_n) \\ &= (I - B_n A_n^t) \epsilon_n + B_n \nu_n. \end{aligned} \tag{17}$$

Iterating (17) yields explicitly

$$\epsilon_{n+1} = \prod_{k=1}^{n} (I - B_k A_k^t) \epsilon_1 + \sum_{k=1}^{n} \prod_{j=k}^{n-1} (I - B_{j+1} A_{j+1}^t) B_k \nu_k. \tag{18}$$

This is the general solution but because of property (c), which states that $A_n^t B_{n-1} = 0$, we get a much simpler solution, which is the chief reason for the rapid convergence, namely,

$$\epsilon_{n+1} = \left(I - \sum_{k=1}^{n} B_k A_k^t\right) \epsilon_1 + \sum_{k=1}^{n} B_k \nu_k$$

$$= \left[I - Z_N^{-1}\left(\sum_{k=1}^{n} A_k A_k^t\right)\right] \epsilon_1 + Z_N^{-1}\left(\sum_{k=1}^{n} A_k \nu_k\right). \tag{19}$$

This simple form results because the product of the matrices in (18) reduces to

$$\prod_{k=1}^{n} (I - B_k A_k^t)\epsilon_1$$

$$= (I - B_1 A_1^t)(I - B_2 A_2^t) \cdots (I - B_n A_n^t)\epsilon_1$$

$$= [(I - B_1 A_1^t)(I - B_2 A_2^t) \cdots -B_n A_n^t]\epsilon_1$$

$$= \left[I - \sum_{k=1}^{n} B_k A_k^t\right] \epsilon_1.$$

The evolution of the error vector $\epsilon_n$ is guided by two components, the transient

$$\tau_n = \left[I - Z_N^{-1}\left(\sum_{k=1}^{n} A_k A_k^t\right)\right] \epsilon_1, \tag{20}$$

and the steady-state component

$$S_n = Z_N^{-1}\left(\sum_{k=1}^{n} A_k \nu_k\right). \tag{21}$$

A most crucial property of the transient solution is that at $n = N$, $\tau_n$ vanishes, and this is the reason for claiming "fast" convergence. Clearly, the transient solution cannot vanish before this time since the inverse doesn't even exist, and therefore we claim the algorithm convergence in the least possible number of steps. This most important property of the algorithm can be seen from (20), since

$$\tau_N = (I - Z_N^{-1} Z_N)\epsilon_1 = 0.$$

Consequently, the error vector at time $N + 1$ consist only of measurement noise, or the steady-state component

$$\epsilon_{N+1} = Z_N^{-1} \left( \sum_{k=1}^{N} A_k \nu_k \right). \tag{22}$$

We have thus demonstrated that the algorithm converges to the true solution in $N$ steps since the transient vanishes at the end of the $N$th iteration and, from that time on, the taps fluctuate around the true value due to measurement noise, $\nu_n$, alone. The variance of the tap fluctuations can be calculated from the variance matrix

$$\begin{aligned} \rho_N &= E\{\epsilon_{N+1}\epsilon_{N+1}^t\} \\ &= Z_N^{-1} \left( \sum_{k,k'=1}^{N} A_k A_{k'}^t E\{\nu_k \nu_{k'}\} \right) Z_N^{-1} \\ &= \sigma^2 Z_N^{-1} = \frac{\sigma^2}{N+1} (I + QQ^t), \end{aligned} \tag{23}$$

where again we assumed that the $\nu_k$'s are identically and independently distributed. The error variance, $\sigma_H^2$, is therefore,

$$\begin{aligned} \sigma_H^2 &= \sigma^2 \text{ Trace } Z_N^{-1} \\ &= \frac{\sigma^2}{N+1} \text{Trace}[I + QQ^t] \\ &= \sigma^2 \frac{2N}{N+1}. \end{aligned} \tag{24}$$

This is precisely the value we obtained from solving the set of linear equations, (3), with pseudorandom input sequences. Thus, iterating $N$ times provides the solution in a simple fashion.

It may turn out in applications that the value of variance obtained in $N$ iterations is not sufficiently small. As seems reasonable, the noise variance can be reduced to any desired value by repeating the pseudorandom sequence of length $N$. To see this, consider a slightly modified tap-adjustment algorithm

$$H_{n+1} = H_n - \alpha Z_N^{-1} A_n e_n, \tag{25}$$

where now the scaler, $\alpha$, is a fixed step size yet to be determined. Proceeding as before, the recursion for the tap error $\epsilon_n = H_n - H$ now becomes

$$\epsilon_{n+1} = (I - \alpha B_n A_n)\epsilon_n + \alpha B_n \nu_n, \tag{26}$$

with the concomitant solution

$$\epsilon_{n+1} = \left[I - \alpha Z_N^{-1}\left(\sum_{k=1}^{n} A_k A_k^t\right)\right]\epsilon_1 + \alpha Z_N^{-1}\left(\sum_{k=1}^{n} A_k \nu_k\right). \tag{27}$$

Again, examine the solution at $n = pN + 1$, where $p$ is a positive integer, to obtain

$$\epsilon_{pN+1} = \left[I - \alpha Z_N^{-1}\left(\sum_{k=1}^{pN} A_k A_k^t\right)\right]\epsilon_1 + \alpha Z_N^{-1}\left(\sum_{k=1}^{pN} A_k \nu_k\right). \tag{28}$$

Since $A_n$ is periodic with period $N$ (property 3), we conclude that

$$\epsilon_{pN+1} = (1 - \alpha p)\epsilon_1 + \alpha Z_N^{-1}\left(\sum_{k=1}^{pN} A_k \nu_k\right), \tag{29}$$

and so we see that the transient component can be made to vanish when $\alpha = 1/p$. A straightforward calculation indicates that with this choice of $\alpha$ the variance is

$$\sigma_H^2 = \frac{2N}{p(N+1)}\,\sigma^2, \tag{30}$$

indicating a reduction by a factor of $p$—the number of times the pseudorandom sequence is repeated.

## REFERENCES


1. M. S. Mueller, "Least-Squares Algorithms for Adaptive Equalizers," B.S.T.J., *60*, No. 8 (October 1981), pp. 1905–25.
2. B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Information and System Science Series, Thomas Kailath, ed., Englewood Cliffs, NJ: Prentice-Hall, 1979.
3. J. J. Stiffler, *Theory of Synchronous Communications*, Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 178–84.


## AUTHOR


**Jack Salz,** B.S.E.E., 1955, M.S.E., 1956, and Ph.D., 1961, University of Florida; Bell Laboratories, 1961—. Mr. Salz first worked on the electronic switching system. Since 1968 he has supervised a group engaged in theoretical studies in data communications and is currently a member of the Communications Methods Research Department. During the academic year 1967–68, he was on leave as Professor of Electrical Engineering at the University of Florida. He was a visiting lecturer at Stanford University in Spring 1981 and a visiting MacKay Lecturer at the University of California, at Berkeley, in Spring 1983.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

# Sample Reduction and Subsequent Adaptive Interpolation of Speech Signals

By R. STEELE* and F. BENJAMIN†

(Manuscript received December 15, 1982)

In this paper we investigate the effect of rejecting every $n$th speech sample and replacing it by means of adaptive interpolation. The interpolation procedure attempts to minimize the mean square interpolation error by recomputing the autocorrelation function of the speech sequence every $W$ samples. We describe three methods of computing the correlation function. An iterative procedure is evaluated for estimating the correlation function of a speech sequence whose every $n$th sample has been discarded. For speech bandlimited to 3.2 kHz, sampled at 8 kHz, and $n = 4$, $W = 256$, the gain in signal-to-noise ratio (s/n) achieved by adaptive interpolation compared to nearest neighbor average interpolation was 14 and 8 dB, depending on whether the correlation function was computed from the original speech, or by using the iterative procedure, respectively. The effect of varying $n$ from 2 to 6 was also investigated. Finally, we applied the interpolation procedures to 8-bit $\mu$-law pulse code modulation (PCM), $\mu = 255$, reducing 64 kb/s transmission to 48 kb/s by rejecting one PCM word in four. The recovered speech after interpolation had a s/n that approximated that of conventional 56 kb/s $\mu$-law PCM speech.

## I. INTRODUCTION

Sampling of speech signals is usually performed at a rate high enough to prevent objectionable aliasing. Thus a speech signal whose bandwidth extends from 0.3 to 3.3 kHz is typically sampled at 8 kHz. After the speech signal has been sampled, the subsequent processing

* University of Southampton, Southampton, England. † Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

of the samples depends on the application. In digital encoding the samples are converted into a digital (usually binary) sequence. Clearly, the lower the sampling rate the lower the bit rate of the digitally encoded speech, and the smaller the required channel capacity. There are numerous techniques for bit-rate reduction that have been compiled and categorized.[1,2] Our intention here is to dwell on a specific subset of bit-rate reduction, namely that arising from discarding speech samples at the transmitter or at a node in a network, and replacing them at the receiver by interpolation. The impetus for this approach arose from two unrelated interests, analog speech scramblers and bit-rate reduction in $\mu$-law pulse code modulation (PCM). Scramblers that offer a reasonable amount of security, such as modulo masking scramblers, expand the bandwidth to half the sampling rate. Thus, operating on 8-kHz sampled speech, we wanted to reduce the sampling rate by discarding every fourth sample such that, after scrambling, the bandwidth would be 3 kHz, i.e., confined to the bandwidth of telephonic speech. The descrambled samples at the receiver would be brought back to the original 8-kHz rate by interpolating the missing samples. In the case of $\mu$-law PCM our interest was to discard a percentage (typically 25 percent) of the words representing encoded speech, enabling other data to have a free ride. At the receiver the data would be removed and the missing speech samples recovered by interpolation.

Interpolation ideas are part of everyday life; we are always filling in the gaps in our perception by interpolation, extrapolation, and prediction. We will avoid the luxury of philosophizing, and briefly review some of the interesting aspects of interpolation that are related to speech encoding. Mathews,[3] in an attempt to make significant reductions in the sampling rate of speech, considered extremal encoding. With this technique the amplitudes of the speech signal at its extremes and the time intervals between these extremes are transmitted in an encoded format. At the receiver the extremes can be widely separated, and the interpolation of the missing samples can be thought of as curve fitting, i.e., passing a quadratic function through the extreme sample and two nearest received samples. In the 1960's there was considerable interest in interpolation, which Kortman[4] defined as "the process of after-the-fact polynomial curve fitting to eliminate redundant data samples." The interpolators[4–6] tended to be concerned with zero-order, first-order, and fan interpolators, and were closely related to aperture predictors.[4–7] Andrews et al.[6] considered straight-line optimum, optimum interpolation filter, $(\sin x)/x$, Lagrange, and Fourier reconstruction interpolators. The use of piecewise polynomials, called splines, have been extensively investigated (see Ref. 8 and its bibliography). More recently, there have been a spate of publications[9] on

digital speech interpolation, although the interpolation there was mainly concerned with Time Assignment Speech Interpolation (TASI)-type systems. Interpolation techniques have also been applied to reduce distortion in packet switching of speech when a packet is lost or discarded.[10] For an understanding of the existing low-pass filtering procedures of interpolation and decimation of digital signals, the reader is directed to the in-depth review presented by Crochiere and Rabiner.[11]

Having commented on some of the existing interpolation methods let us now proceed to the issues to be addressed here. Consider the situation of being presented with the speech samples, or say $\mu$-law PCM words, at some point in a communications network, together with a system control demand to eliminate $J$ samples (or words) per $W$ samples (or words). This could happen, for example, if there were a sudden increase in traffic. Rejection of these $J$ samples might precipitate unacceptable degradation in the recovered speech unless we introduced at the receiver replacement samples that closely approximate those samples rejected. We must therefore decide at the outset on the means of reinserting the $J$ samples, for example, by prediction or interpolation, using those samples not discarded. Having replaced the missing samples, we need to establish criteria for judging the quality of the recovered speech signal. Is a single criterion such as mean square error sufficient, or is a combination of objective and subjective measures required? Coupled with the issues of reinsertion and quality are the criteria of how to select which $J$ samples are to be rejected in every $W$ samples. Should we determine if speech is present as distinct from silence, and if so, whether it is voiced or unvoiced? Is it better to reject small groups of samples (e.g., in silence periods), leaving clusters of samples intact because rejection followed by subsequent interpolation of one of these samples might cause significant speech degradation, or should we always endeavor to retain samples on either side of a rejected sample, and so on?

Clearly, the choices are legion, and faced with this situation we have imposed a set of guidelines that are not concerned with what the samples (or words) represent in the speech signal. For example, we make no attempt to recognize if the talker is male or female, whether the speech is voiced or unvoiced, or in transition, and so on. Instead, we simply reject samples on a periodic basis, and reinsert them at their destination by means of adaptive interpolation. By making the interpolation adaptive we take cognizance of the local statistics of the speech signal, specifically, the utilization of the signal's correlative properties. This is the only characteristic of the speech signal that is exploited. Our approach is, therefore, oriented to ease implementation rather than to squeeze the maximum advantage from the properties

of speech. Indeed, the method is applicable to other types of analog signals (e.g., modem-generated data signals), provided the signals possess correlative properties that are capable of exploitation in the interpolation processes. Our strategy is as follows:

1. Discard samples on a periodic basis, with no two consecutive samples being rejected.
2. Process blocks of $W$ samples at a time.
3. Exploit the local statistics of the speech samples by determining their correlation function over each block. Approximate methods for determining the correlation function are required when the sequence has had $J$ of its $W$ samples rejected.
4. Apply adaptive interpolation.
5. Employ quality criteria based on the minimization of the mean square error, justified by the prior knowledge that the mean square error values achieved will conform to perceptual standards based on informal listening tests.

Having introduced the notion of sample or word rejection to reduce the baud or bit rate, and an outline of how to discard the samples or words and reintroduce them at their destination, we will now formulate the problem and its solution in detail.

## II. THE PROBLEM

Consider a speech signal $x(t)$, bandlimited to $f_c H_z$ and sampled at $f_s H_z$ to yield the sequence $\{x_k\}$, where $f_s$ satisfies

$$f_s \geq 2f_c. \tag{1}$$

This sequence $\{x_k\}$ could be encoded into binary words, or we could be given binary words at a node in the network. In such situations our description of the problem would be in data words. However, for ease of explanation we will confine our discussion to operations on samples unless otherwise stated.

To reduce the number of samples per second in $\{x_k\}$, we discard every $n$th sample. This is achieved by clocking the speech samples into a gear-down changing buffer under the auspices of a clock operating at $f_s$ samples per second, such that, after every $n$ samples are inserted into the buffer, the clock is inhibited for one clock period. The arrangement is shown in Fig. 1. The samples are clocked out of the buffer at a rate

$$F_s < 2f_c \tag{2}$$

to yield a sequence $\{y_k\}$ whose components are uniformly spaced in time by $1/F_s$. Figure 2 shows $\{x_k\}$ and $\{y_k\}$ for arbitrary segments of sampled speech. Observe that $\{y_k\}$ has its parameter $F_s$ related to $f_s$ by

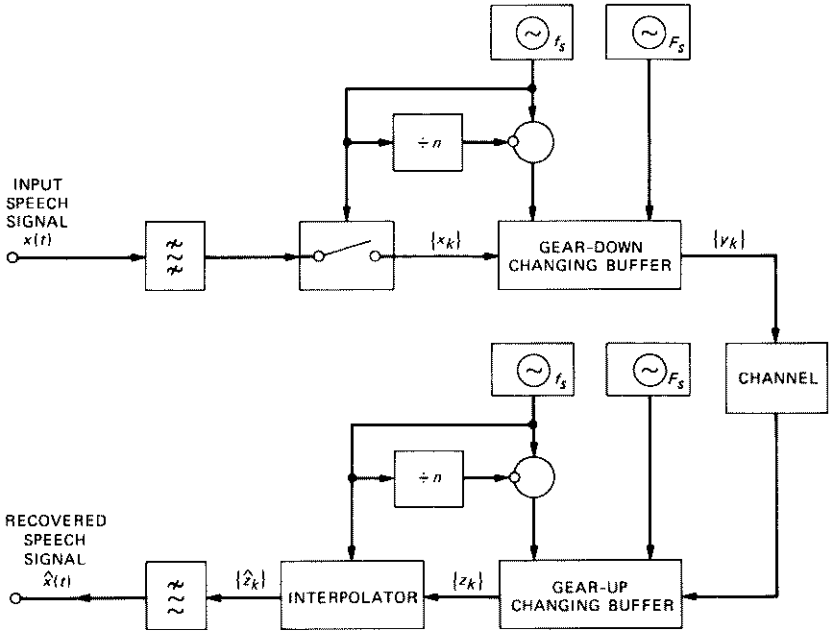



Fig. 1—Arrangement for decreasing the sample rate.

$$F_s = \left(\frac{n-1}{n}\right) f_s. \tag{3}$$

Suppose $\{y_k\}$ with its symbol rate lower than $\{x_k\}$ is transmitted over an ideal channel. Let us further assume that the receiver is able to return the samples in $\{y_k\}$ to the relative time positions they occupied in the original speech sequence $\{x_k\}$ using the gear-up changing buffer shown in Fig. 1. This newly formed sequence, $\{z_k\}$, generated at a rate $1/f_s$, has one out of every $n$ samples absent owing to the sample reduction process at the transmitter. A small, arbitrary segment of this sequence $\{z_k\}$, corresponding to a particular $\{x_k\}$ and $\{y_k\}$, is displayed in Fig. 2. Our problem is to replace those samples rejected at the transmitter with substitutes of acceptable accuracy whose generation is not excessively complex. The approach employed is adaptive interpolation.

## III. ADAPTIVE INTERPOLATION

The speech sequence $\{z_k\}$ is divided into sequential blocks having $W$ sample positions spaced $1/f_s$ seconds apart. The sequence $\{z_k\}$ is therefore composed of $W$ samples from $\{x_k\}$ with every $n$th sample absent. Our purpose is to interpolate the missing samples to produce

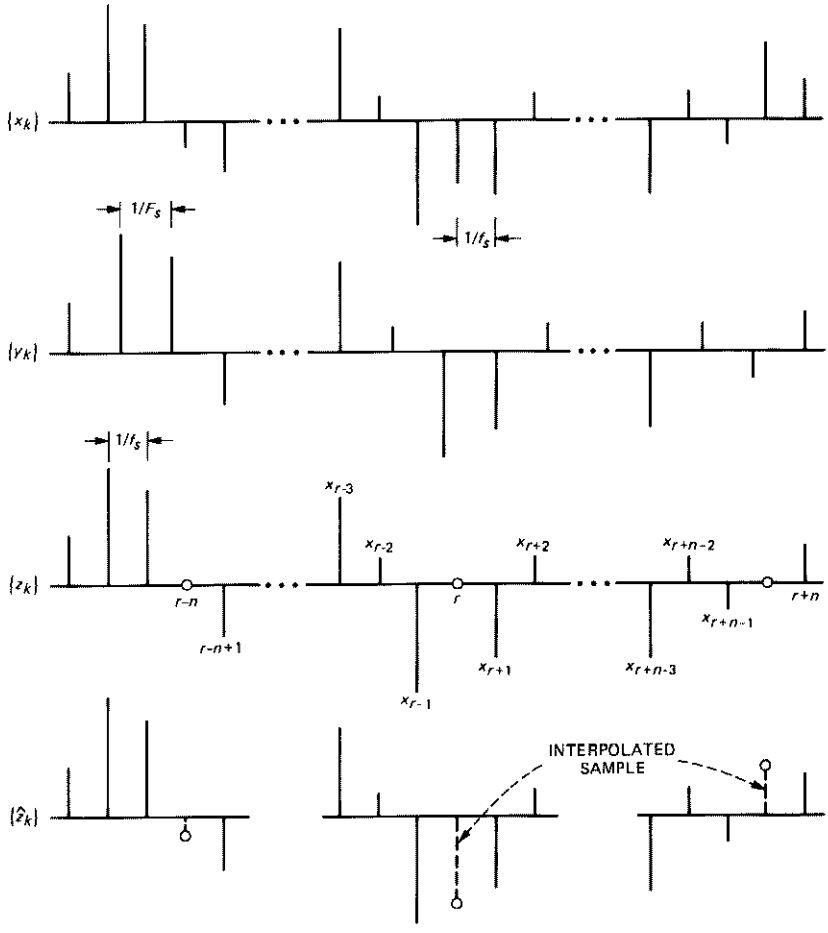



Fig. 2—Sampled sequences for arbitrary segment of speech; $\{x_k\}$ is the original speech sequence; $\{y_k\}$ is the sequence after sample reduction and gearing changing; $\{z\}$ is $\{x_k\}$ with every $n$th sample absent; $\{\hat{z}\}$ is the recovered speech sequence.

a sequence for the first block having components

$$x_1, \cdots, x_{n-1}, \hat{z}_n, x_{n+1}, \cdots, x_{W-2}, x_{W-1}\hat{z}_W,$$

where

$$\hat{z}_r; \qquad r = n, 2n, \cdots, W - n, W$$

are the interpolated samples. When interpolating each speech sample, we will use $\lambda$ past and $\lambda$ previous samples, and restrict $\lambda$ to be

$$\lambda \leq n - 1. \tag{4}$$

Thus, a missing sample at the $r$th sampling instant (see Fig. 2) is

formed by interpolation according to

$$\hat{z}_r = \sum_{i=-\lambda}^{-1} a_i x_{r+i} + \sum_{i=1}^{\lambda} a_i x_{r+i}, \tag{5}$$

where $a_i$ are the interpolation parameters. Equation (5) may be written as

$$\hat{z}_r = \sum_{i=-\lambda}^{\lambda} a_i x_{r+i}, \tag{6}$$

where $a_0 = 0$. The interpolation error of the $r$th sample is

$$e_r = x_r - \hat{z}_r \tag{7}$$

and the square of this error is

$$\begin{aligned} e_r^2 &= x_r^2 - 2x_r \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} + \left( \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} \right)^2 \\ &= x_r^2 - 2x_r \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} + \sum_{i=-\lambda}^{\lambda} a_i^2 x_{r+i}^2 + 2 \sum_{i=-\lambda}^{\lambda} \sum_{j=-\lambda}^{\lambda} a_i x_{r+i} a_j x_{r+j} \end{aligned} \tag{8}$$

with $i \neq j$ and $j > i$. To determine the interpolation parameters we proceed as follows. The square of the error is summed over the block of samples

$$\begin{aligned} E_r^2 = \sum_{r=n}^{W} e_r^2 = \sum_{r=n}^{W} x_r^2 - 2 \sum_{r=n}^{W} \left\{ x_r \sum_{i=-\lambda}^{\lambda} a_i x_{r+i} \right\} \\ + \sum_{r=n}^{W} \sum_{i=-\lambda}^{\lambda} a_i^2 x_{r+i}^2 + 2 \sum_{r=n}^{W} \sum_{i=-\lambda}^{\lambda} \sum_{j=-\lambda}^{\lambda} a_i x_{r+i} a_j x_{r+j}, \end{aligned} \tag{9}$$

where $r = n, 2n, \cdots, W$. To select coefficients that minimize this summation, we partially differentiate $E_r^2$ with respect to each of the coefficients and set the result to zero. After rearranging the equations we have

$$\sum_{r=n}^{W} x_r x_{r+j} = \sum_{i=-\lambda}^{\lambda} \sum_{r=n}^{W} a_i x_{r+i} x_{r+j}, \tag{10}$$

where

$$\begin{aligned} j &= \pm 1, \pm 2, \cdots, \pm\lambda \\ i &= -\lambda, -\lambda + 1, \cdots, \lambda \\ r &= n, 2n, \cdots, W - n, W \\ W &\geq 2n. \end{aligned}$$

The interpolation coefficients in eq. (10) can be represented in vector form as

$$\alpha = A^{-1}C, \tag{11}$$

where

$$\alpha = [a_{-\lambda}, a_{-\lambda+1}, \cdots, a_{-1}, a_1, \cdots, a_{\lambda-1}, a_\lambda]^T, \tag{12}$$

$$C = [R(0, -\lambda), R(0, -\lambda + 1), \cdots, R(0, -1), R(0, 1), \cdots, R(0, \lambda - 1), R(0, \lambda)]^T, \tag{13}$$

and the superscripts $-1$ and $T$ represent inverse and transpose operations, respectively. The rectangular matrix $A$ is of order $2\lambda$, and its elements are presented in Table I. The concentric dotted enclosures, commencing with the inner one, refer to the $A$ matrix for $\lambda = 1, 2, 3, \cdots$. The elements are given by

$$R(k, j) = \frac{\sum_{r=n}^{W} x_{r+k}x_{r+j}}{\sum_{r=n}^{W} x_{r+j}^2}$$

$$k = \pm 1, \pm 2, \cdots, \pm\lambda$$

$$j = \pm 1, \pm 2, \cdots, \pm\lambda$$

$$r = n, 2n, \cdots, W, \tag{14}$$

and $R(k, j)$ will be referred to as the correlation function $R(k, j)$. The vector $C$ has elements $R(0, j)$, i.e., the elements are given by eq. (13) with $k$ always zero.

As an example of the application of eq. (11), consider the case of $W = 32$, $n = 4$, $\lambda = 2$, whence

$$\begin{bmatrix} a_{-2} \\ a_{-1} \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & R(-1, -2) & R(1, -2) & R(2, -2) \\ R(-2, -1) & 1 & R(1, -1) & R(2, -1) \\ R(-2, 1) & R(-1, 1) & 1 & R(2, 1) \\ R(-2, 2) & R(-1, 2) & R(1, 2) & 1 \end{bmatrix}^{-1} \begin{bmatrix} R(0, -2) \\ R(0, -1) \\ R(0, 1) \\ R(0, 2) \end{bmatrix} \tag{15}$$

and eq. (14) becomes for the $A$ and $C$ matrices

$$R(k, j) = \frac{\sum_{r=4}^{32} x_{r+k}x_{r+j}}{\sum_{r=4}^{32} x_{r+j}^2}; \quad \begin{array}{l} k = 0, \pm 1, \pm 2 \\ j = \pm 1, \pm 2 \\ r = 4, 8, 12, 16, 20, 24, 28, 32 \end{array}$$

Table I—The $A$ matrix

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | $R(-\lambda+1,-\lambda)$ | . . | $R(-5,-\lambda)$ | $R(-4,-\lambda)$ | $R(-3,-\lambda)$ | $R(-2,-\lambda)$ | $R(-1,-\lambda)$ | $R(1,-\lambda)$ | $R(2,-\lambda)$ | $R(3,-\lambda)$ | $R(4,-\lambda)$ | $R(5,-\lambda)$ | . . | $R(\lambda-1,-\lambda)$ | $R(\lambda,-\lambda)$ |
| $R(-\lambda,-\lambda+1)$ | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | $R(\lambda,-\lambda+1)$ |
| . | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | . |
| . | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | . |
| $R(-\lambda,-5)$ | . . | . . | 1 | $R(-4,-5)$ | $R(-3,-5)$ | $R(-2,-5)$ | $R(-1,-5)$ | $R(1,-5)$ | $R(2,-5)$ | $R(3,-5)$ | $R(4,-5)$ | $R(5,-5)$ | . . | . . | $R(\lambda,-5)$ |
| $R(-\lambda,-4)$ | . . | . . | $R(-5,-4)$ | 1 | $R(-3,-4)$ | $R(-2,-4)$ | $R(-1,-4)$ | $R(1,-4)$ | $R(2,-4)$ | $R(3,-4)$ | $R(4,-4)$ | $R(5,-4)$ | . . | . . | $R(\lambda,-4)$ |
| $R(-\lambda,-3)$ | . . | . . | $R(-5,-3)$ | $R(-4,-3)$ | 1 | $R(-2,-3)$ | $R(-1,-3)$ | $R(1,-3)$ | $R(2,-3)$ | $R(3,-3)$ | $R(4,-3)$ | $R(5,-3)$ | . . | . . | $R(\lambda,-3)$ |
| $R(-\lambda,-2)$ | . . | . . | $R(-5,-2)$ | $R(-4,-2)$ | $R(-3,-2)$ | 1 | $R(-1,-2)$ | $R(1,-2)$ | $R(2,-2)$ | $R(3,-2)$ | $R(4,-2)$ | $R(5,-2)$ | . . | . . | $R(\lambda,-2)$ |
| $R(-\lambda,-1)$ | . . | . . | $R(-5,-1)$ | $R(-4,-1)$ | $R(-3,-1)$ | $R(-2,-1)$ | 1 | $R(1,-1)$ | $R(2,-1)$ | $R(3,-1)$ | $R(4,-1)$ | $R(5,-1)$ | . . | . . | $R(\lambda,-1)$ |
| $R(-\lambda,1)$ | . . | . . | $R(-5,1)$ | $R(-4,1)$ | $R(-3,1)$ | $R(-2,1)$ | $R(-1,1)$ | 1 | $R(2,1)$ | $R(3,1)$ | $R(4,1)$ | $R(5,1)$ | . . | . . | $R(\lambda,1)$ |
| $R(-\lambda,2)$ | . . | . . | $R(-5,2)$ | $R(-4,2)$ | $R(-3,2)$ | $R(-2,2)$ | $R(-1,2)$ | $R(1,2)$ | 1 | $R(3,2)$ | $R(4,2)$ | $R(5,2)$ | . . | . . | $R(\lambda,2)$ |
| $R(-\lambda,3)$ | . . | . . | $R(-5,3)$ | $R(-4,3)$ | $R(-3,3)$ | $R(-2,3)$ | $R(-1,3)$ | $R(1,3)$ | $R(2,3)$ | 1 | $R(4,3)$ | $R(5,3)$ | . . | . . | $R(\lambda,3)$ |
| $R(-\lambda,4)$ | . . | . . | $R(-5,4)$ | $R(-4,4)$ | $R(-3,4)$ | $R(-2,4)$ | $R(-1,4)$ | $R(1,4)$ | $R(2,4)$ | $R(3,4)$ | 1 | $R(5,4)$ | . . | . . | $R(\lambda,4)$ |
| $R(-\lambda,5)$ | . . | . . | $R(-5,5)$ | $R(-4,5)$ | $R(-3,5)$ | $R(-2,5)$ | $R(-1,5)$ | $R(1,5)$ | $R(2,5)$ | $R(3,5)$ | $R(4,5)$ | 1 | . . | . . | $R(\lambda,5)$ |
| . | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | . |
| . | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | . |
| . | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | . |
| $R(-\lambda,\lambda-1)$ | . . | . . | . | . | . | . | . | . | . | . | . | . | . . | . . | $R(\lambda,\lambda-1)$ |
| $R(-\lambda,\lambda)$ | $R(-\lambda+1,\lambda)$ | . . | $R(-5,\lambda)$ | $R(-4,\lambda)$ | $R(-3,\lambda)$ | $R(-2,\lambda)$ | $R(-1,\lambda)$ | $R(1,\lambda)$ | $R(2,\lambda)$ | $R(3,\lambda)$ | $R(4,\lambda)$ | $R(5,\lambda)$ | . . | $R(\lambda-1,\lambda)$ | 1 |

When $r = 32$ and $|k|$ or $|j| \geq 1$, samples are used that reside in the subsequent block.

When the block size $W$ is large, typically in excess of 256 samples, the computations required to solve eq. (11) can be reduced at the expense of a small increase in interpolation noise power. This is accomplished by replacing $R(k, j)$ of eq. (14) by the correlation function

$$R(\tau) = \frac{(W-\tau)\sum_{i=1}^{W-\tau} x_i x_{i+\tau} - \left(\sum_{i=1}^{W-\tau} x_i\right)\left(\sum_{i=1}^{W-\tau} x_{i+\tau}\right)}{\left\{\left[(W-\tau)\sum_{i=1}^{W-\tau} x_i^2 - \left(\sum_{i=1}^{W-\tau} x_i\right)^2\right] \cdot \left[(W-\tau)\sum_{i=1}^{W-\tau} x_{i+\tau}^2 - \left(\sum_{i=1}^{W-\tau} x_{i+\tau}\right)^2\right]\right\}^{1/2}}, \tag{16}$$

where

$$\tau = |k - j| \tag{17}$$

and $k$ and $j$ have values prescribed in eq. (14). For a further small increase in interpolation noise, which typically reduces the recovered signal-to-noise ratio (s/n) by a couple of decibels compared to when eq. (16) is used, $R(\tau)$ can be simplified to

$$R(\tau) = \frac{\sum_{r=1}^{W-\tau} x_r x_{r+\tau}}{\sum_{r=1}^{W} x_r^2}, \tag{18}$$

where $r$ is integer-valued. Observe that, in computing the interpolation coefficients with the aid of eqs. (16) or (18), only $\lambda$ values need be determined as

$$a_{-p} = a_p; \qquad p = 1, 2, \cdots, \lambda. \tag{19}$$

By contrast, when $R(k, j)$ is used in preference to $R(\tau)$, eq. (19) does not apply and $2\lambda$ coefficients must be computed.

Thus, provided we can compute $R(k, j)$ or $R(\tau)$, over a duration of $W/f_s$, where $W$ is the block length, we can determine the interpolation parameters contained in the vector $\alpha$. Employing eq. (5) we can estimate the missing samples by means of interpolation. The mean squared error between $\{x_k\}$, and the recovered sequence $\{\hat{z}_k\}$ containing the interpolated samples (see the example shown in Fig. 2), is approximately minimized, provided $W$ is sufficiently large. Practical values of $W$ are determined in Section VI.

## IV. INTERPOLATION PARAMETERS DERIVED FROM THE INPUT DATA

The interpolation parameters are computed by first finding the correlation function $R(\theta)$, where $\theta$ is $k$, $j$, or $\tau$. Equation (14) shows $R(\theta)$ as dependent on the input sequence $\{x_k\}$. Clearly, as $\{x_k\}$ is only known at the transmitter, it follows that the interpolation vector $\alpha$ must be computed at the transmitter and multiplexed with the slowed-down speech samples $\{y_k\}$. Consequently,

$$Y = \left(\frac{n-1}{n}\right) W + \nu \tag{20}$$

samples are transmitted every $W/f_s$ seconds, where $Y$ is composed of $W(n-1)/n$ speech samples and $\nu$ interpolation parameter samples. The value of $\nu$ is $\lambda$ or $2\lambda$, depending on whether $\theta$ is $\tau$ or $k$, $j$, respectively. This means that $F_s$ of eq. (3) is modified to

$$F_s' = F_s + \frac{\nu}{W} f_s. \tag{21}$$

For example, if $f_s = 8$ kHz, $n = 4$, $F_s = 6$ kHz, $W = 256$, $\lambda = 3$, $\theta = \tau$, then $F_s' = 6.093$ kHz. We also observe from eq. (20) that three interpolation samples are sent for every 192 speech samples. The values of $\lambda$ and $W$ as a function of s/n are present in Section VI.

## V. INTERPOLATION PARAMETERS DERIVED FROM THE RECEIVED DATA

The receiver produces the sequence $\{z_k\}$ whose samples are spaced apart by $1/f_s$, and in every $n$th sample position one sample is missing, as shown in Fig. 2. The receiver has the task of estimating the interpolation coefficients from $\{z_k\}$, and must therefore commence by calculating the correlation function $R(\tau)$ without the full knowledge of the original speech sequence $\{x_k\}$. In this situation we proceed as follows. The missing samples are found as an average of adjacent speech samples

$$\bar{x}_r = (x_{r-1} + x_{r+1})/2 \tag{22}$$

until a sequence $\{\bar{z}_k\}$ consisting of $W(n-1)/n$ original speech samples and $W/n$ interpolated samples is formed, where each sample is equally spaced from its neighbor by $1/f_s$ seconds. This sequence corresponds to a recovered speech signal that has considerable distortion, particularly for female speakers. Instead of accepting $\{\bar{z}_k\}$ as the recovered speech sequence, we use it solely for the purpose of computing $R(\tau)$, and hence the vector $\alpha$ containing $a_{-1}$ and $a_1$. We now remove the samples introduced by eq. (22) and replace them by using adaptive interpolation based on the two nearest neighbors, viz:

$$ {}_2x_r = a_{-1}x_{r-1} + a_1 x_{r+1}, \tag{23} $$

where $a_{-1}$ and $a_1$ are found using eq. (11) with $R(k, j)$ replaced by $R(\tau)$ of eq. (18). The new recovered speech sequence $\{{}_2z_k\}$ has interpolated samples formed according to eq. (23), and in general contains less distortion than $\{\tilde{z}_k\}$. However, we can further reduce the distortion. The function $R(\tau)$ is again computed, this time from $\{{}_2z_k\}$. The interpolated samples in $\{{}_2z_k\}$, derived with the aid of eq. (23), are rejected and replaced by

$$ {}_4x_r = a_{-2}x_{r-2} + a_{-1}x_{r-1} + a_1 x_{r+1} + a x_{r+2} \tag{24} $$

to yield the speech sequence $\{{}_4z_k\}$. The correlation function $R(\tau)$ of sequence $\{{}_4z_k\}$ is found, and those interpolated samples previously formulated with the aid of eq. (24) are exchanged for

$$ {}_6x_r = \sum_{i=-3}^{3} a_i x_{r-i}, \qquad a_0 = 0. \tag{25} $$

This process of using two more samples per iteration in the interpolation procedure continues until the limits of the summation in eq. (25) become $n - 1$, when the final recovered speech sequence $\{\hat{z}_k\}$ is obtained. It should be noted that more than $2(n - 1)$ samples can be used in each interpolation process, but the improvement in interpolation accuracy results in a significant increase in complexity.

Often it is sufficient to produce the sequence $\{\tilde{z}_k\}$, compute $R(\tau)$ from $\{\tilde{z}_k\}$, remove the samples formed by averaging the adjacent two samples, and with the aid of $\lambda \leq n - 1$ samples on either side of each discarded sample, insert the missing samples by adaptive interpolation. When this process is adopted, it will be referred to as $\lambda$-interpolation. However, when the same $\lambda$ samples are used in the interpolation process, and the iteration procedure of eqs. (22) to (25) activated, we will refer to this interpolation scheme as $\lambda$-with-iteration.

Observe that $R(\tau)$ is used in the $\lambda$-with-iteration scheme. If $R(k, j)$ is employed instead of $R(\tau)$, the final interpolation is not better than that of the sample used to approximate $x_r$. For example, if we replaced $x_r$ by $\tilde{x}_r$, the iterative algorithm would merely attempt to minimize the error power between the $\tilde{x}_r$ samples and the interpolated samples. Unlike the application of $R(k, j)$, the $R(\tau)$ function does not enable an exact minimization of interpolation error power to be achieved at the transmitter. However, it transpires that by computing $R(\tau)$ at the receiver, and using the iterative procedure to interpolate the missing samples, the interpolation noise is significantly reduced (see Section 6.2).

## VI. RESULTS

The speech signal used in our experiments consisted of two con-

catenated sentences, "Live wires should be kept covered," and "To reach the end he needs much courage." These were spoken by a male and female, respectively. The signal was bandlimited from 0.3 to 3.2 kHz and sampled at 8 kHz to give the input speech sequence $\{x_k\}$. This sequence is displayed in Fig. 3, together with its spectrogram, where the higher frequencies have been preemphasized.

### *6.1 Interpolation parameters generated from the original speech*

The gear-down changing procedure was invoked (see Figs. 1 and 2) whereby the sampling rate of 8 kHz was decreased to a uniform $F_s$-kHz rate by rejecting every $n$th sample, and adjusting the sample spacing to provide the output sequence $\{y_k\}$. This sequence was assumed to be transmitted through an ideal channel, and after passing through the receiver's gear-up changing buffer the sequence $\{z_k\}$ was formed. The sequence $\{z_k\}$ had a symbol rate of 8 kHz, with every $n$th sample absent. The absentee samples rejected at the transmitter were then formulated according to eq. (6). The interpolation parameters $a_i$ were found with the aid of eq. (11), which used correlation functions $R(k, j)$ related to the input speech sequence $\{x_k\}$. Thus, in our first experiment we were concerned with how successfully we could discard every $n$th sample in $\{x_k\}$, and replace the discarded samples using interpolation parameters based on $\{x_k\}$.

We used as a performance criterion segmental signal-to-noise ratio[12] (SEG-s/n), computed using the input sequence $\{x_k\}$, and the error sequence $\{e_k\}$ whose components are given by eq. (7). In our initial experiment the variation of SEG-s/n as a function of block size $W$ was found using practical values of $W$ extending from 64 to 1024, and $n = 4$. This value of $n$ is a compromise between providing adequate SEG-s/n and a reasonable reduction in the number of transmitted samples. Applying the correlation function $R(k, j)$ in the determination of the interpolation parameters, we obtained the solid curves in Fig. 4. Curves a, b, and c apply for the case of $\lambda = 3$, 2, and 1, respectively. Curve d is the SEG-s/n when the interpolation was performed using the average of adjacent samples. Increasing $\lambda$ to $n - 1$, i.e., to 3, resulted in an increase in SEG-s/n compared to lower values of $\lambda$, and for a block size of 256 and $\lambda = 3$, a gain of 13.7 dB in SEG-s/n was obtained compared to the simple interpolation averaging method of eq. (22). When the correlation function $R(\tau)$ of eq. (18) was employed to compute the interpolation parameters, curves e, f, and g were obtained corresponding to $\lambda = 3$, 2, and 1, respectively. The improvement in SEG-s/n derived from employing $R(k, j)$ rather than $R(\tau)$ increased with increasing $\lambda$, being 0.1, 1, and 3 dB for $\lambda = 1$, 2, and 3, respectively, and with $W = 256$.

As a means of providing further insight into the performance of the

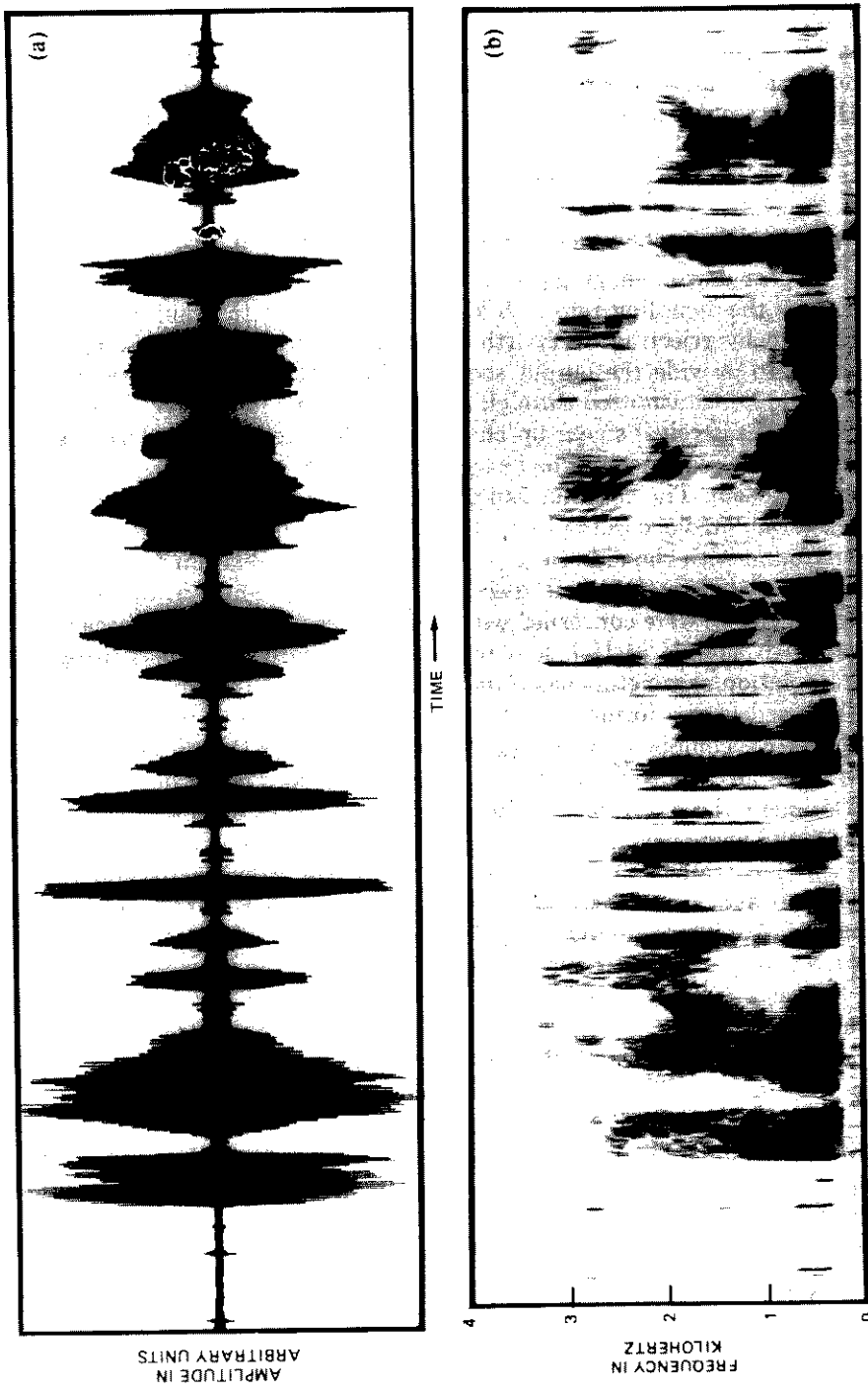



Fig. 3—(a) Input speech sequence $\{x_k\}$ and (b) its preemphasized spectrogram.

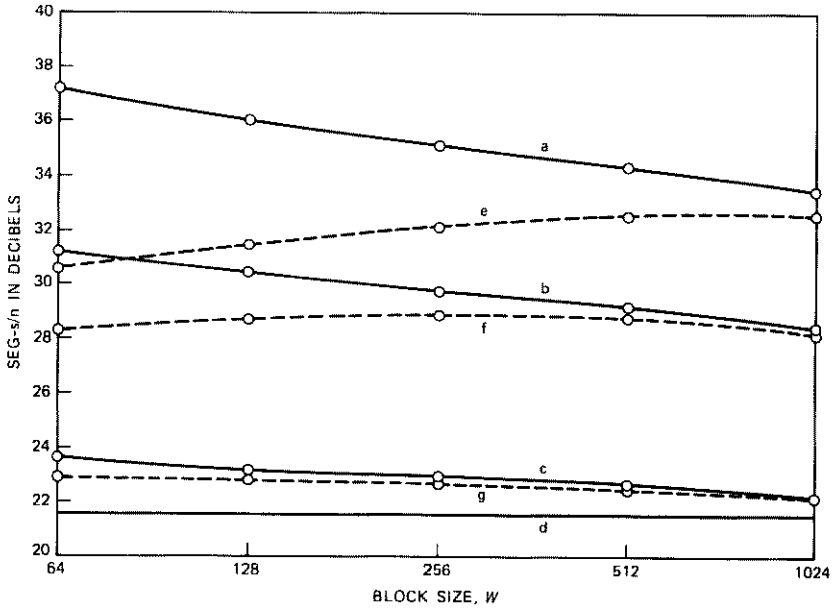



Fig. 4—SEG-s/n versus block size $W$. Curves a, b, and c apply for $\lambda = 3$, 2, and 1; and $\hat{R}(k, j)$ was used in calculating the interpolation parameters. Curve d relates to interpolation using nearest neighbor averaging. Curves e, f, and g apply for $\lambda = 3$, 2, and 1; and $R(\tau)$, given by eq. (18), was used in calculating the interpolation parameters.

interpolation system, we show in Fig. 5 the variation of the s/n of each block in the speech signal as a function of successive blocks for $W = 256$, $n = 4$. The average of these s/n values constitutes the SEG-s/n points in Fig. 4 for $W = 256$. As expected, we found that the s/n in every block was greater, if only by an infinitesimal amount, when more samples were used in the interpolation process, i.e., when larger values of $\lambda$ were employed. Interpolation by adjacent sample averaging always provided the lowest s/n. In some blocks the advantage of using $\lambda = 3$ compared to $\lambda = 2$, $\lambda = 1$, and nearest neighbor averaging, provided s/n gains as large as 13, 31, and 33 dB, respectively.

Returning to Fig. 4, the SEG-s/n of over 35 dB, $\lambda = 3$, was found to be approximately 3 dB greater than the coventional s/n computed by measuring the mean square values of the components in $\{x_k\}$ and $\{e_k\}$ over the entire speech input signal. Our SEG-s/n measurements therefore indicate that the recovered speech is similar to toll quality speech.[2] Informal listening experiences for the recovered speech sequence $\{\hat{z}_k\}$ when $\lambda = 3$, $W = 256$, tended to confirm the SEG-s/n findings that the quality of the recovered speech sequence $\{\hat{z}_k\}$ was judged to be very similar to that of the original bandlimited sequence $\{x_k\}$. We observed that although the distortion in $\{\hat{z}_k\}$ for $\lambda = 1$ was

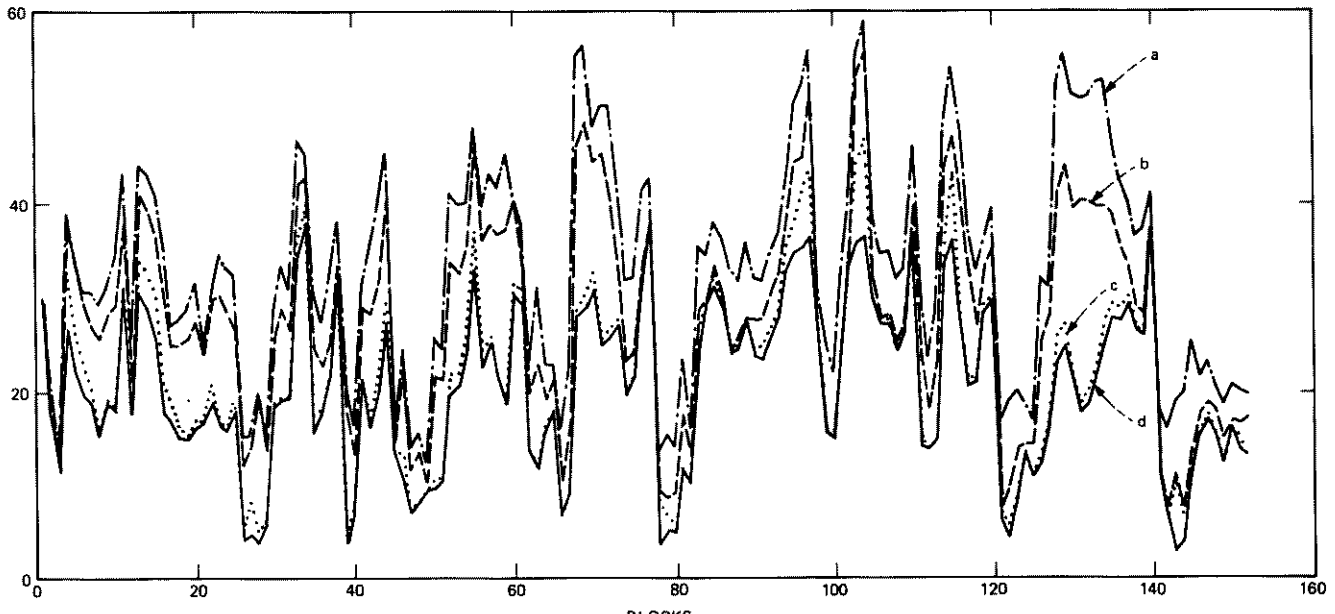



Fig. 5—Variation of block s/n as a function of block number, $W = 256$, $n = 4$. Interpolation procedure used (a) $\lambda = 3$, (b) $\lambda = 2$, (c) $\lambda = 1$, and (d) nearest neighbor averaging.

annoying, by making $\lambda = 2$ the interpolation noise was barely noticeable.

In Section III we present different expressions for the correlation function $R(\theta)$, namely eq. (14), where $\theta = k, j$, and eqs. (16) or (18) having $\theta = \tau$. The SEG-s/n and block s/n values shown in Figs. 4 and 5 were determined using $\theta = k, j$. We now demonstrate the loss in SEG-s/n due to using $\theta = \tau$ compared to when $\theta = k, j$ as a function of $W$, for the conditions of $n = 4$ and $\lambda = 3$. It will be recalled that $\theta = k, j$ enables the interpolation samples to be formulated that minimize $E_r^2$ over a block of $W$ samples. When $\theta = \tau$ a low but not minimum value of $E_r^2$ is produced over the working range of $W$. The application of eq. (16) gives a more accurate measure of $R(\tau)$ than the simpler expression of eq. (18). Figure 6 shows the variation of SEG-s/n with block size when $R(\theta)$ is computed using eqs. (14), (16), and (18), $n = 4$, $\lambda = 3$. When $R(k, j)$ was used, the effect of increasing $W$ was to decrease the SEG-s/n. This is to be expected because the interpolation parameters are fixed for a block, and we may think of a large block as composed of many smaller blocks, each with its optimum interpolation parameters. Thus, if one set of parameters is selected for the large block, these parameters are inevitably suboptimum for the smaller subblocks, and hence the SEG-s/n is lower for the larger

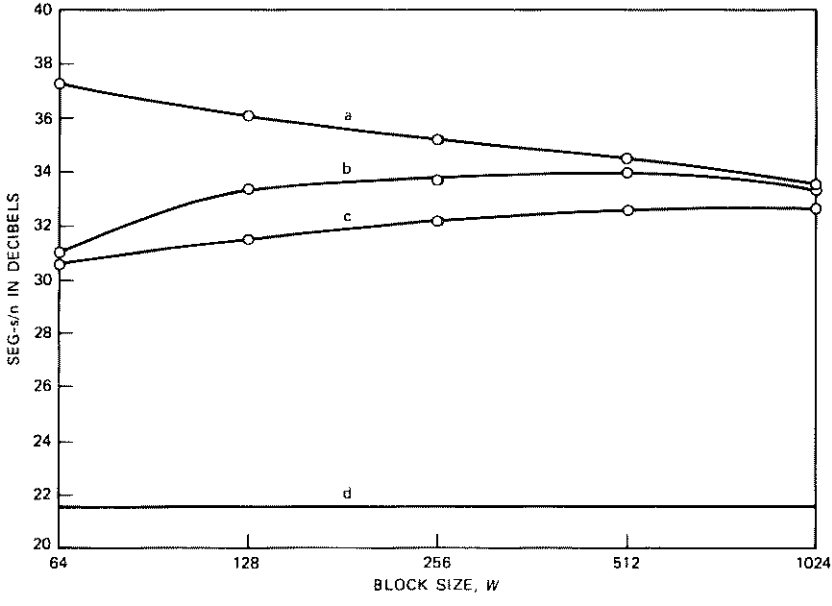



Fig. 6—Variation of SEG-s/n versus block size $W$ for $\lambda = 3$, $n = 4$. The correlation function was computed for curves a, b, and c using eqs. (12), (16), and (18), respectively. Curve d applies to nearest neighbor averaging.

blocks. By contrast the SEG-s/n determined using $R(\tau)$ deteriorates with decreasing $W$. Now $R(\tau)$ is the conventional correlation function that assumes the speech signal to have stationary statistics. For the larger block sizes shown in Fig. 6, local statistical departures from stationarity tend to be smoothed by the $R(\tau)$ equations, but at low values of $W$ a considerable number of interpolation errors occur in some blocks to yield a low SEG-s/n. When $W = 1024$ the SEG-s/n values computed using the different $R(\theta)$ expressions are very similar. We do not plot curves for $W < 64$ as the side information to transmit the interpolation parameters is unacceptably high [see eq. (20)], and for $W > 1024$ the delay is excessive (>250 ms). Thus, by using $R(k, j)$ we obtain higher and better interpolation performance compared with employing the conventional $R(\tau)$, but the computational complexity is greater.

### 6.2 *Interpolation parameters generated from the received sequence*

Having concluded that every fourth speech sample can be discarded and replaced by an interpolated sample to yield speech with negligible perceptual degradation, we next considered the performance of our scheme when the interpolation parameters were derived from the received data. The problem in this case was how to obtain a reliable estimate of the autocorrelation function $R(\tau)$. The procedures described in Section V for determining $R(\tau)$, and thence the interpolation parameters, were tried, and the variation of SEG-s/n with block size

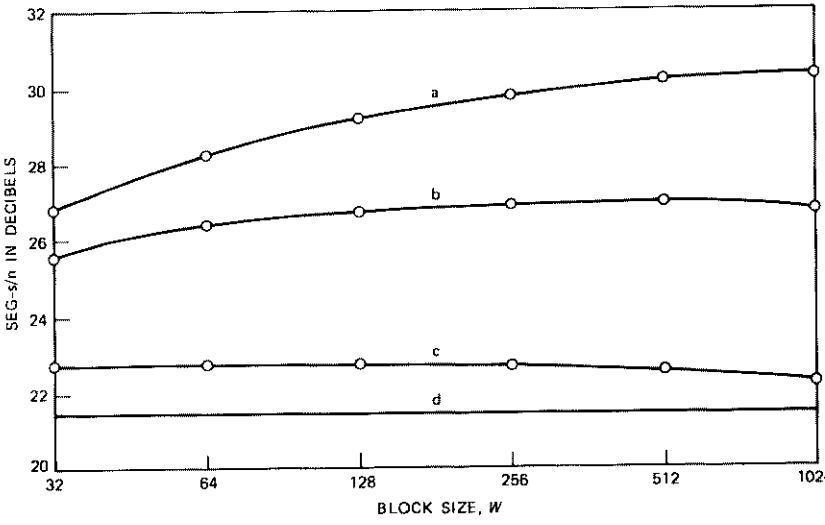



Fig. 7—SEG-s/n versus block size for interpolation parameters derived from the received sequence. Curves a, b, c, and d apply for λ-with-iteration, λ = 3; λ-interpolation, λ = 3; λ-interpolation, λ = 1; and average of adjacent sample interpolation, respectively.

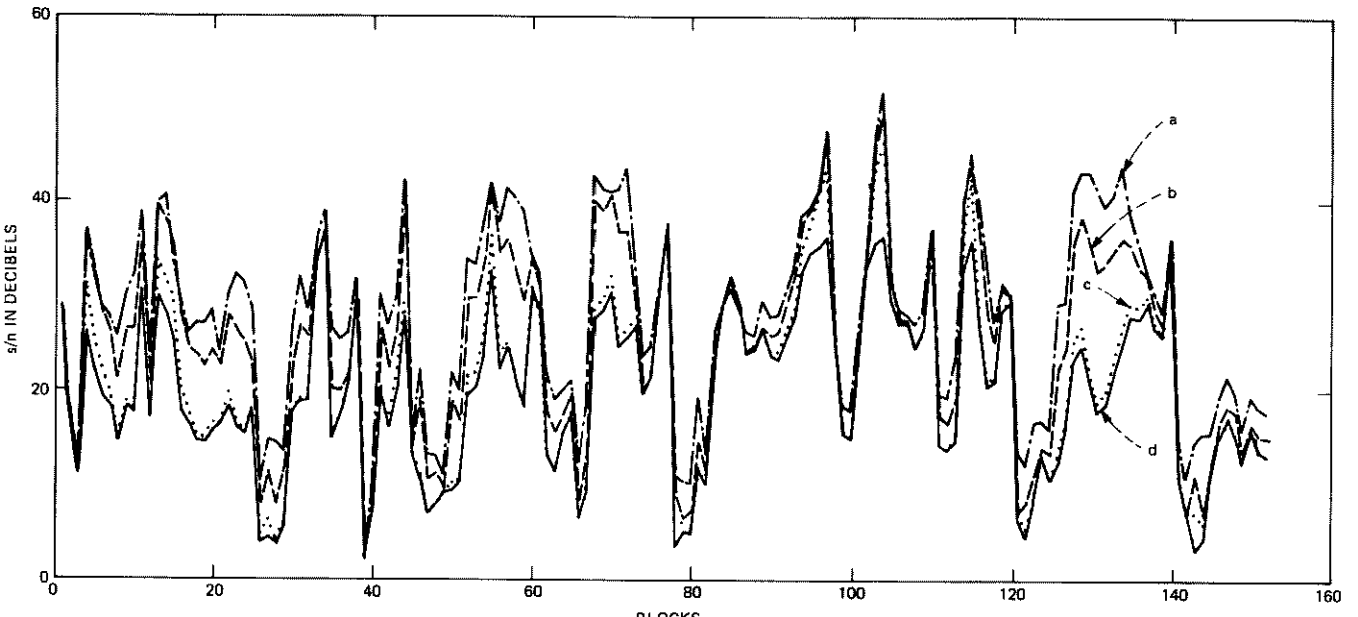



Fig. 8—Variation of block s/n versus block number for the iterative interpolation procedure, $W = 256$, $n = 4$. Curves a, b, c, and d apply for $\lambda = 3$, 2, 1, and nearest neighbor averaging, respectively.

$W$ found for different values of $\lambda$. Curve d in Fig. 7 shows that a SEG-s/n of 21.5 dB was obtained when the interpolation process used nearest neighbor averaging, and was employed as a reference level. When the nearest neighbor interpolation was made adaptive, eq. (23) was used for which $\lambda = 1$, and curve c obtained. By using the iteration procedure where the interpolation was made according to eq. (25), $\lambda = 3$, a s/n $> 30$ dB was achieved for $N = 512$ (see curve a).

Also shown in Fig. 7 is curve b, which was obtained by formulating the sequence $\{\hat{z}_k\}$ based on first computing $\{\bar{z}_k\}$ according to eq. (22), formulating $R(\tau)$, and then removing samples $\bar{z}_r$ and replacing them with interpolated samples derived by using $\lambda = 3$. Thus, in this $\lambda$-interpolation method we do not progress from the average sequence to those derived with $\lambda = 1$ and 2, but proceed directly from the average sequence to compute the parameters with $\lambda = 3$. The result is a s/n of 27 dB for $W = 256$ to 1024, a 3-dB reduction compared to the $\lambda$-with-iteration ($\lambda = 3$) case, and a diminution in complexity. Informal listening experiences showed that the $\lambda$-interpolation and $\lambda$-with-iteration schemes, both with $\lambda = 3$, produced speech whose impairments were barely perceptible.

The variation of block s/n with block number for the iterative interpolation procedure is displayed in Fig. 8, $n = 4$, $W = 256$. The average values of these block s/n's give the segmental s/n in Fig. 7 for $W = 256$. Close inspection of the curves in Fig. 8 reveal that progressive iteration does not always give the highest block s/n. However, $\lambda$-with-iteration, $\lambda = 3$, achieved the highest s/n for most blocks with gains up to 25 dB compared to nearest neighbor averaging interpolation.

In Fig. 9, the variation of block s/n with block number is displayed for the $\lambda = 3$ condition, $n = 4$, $W = 256$. Curves a and b are those previously displayed in Figs. 5a and 8a, and refer to interpolation parameters computed from the original speech sequence, and by $\lambda$-with-iteration procedure, respectively. Curve c applies for the $\lambda$-interpolation method, while curve d corresponds to nearest neighbor averaging interpolation, and is included as a reference level. The curves in Fig. 9 illustrate that by deriving the interpolation parameters from the original speech instead of from the received data, the large dips in block s/n are mitigated.

#### *6.2.1 Interpolation errors*

We will now consider the interpolation error sequences and their spectra when the interpolation parameters are generated from the received sequence $\{z_k\}$. To illustrate the error performance we selected an arbitrary segment of our input speech signal (see Fig. 3) that had high-level and low-level voiced speech, and unvoiced speech. The speech segment and its spectrogram are displayed in Fig. 10. As before,

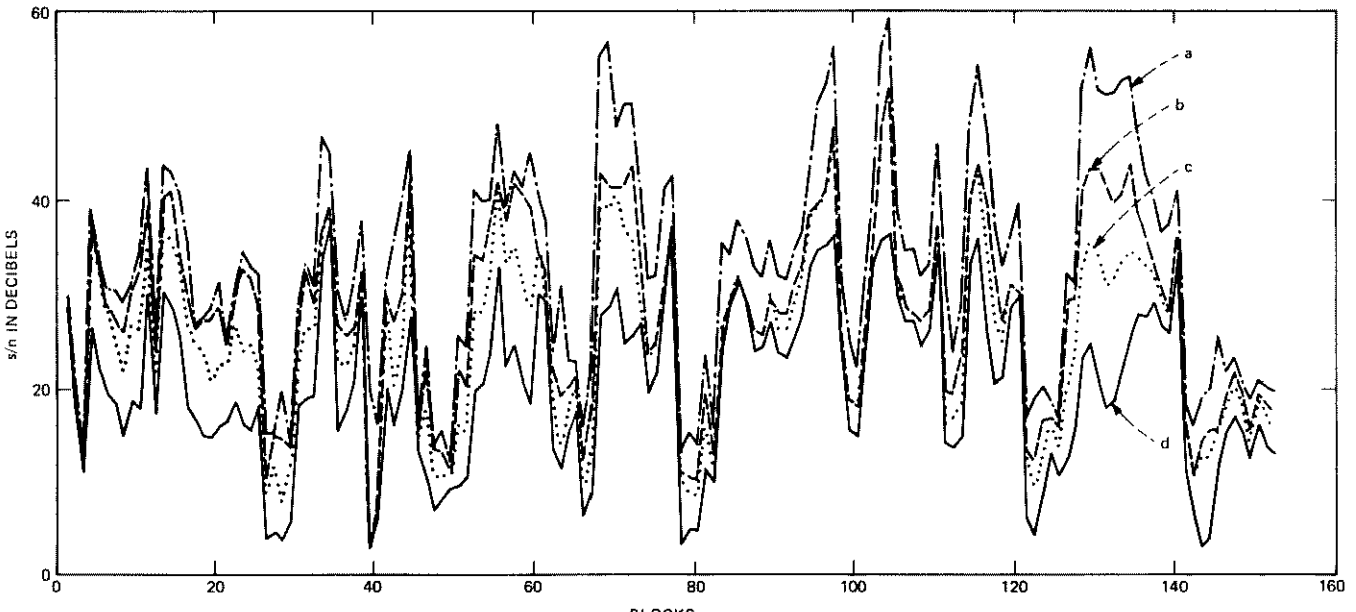



Fig. 9—Variation of block s/n as a function of block number for the $\lambda = 3$ condition, $W = 256$, $n = 4$. Curves a, b, c, and d are for interpolation parameters computed from original speech, $\lambda$-with-iteration, $\lambda$-interpolation, and nearest neighbor averaging, respectively.

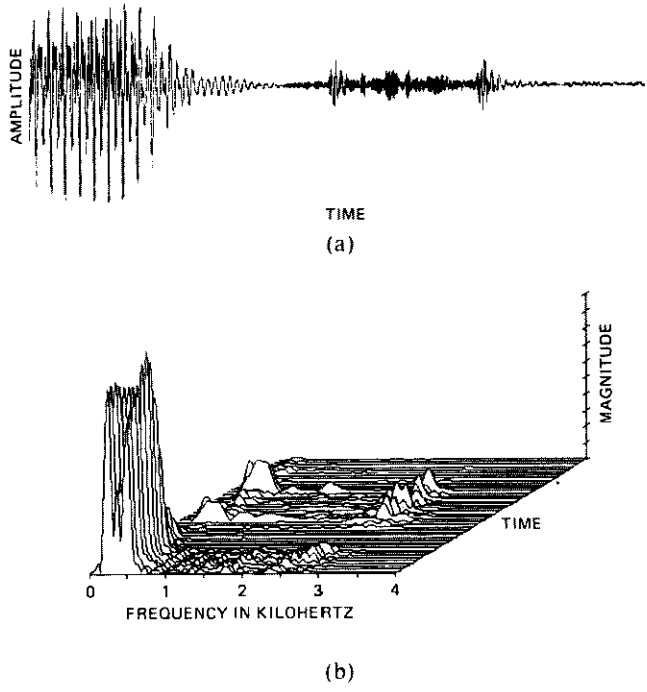



Fig. 10—Speech segment: (a) time waveform, and (b) spectrogram.

every fourth sample in the segment was removed and replaced by a sample produced by interpolation from neighboring samples, where the interpolation parameters were derived from the received data as described in Section V. We avoid displaying the spectrograms of the recovered speech segments associated with the four interpolation conditions used in Fig. 7 because of their similarity. Instead we show in Fig. 11a, b, c, and d the error signals determined as the difference between the input speech segment shown in Fig. 10a and the recovered waveforms produced using interpolation methods of: average of adjacent samples; λ-interpolation with $\lambda = 1$ and 3; λ-with-iteration, $\lambda = 3$; respectively. The error signals in Fig. 11 are small for the low-level highly correlated voiced speech section, and are much greater in the high-level voiced section and for the unvoiced speech. The block s/n values for the speech signal in Fig. 10a are therefore higher for the voiced speech than for the unvoiced speech. Inspection of Fig. 11 shows that the smallest error signal amplitudes generally occurred when the interpolation procedure used three samples on either side of each missing sample. For this segment of speech the advantage of using the full iteration procedure is small compared to $\lambda = 3$, no

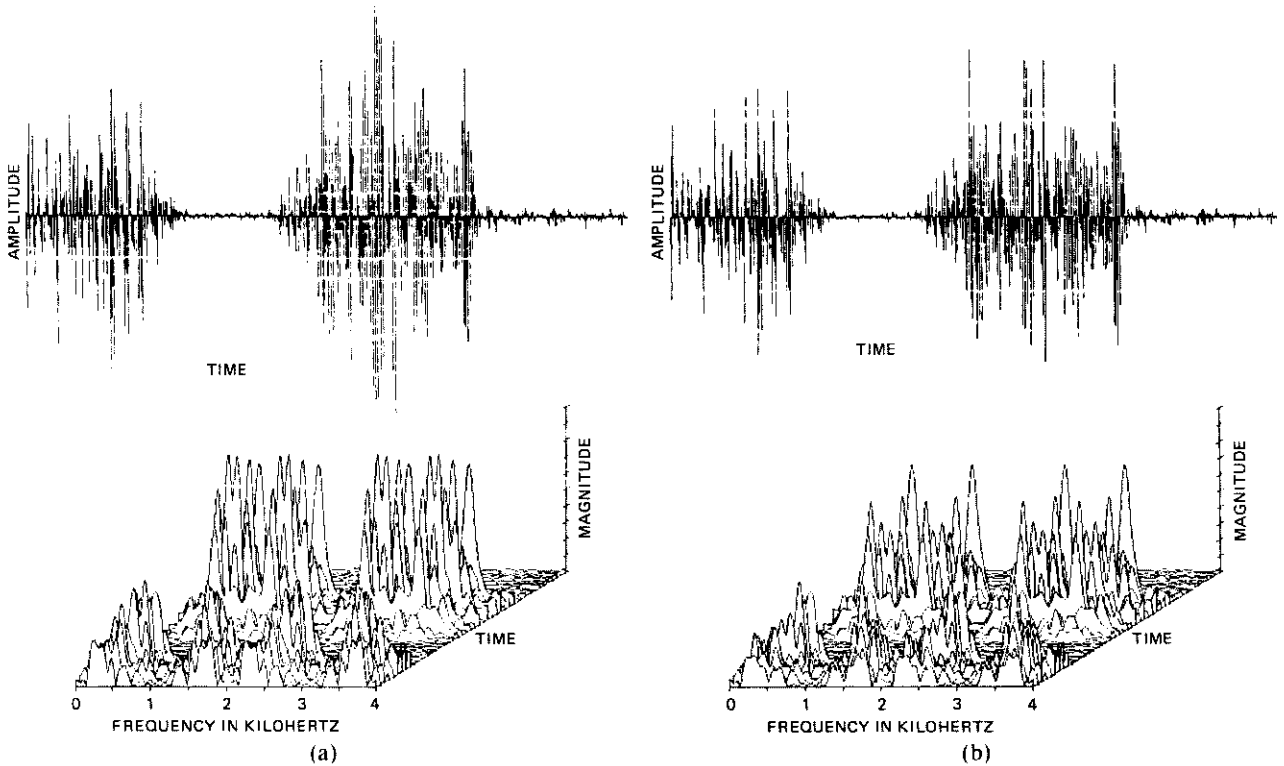



Fig. 11 (a) and (b)—Error waveforms and their spectrograms.

(Figure is continued on next page.)

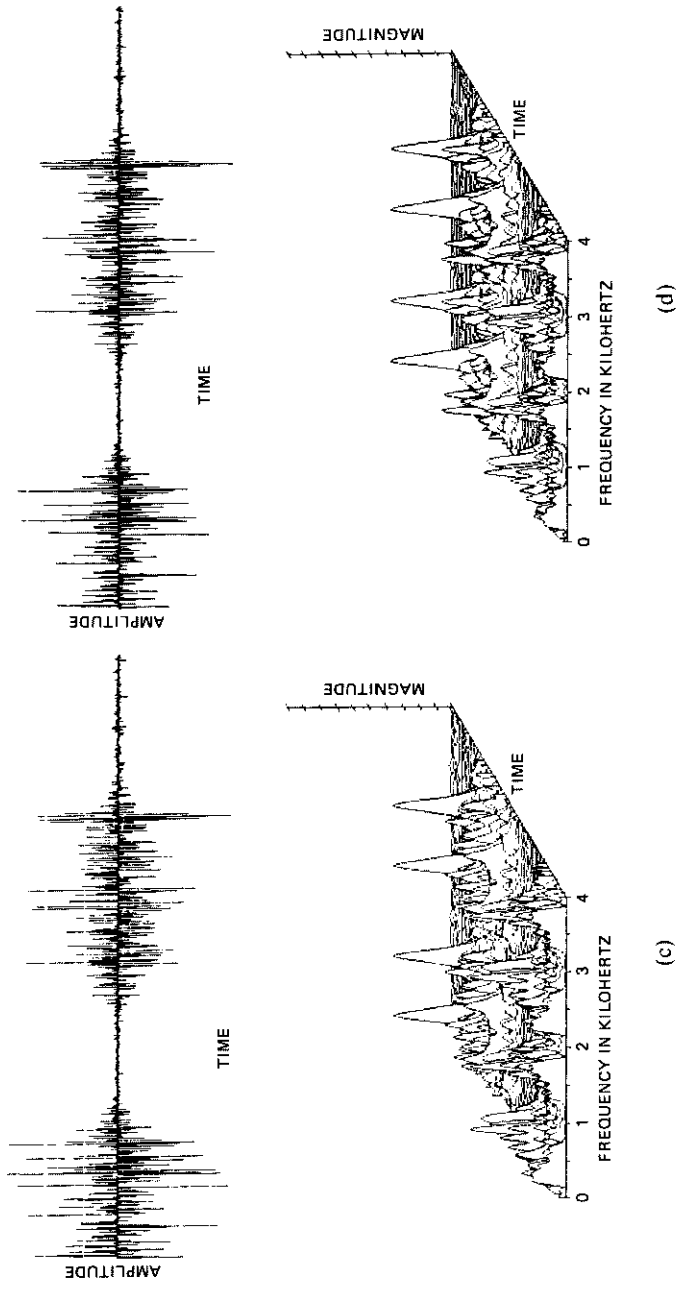
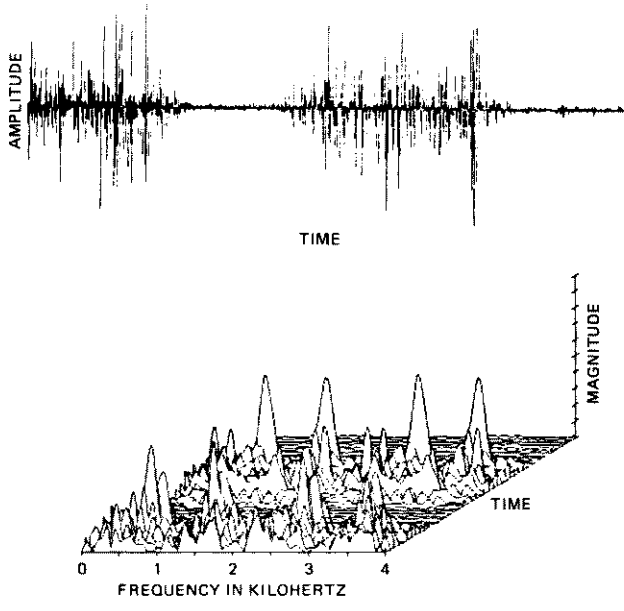



Fig. 11 (c) and (d)—Error waveforms and their spectrograms.
(Figure is continued on next page.)




(e)

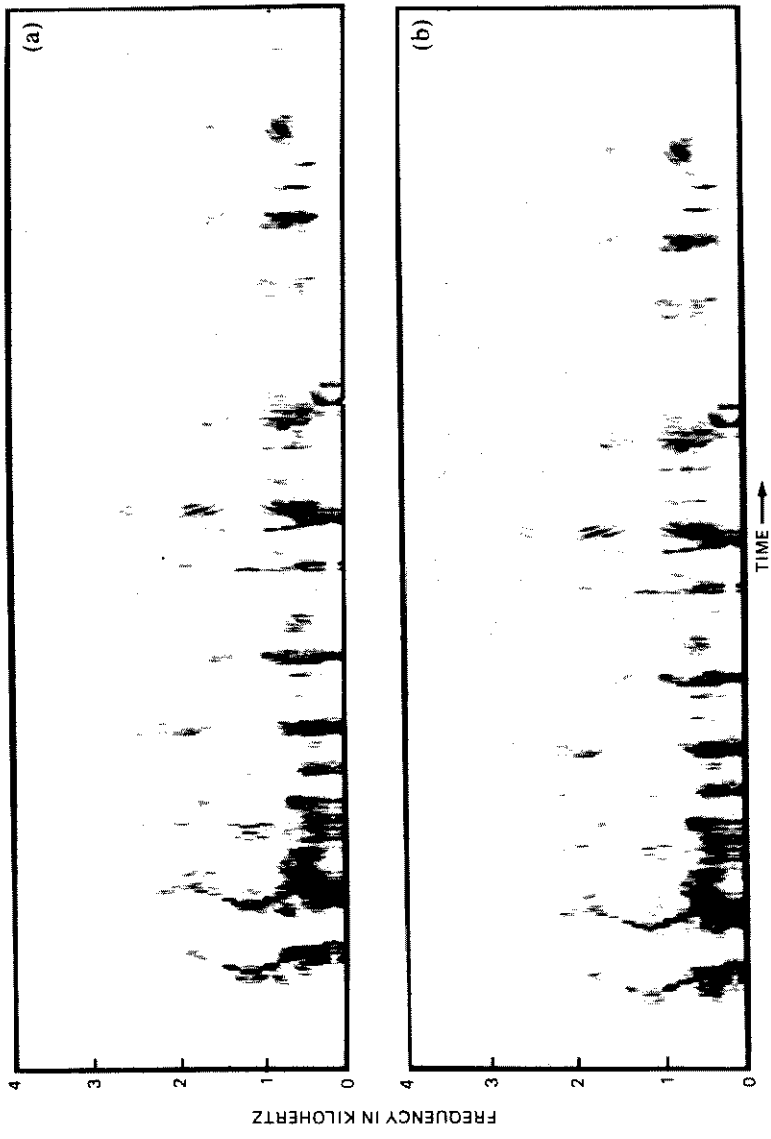
Fig. 11 (e)—Error waveforms and their spectrograms. The error waveforms in a, b, c, and d are the difference between the input speech segment of Fig. 10a and the recovered signals formulated using the interpolation method of: average of adjacent samples, λ-interpolation, $\lambda = 1$, $\lambda = 3$, and λ-with-iteration, $\lambda = 3$, respectively, $W = 256$. The waveform in e shows the effect of $\mu$-law PCM encoding the speech samples prior to interpolation, λ-with-iteration, $\lambda = 3$, $W = 256$.

iteration. The spectrograms of the error signals are displayed above these signals in Fig. 11.

As the error sequence $\{e_k\}$ is the difference between the original speech sequence $\{x_k\}$ and the recovered speech sequence $\{\hat{z}_k\}$, it is composed of $n - 1 = 3$ components of zero magnitude followed by a nonzero component due to the interpolation error. The components in the original and recovered speech sequences occur at a rate $f_s = 8$ kHz, while the components in $\{e_k\}$ are generated at $f_s/n = 2$ kHz in Fig. 11. Close examination of the error spectra in this figure shows a symmetry about 2 kHz over the range dc to 4 kHz, and further symmetries about 1 kHz and 3 kHz for the frequency ranges of dc to 2 kHz, and 2 kHz to 4 kHz, respectively.

The noise components above 3.3 kHz were removed by the output filter shown in Fig. 1. When this was done the improvement in speech quality was minimal, as can be anticipated from the spectrograms of Fig. 11. Also the SEG-s/n values shown in Fig. 7 were only increased by a fraction of a decibel.

Figures 12a through d show the error spectrograms obtained when


(a)
(b)
4
3
2
1
0
TIME
FREQUENCY IN KILOHERTZ

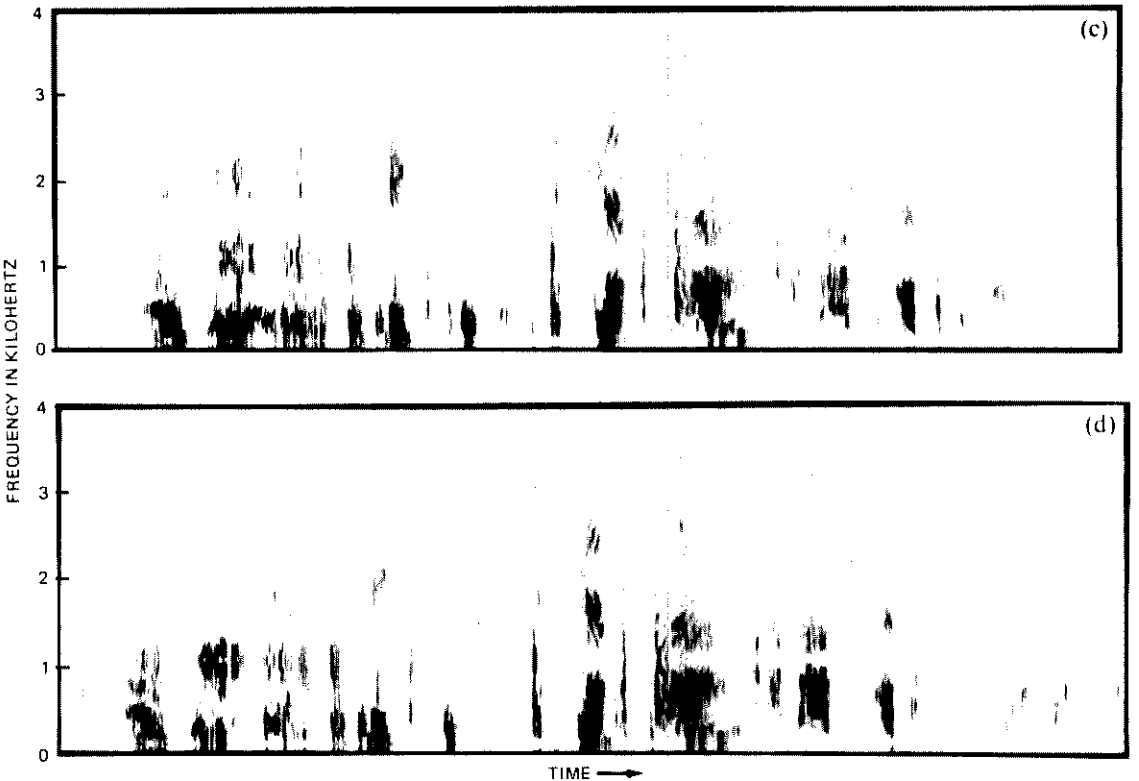



Fig. 12—Error spectograms for the two sentences shown in Fig. 3. The spectrograms a, b, c, and d relate to the conditions a, b, c, and d in Fig. 11, respectively.

the interpolation procedure used adjacent sample averaging, λ-interpolation with $\lambda = 1$, $\lambda = 3$, and λ-with-iteration, $\lambda = 3$ for the entire speech signal of Fig. 3. The spectrograms are not preemphasized. The effect of using $\lambda = 3$ compared to $\lambda = 1$ is to alter the nature of the noise spectrum, increasing its tendency to be more random. When we connected the error signal appertaining to the interpolation procedure having $\lambda = 3$ to a pair of earphones, it was found to be unintelligible and noise-like. With adjacent sample averaging the error signal was significantly more correlated with the original speech, and when we listened to this error signal the two sentences were comprehendible.

### 6.3 Variation of n

The effect of $n$ on SEG-s/n is displayed in Fig. 13 for a block size of 256. When the interpolation parameters were derived from the original speech sequence using the correlation function given by eq. (14), $\lambda = n - 1$, the SEG-s/n increased by 9 dB to 28.6 dB when $n$ was increased from 2 to 3, as shown in Fig. 13a. Thus, by rejecting every third sample we are able to reconstitute the speech by means of interpolation with only a slight perceptual impairment. For $n = 4$ the distortion in the interpolated speech was imperceptible. Curve b in Fig. 13 applies to λ-with-iteration, $\lambda = n - 1$. When $n = 2$ there is negligible difference between the two curves in Fig. 13, but as $n$ is increased the curves diverge and curve a maintains a minimum mean square interpolation error. We conclude from Fig. 13 that if the block size is 256, the recovered speech will have only minor impairments for $n = 3$ when the interpolation parameters are derived at the transmitter, and $n = 4$ is acceptable when the parameters are computed by the iterative procedure.

### 6.4 Adaptive interpolation results for PCM encoded speech

The sampled speech sequence $\{x_k\}$ was binary encoded by an 8-bit $\mu$-law PCM encoder having $\mu = 255$. From Figs. 4 through 13 we concluded that $n = 4$ was a good compromise, offering the prospect of an acceptable s/n while reducing the 64-kb/s transmission rate to 48 kb/s. With this bit-rate reduction, 16 kb/s of data can be added to the 48 kb/s of speech to give the conventional transmission rate. At the destination (or at some convenient point along the transmission path) the data can be removed, and each of the 6 k-words/s comprising the $\mu$-law PCM signal decoded. In our experiments we assumed that the bits would be regenerated without error, and the 8-kHz sampling rate established by reinserting the missing samples by the interpolation techniques previously described. However, the accuracy of the interpolation process was reduced by the quantization noise produced in the digital encoder.

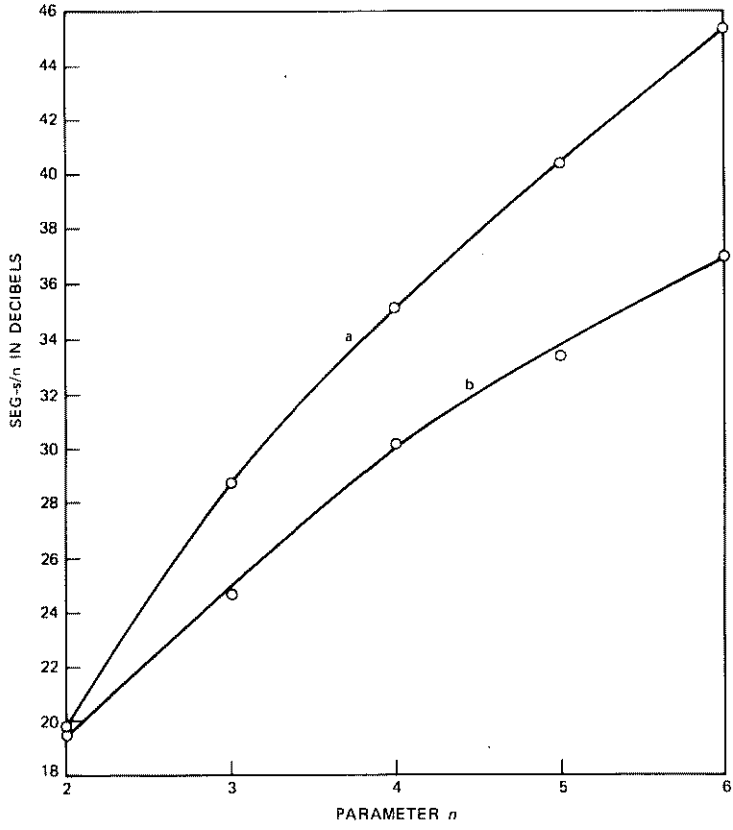



Fig. 13—Variation of SEG-s/n versus parameter $n$, for $\lambda = n - 1$, and the interpolation parameters derived from the original speech sequence, and by the $\lambda$-with-iteration scheme at the receiver. The block size is 256.

Deriving the interpolation parameters from the received data, as described in Section V, yielded the results shown in Fig. 14. The interpolation procedures applicable to curves a, b, c, and d were identical to those employed for curves a, b, c, and d, respectively, in Fig. 7. The respective curves in Fig. 14 are lower than those in Fig. 7 owing to the effect of quantization noise encountered in $\mu$-law PCM encoding. We observe that the quantization noise causes a loss in SEG-s/n of 0.5 dB when the interpolation is performed by adjacent sample averaging. For the cases of $\lambda = 1$ and $\lambda = 3$ the losses in SEG-s/n owing to the effect of quantization noise become approximately 1 and 1.7 dB, respectively, when $W = 256$. As the interpolation process improves, the SEG-s/n becomes relatively more affected by the quantization noise. We observe that the greatest loss in SEG-s/n relative to when there is no quantization noise occurs for the case of $\lambda$-with-

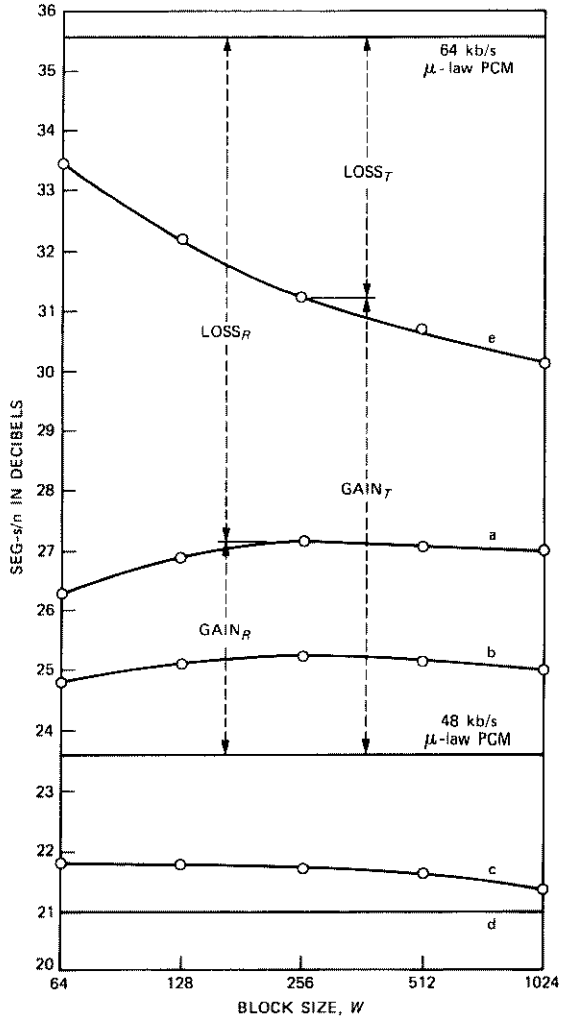



Fig. 14—8-bit $\mu$-law PCM encoding, $\mu = 255$. The effect of block size on SEG-s/n for different conditions. Curves a, b, c, and d apply for $\lambda$-with-iteration, $\lambda = 3$; $\lambda$-interpolation, $\lambda = 3$; $\lambda$-interpolation, $\lambda = 1$; and average of adjacent sample interpolation, respectively. Curve e relates to the interpolation parameters derived at the transmitter using $\lambda = 3$ and $R(k, j)$.

iteration, $\lambda = 3$, and is 2.7 dB for $W = 256$. Also shown in Fig. 14 are the SEG-s/n values for the encoded speech when the sampling rate is 8 kHz, $\mu = 255$, and the number of bits per code word is 8 and 6, i.e., the transmission bit rates are 64 kb/s and 48 kb/s, respectively. The loss in SEG-s/n due to transmitting at only 48 kb/s and reinserting the discarded samples using $\lambda$-with-iteration, $\lambda = 3$, compared to not discarding samples and transmitting at 64 kb/s is given by $LOSS_R$ in

Fig. 14. The subscript $R$ indicates that the interpolation parameters were derived from the received data. The gain in SEG-s/n, $GAIN_R$, is due to interpolating the missing samples compared to the 48 kb/s $\mu$-law PCM encoding where no samples are discarded. Although the 48 kb/s has the 8-kHz word rate, the quantization noise is higher because there are 6 bits per code word. The values of $LOSS_R$ and $GAIN_R$ for $W = 256$ are 8.4 and 3.6 dB, respectively.

The interpolation performance can be enhanced if the interpolation parameters are derived from the locally decoded $\mu$-law PCM signal at the transmitter. The interpolation parameters must be conveyed to the receiver in a binary format as side information. This is not a difficult task in a digital transmission system as fewer than 30 bits per $W$ PCM words are required to be transmitted. When $W$ is 256, the side information increases the bit rate by 2 percent. The bit rate can be maintained at 48 kb/s if the side information replaces the least significant bit in every sixth word transmitted. This will have only a marginal effect on the quality of the recovered speech. When the interpolation parameters for $\lambda = 3$ were determined at the transmitter using eqs. (11) through (14) and subsequently transmitted as side information, curve e in Fig. 14 was obtained. Comparing curve a in Fig. 6 with this curve shows that the presence of quantization noise reduces the SEG-s/n of the interpolated speech signal by 4 dB. Nevertheless, curve e is significantly higher than curve a, and the disparity increases to 7 dB for $W = 64$. The $LOSS_T$ and $GAIN_T$ factors, where the subscript $T$ implies the generation of the parameters at the transmitter, had values of 4.3 and 7.7 dB, respectively, for $W = 256$. The effect of interpolation is equivalent to saving more than one bit per word.

For $\mu$-law PCM encoding and the adaptive interpolation procedure of $\lambda$-with-iteration, $\lambda = 3$, $W = 256$, resulted in the error waveform and its spectrogram displayed in Fig. 11e for the speech segments shown in Fig. 10. Figures 11e and d show that the effect of quantization on the error spectrum is small.

## VII. DISCUSSION

Our intention at the outset of this investigation was to discard one speech sample (or PCM word) in every four, $n = 4$, and to replace the missing samples or words by an interpolation process such that the degradation in speech quality was virtually imperceptible. Further, the implementation algorithm was to be inherently simple. These goals have been reached in good measure.

The central issue in any interpolation process is determining the interpolation parameters. Our approach is to attempt to minimize the mean square error, a nonoptimum procedure for speech signals where

the perception of interpolation noise may be modified by the spectral composition of the speech signal over some 20 ms interval, and temporal effects lasting approximately 200 ms. The justification of the mean square error is based on simplicity, and for $n = 4$ gives good results. In deriving the interpolation parameters of eq. (11) based on $R(k, j)$, we made no assumptions concerning the statistic of the speech signal. The selection of block size $W$ depends upon an acceptable s/n, the need to avoid excessive signal delays resulting from too high a value of $W$, and the amount of side information permitted, where appropriate, when $W$ is small. Our suggested range of $W$ is from 64 to 1024 (see Figs. 4 and 7). These values of $W$ correspond to durations of 8 to 128 ms, i.e., ranging from approximately a pitch to a syllable period. Computing the interpolation parameters using eqs. (11) and (14), we were able to achieve gains in s/n of 16, 14, and 12 dB compared to interpolation using nearest neighbor linear interpolation for block sizes of 64, 256, and 1024, respectively.

When the estimate of the autocorrelation function could not be based on the original speech sequence, but had to be estimated from the received speech samples where every $n$th sample was missing, an iterative estimation procedure was employed. By making a crude estimation of the missing samples, the autocorrelation function $R(\tau)$ was computed, and in general a more accurate set of interpolated samples were found. The autocorrelation function was again determined, and the accuracy of the interpolated samples nearly always improved. By this iterative approach, for $n = 4$, $\lambda$-with-iteration, $\lambda = 3$ had an interpolation gain over nearest neighbor averaging of 8 dB for $W = 256$, as displayed in Fig. 7.

The effect of discarding every $n$th sample, $n = 2, 3, 4$, and 5, showed that it is advisable to maintain $n \geq 4$ for imperceptible perceptual degradation. For highly correlated sounds and where some masking of the interpolation noise occurs, we can satisfactorily deploy $n = 3$ and even $n = 2$. However, in general there is considerable distortion power when $n = 2$, which is hardly surprising as half the samples had been rejected. Nevertheless, for $n = 2$ adaptive interpolation yielded a s/n of approximately 19.5 dB, which is noisy but intelligible speech.

Finally, we found that when conventional 8-bit $\mu$-law PCM encoded speech, $\mu = 255$, had its bit rate reduced from 64 kb/s to 48 kb/s to enable 16 kb/s of other data to be transmitted, the recovered speech after decoding and interpolation had a s/n that approximated that of 56-kb/s $\mu$-law PCM.

## VIII. ACKNOWLEDGMENTS

The authors thank D. Vitello for her advice with computer simulation, and D. J. Goodman for his constructive criticism of this work.

## REFERENCES


1. B. G. Haskell and R. Steele, "Audio and Video Bit-Rate Reduction," Proc. IEEE, *69*, No. 2 (February 1981), pp. 252–62.
2. J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, and J. M. Tribolet, "Speech Coding," IEEE Trans. Commun., *COM-27* (April 1979), pp. 710–37.
3. M. V. Mathews, "Extremal Coding for Speech Transmission," IRE Trans. Inform. Theory, *IT-5* (September 1959), pp. 129–36.
4. C. M. Kortman, "Redundancy Reduction—A Practical Method of Data Compression," Proc. IEEE, *55*, No. 3 (March 1967), pp. 253–63.
5. L. Ehrman, "Analysis of Some Redundancy Removal Bandwidth Compression Techniques," Proc. IEEE, *55*, No. 3 (March 1967), pp. 275–87.
6. C. A. Andrews, J. M. Davis, and G. R. Schwarz, "Adaptive Data Compression," Proc. IEEE, *55*, No. 3 (March 1967), pp. 267–77.
7. N. S. Jayant, "Adaptive Aperture Coding for Speech Waveforms-1," B.S.T.J., *58*, No. 7 (September 1979), pp. 1631–45.
8. H. S. Hou and H. C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," IEEE Trans. ASSP, *ASSP-26*, No. 6 (December 1978), pp. 508–17.
9. "Special Issue on Bit-Rate Reduction and Speech Interpolation," IEEE Trans. Commun., *COM-30*, No. 4 (April 1982), pp. 728–80.
10. N. S. Jayant, "Effect of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure," IEEE Trans. Commun., *COM-29*, No. 2 (February 1981), pp. 101–9.
11. R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals—A Tutorial Review," Proc. IEEE, *69*, No. 3 (March 1981), pp. 300–31.
12. P. Noll, "Adaptive Quantization in Speech Coding Systems," Int. Zurich Seminar, Zurich, Switzerland, April 1974.


## AUTHORS


**Frank Benjamin**, B.A. (Music Education), 1980, B.S. (Electronic Engineering), 1983, Valedictorian (both cum laude), Monmouth College, West Long Branch, NJ; Monmouth College, 1980-1981; United Telecontrol Electronics, 1981; Bell Laboratories, 1981—. While Mr. Benjamin was attending college, he worked as a recording engineer and studio musician on albums for RCA-Columbia and EPIC studios, and on promotional sound tracks for the motion picture *STAR TREK*.® He taught privately in the fields of music, engineering, mathematics and physics. At Monmouth College he worked as a laboratory instructor and acoustical consultant. While at United Telecontrol Electronics, he helped develop a missile guidance control system. In 1981 he joined Bell Laboratories as a member of the Radio Communications Research Department, designing and writing software simulation systems for speech compression and interpolation techniques. Past President, Lambda Sigma Tau; member, Eta Kappa Nu; nominee, Who's Who Among Students in American Universities and Colleges; New Jersey State Teacher's Certificate.

**Raymond Steele**, (SM '80), B.S. (Electrical Engineering) from Durham University, Durham, England, in 1959, and the Ph.D. degree in 1975. Prior to his enrollment at Durham University, he was an indentured apprenticed Radio Engineer. After research and development posts at E. K. Cole Ltd., Cossor Radar and Electronics, Ltd., and The Marconi Company, all in Essex, England, he joined the lecturing staff at the Royal Naval College, Greenwich, London, England. Here he lectured in telecommunications to NATO and the External London University degree courses. His next post was as Senior Lecturer in the Electronic and Electrical Engineering Department of Loughborough University, Loughborough, Leics., England, where he directed a research group in digital encoding of speech and television signals. In 1975 his book, *Delta*

*Modulation Systems* (New York: Halsted), was published. He was a consultant to the Acoustics Research Department at Bell Laboratories in the summers of 1975, 1977, and 1978, and in 1979 he joined the company's Communications Methods Research Department, Crawford Hill Laboratory, Holmdel, NJ. In 1981 Mr. Steele was given *The Bell System Technical Journal* Best Paper Award in the category of Mathematics, Communication Techniques, Computing and Software, and Social Sciences. In 1983 he joined The University of Southampton, Southampton, England, as a Professor of Communications.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

# Practical Design Considerations for Coupled-Single-Amplifier-Biquad Active Bandpass Filters

By J. TOW*

(Manuscript received December 10, 1982)

This paper presents a synthesis method and practical design considerations for the Coupled-Single-Amplifier-Biquad (CSAB) realization of all-pole symmetrical bandpass (BP) filters. The CSAB topology consists of a cascade of second-order SAB bandpass sections, together with negative feedback around adjacent sections. A straightforward procedure that leads to the block diagram representation of the CSAB is shown. Explicit design formulas are given for the optimum element values of the Deliyannis-Friend SAB bandpass section, as well as for the feedback resistors. This CSAB design offers improved performance over the cascade SAB approach without using additional operational amplifiers. Also described are the effects on the filter response due to finite amplifier gain, capacitor dissipations, noninfinite pole-$Q$ sections, and their compensation techniques. These are followed by discussions on maximizing the filter dynamic range and tuning.

## I. INTRODUCTION

It is commonly known that in the realization of high-order active filters, properly designed multiple-loop-feedback topologies offer far superior sensitivity performance than the approach of cascading biquadratic filter blocks.[1-6] The multiple-loop-feedback structure is particularly useful in the design of bandpass filters, where reduced sensitivity design is most often needed. Compared with the cascade biquad approach, two drawbacks are usually attributed to these topologies,

* Bell Laboratories, Holmdel, N.J.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

namely, the more complicated design procedures and the use of more op amps. However, for the class of all-pole geometrically symmetrical Bandpass (BP) filters, these drawbacks do not exist for a particular multiple-loop-feedback structure commonly referred to as leap-frog, active-ladder, or coupled-biquad.[7–13] The coupled-biquad topology consists of a cascade of second-order sections together with negative feedback around adjacent biquad sections.

Most of the existing coupled-biquad descriptions assumed multiple op-amp biquads, but details on the use of Single-Amplifier Biquads (SABs) are scarce. Our discussion here focuses on the use of SAB and is further restricted to voice-frequency applications. Note that the Coupled-Single-Amplifier-Biquad (CSAB) topology described here requires $n$ op amps for a $2n$-order BP filter.

The first part of this paper presents a general design method for the CSAB realization of all-pole symmetrical BP filters. A straightforward procedure is given that leads to the block diagram representation of the CSAB. Each of the second-order sections is then implemented by the Deliyannis-Friend[14] SAB configuration. Optimum element values for these SABs are computed according to Fleischer's results.[15] The second part of the paper discusses the effects on the filter response due to finite op-amp gain, capacitor dissipations, noninfinite pole-$Q$ sections, and their compensation techniques. These are followed by discussions on maximizing the filter dynamic range and tuning.

## II. CSAB DESIGN PROCEDURE

This section presents a straightforward design procedure for the CSAB realization of all-pole symmetrical BP filters. Optimum design equations are given for the Deliyannis-Friend SAB bandpass section and the feedback resistors.

### *2.1 Block diagram representation of CSAB configuration*

The starting point for the CSAB realization of an all-pole symmetrical BP filter, e.g., a Bessel-, Butterworth-, or Chebychev-type BP filter, is its normalized low-pass (LP) prototype ladder configuration. The ladder configuration is readily available from many existing handbooks and is shown in Fig. 1.

Let $\omega_0$ = center frequency of the BP filter (in rad/s)

$B$ = passband bandwidth of the BP filter (in rad/s)

A block diagram representation of the CSAB topology for the BP filter is given in Fig. 2, where the coupled biquadratic transfer functions,

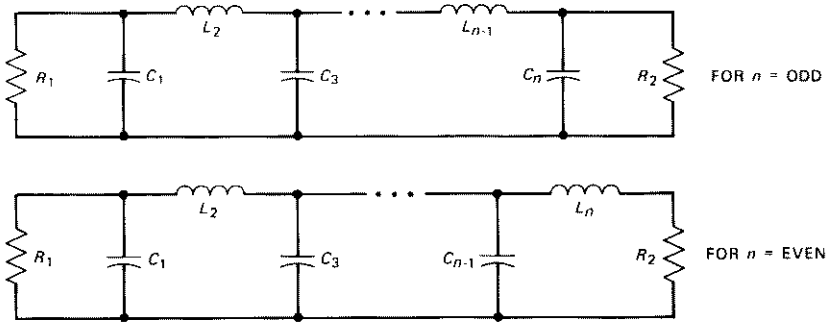



Fig. 1—LP prototype ladder configuration.

$T_i(s)$ and $T_i'(s)$, are related to the element values of the LP ladder by the following equations:

$$T_1(s) = K \cdot \frac{\dfrac{R_1 + R_2}{R_2} \cdot \dfrac{B}{R_1 C_1}\, s}{s^2 + \dfrac{B}{R_1 C_1}\, s + \omega_0^2} \tag{1}$$

$$T_1'(s) = \frac{\dfrac{B}{C_1}\, s}{s^2 + \dfrac{B}{R_1 C_1}\, s + \omega_0^2} \tag{2}$$

$$T_i(s) = T_i'(s) = \frac{\dfrac{B}{X_i}\, s}{s^2 + \omega_0^2} \qquad i = 2, 3, \cdots, n-1 \tag{3}$$

$$\text{and} \quad X_i = \begin{cases} L_i \text{ for } i \text{ even} \\ C_i \text{ for } i \text{ odd} \end{cases}$$

$$T_n(s) = \begin{cases} \dfrac{\dfrac{B}{C_n}\, s}{s^2 + \dfrac{B}{R_2 C_n}\, s + \omega_0^2} & \text{for } n \text{ odd} \\ \dfrac{\dfrac{B}{L_n}\, s}{s^2 + \dfrac{B R_2}{L_n}\, s + \omega_0^2} & \text{for } n \text{ even,} \end{cases} \tag{4}$$

where $K$ determines the overall gain of the filter. For unity (0 dB) voltage gain, set $K = 1$.

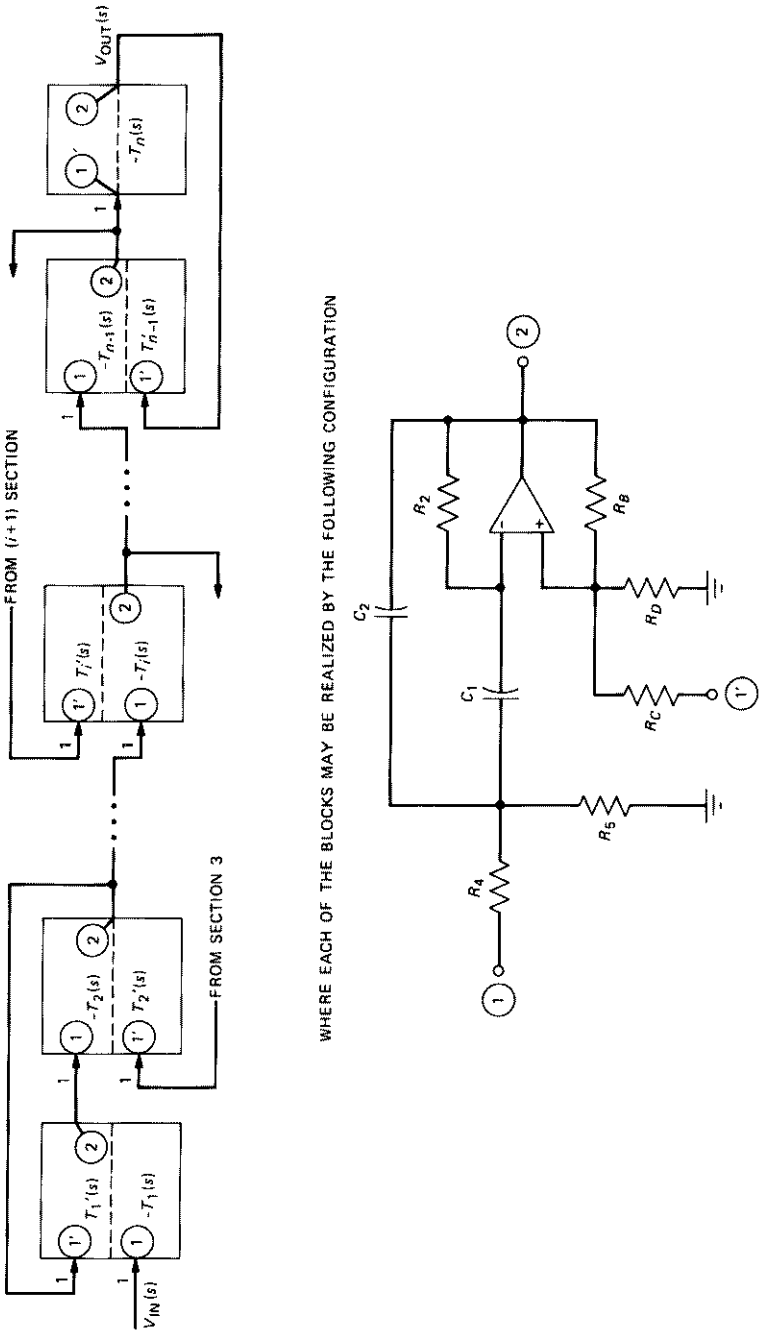



Fig. 2—Coupled SAB configuration for all-pole symmetrical BP filters.

The above derivation is straightforward and is omitted here. For the three-section case, discussions can be found in pages 726 to 729 of Ref. 3 and pages 348 to 351 of Ref. 6. Note that in Fig. 2, the Deliyannis-Friend SAB configuration is also shown. Besides its good sensitivity properties, this particular SAB can simultaneously realize both the forward and the feedback paths as required by Fig. 2.

One final step in the CSAB realization is to obtain the element values for the individual SAB blocks. The optimum formulas are given in the next section.

### *2.2 Optimum CSAB element values*

For each of the second-order blocks shown in Fig. 2, let the forward and feedback voltage transfer functions be represented by:

$$\frac{V_2(s)}{V_1(s)} = -T_i(s) = \frac{-n_1 s}{s^2 + d_1 s + d_0} \tag{5}$$

and

$$\frac{V_2(s)}{V_1'(s)} = T_i'(s) = \frac{n_1' s}{s^2 + d_1 s + d_0}. \tag{6}$$

As shown in Fig. 2 and eqs. (1) to (3), the internal sections, i.e., $i = 2, 3, \cdots, n-1$, have an infinite pole-$Q$ value or $d_1 = 0$. For improved filter performances, a later section suggests the use of a very high-$Q$ value, say several hundreds, instead of the infinite value. This corresponds to the use of a small value for $d_1$ in eqs. (5) and (6). The high but fixed $Q$-value can be designed into the CSAB configuration by the familiar predistortion technique.

Element values are first obtained for the forward transfer function, $T_i(s)$, by the formulas given in Ref. 14, where the conductance value of a fictitious resistor, $R_1$, is first computed and the values of $C_1$, $C_2$, $R_A$, and $R_B$ can be conveniently chosen such that

$$G_1 = \frac{C_2 R_B}{2R_A}\left\{-d_1 + \left[d_1^2 + 4d_0\left(1 + \frac{C_1}{C_2}\right)\frac{R_A}{R_B}\right]^{1/2}\right\} \tag{7}$$

$$R_2 = \frac{G_1}{C_1 C_2 d_0} \tag{8}$$

$$R_4 = \frac{1 + \dfrac{R_A}{R_B}}{n_1 C_2} \tag{9}$$

$$R_5 = \frac{1}{G_1 - G_4} \tag{10}$$

$$R_B = \text{arbitrary [can be chosen according to eq. (15)]} \tag{11}$$

$$R_D = R_A \tag{12}$$

$$R_C = \text{infinite.} \tag{13}$$

As shown by Fleischer,[15] the value of $R_B$ or the ratio of $R_A/R_B$ should be chosen so as to minimize the overall SAB variability (sensitivity) due to the combined active and passive elements variations. His results are given below:*

$$Q_0 = \left[ |A(s_0)|^2 \frac{8\sigma_R^2 + \sigma_C^2}{8\sigma_{A(s_0)}^2} \right]^{1/4} \tag{14}$$

$$\frac{R_A}{R_B} = \frac{C_2}{C_1 + C_2} \frac{1}{Q_0} \left( \frac{1}{Q_0} - \frac{d_1}{\sqrt{d_0}} \right), \tag{15}$$

where $A(s)$ represents the gain of the op amp, $\sigma_R^2$, $\sigma_C^2$, and $\sigma_{A(s)}^2$ correspond to the variances of $\Delta R/R$, $\Delta C/C$, and $\Delta A(s)/A(s)$, respectively, and $s_0$ is the pole location of the particular SAB transfer function.

The values of $R_B$ and $R_C$ are next modified according to the following formulas to implement the feedback transfer function as well [see Fig. 2 and eq. (6)]:†

$$K_F = \frac{n_1'}{\left( \frac{G_1 + G_2}{C_2} + \frac{G_2}{C_1} \right)} \tag{16}$$

$$\text{(new) } R_C = \frac{1 - K_F}{K_F} R_D \tag{17}$$

$$\text{(new) } R_B = \frac{R_B R_C}{R_C + R_D} \tag{18}$$

$$R_D = R_A \text{ (as before).} \tag{19}$$

In summary, the optimum element values can be computed from eqs. (14), (15), (7) to (12), and (16) to (18).

---

* Equations (57), (61), (13b), (2) and (7) of Ref. 15 were used. For the case of $C1 = C2$ and single-pole op-amp characteristics, simpler formulas are given in page 323 of Ref. 2.

† The derivation is given in the appendix. Note that, as shown in pages 362 to 364 of Ref. 6, the feedback transfer function realized is not exactly a BP function but has negative and real transmission zeros. However, in the vicinity of the pole frequency, where the effect of feedback is of interest, the function behaves like a BP function. In any event, the error introduced is negligible.

## III. PRACTICAL DESIGN CONSIDERATIONS

The preceding section describes one practical design consideration, namely, that component statistics are used to obtain an optimum design parameter, $R_A/R_B$, for the individual SAB blocks. This section further discusses some important factors that cause the SAB or CSAB response to deviate from its ideal characteristic. Compensations in the form of predistortion techniques are described. These compensations can be achieved by modifying the pole locations of the individual second-order blocks as given by eqs. (5) and (6). In a good design, they must be considered as part of the original design and optimum element values are to be computed from the modified transfer functions.

### *3.1 Nonideal op amp characteristics*

For a given SAB realizing the transfer functions (5) and (6), the major effect of finite op-amp gain is to shift the desired pole location, or the roots of

$$s^2 + d_1 s + d_0 = s^2 + \frac{\omega_0}{Q} s + \omega_0^2 = (s - s_0)(s - s_0^*) \tag{20}$$

from $s_0$ to $s_0 + \Delta s_0$. Fleischer[15] has shown that this deviation is approximately given by

$$\Delta s_0 = -\left(1 + \frac{R_A}{R_B}\right) \frac{1}{A(s_0)} \frac{D_2(s_0)}{s - s_0^*}, \tag{21}$$

where

$$D_2(s) = s^2 + \left(\frac{C_1 + C_2}{C_1 C_2} \frac{1}{R_2} + \frac{G_1}{C_2}\right) s + \frac{G_1}{R_2 C_1 C_2}. \tag{22}$$

In the actual design, one can apply the negative of this shift to the nominal transfer function. Equivalently, as shown in pages 415 to 416 of Ref. 2, the shift in $\Delta s_0$ can be represented by:

$$\frac{\Delta \omega_0}{\omega_0} = Re\left(\frac{\Delta s_0}{s_0}\right) \tag{23}$$

$$\frac{\Delta Q}{Q} = -2QIm\left(\frac{\Delta s_0}{s_0}\right). \tag{24}$$

Hence, instead of using the design parameters $\omega_0$ and $Q$ in eq. (20), one can use the modified parameters $\omega_0' = \omega_0 - \Delta\omega_0$ and $Q' = Q - \Delta Q$.

As pointed out by Fleischer, eq. (21) shows that the fractional change in the complex pole location is inversely proportional to the magnitude of the amplifier gain at the pole frequency. Hence, the use of a two-pole, one-zero compensated op amp would provide at least an

order of magnitude smaller pole shift magnitude than that obtained from using a single-pole compensated op amp in most of the audio frequency band.* For many practical voice-frequency-band applications, this pole shift is quite small for the former compensation and its effect can often be ignored.

### *3.2 Nonideal capacitor characteristics*

A nonideal capacitor is usually associated with a finite dissipation factor or tan $\delta$. It is commonly represented by introducing a resistor (with conductance $G_i$) in parallel with the capacitor $C_i$, where

$$G_i = \omega_0 C_i \tan \delta_i$$

and $\omega_0$ corresponds to the pole frequency of the SAB block, since we are particularly interested in the variations of the transfer function for frequencies near $\omega_0$.

As with the op-amp finite gain, the major effect of the capacitor dissipation factor is to cause a shift in the desired pole locations, which again can be compensated for. Weyten's approach[16] is described here. For a two-capacitor biquad section realizing eq. (5), Weyten has shown that

$$\frac{\Delta d_1}{d_1} = Q(\tan \delta_1 + \tan \delta_2) \tag{25}$$

$$\frac{\Delta d_0}{d_0} = \tan \delta_1 \tan \delta_2 - \frac{1}{Q}\left(S_{C_1}^{d_1} \tan \delta_2 + S_{C_2}^{d_1} \tan \delta_1\right), \tag{26}$$

where

$$-S_{C_i}^{d_1} = \tfrac{1}{2} + S_{C_i}^{Q} \tag{27}$$

and the coefficients in eq. (20) move from $d_i$ to $d_i + \Delta d_i$. For the SAB under consideration (Fig. 2), Ref. 15 gives

$$S_{C_1}^{Q} = -S_{C_2}^{Q} = \frac{C_2}{C_1 + C_2}\frac{Q}{Q_0} - \frac{1}{2}, \tag{28}$$

where $Q_0$ is as given before.

Hence, instead of using the design parameters $d_1$ and $d_0$ in eqs. (5) and (6), the modified parameters $d_1' = d_1 - \Delta d_1$ and $d_0' = d_0 - \Delta d_0$ can be used. Note that the fractional change of $d_0$ is usually very small and can practically be ignored. Alternately, eqs. (25) and (26) can be rewritten as

$$\frac{\Delta Q}{Q} \simeq -\frac{\Delta d_1}{d_1} = -Q(\tan \delta_1 + \tan \delta_2) \tag{29}$$

* See for example, pages 84 to 85 of Ref. 6.

$$\frac{\Delta\omega_0}{\omega_0} \simeq \frac{1}{2} \tan \delta_1 \tan \delta_2 - \frac{1}{2Q} (S_{C_1}^{d_1} \tan \delta_2 + S_{C_2}^{d_1} \tan \delta_1) \qquad (30)$$

since the variations of $\omega_0$ are usually very small. The decrease in pole-$Q$ value due to capacitor dissipation factors is usually appreciable and should be compensated for. As an illustration, for a medium-$Q$ BP section, say $Q = 20$, and for a capacitor dissipation factor of 0.0015, such as that encountered with thin-film capacitors, this fractional change in the pole-$Q$ value is 6 percent.

### *3.3 Infinite pole-Q sections*

As we see in Fig. 2, all of the second-order sections except the first and the last have an infinite pole-$Q$ value, i.e., $d_1 = 0$ in eq. (5). Note that the negative feedbacks in the coupled-biquad configuration move these poles away from the $j\omega$-axis and into the desired pole locations. Except for very high-$Q$ BP filters, realization of the infinite pole-$Q$ sections in the CSAB configuration is not very critical as long as the value of these pole-$Q$s is high, say several hundreds. The effects of using a high but finite pole-$Q$ value in these sections are a lower overall gain of the filter and a reduction of the effective passband bandwidth.[17] A decrease in gain is easily compensated for in active filter design, while the reduced passband bandwidth may usually be absorbed in the original design margin.

On the other hand, a closer approximation to the desired response is obtained if each of the internal sections is a priori designed to have a high but finite pole-$Q$ value. This value can be chosen to be the highest and practically realizable pole-$Q$ value, say $Q_M$. For the SAB, this is in the order of a few hundreds. Note that in the actual realization, these pole-$Q$ values will deviate a great deal (higher or lower than $Q_M$). Computer simulations have shown that the overall circuit variability is smaller for the finite internal pole-$Q$ design than the design with infinite pole-$Q$s.

Having presented the virtues of noninfinite internal pole-$Q$ sections, we show now how this can be achieved with the classical predistortion technique. If we refer back to Section 2.1, instead of starting with the ladder configuration, we see that the overall transfer function for the normalized LP prototype is used (again this is available from many handbooks). A shift $\alpha$ is introduced to the complex frequency variable $s$, $s = p - \alpha$, where $p$ represents the new complex frequency variable, and

$$\alpha = \frac{\omega_0}{BQ_M}. \qquad (31)$$

In eq. (31) all variables are as defined before. The new transfer function

in $p$ is used to realize the ladder configuration shown in Fig. 1. With the following modifications to eqs. (1) through (4), where $R_1'$ and $R_2'$ are used in place of $R_1$ and $R_2$, respectively, the CSAB configuration is again as shown in Fig. 2:

$$R_1' = \frac{1}{\frac{1}{R_1} + \alpha C_1} \tag{32}$$

$$R_2' = \begin{cases} \dfrac{1}{\frac{1}{R_2} + \alpha C_n} & \text{for } n \text{ odd} \\ R_2 + \alpha L_n & \text{for } n \text{ even} \end{cases} \tag{33}$$

$$T_i(s) = T_i'(s) = \frac{\frac{B}{X_i}\, s}{s^2 + \frac{\omega_0}{Q_M}\, s + \omega_0^2} \qquad i = 2, 3, \cdots, n-1. \tag{34}$$

In addition, the value of $K$ in eq. (1) must be modified to obtain the desired overall gain of the filter. This value is easily determined by computing the gain of the ladder in Fig. 1 after replacing each inductor $L_i$ with a resistor of value $\alpha^* L_i$ and each capacitor $C_i$ with a resistor of conductance $\alpha^* C_i$.

The ladder realization of the shifted LP prototype transfer function can be obtained by the classical synthesis technique, or with a filter synthesis program.[18] For best sensitivity performance, this ladder must correspond to the maximum power transfer design.[19]

### *3.4 Maximizing the filter dynamic range*

A rule-of-thumb design procedure for maximizing the dynamic range of a high-order filter is to make the maximum voltage output level at the various amplifiers equal. Variations among the various amplifier outputs in the CSAB are generally much smaller than in the corresponding cascade SAB design. For many applications, one may find the CSAB design as obtained before to be satisfactory.

The maximum voltage output levels at the amplifiers usually occur at frequencies within the filter passband or the transition bands. A practical procedure to maximize the filter dynamic range is to evaluate the filter frequency responses (via a computer program) at each of the amplifier outputs and at a set of discrete frequencies chosen from the passband and transition bands. The maximum output values thus obtained are used to rescale the gain of each of the transfer functions given by eqs. (1) through (4). More formally, let

$$G_i = \text{Max}(V_i) - \text{Max}(V_{\text{out}}) \text{ in dB},$$

where $V_i$ (in dB) is the voltage level at the output of the $i$th amplifier and $\text{Max}(V_i)$ is the maximum value over the chosen set of frequencies.

Compute

$$H_i = 10^{-\frac{G_i}{20}} \qquad i = 1, 2, \cdots, n-1$$

with $H_0 = H_n = 1$.

Multiply each of the forward transfer functions, $T_i(s)$ by $K_i$, where

$$K_i = H_i/H_{i-1} \qquad i = 1, 2, \cdots, n,$$

and multiply each of the feedback transfer functions, $T_i'(s)$, by $K_i'$, where

$$K_i' = H_i/H_{i+1} \qquad i = 1, 2, \cdots, n-1.$$

### *3.5 CSAB tuning*

When the STAR realization[14] is used to implement the SAB, the manufacturing tuning procedure is to measure values of the two capacitors on the substrate and then compute the resistor values, based on these measurements, from the predistorted transfer function using the equations given in Section 2.2. The resistors are laser-trimmed to these values. The individual SAB blocks are then connected as shown in Fig. 2. In general, this procedure is sufficient for all practical purposes.

For extremely high-precision filter applications where functional tuning may be desirable, the SAB, like any other single-amplifier-biquad configuration, does not exhibit an orthogonal set of tuning parameters. However, when the desired tuning range is small, e.g., during final mop-up trimming, the following tuning sequence is suggested: For the forward BP transfer function, with $R_C$ connected to ground, use $R_4$ to adjust for the gain at the pole frequency, $R_5$ for the pole-$Q$, and then $R_2$ for the pole frequency. The adjustments for the pole-$Q$ and pole frequency can be monitored by the 45-degree phase-shift points and the 180-degree phase-shift point, respectively. Finally, with $R_4$ connected to ground, the feedback factor can be adjusted by $R_C$. As in any coupled-biquad configuration, the CSAB overall response is relatively insensitive to these feedback resistors.

## IV. BP FILTERS WITH FINITE TRANSMISSION ZEROS

Extensions of the CSAB design to BP filters with finite transmission zeros, e.g., elliptic-type BP filters, are available. There are two approaches here. The first is to realize these transmission zeros within the individual SAB blocks;[11,12] however, their design procedures are

rather complicated. A second approach is to form these transmission zeros by a weighted sum of the individual SAB BP sections (Fig. 2) with an additional summing amplifier (in a manner analogous to the feedforward technique described in Ref. 9). Simple design formulas exist for this purpose. The second approach is particularly useful and exhibits excellent sensitivity properties for low-order filters, say fewer than five sections.

## V. CONCLUSIONS

A straightforward design procedure is given for the coupled-single-amplified-biquad realization of high-order, all-pole, symmetrical BP filters. Practical limitations and their compensation techniques are discussed. Many of these considerations, i.e., optimum choice of element values, nonideal op-amp and capacitor characteristics, are the same for the cascade or coupled designs. With this in mind, the CSAB design procedure is seen to be not more (if not less) complicated than the cascade design, since the individual second-order transfer functions are more readily computed.

The CSAB approach uses the same number of op amps as that of the cascade SAB approach. This number is equal to $n$ for a $2n$-order BP filter and is approximately half the number required by the many inductance simulation techniques, e.g., the two-op-amp Generalized Impedance Converter (GIC) designs. Sensitivity performances of the coupled-biquad may be considered as the best among all the various multiple-loop-feedback topologies[20] and are far superior to the cascade biquad. These observations, together with the fact that the Deliyannis-Friend SAB, from a sensitivity point of view, is as good as any other circuit in the audio frequency band,[15] suggest that the CSAB described here should be the choice for the design of low to medium-high $Q$ (say, $Q < 60$) BP filters in the audio frequency band.*

## VI. ACKNOWLEDGMENT

The author would like to thank R. N. Gadenz for the review of this manuscript.

## REFERENCES

1. G. S. Moschytz, *Linear Integrated Networks Design*, New York, NY: Van Nostrand Reinhold Company, 1975, pp. 623–53.
2. G. Daryanani, *Principles of Active Network Synthesis and Design*, New York, NY: John Wiley and Sons, 1976, Chapter 11.

* Practical element values constraints and the lack of an orthogonal tuning sequence for the SAB usually limit its maximum realizable stable pole-$Q$ to 30. Note that in the CSAB design, the maximum pole-$Q$ value for the two end sections is usually less than or equal to one-half the maximum pole-$Q$ of the overall design.

3. A. S. Sedra and P. O. Brackett, *Filter Theory and Design: Active and Passive*, Portland, OR: Matrix Publishers, Inc., 1978, Chapters 10–12.
4. L. T. Bruton, *RC-Active Circuits: Theory and Design*, Englewood Cliffs, NJ: Prentice-Hall, 1980, Chapters 9 and 10.
5. L. P. Huelsman and P. E. Allen, *Introduction to the Theory and Design of Active Filters*, New York, NY: McGraw-Hill Book Company, 1980, pp. 292–307.
6. M. S. Ghausi and K. R. Laker, *Modern Filter Design—Active RC and Switched Capacitor*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981, Chapter 5.
7. F. E. J. Girling and E. F. Good, "Active Filters," Part 12, Wireless World, *76* (July 1970), pp. 341–5; "The Leapfrog or Active-Ladder Synthesis," Part 13, Wireless World, *76* (September 1970), pp. 445–50; "Applications of the Active-Ladder Synthesis," and "Bandpass Types," Part 14, Wireless World, *76* (October 1970), pp. 505–10.
8. R. L. Adams, "On Reduced Sensitivity Active Filters," in *Proc. 14th Midwest Symp. Circuit Theory*, May 1971, University of Denver, Denver, Colorado, pp. 14.3.1–8.
9. J. Tow and Y. L. Kuo, "Coupled Biquad Active Filters," in Proc. IEEE Int. Symp. Circuit Theory, IEEE Catalog No. 72CHO594-2CT (April 1972), pp. 164-8.
10. L. T. Bruton, "Topological Equivalence of Inductorless Ladder Structures Using Integrators," IEEE Trans. Circuit Theory, *CT-20* (July 1973), pp. 434–7.
11. G. Szentirmai, "Synthesis of Multiple-Feedback Active Filters," B.S.T.J., *52* (April 1973), pp. 527–55.
12. D. Dubois and J. J. Neirynck, "Synthesis of a Leapfrog Configuration Equivalent to an LC-Ladder Filter Between Generalized Terminations," IEEE Trans. Circuit Syst., *CAS-24* (November 1977), pp. 590–7.
13. K. R. Laker, M. S. Ghausi, and J. J. Kelly, "Minimum Sensitivity Active (Leap Frog) and Passive Ladder Bandpass Filters," IEEE Trans. Circuits Syst., *CAS-22* (August 1975), pp. 670–7.
14. J. J. Friend, C. A. Harris, and D. Hilberman, "STAR: An Active Biquadratic Filter Section," IEEE Trans. Circuits Syst., *CAS-22* (February 1975), pp. 115–21.
15. P. E. Fleischer, "Sensitivity Minimization in a Single Amplifier Biquad Circuit," IEEE Trans. Circuits Syst., *CAS-23* (January 1976), pp. 45–55.
16. L. Weyten, "Variation of the Poles of a Second Order RC Active Filter due to a Finite tanδ of the Filter Capacitors," AEU-Arch Elektron Uebertrag, *28* (March 1974), pp. 140–1.
17. V. Belevitch and C. Wellekens, "Internal Equalization in Filters," Circuit Theory and Applications, *1* (1973), pp. 179–86.
18. G. Szentirmai, "FILSYN: A General Purpose Filter Synthesis Program," Proc. IEEE, *65* (October 1977), pp. 1443–58.
19. H. J. Orchard, "Inductorless Filters," Electron. Lett., *2* (June 1966), pp. 224–5.
20. K. R. Laker, R. Schaumann, and M. Ghausi, "Multiple-Loop Feedback Topologies for the Design of Low-Sensitivity Active Filters," IEEE Trans. Circuits Syst., *CAS-26* (January 1979), pp. 1–21.

## APPENDIX

### *Derivation of the Element Values for the Feedback Transfer Function*

With ideal op amp, the voltage transfer functions for the SAB circuit of Fig. 2 are given by:

$$\frac{V_2(s)}{V_1(s)} = -\frac{n_1 s}{s^2 + d_1 s + d_0} \tag{35}$$

and

$$\frac{V_2(s)}{V_1'(s)} = \frac{K_F(s^2 + Fs + d_0)}{s^2 + d_1 s + d_0}, \tag{36}$$

where

$$d_1 = \left(\frac{C_1 + C_2}{C_1 C_2}\right) G_2 - \frac{(G_4 + G_5)}{(G_C + G_D)} \frac{G_B}{C_2} \tag{37}$$

$$d_0 = \frac{(G_4 + G_5)G_2}{C_1C_2} \tag{38}$$

$$n_1 = \left(\frac{G_4}{C_2}\right)\left(1 + \frac{G_B}{G_C + G_D}\right) \tag{39}$$

$$K_F = \frac{G_C}{G_C + G_D} \tag{40}$$

$$F = \left(\frac{G_4 + G_5 + G_2}{C_2} + \frac{G_2}{C_1}\right). \tag{41}$$

The element values as given by eqs. (7) through (13) satisfy the forward transfer function, eqs. (35) and (37) through (39). Note that $G_C = 0$.

In the vicinity of the pole frequency, the feedback transfer function, eq. (36), is closely approximated by

$$\frac{V_2(s)}{V_1'(s)} \simeq \frac{n_1's}{s^2 + d_1s + d_0}, \tag{42}$$

where

$$n_1' = K_F \cdot F. \tag{43}$$

To realize the feedback transfer function, the value of $G_C$ must be finite, say $G_C'$. If we let the value of $G_B$ take on a new value, $G_B'$, and, furthermore, let all the remaining elements take on the values as given before, then the forward and feedback transfer functions, Eqs. (35) and (37) through (43), can be simultaneously satisfied if the following two conditions are met:

$$\frac{G_B}{G_D} = \frac{G_B'}{G_C' + G_D} \tag{44}$$

and

$$K_F = \frac{n_1'}{F} = \frac{G_C'}{G_C' + G_D}. \tag{45}$$

Note that in eq. (45) the coefficient $n_1'$ corresponds to the like coefficient of eq. (6).

The new value of $G_C$ is obtained from eq. (45) and is given by:

$$G_C' = \frac{K_F G_D}{1 - K_F}. \tag{46}$$

Equations (44) and (46) yield the new value of $R_B$, which is given by:

$$G_B' = \frac{G_B(G_C' + G_D)}{G_D}. \tag{47}$$

Equations (46) and (47) correspond to eqs. (17) and (18), respectively.

## AUTHOR

**James Tow,** B.S. (E.E.), 1960, M.S. (E.E.), 1962, Ph.D. (E.E.), 1966, University of California, Berkeley; Bell Laboratories, 1966—. Mr. Tow has been concerned with computer-aided network analysis and design, and the implementation of practical active filters for telecommunication systems. He is currently engaged in the application of digital signal-processing techniques and the design of digital signal processor integrated circuits. Member, IEEE, Eta Kappa Nu, Phi Beta Kappa.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

# Modal Structure of an MCVD Optical Waveguide Fiber

By A. CARNEVALE* and U. C. PAEK†

(Manuscript received September 1, 1982)

In this paper we compare modal analyses of a Modified Chemical Vapor Deposition (MCVD) optical waveguide fiber and an idealized fiber. The pertinent profile parameters of the idealized fiber are obtained from the MCVD profile. We have validated the concept of an optimum $\alpha$ developed in our previous work. The reduction of the bandwidth of the MCVD fiber is shown to be directly related to the imperfections in the profile inherent in the MCVD manufacturing process.

## I. INTRODUCTION

Multimode lightguides, produced by the Modified Chemical Vapor Deposition (MCVD) process, are expected to be of continuing importance to the Bell System. Such lightguides with bandwidths exceeding 1 GHz·km have been produced in the manufacturing plant.

The unique MCVD process produces the index gradient in the core by depositing many layers of material with different compositions, leading to a ripple effect in the profile. Other imperfections in the MCVD process include certain mechanical limitations, difficulties at the core-cladding interface, and the altering of the core center index (burn-off) during the collapse of the preform. These imperfections in the MCVD process lead to a unique modal structure in the fiber, which

* Bell Laboratories, Murray Hill, N.J. † Western Electric Company, Princeton, N.J.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

may differ radically from the modal structure obtained from a perfect power law profile. These differences, for a particular fiber preform and idealization, are analyzed and compared. The effects of the imperfections on dispersion and bandwidth are evaluated.

## II. EXPERIMENT

Our method[1] for solving Maxwell's equations has been described earlier and we shall not repeat it here. A brief schematic of the process is given in Fig. 1. In the experiment, the perfect power law profile ($N_i$)

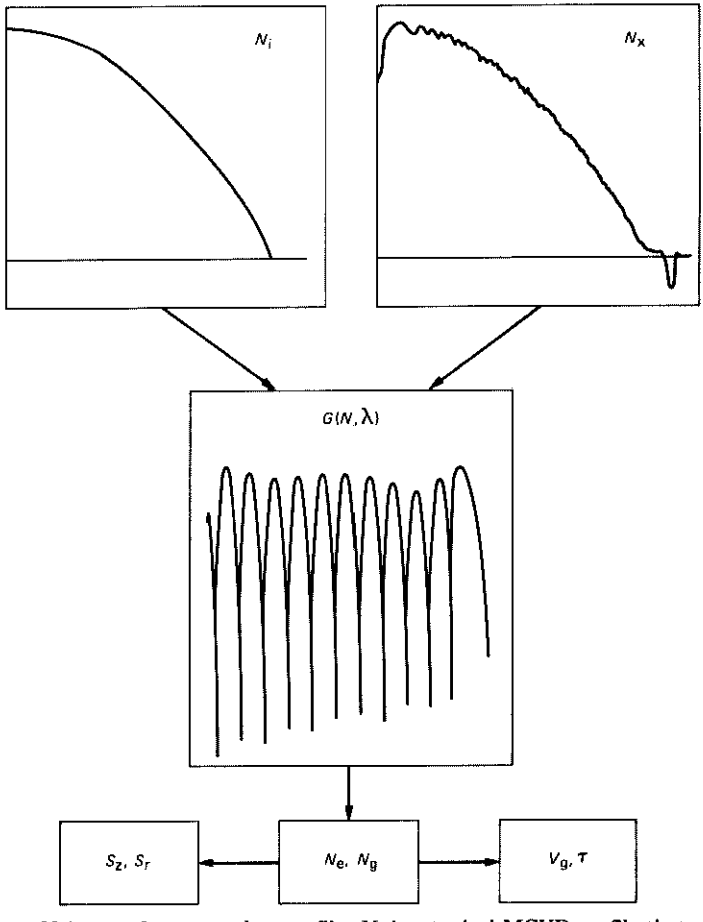



Fig. 1—$N_i$ is a perfect power law profile. $N_x$ is a typical MCVD profile that must be normalized in a particular way to be compatible with the calculations described in Ref. 1. From these calculations we obtain the $G$ function, from which one further determines the effective indices, $N_e$, and the group indices, $N_g$. Further processing leads to the calculation of the Poynting vector, the field functions, the delay times, etc.

is replaced with the actual MCVD profile ($N_x$), which differs from the perfect profile. Thereafter, the data are processed identically to find the effective indices $N_e$, group indices $N_g$, the Poynting vector, and other propagation parameters.

An MCVD preform was given to us by S. Jang of the Western Electric Research Center in Princeton, N.J. This preform had been used earlier to produce fiber for experiments. It was typical in the sense that the bandwidths averaged 1 GHz·km. By means of the laser beam refraction method[2] the refractive index profile was determined and is shown in Fig. 2.

We note in this profile the familiar characteristics of the MCVD preform, i.e., a burn-off region in the center, an undefined core-cladding interface region, and ripples on the profile, which are more pronounced as we near the core center.

The pertinent data required for our analytical testing are the approximate $\bar{\alpha}$ and the maximum $\Delta N$. These data can be obtained from known procedures,[2] or alternatively, as in our case, by utilization of the technique described in one of our earlier publications.[3] Briefly, at a selected wavelength, one obtains the modal display for zero azimuthal numbers, fits the data to a linear least squares equation for $N_g$ vs. $N_e$,

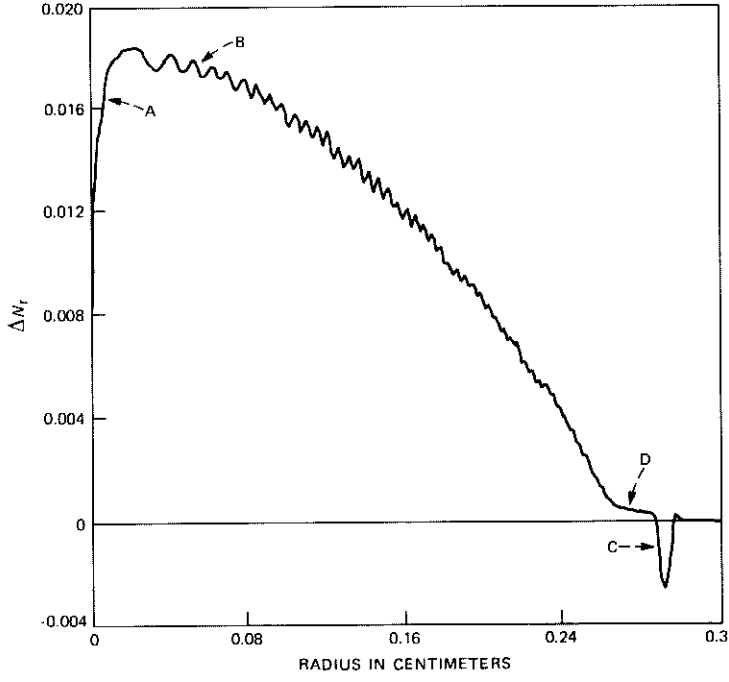



Fig. 2—Typical MCVD profile before normalization.

and compares this slope to the slope vs. $\alpha$, which has been previously calculated for that wavelength. The parameters to be used in this case were $\alpha = 2$, $\Delta N$ is typical for an 11.5-percent $GeO_2$ dopant, and radius $(R_{cc}) = 25$ $\mu$m.

These data are then used to produce modal displays for perfect power law profiles (IDEAL) at the wavelengths and polar indices used for probing the experimental (MCVD) profile. The programs are run in parallel, to be used eventually for comparison purposes only, i.e., there is no communication between the two series of calculations.

Inherent in our calculations is the automatic detection of significant mode splitting between the EH vs. HE, or TE vs. TM, and in the displays to follow, the splittings are shown, wherever possible.

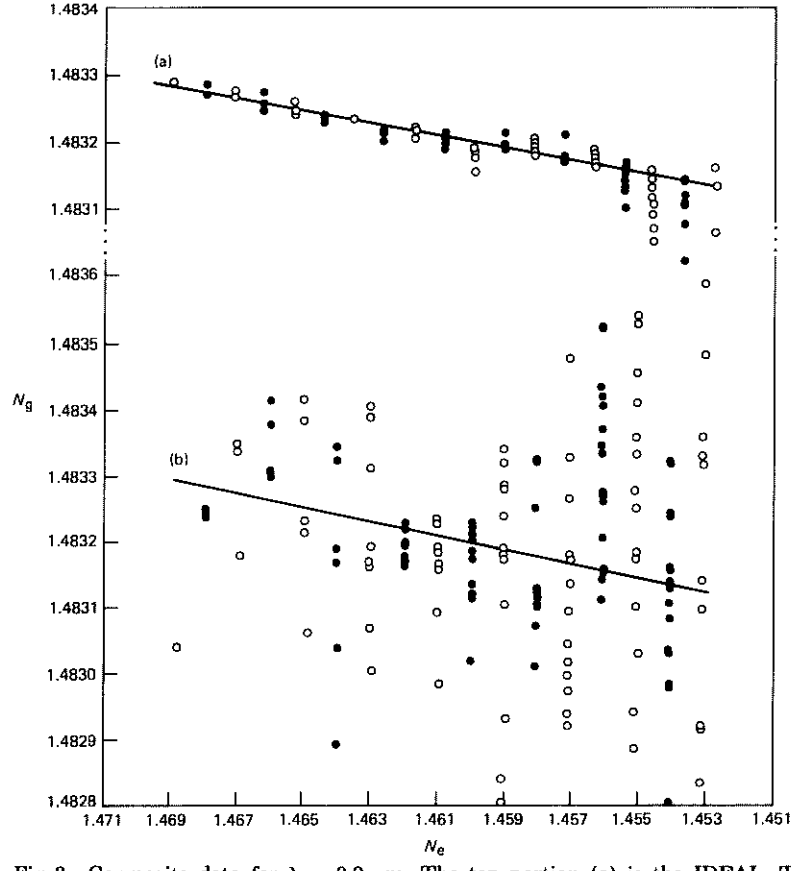



Fig. 3—Composite data for $\lambda = 0.9$ $\mu$m. The top portion (a) is the IDEAL. The bottom portion (b) is the MCVD. The straight lines are the linear least-squares curves fitted to all the data in each case, exclusive of those modes near the cladding, and for those modes widely divergent from the line. In general, the open circles are for polar indices $m = 1, 3, 5, \cdots$ last, while the solid circles are for polar indices $m = 0, 2, 4, \cdots$ last. Split modes are not readily identifiable in this display.

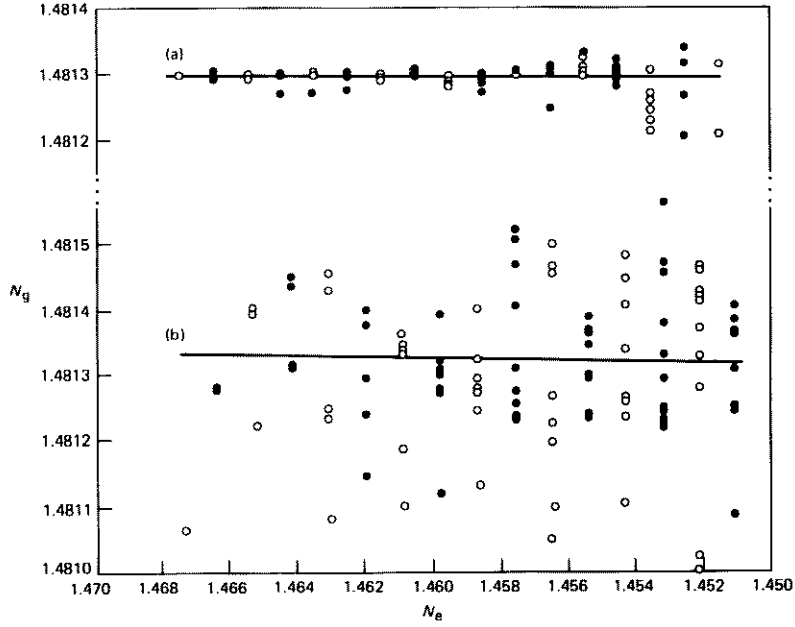



Fig. 4—Same as Fig. 3, $\lambda = 1.0\ \mu m$.

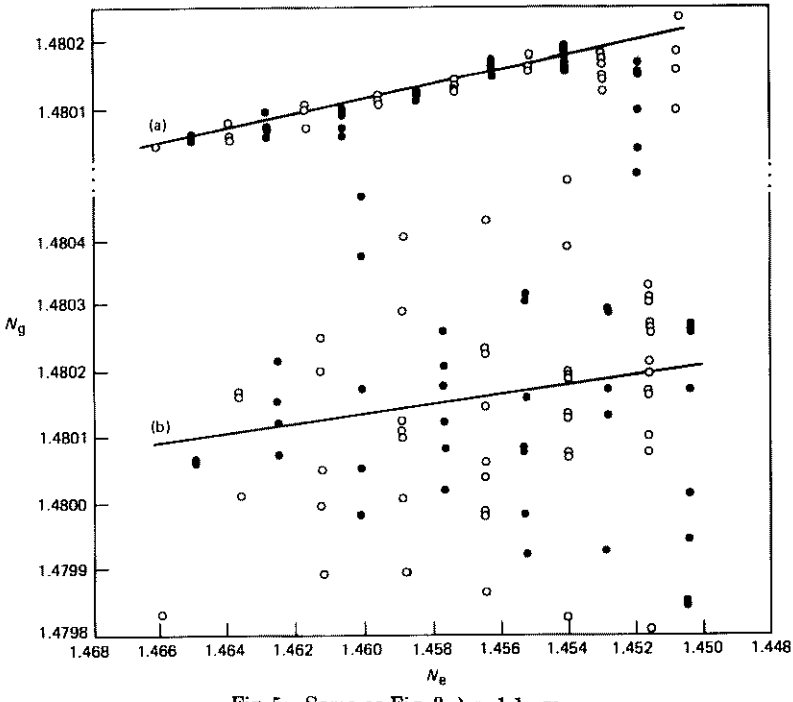



Fig. 5—Same as Fig. 3, $\lambda = 1.1\ \mu m$.

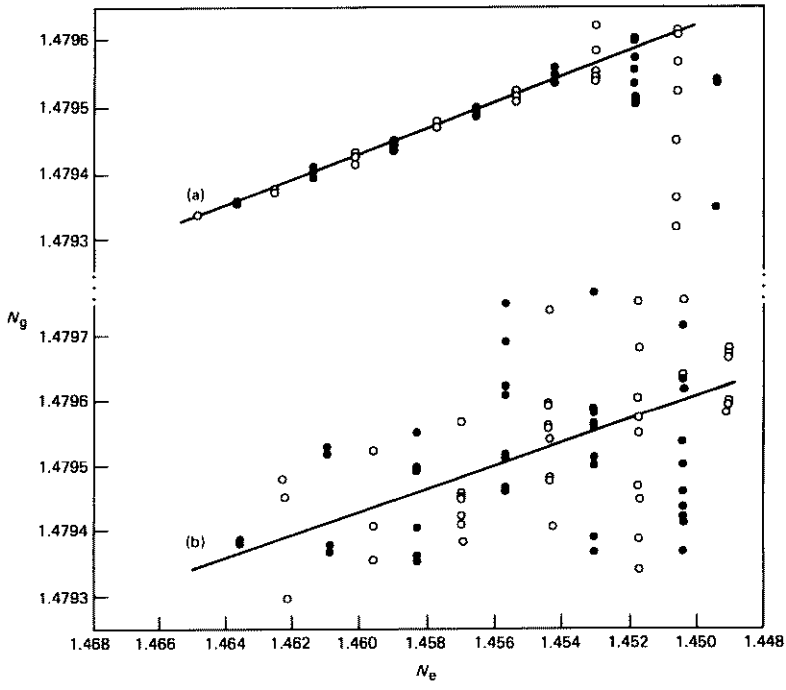



Fig. 6—Same as Fig. 3, $\lambda = 1.2$ μm.

We selected eight wavelengths ($\lambda$) spanning the region

$$0.9 \leq \lambda \leq 1.55 \ \mu\text{m}.$$

Each wavelength was checked for all possible modes ($m = 0, 1, 2, \cdots$ last) for both the IDEAL and the MCVD profiles. Because the extremely large amounts of data required in this analysis involved exorbitantly large amounts of computer time, we reduced the required accuracy to a minimum sufficient for our purposes, thereby saving considerable time and expense. Nevertheless, much computational time was required to accumulate the data given in this report.

Our results on the foregoing procedure are displayed in Figs. 3 through 10. In each figure, the dots represent the modes actually determined, the line is the linear least-squares curve fitted to the data, excluding some of the modes near cutoff, the upper portion (a) is for the IDEAL profile, and the lower portion (b) is for the MCVD profile. Again, some modes that are well off the chart are not shown because of space limitations and were not used to calculate the fitted curves in any case. All the fitted curves, translated where necessary but not rotated, have been condensed into the single plot given in Fig. 11.

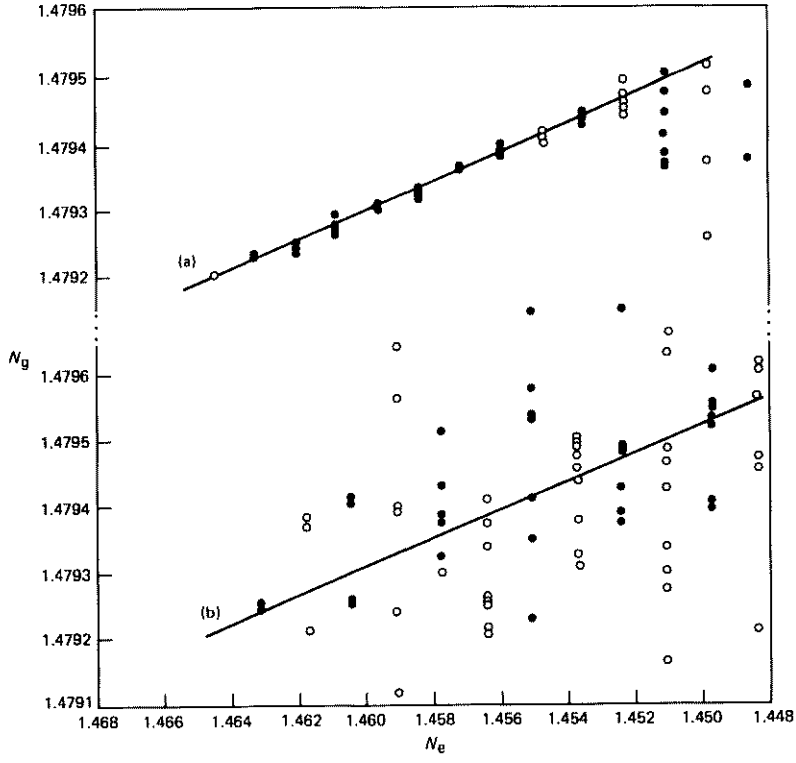



Fig. 7—Same as Fig. 3, $\lambda = 1.23$ $\mu$m.

## III. DISCUSSION

The degree of scatter in the modes for the IDEAL case (Figs. 3 through 10, Part a) is due in part to the inherent splitting of the HE-EH modes, which has been further compounded by the reduced accuracy used in this analysis.

The scattering of the modes for the MCVD case (Figs. 3 through 10, Part b) is much greater than the IDEAL series, perhaps by a factor of 5 to 10. Obviously, the greatly increased scattering for the MCVD series is due to the imperfections in the MCVD profile, such as burn-off and ripple. The contribution of each of these imperfections to increased modal dispersion (poor bandwidth) has been determined and will be the topic of a forthcoming paper.

Further, within the statistical precision and accuracy employed by the authors, one can calculate that the IDEAL and MCVD data will both extrapolate to the same index at the core. We should realize that the MCVD index of refraction data of Fig. 2 was obtained at $\lambda = 0.6328$ $\mu$m. This must be converted to the proper $\Delta N$ at the probing

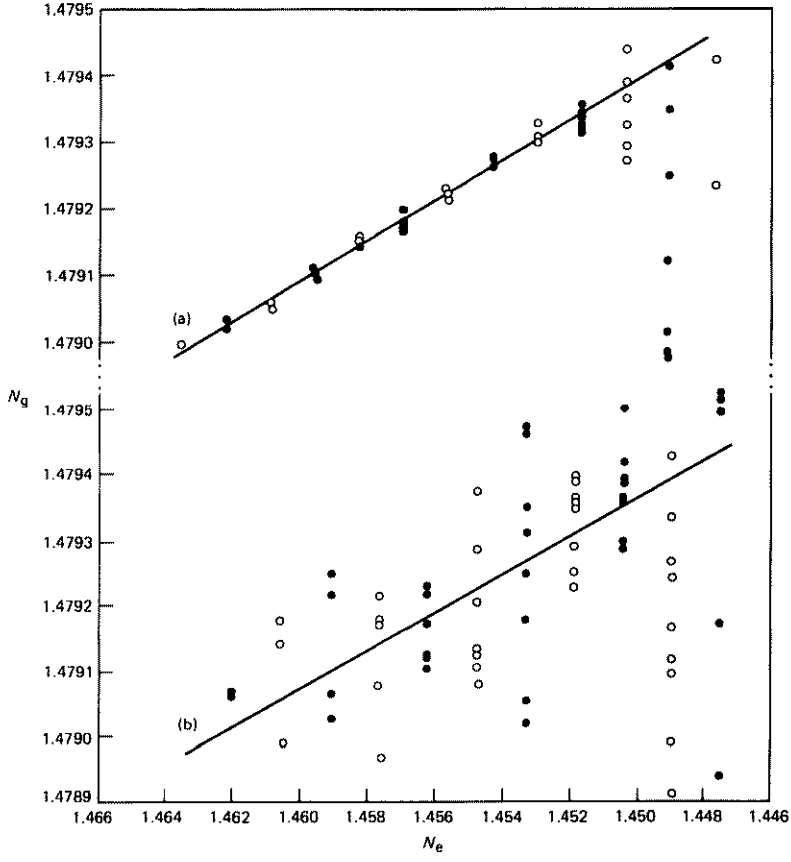



Fig. 8—Same as Fig. 3, $\lambda = 1.32$ $\mu$m.

$\lambda$'s used in any analysis, and was done in this analysis. The results indicate that our parameter $\Delta N$ as calculated at each $\lambda$ is reliable.

The parallelism of the least-squares fitted lines, seen in Fig. 11, indicates that our calculations of the parameter $\alpha$ from the MCVD profile are also reliable, and that an idealized profile may be used to predict characteristics of an MCVD profile. We must realize that $\alpha$ does not vary with $\lambda$, and this is true regardless of whether we are using the IDEAL or MCVD profile. We are immediately struck by the rotation of these curves as we change $\lambda$. We have noted a similar effect[3] at a fixed $\lambda$ for a changing $\alpha$. It should be noted that the previous work[3] and the present work agree closely.

We have one further observation in the present work, which is not readily seen in Figs. 3 through 10: although both the IDEAL and MCVD profiles were normalized to a 25-$\mu$m radius, the IDEAL profile

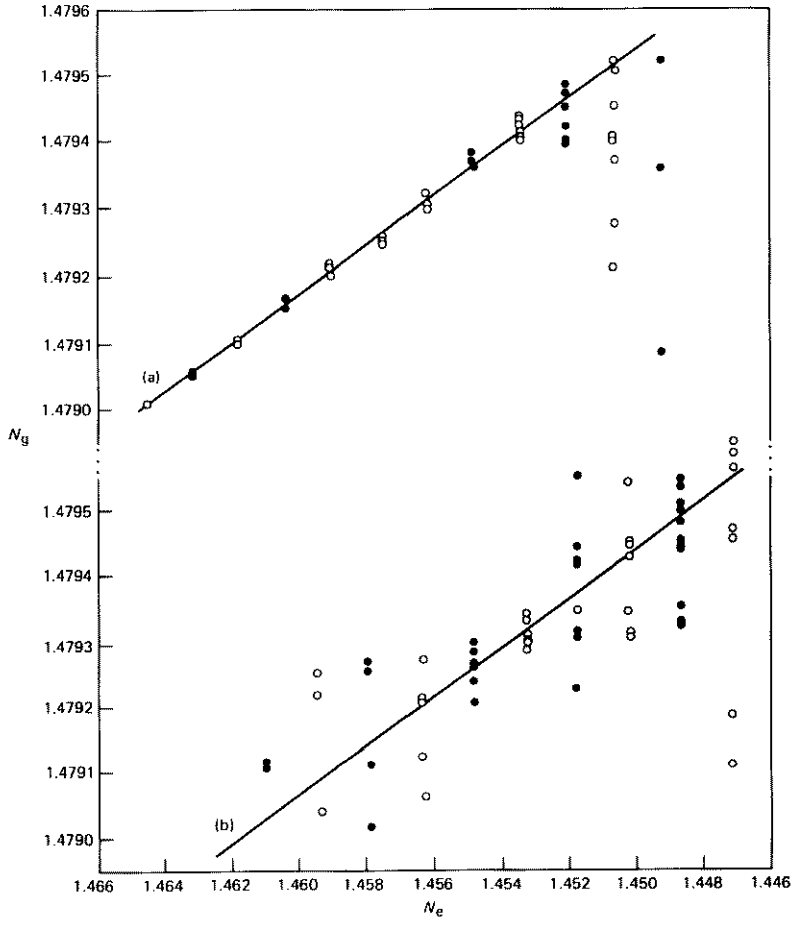



Fig. 9—Same as Fig. 3, $\lambda = 1.40$ $\mu$m.

generated more modes than the MCVD, without exception. To account for the fewer MCVD modes a crude estimation of the effective radius for the MCVD indicates that

$$r_e \approx 23 \ \mu\text{m}.$$

It may be only coincidence that this difference of 2 $\mu$m is of approximately the same magnitude as the width of the burnout.

## IV. REDUCTION OF DATA

Two useful measures of dispersion were derived for this report. The first, which is relatively simple, is the calculation of the dispersion due to the rotation of the line as a function of $\lambda$. This implies that every

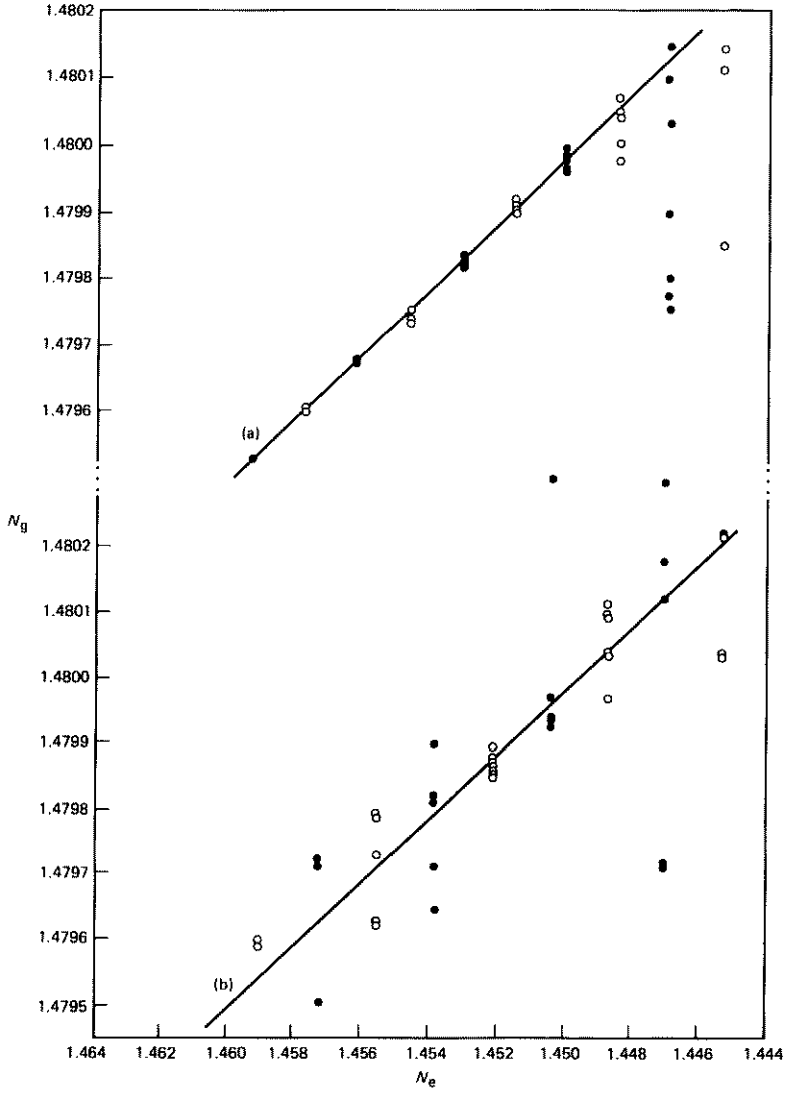



Fig. 10—Same as Fig. 3, $\lambda = 1.55$ $\mu$m.

group index ($N_g$) would be *on* the line. However, the total spread in $N_g$ is now a function of the slope of the line, i.e., the spread in $N_g$ between the $HE_{1,1}$ and the last possible mode that can be obtained on that line. We have given this measure the name "sigma of rotation", ($\sigma_R$). Of course, we must count the number of modes at each $N_e$ to find the mean $N_g$ and then calculate $\sigma_R$ in the conventional manner.

The second measure of dispersion we have utilized is the standard deviation of the group indices determined from the indicated least-

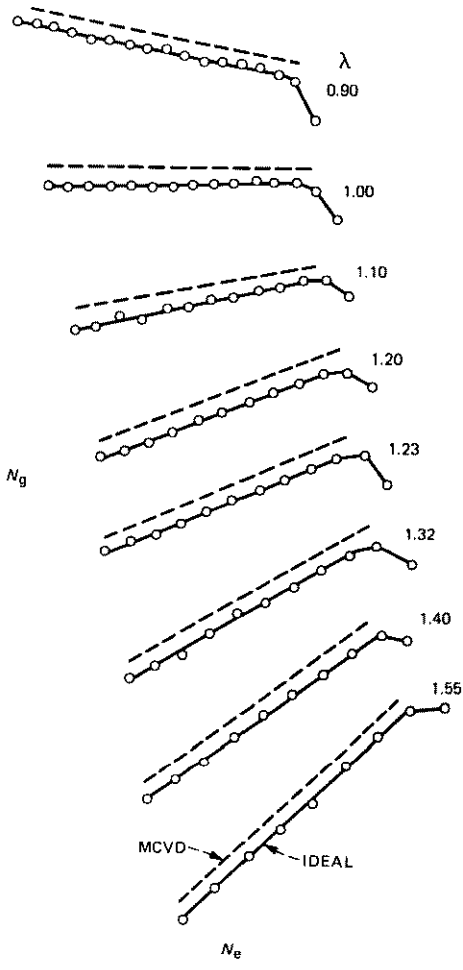



Fig. 11—Composite data. Least-squares fitted lines (IDEAL and MCVD) from the data of Figs. 3 through 10, identifiable in this display.

squares fitted line for $N_g$ vs. $N_e$. Thus, if the curve value of $N_g$ is denoted by $N_g(\text{curve})$, then

$$\sigma_{N_g} = \sqrt{\frac{\Sigma[N_g - N_g(\text{curve})]^2}{n-1}},$$

where $n$ is the total number of modes.

The results of these calculations for $\sigma_R$ are given in Fig. 12, and as indicated in Fig. 11, are about the same for the IDEAL and MCVD fibers. Further, if this were the only measure of dispersion employed,

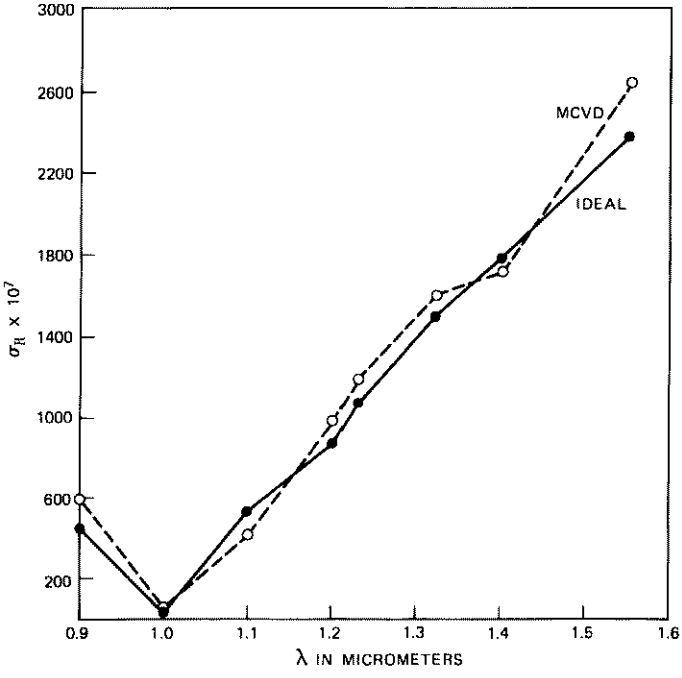



Fig. 12—$\sigma_R$ vs. λ for IDEAL or MCVD.

we would expect this fiber to yield the maximum bandwidth at $\lambda = 1.00\ \mu m$. This agrees precisely with our previous work.[3] The results for the calculation of $\sigma_{N_g}$ are given in Fig. 13 and we can see that:

$$\sigma_{N_g}(\text{MCVD}) \approx 10\ \sigma_{N_g}(\text{IDEAL}).$$

Also, we note further that $\sigma_{N_g}$ is a decreasing function for larger values of λ, and becomes virtually constant when $\lambda \geq 1.2\ \mu m$.

A better understanding of the roles of the two measures of dispersion previously described is given in Figs. 14 and 15. In Fig. 14, $\sigma_R$ and $\sigma_{N_g}$ are plotted for the IDEAL case, and in Fig. 15 the same data are plotted for the MCVD case. Apparently, the $\sigma$ that is dominant at any λ will be the principal cause of inability to maximize the bandwidth.

From theory[4] we have calculated the total dispersion $\sigma_c$ by:

$$\sigma_c = \sqrt{\sigma_R^2 + \sigma_{N_g}^2},$$

and these data are given in Fig. 16. We see that in the region where $\sigma_R$ is dominant ($\lambda \geq 1.2\ \mu m$), the curves are roughly parallel. The larger $\sigma_c$ for the MCVD in this region comes about because the $\sigma_{N_g}$ of the MCVD is greater than $\sigma_{N_g}$ of the IDEAL.

Over the range $0.9\ \mu m \leq \lambda \leq 1.2\ \mu m$, the IDEAL case peaks ($\sigma_c =$

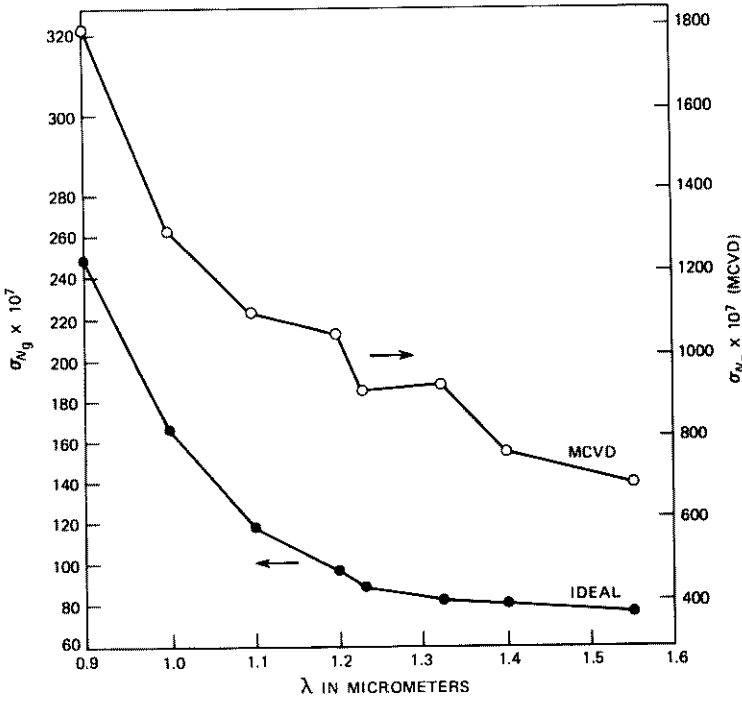



Fig. 13—$\sigma_{N_g}$ vs. λ for IDEAL and MCVD.

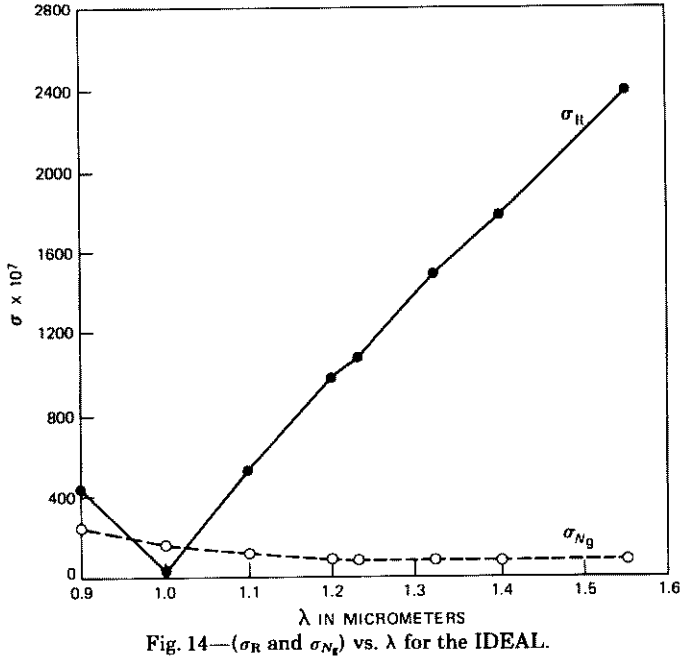



Fig. 14—($\sigma_R$ and $\sigma_{N_g}$) vs. λ for the IDEAL.

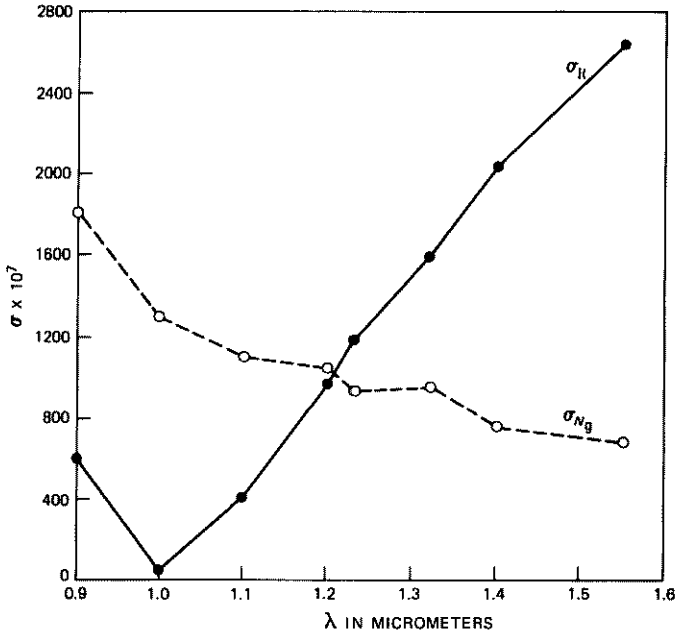



Fig. 15—($\sigma_R$ and $\sigma_{N_g}$) vs. λ for the MCVD.

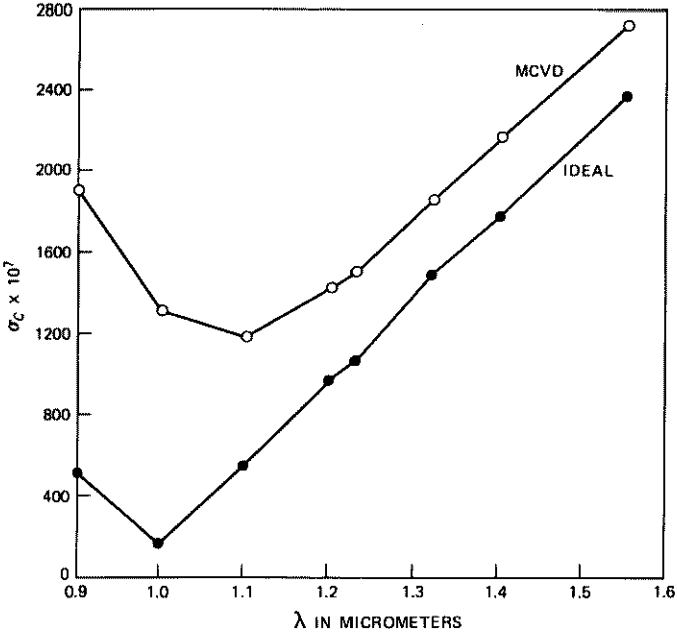



Fig. 16—$\sigma_c$ vs. λ for IDEAL and MCVD.

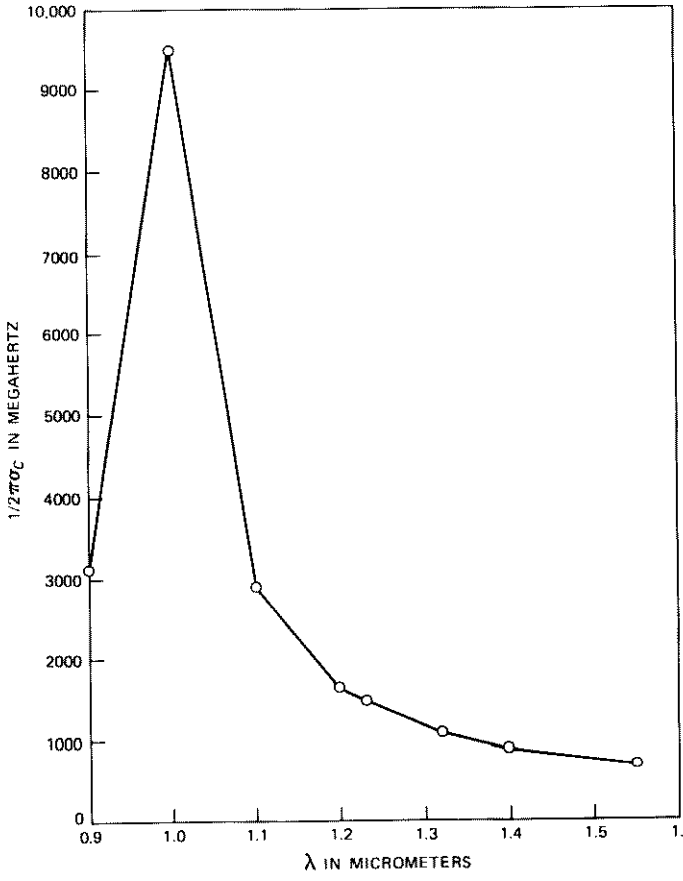



Fig. 17—$1/2\pi\sigma_c$ for the IDEAL case.

minimum) sharply at about 1.0 $\mu$m, while the MCVD displays a very broad response that appears to peak at about 1.1 $\mu$m. A qualitative measure to estimate the pulse shape can be realized by calculating $W(h/2)$; $W(h/2) = 1/2\pi\sigma_c$ (MHz).

These data are given in Figs. 17 and 18 for the IDEAL and the MCVD, respectively. The IDEAL gives a bandwidth of 9 to 10 GHz/km at 1.0 $\mu$m and falls off to half-height at $\pm 0.7$ $\mu$m. The MCVD obtains a maximum of about 1.5 GHz·km at $\approx 1.1$ $\mu$m, and falls off to half height at $\pm 0.2$ $\mu$m.

## V. CONCLUSION

There can be no doubt that the concept of an optimum $\alpha$ is valid for perfect power law and for experimental (manufactured) profiles.[3] The reduction of the bandwidth is directly related to the imperfections

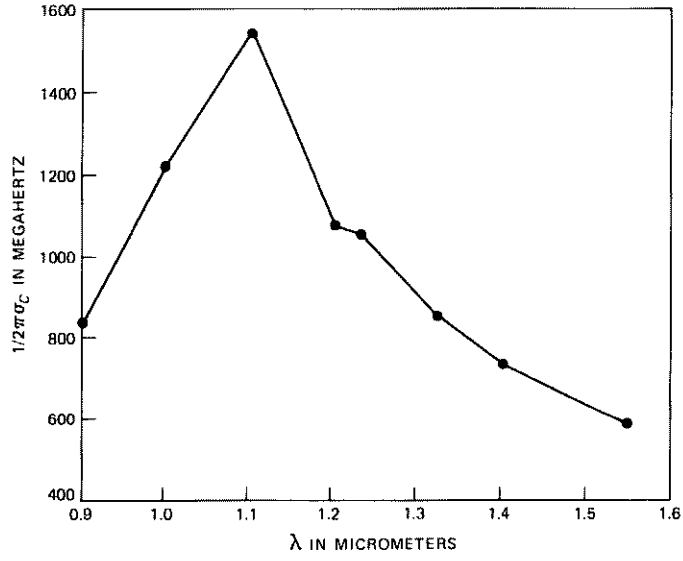



Fig. 18—$1/2\pi\sigma_c$ for the MCVD case.

of the MCVD profile such as burnout, ripples, etc. The direct role of each of these deviations from an idealized profile, in the reduction of bandwidth, has been calculated, and will be the subject of a forthcoming paper. These observations have been substantiated in many respects from direct communication with the materials scientists at Bell Laboratories in Murray Hill and Atlanta, and the Western Electric Company Engineering Research Center. We have also ascertained that we can improve the calculation of $\sigma_{N_g}$ by about a factor of 2. Thus we find the bandwidth for the idealized fiber is about 18 GHz·km, while for the MCVD fiber it is about 2.8 GHz·km. These bandwidths can be further improved by careful computer-aided design of the profile, for which investigations are currently in progress.

## VI. ACKNOWLEDGMENTS

We appreciate the continued support and encouragement given us in this work by M. I. Cohen, R. J. Klaiber, and L. S. Watkins. We are also encouraged by the enthusiasm of the materials scientists engaged in the lightguide program at Bell Laboratories in Murray Hill and Atlanta and Western Electric, as evidenced in a number of informal discussions with them.

## REFERENCES

1. G. E. Peterson, A. Carnevale, U. C. Paek, and D. W. Berreman, "An Exact Numerical Solution to Maxwell's Equations for Lightguides," B.S.T.J., *59*, No. 7 (September 1980), pp. 1175–96.

2. L. S. Watkins, "Laser beam refraction traversely through a graded-index preform to determine refractive index ratio and gradient profile," Applied Optics, *18* (July 1, 1979), pp. 2214–22.
3. G. E. Peterson, A. Carnevale, U. C. Paek, and J. W. Fleming, "Numerical Calculation of Optimum $\alpha$ for a Germania-Doped Silica Lightguide," B.S.T.J., *60*, No. 4 (April 1981), pp. 455–70.
4. J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, New York, NY: John Wiley & Sons, p. 93.

## AUTHORS

**Un-Chul Paek,** B.S. (Engineering), 1957, Korea Merchant Marine Academy, Korea; M.S., 1965, Ph.D., 1969, University of California, Berkeley; Western Electric, 1969—. At the Western Electric Engineering Research Center, Princeton, N.J., Mr. Paek has been engaged primarily in research on laser material interaction phenomena and fiber optics. Member, Optical Society of America, American Ceramic Society, Sigma Xi.

**Anthony Carnevale,** B.S. (Physics), 1960, Fairleigh Dickinson University; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Carnevale has been engaged in research on nuclear magnetic resonance, electron paramagnetic resonance, and computer software. For the last four years, his work has been devoted to fiber optics.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

# Asymptotic Analysis of a Queueing Model With Bursty Traffic

By D. Y. BURMAN* and D. R. SMITH*

(Manuscript received September 24, 1982)

Assuming a particular model for "bursty" traffic at a packet-switching node, we find expressions for the expected delay of packets that are valid in light and heavy traffic. Each expression consists of a "correction factor" multiplied by the expected delay experienced by packets when the arrivals are "smooth" (Poisson) and of the same average rate. Approximate values for the correction factor in arbitrary traffic can be obtained by interpolation. This provides an example of a method that often gives fast approximate solutions for bursty traffic models that are not themselves tractable but become so when the offered traffic is assumed to be Poisson.

## I. INTRODUCTION

Many models of queueing systems assume that arrivals occur according to a Poisson process. Intuitively, the Poisson process may be characterized by the properties that events occur one at a time and do not depend on the past history of events. Typically, this situation arises when there are large numbers of users of a system, as in the case of arrivals of calls to a central office, since one arrival does not significantly affect the probability of another. Fortunately, these models are often mathematically tractable.

For other systems the Poisson assumptions are not realistic. Often arrivals are indicative of overall activity and give information about the probability of future arrivals. For example, suppose that all arrivals

* Bell Laboratories, Holmdel, N.J.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

are generated by a single user who is alternately active and inactive. An arrival indicates that the user is active and hence that there is a greater than average probability of another arrival shortly thereafter. A second example concerns the arrival of packets to a node in a packet-switching network. When virtual calls are employed, the route packets travel for a particular call is fixed for the duration of the call. If we make the assumption that an individual virtual call generates packets according to a Poisson process, then the instantaneous arrival intensity at any node is equal to the sum of the intensities for the calls routed through the node, which varies probabilistically with time. In these examples the arrivals are correlated and the traffic is said to be bursty.

Even the simplest models involving bursty traffic tend to be difficult to solve analytically. Several authors have given approximations. Heffes[1] matched the first three moments of the arrival process to those of an Interrupted Poisson Process. Assuming that an arriving packet requires an exponentially distributed amount of time to be served, he was then able to use the results of Kuczura[2] to analyze this system. Laue,[3] making a similar approximation for the arrival process, assumed that an arriving packet requires a constant service time. The mean waiting time of a packet could then be calculated using the numerical matrix techniques developed by Neuts[4] and Lucantoni and Neuts.[5] A third approach developed by Anick, Mitra, and Sondhi[6] treats a different but related model. The models of Heffes and Laue are based on the assumption that an individual customer generates packets according to a Poisson process. Anick, Mitra, and Sondhi assume that a customer generates packets at a constant rate for a random time. The resulting fluid model was treated numerically by calculating the eigenvalues of the resulting equations.

All of the above works derive numerical techniques to estimate the traffic statistics of interest over a wide range of traffic parameters. It is the goal of this paper to provide simple, closed-form expressions that give insight into the effect of burstiness on delays. This is done by studying queueing systems offered bursty traffic (see Section II for a complete description) in light and heavy traffic.

In light (heavy) traffic, it is obvious that the expected delay tends to zero (infinity). Surprisingly, when the expected delay is divided by the expected delay for the same system offered Poisson traffic of equal average intensity, the ratio goes to a nonzero finite limit in both light and heavy traffic. These limits may be thought of as "correction factors" by which the expected delay for the solvable Poisson system should be multiplied to obtain the expected delay for the bursty system. By interpolating between the light- and heavy-traffic results, one can obtain insight into the approximate effect of burstiness for all values

of traffic. Indeed, similar light- and heavy-traffic limits for the $M/E_k/c$ system have been used to obtain very accurate approximate values of the delay for all values of traffic (see Lauber and Smith[7]).

The remainder of this paper is organized as follows: The queueing model for bursty traffic and our results are discussed in detail in Section II. Rigorous proofs of the light-traffic results are given in Section III when the service times are of the phase type. (Appendix A presents a brief background on phase-type distributions.) Since any service-time distribution can be approximated arbitrarily closely by one of phase type, it is sufficient for practical purposes to establish the light-traffic results for the latter. Appendix B presents an intuitive approach for deriving the light-traffic results. Heavy-traffic results are presented in Section IV and concluding remarks in Section V.

## II. QUEUEING MODEL AND RESULTS

The specific queueing model treated here is one in which the arrival process is a nonhomogeneous Poisson process whose rate equals $\lambda m$, where m is the state of an $M/M/\infty$ queue with birth rate $\alpha$ and service rate $\beta$. The arrivals are offered to a single server whose successive services are assumed to be independent and identically distributed according to some general distribution with mean equal to $\mu^{-1}$. Blocked arrivals queue up and are served on a first-in-first-out basis.

This queueing model supports either of two (essentially identical) scenarios for the queueing of packets of information at a packet switch. In both scenarios, the switch is modeled as a single server with an infinite buffer for queued packets, and the service time of a packet is its length (in bits) divided by the line speed. In the first scenario, an individual virtual call generates packets according to a Poisson process with rate $\lambda$, for an exponentially distributed length of time with mean $\beta^{-1}$. The distribution of requests for virtual calls is Poisson with rate $\alpha$, and the number of simultaneous virtual calls that can be supported by the switch is unlimited. The second scenario is similar to a fluid model treated by Mitra and Anick[8] and Kosten.[9] In this case an individual customer is in one of two states, either "active" or "inactive." It is assumed that the time in each of the states is exponentially distributed with rates $\beta$ and $\gamma$, respectively. While in the "active" state, the customer transmits packets according to a Poisson process with rate $\lambda$. If there are $N$ (large) such customers with $N\gamma \sim \alpha$ and $\gamma\beta^{-1} \sim 0$, then the number of "active" customers is distributed like the number of customers in the $M/M/\infty$ queue described above.

Throughout the remainder of this paper, we will refer to the entities queued at the single server as packets and the entities in the $M/M/\infty$ system as virtual calls or calls. Our analysis will focus on the limiting form for the mean delay of a packet in a lightly or heavily loaded

system. The total number of packets generated during a virtual call is geometrically distributed with mean $\lambda\beta^{-1}$ and the mean total rate of packet generation is $\lambda(\alpha/\beta)$.

There is a technical problem in the analysis for this model since the packet arrival rate is unbounded. Hence, consider first the system where the number of virtual calls is limited to $N$, i.e., the rate of arrivals of packets is $\lambda$ times the number of calls present in an M/M/N/N queueing system. The average rate of arrival of packets to the queue is

$$\lambda_N \stackrel{\text{def}}{=} \lambda(\alpha/\beta)[1 - B(N, \alpha/\beta)], \tag{1}$$

where $B(N, \alpha/\beta)$ is the Erlang Blocking formula. Let $D_B^{(N)}$ be the expected delay of a packet in this system, and let $D_M^{(N)}$ be the delay in an M/G/1 with arrival rate $\lambda_N$ and the same service-time distribution.

Our key light-traffic result (valid for phase-type service distributions) is

$$\lim_{\lambda \to 0} \frac{D_B^{(N)}}{D_M^{(N)}} = k_l^{(N)}, \tag{2}$$

where $0 < k_l^{(N)} < \infty$, and

$$\lim_{N \to \infty} k_l^{(N)} = 1 + \frac{2\mu}{\alpha} \frac{[1 - \phi(\beta)]}{1 + C^2} \stackrel{\text{def}}{=} k_l, \tag{3}$$

where $\mu^{-1} = E(S)$, $C^2 = \text{var}(S)/E(S)^2$ ($S$ is a service-time random variable), and $\phi(\cdot)$ is the Laplace transform of the stationary excess of $S$. The limit (2) is proved by using a simple extension of a lemma established in Burman and Smith.[10] The exact value of $k_l^{(N)}$ is difficult to compute; however, the limit $k_l$ as $N \to \infty$ is computable and can be interpreted as the light-traffic limit of the ratio of the delay ($D_B$) for the bursty system described earlier (with no limit on the number of virtual calls) and the delay ($D_M$) for the M/G/1 queue with arrival rate $\lambda(\alpha/\beta)$. This statement can be summarized (although not explicitly proved) as

$$\lim_{\lambda \to 0} \frac{D_B}{D_M} = k_l, \tag{4}$$

where $k_l$ is given in (3).

In the heavy-traffic case no rigorous limit is available. Nevertheless, diffusion analysis gives the following approximation in heavy traffic:

$$\frac{D_B^{(N)}}{D_M^{(N)}} \approx k_h^{(N)}, \tag{5}$$

where $k_h^{(N)}$ is known explicitly [see eq. (40)] and

$$\lim_{N\to\infty} k_h^{(N)} = 1 + \frac{2\mu}{\alpha}\frac{1}{1+C^2} \stackrel{\text{def}}{=} k_h. \tag{6}$$

This supports the conjecture that

$$\lim_{\lambda\to\frac{\beta\mu}{\alpha}} \frac{D_B}{D_M} = k_h. \tag{7}$$

It should be noted that $1 < k_l < k_h$, and while no proof is available, it is reasonable to conjecture that $k_l < D_B/D_M < k_h$ for all stable values of $\lambda$. (Indeed it is conjectured that $D_B/D_M$ is monotone, as suggested by a similar analysis for the delay in an M/G/c queue normalized by the delay in an M/M/c queue (see Lauber and Smith[7]).

At this point it is worth noting that the manner in which the traffic intensity $\rho = (\lambda/\mu)(\alpha/\beta)$ approached 0 or 1 affects the limiting value of $D_B/D_M$. The previously described results are based on the variation of $\lambda$ only; one can also allow $\alpha$, the rate of arrival of calls, to vary. For an intuitive example of the difference, note that $\lambda \to 0$ corresponds to light traffic with one packet per call, while $\alpha \to 0$ corresponds to light traffic with a geometrically distributed (with mean $\lambda/\beta$) number of packets per call. It can be shown (although it is not explicitly reported here) that $D_B/D_M$ is completely different in the two cases.

We now focus our attention on the behavior of $k_l$ and $k_h$ for a fixed mean number of calls in the system $(\alpha/\beta)$. As $\alpha, \beta \to \infty$, $(\alpha/\beta$ fixed) the arrival process of packets approaches a Poisson process and indeed, by examination of (3) and (6), $k_l$ and $k_h$ both go to 1. As $\alpha, \beta \to 0$ ($\alpha/\beta$ fixed), $k_h$ goes to $\infty$, and $k_l$ goes $1 + \beta/\alpha$. To understand these limits, note that in this case the number of calls remains constant for longer and longer periods of time and steady-state effects become significant. In heavy traffic, the process remains in states for which the packet generation rate is faster than the service rate, so that the delays become large. In light traffic, one may obtain the $1 + \beta/\alpha$ limit in an intuitive fashion by conditioning on the number of calls $m$ at packet generation times and computing the conditional delay (using known expressions for delay) assuming a constant arrival rate $\lambda m$.

In (4) and (7), the delay for the bursty system was normalized by the delay for the M/G/1 system. If instead we choose to normalize by the delay for an M/M/1 system with the same arrival and service rates (denoted by $\tilde{D}_m$), then the limits become

$$\lim_{\lambda\to 0} \frac{D_B}{\tilde{D}_M} = 1 + \frac{C^2-1}{2} + \frac{\mu}{\alpha}[1 - \phi(\beta)] \tag{8}$$

and

$$\lim_{\lambda \to \frac{\beta\mu}{\alpha}} \frac{D_B}{\tilde{D}_M} = 1 + \frac{C^2 - 1}{2} + \frac{\mu}{\alpha}. \tag{9}$$

These results are interesting in that they suggest separation of the effects of variability of the service time $(C^2 - 1)/2$ and the variability of the arrival process (the third term).

In addition to the results for the mean delay $D_B$, we show that the light-traffic limit of the distribution of the delay $D$, given $D > 0$, is

$$\lim_{\lambda \to 0} P(D > t \mid D > 0) = \frac{\int_t^\infty \int_y^\infty [\alpha + \beta e^{-\beta(x-t)}] dH(x) dy}{\frac{\alpha}{\mu} + 1 - \int_0^\infty e^{-\beta x} dH(x)}. \tag{10}$$

Again, in order for this to be rigorously stated it should be in terms of the limit of similar quantities for systems allowing only a finite number of virtual calls.

## III. DERIVATION OF THE LIGHT-TRAFFIC RESULTS

In this section, we derive the light-traffic result stated in (4). Our approach is to show that as the traffic intensity goes to 0, the probability of having $i$ (greater than 0) packets in the system goes to 0 asymptotically as $\lambda^i$. The exact rate of convergence can be derived by a detailed study of the state equations and from there (4) follows trivially.

Consider a single-server queue whose service times are of phase type (see Appendix A). Let the arrival process be a nonhomogeneous Poisson process whose rate is $\lambda$ times a function of the state of a Markov process. Then, the multidimensional process consisting of the number $i$ of packets in queue, the state $j$ of the arrival Markov process, and the phase $k$ of the packet in service is itself a Markov process. A typical state will be denoted by $(i, j, k)$ for $i > 0$ and $(0, j)$ for $i = 0$, where $j \geq 0$ and $k = 1, \cdots, m$, the number of phases. The ergodic distribution will be denoted by $p(\cdot, \cdot, \cdot)$ and define

$$p(i) = \sum_{j,k} p(i, j, k),$$

with the obvious definition for $i = 0$. When there is an upper bound on the arrival rate (as when the arrival Markov process is finite), then the technique used in Burman and Smith[10] to prove Theorem 3.1 therein can be employed to prove:

*Lemma 1: There exists a constant $R > 0$ such that*

$$p(i) \leq \lambda^i R^i.$$

One may define

$$\tilde{\rho}(i, j, k) = \lim_{\lambda \to 0} \lambda^{-1} \rho(i, j, k) \tag{11}$$

with an analogous definition for $i = 0$, and these limits may be recursively related and shown to exist by examination of the balance equations. (See Smith[11] and Burman and Smith[10] for examples of this technique.) Thus, when the arrival rates are bounded, the light-traffic methodology is straightforward. It is quite possible, however, that the resulting equations are difficult to solve.

This is exactly the case when the arrival Markov process is an M/M/N/N queue (finite number of virtual calls). It is difficult to explicitly solve for $\tilde{\rho}^{(N)}$ [the limiting light-traffic normalized probabilities for this system, see (11)], although it can be shown that $\tilde{\rho}^{(N)} \to \tilde{\rho}$ as $N \to \infty$, where $\tilde{\rho}$ is the solution to the equations (assuming Lemma 1) when the birth-death process is the M/M/$\infty$ queue. At the core of this argument (not presented here in detail) are the facts that the equations involving $\tilde{\rho}^{(N)}(\cdot, j, \cdot)$ for $j < N$ are identical with those involving $\tilde{\rho}^{(N+1)}(\cdot, j, \cdot)$ and that

$$\lim_{N \to \infty} \tilde{\rho}^{(N)}(0, j) = \frac{1}{j!} (\alpha/\beta)^j e^{-\alpha/\beta}.$$

The existence of the limits [in (11)] can be shown by recursion on $i$ and the fact that the limit was previously established for $i = 0$. Thus $\lim_{N \to \infty} \tilde{\rho}^{(N)} (= \tilde{\rho})$ may be calculated by studying the system with $N = \infty$.

We now turn our attention to calculating $\tilde{\rho}$ by studying the bursty system with arrival rate $\lambda$ times the number of calls in an M/M/$\infty$ queue. For this system it is not hard to show that $\rho$, the steady-state probability satisfies

$$\begin{aligned} &-(j\lambda + \alpha + j\beta)\rho(0, j) + \alpha\rho(0, j-1) \\ &+ (j+1)\beta\rho(0, j+1) + \sum_n \rho(1, j, n)\mathrm{E}_n = 0, \end{aligned} \tag{12}$$

$$\begin{aligned} &-(j\lambda + \alpha + j\beta - T_{kk})\rho(1, j, k) + \alpha\rho(1, j-1, k) \\ &+ (j+1)\beta\rho(1, j+1, k) + \sum_n \rho(1, j, n)T_{nk} \\ &+ j\lambda\omega_k\rho(0, j) + \omega_k \sum_n \rho(2, j, n)\mathrm{E}_n = 0, \end{aligned} \tag{13}$$

and

$$\begin{aligned} &-(j\lambda + \alpha + j\beta - T_{kk})\rho(i, j, k) + \alpha\rho(i, j-1, k) \\ &+ (j+1)\beta\rho(i, j+1, k) + \sum_n \rho(i, j, n)T_{nk} \\ &+ j\lambda\rho(i-1, j, k) + \omega_k \sum_n \rho(i+1, j, n)\mathrm{E}_n = 0, \quad \text{for} \quad i \geq 2, \end{aligned} \tag{14}$$

where $j \geq 0$ and $\omega\cdot$, $T\cdot\cdot$, and $\mathrm{E}\cdot$, are the initial, transition, and exit rates defining the phase-type distribution of the service-time process. (These are discussed in greater detail in Appendix A.) We also assume that Lemma 1 holds.

Next define the generating functions

$$q(i, z, k) = \sum_j z^j \tilde{\rho}(i, j, k), \qquad i > 0,$$

and

$$q(0, z) = \sum_j z^j \tilde{\rho}(0, j).$$

From (12) to (14), we see that $q$ satisfies the following equations

$$-\alpha(1 - z)q(0, z) + \beta(1 - z)q_z(0, z) = 0, \tag{15}$$

$$q(1, z, \cdot)[T - \alpha(1 - z)I] + q_z(1, z, \cdot)\beta(1 - z)I = -\omega z q_z(0, z), \tag{16}$$

and

$$q(i, z, \cdot)[T - \alpha(1 - z)I] + q_z(i, z, \cdot)\beta(1 - z)I = -z q_z(i - 1, z, \cdot) \quad \text{for} \quad i > 1. \tag{17}$$

Equation (15) immediately gives

$$q(0, z) = e^{-\alpha/\beta(1-z)}. \tag{18}$$

The next lemma relates $D_B$, the expected delay in this system, to $q(1, z, \cdot)$.

*Lemma 2:*

$$\lim_{\lambda \to 0} \lambda^{-1} D_B = -\frac{\beta}{\alpha} q_z(1, 1, \cdot) T^{-1} e,$$

*where e is the vector of ones.*

*Proof:* The mean number $L$ of packets in the queue is given by

$$L = \sum_{i>1} (i - 1) \sum_{j,k} \rho(i, j, k).$$

By Lemma 1,

$$\lim_{\lambda \to 0} \lambda^{-2} L = \sum_{j,k} \tilde{\rho}(2, j, k) = q(2, 1, \cdot)e.$$

From (17), we get that

$$q(2, 1, \cdot) = -q_z(1, 1, \cdot)T^{-1}$$

and by Little's Law we are done. □

We now establish (3). The previous lemma shows that the key

quantity is $q_z(1, 1, \cdot)$. Differentiating (16) with respect to $z$ and evaluating at $z = 1$ gives

$$\alpha q(1, 1, \cdot)I + q_z(1, 1, \cdot)(T - \beta I) = \omega \frac{\alpha}{\beta}\left(\frac{\alpha}{\beta} + 1\right), \tag{19}$$

where we used (18) to give us $q(0, z)$. Substituting for $z = 1$ into (16) gives that

$$q(1, 1, \cdot) = \frac{\alpha}{\beta}(-\omega T^{-1}) = \frac{\alpha}{\beta\mu}\xi,$$

where $\xi$ is the stationary distribution of the service-time process [see (42) and (43)]. Rearranging (19) we get

$$\begin{aligned} q_z(1, 1, \cdot) &= \frac{\alpha^2}{\beta}\omega T^{-1}(\beta I - T)^{-1} + \frac{\alpha}{\beta}\left(\frac{\alpha}{\beta} + 1\right)\omega(\beta I - T)^{-1} \\ &= -\frac{\alpha}{\beta^2}[\alpha\beta\omega T^{-1}(\beta I - T)^{-1} - (\alpha + \beta)\omega(\beta I - T)^{-1}] \\ &= -\frac{\alpha}{\beta^2}[\alpha\omega T^{-1} - \beta\omega(\beta I - T)^{-1}] \\ &= \left(\frac{\alpha}{\beta}\right)^2(-\omega T^{-1}) + \left(\frac{\alpha}{\beta}\right)\omega(\beta I - T)^{-1}. \end{aligned} \tag{20}$$

Finally, from Lemma 2 and Corollary 1, we get

$$\begin{aligned} \lim_{\lambda \to 0} \lambda^{-1} D_B &= -\frac{\beta}{\alpha} q_z(1, 1, \cdot)T^{-1}e \\ &= \frac{\alpha}{\beta}\frac{1}{\mu}(-\xi T^{-1}e) - \frac{1}{\mu}\xi T^{-1}(\beta I - T)^{-1}E \\ &= \frac{\alpha}{\beta}\frac{\mu_2}{2} + \int_0^\infty e^{-\beta x}\int_x^\infty H^c(s)dsdx. \end{aligned}$$

Normalizing by the expected delay in the M/G/1 queue and integrating by parts gives us (3). □

We next calculate the Laplace Transform $E(e^{-sD} \mid D > 0)$. In light traffic, a customer who is delayed ($D > 0$) will usually see only one customer in the system and will just wait for the service completion. This is made rigorous by Lemma 1. The probability that such a customer arrives and finds the server in phase $k$ is given by

$$\frac{\sum_m m\rho(1, m, k)}{\sum_{m,n} m\rho(1, m, n)},$$

or in terms of $q$ this is

$$\frac{q_z(1, 1, k)}{\sum_n q_z(1, 1, n)}. \tag{21}$$

From (21) we get that

$$E(e^{-sD} \mid D > 0) = \frac{\sum_k q_z(1, 1, k) E_k(e^{-s\tau})}{\sum_n q_z(1, 1, n)}$$

$$= \frac{q_z(1, 1, \cdot)(sI - T)^{-1}\mathbf{E}}{q_z(1, 1, \cdot)e}, \tag{22}$$

where $E_k(e^{-s\tau})$ is the Laplace transform of the remaining service time given that the current phase is $k$, and we have used proposition (1) from Appendix A.

Evaluating the denominator first, from (20) and Corollary 1 we see that

$$[q_z(1, 1, \cdot), e] = \left(\frac{\alpha}{\beta}\right)^2 \frac{1}{\mu}(\xi, e) + \left(\frac{\alpha}{\beta}\right) \frac{1}{\mu} \xi(BI - T)^{-1}\mathbf{E}$$

$$= \left(\frac{\alpha}{\beta}\right)^2 \frac{1}{\mu} + \left(\frac{\alpha}{\beta}\right) \int_0^\infty e^{-\beta x} H^c(x) dx$$

$$= \left(\frac{\alpha}{\beta}\right)^2 \frac{1}{\mu} + \left(\frac{\alpha}{\beta}\right) \frac{1 - \tilde{H}(\beta)}{\beta},$$

where $\xi$ is the vector of ergodic probabilities for the service-time variable and $\tilde{H}(s) = \int e^{-sx} dH(x)$ is the Laplace Transform of the service-time density. The numerator, after some algebraic reduction, becomes

$$q_z(1, 1, \cdot)(sI - T)^{-1}\mathbf{E} = \left(\frac{\alpha}{\beta}\right)^2 \omega[(-T)^{-1}(sI - T)^{-1})\mathbf{E}$$

$$+ \left(\frac{\alpha}{\beta}\right) \frac{\omega(\beta I - T)^{-1}\mathbf{E} - \omega(sI - T)^{-1}\mathbf{E}}{s - \beta}$$

$$= \left(\frac{\alpha}{\beta}\right)^2 \frac{1 - \tilde{H}(s)}{s} + \left(\frac{\alpha}{\beta}\right) \frac{\tilde{H}(\beta) - \tilde{H}(s)}{s - \beta}.$$

Combining the two calculations, we get

$$E(e^{-sD} \mid D > 0) = \frac{\alpha\left[\frac{1 - H(s)}{s}\right] + \frac{\beta}{s - \beta}[\tilde{H}(\beta) - \tilde{H}(s)]}{\alpha \frac{1}{\mu} + 1 - \tilde{H}(\beta)}. \tag{23}$$

It is not difficult to show that (23) is the transform of (10).

## IV. DERIVATION OF THE HEAVY-TRAFFIC RESULTS

In this section we describe the technique that allows us to arrive at (5) and (6). This approximation is derived by first proving that the number of packets in the system, when appropriately scaled, converges to a diffusion process $X(t)$ as $\rho^{(N)}$ converges to one. Given the drift and infinitesimal variance of the diffusion process, one can obtain the steady-state mean of $X(t)$, and by using Little's Theorem, we get the approximation given in (5).

The main theorem of this section, which can be proved rigorously using the techniques in Burman,[12] forms the theoretical basis for (6). We will outline its proof and show in detail how to calculate the mean and variance of the resulting diffusion, which from an application point of view is the difficult part of developing these approximations.

Consider a sequence of processes $X_n(t)$, where the $n$th process represents the number of packets at time $t$ in a single-server system to which packets arrive at an instantaneous rate $\lambda_n$ times the state of an M/M/N/N queueing system (representing the number of virtual calls). The heavy-traffic limits are found by defining $\lambda_n$ as follows:

$$\lambda_n E^{(N)}V = \mu - \frac{\delta}{\sqrt{n}}, \tag{24}$$

where $\delta$ is a positive constant and $E^{(N)}V$ is the expected number of calls in the M/M/N/N system. We are initially interested in studying the sequence of scaled processes $n^{-1/2}X_n(nt)$ as $n \to \infty$.

We start by investigating the limiting behavior of the infinitesimal generators of a sequence of Markov processes created by appending supplementary variables to $X_n(t)$. A limiting generator is identified and the Trotter-Kato Theorem (see Kato[13]) implies that the finite-dimensional distributions converge. The form of the limiting generator completely determines the limiting process. Weak convergence can be established via a theorem of Stroock and Varadhan.[14]

Let $V(t)$ be the number of virtual calls at time $t$ and let $Y(t)$ be the time since the packet currently in service entered service. The multi-dimensional process

$$M_n(t) = [n^{-1/2}X_n(nt),\ Y(nt),\ V(nt)]$$

is Markov. We denote a typical element of the state-space as $(x, y, j)$, where $x = 0, n^{-1/2}, 2n^{-1/2}, \cdots, y \in [0, \infty)$ and $j = \{0, 1, \cdots, N\}$. For the remainder of this paper, let $h(y)$ be the density of the service-time distribution,

$$H^c(y) = \int_y^\infty h(s)ds, \qquad \mu(y) = \frac{h(y)}{H^c(y)}$$

and assume that for some constants $\gamma$ and $\Gamma$,

$$0 < \gamma \leq \mu(y) \leq \Gamma \leq \infty. \tag{25}$$

The implications of and conditions for relaxing (25) are discussed in Burman.[12] These conditions are required for the method of proof and do not pose any significant restrictions on the results.

Let $B$ and $Q$ be the infinitesimal generators of the processes $V(t)$ and $Y(t)$, respectively. The generator $B$ is an $(N+1) \times (N+1)$ matrix given by

$$B = \begin{bmatrix} -\alpha & \alpha & & & & \\ \beta & -(\beta+\alpha) & \alpha & & & \\ & 2\beta & -(2\beta+\alpha) & \alpha & & \\ & & & \ddots & & \\ & & & & N\beta & -N\beta \end{bmatrix}$$

and $Q$ is a first-order linear differential operator given by

$$Qf(y) = f'(y) + \mu(y)[f(0) - f(y)].$$

The following two lemmas are needed to carry out the calculations needed in the main theorem of the section (to be stated below):

*Lemma 3: The equation*

$$Bv = w \tag{26}$$

*has a solution $v$ if and only if*

$$(e, w) = 0, \tag{27}$$

*where $e_k = 1/k!(\alpha/\beta)^k = 1/k!\ \rho^k$. In this case $v$ is unique to an additive constant and is given by*

$$v(k+1) = v(0) + \frac{1}{\alpha} \sum_{j=0}^{k} \frac{j!}{\rho^j} \sum_{i=0}^{j} \frac{\rho^i}{i!} w(i). \tag{28}$$

*Proof:* Since $B$ is a finite-dimensional linear operator corresponding to an ergodic Markov process, it has a one-dimensional null space spanned by the vector $e$. Hence, (27) is both necessary and sufficient for a unique solution of (26) to exist. This solution is easily seen to be given by (28). □

*Lemma 4: The equation*

$$Qf = g \tag{29}$$

*has a solution $f$ if and only if*

$$\int_0^\infty g(y) H^c(y) dy = 0. \tag{30}$$

*In this case, $f$ is unique to an additive constant and is given by*

$$f(y) = \frac{1}{H^c(y)} \int_0^y H^c(s)g(s)ds. \tag{31}$$

*Proof:* The operator $Q$ is the generator of an ergodic Markov process with the stationary density $\mu H^c(y)$. A unique solution exists if and only if (30) holds. Solving the first-order linear differential equation gives (31). □

We are now ready to state the main theorem of this section:

*Theorem 1: Under assumption* (25),

$$n^{-1/2}X_n(nt) \Rightarrow X(t),$$

*where $X(t)$ is a diffusion on $R^+$ with pure reflection at the origin, downward drift $\delta$ and infinitesimal variance $s^2$ given by*

$$s^2 = \frac{1}{2}\left\{\lambda E^{(N)}V + \mu^3\sigma^2 + \sum_{k=0}^{N} \frac{1}{\alpha P_k}\left[\sum_{j=0}^{k-1} P_j(\mu - j)\right]^2\right\}, \tag{32}$$

*where*

$$P_k = \frac{1}{k!}\left(\frac{\alpha}{\beta}\right)^k \Big/ \sum_{j=0}^{N} \frac{1}{j!}\left(\frac{\alpha}{\beta}\right)^j$$

*is the steady-state probability of finding $k$ calls in the $M/M/N/N$ blocking system.*

*Proof of Theorem:* As mentioned earlier, we will show how to identify the mean and variance of (32). The infinitesimal generator for $M_n(t)$ is given by

$$A_nf(x, y, j) = n(B + Q)f + \lambda j\left[f\left(x + \frac{1}{\sqrt{n}}, y, j\right) - f(x, y, j)\right]$$
$$+ \mu(y)\left[f\left(x - \frac{1}{\sqrt{n}}, 0, j\right) - f(x, 0, j)\right] \quad \text{for} \quad x \geq \frac{1}{\sqrt{n}}, \tag{33}$$

and

$$A_nf(0, 0, j) = nBf + \lambda j\left[f\left(\frac{1}{\sqrt{n}}, 0, j\right) - f(0, 0, j)\right]. \tag{34}$$

For $f(x)$ twice continuously differentiable with $f'(0) = 0$, set

$$f_n(x, y, j) = f(x) + \frac{1}{\sqrt{n}} f'(x)g(y, j) + \frac{1}{n} f''(x)h(y, j). \tag{35}$$

We construct bounded functions $g$ and $h$ so that

$$A_nf_n \to Af$$

uniformly, where

$$Af(x) = mf'(x) + s^2 f''(x).$$

The constants $m$ and $s^2$ are determined in the process. Given (25), these constructions will complete the proof (see Burman[12]).

From (35) and (33) we see that

$$\begin{aligned} A_n f_n = {} & n(B+Q)f + \sqrt{n} f'(x)[\lambda_n j - \mu(y) + (B+Q)g] \\ & + f''(x)\left\{\frac{1}{2}[\lambda_n j + \mu(y)] + \lambda_n j g(y,j) - \mu(y) g(0,j)\right. \\ & \left. + (B+Q)h(y,j)\right\} + 0\left(\frac{1}{\sqrt{n}}\right). \end{aligned} \tag{36}$$

First note that since $f$ is independent of $y$ and $j$, $(B+Q)f = 0$. Set $g$ equal to the solution of the equation

$$(B+Q)g = -[\lambda_n j - \lambda_n E^{(N)} V - \mu(y) + \mu]. \tag{37}$$

Lemmas 3 and 4 imply that $g$ exists and

$$\begin{aligned} g(y,j) = {} & \frac{1}{H^c(y)} \int_0^y H^c(s)[\mu(s) - \mu] ds \\ & - \frac{\lambda_n}{\alpha} \sum_{k=0}^{j-1} \frac{k!}{\rho^k} \sum_{i=0}^{k} \frac{\rho^i}{i!} [j - E^{(N)} V]. \end{aligned} \tag{38}$$

Let $h$ be the solution of the equation

$$(Q+B)h = -[w(y,j) - s^2], \tag{39}$$

where $w(y,j) = 1/2[\lambda_n j + \mu(y)] + \lambda_n j g(y,j) - \mu(y) g(0,j)$. The constant $s^2$ [see (26)] can be easily calculated from conditions (27) and (30), i.e.,

$$\sum_{i=0}^{N} \frac{\rho^i}{i!} \int_0^\infty H^c(y) w(y,i) dy = 0.$$

Using $g$ and $h$ calculated above we have

$$A_n f_n = -\delta f'(x) + s^2 f''(x) + 0\left(\frac{1}{\sqrt{n}}\right).$$

When $x = 0$ analogous calculations give that $A_n f_n$ converges provided $f'(0) = 0$, and the proof is complete. □

The limit theorem for the M/G/1 queue with arrival rate $\lambda_n E^{(N)} V$ (see Iglehart and Whitt[15]) gives the same mean $\delta$ with the variance $\lambda E^{(N)} V + \mu^3 \mu_2$ for the limiting diffusion. Calculating the steady-state expected number in the limiting systems and using Little's Theorem

(see Cooper[16]) yields the asymptotic expression for the ratio of $D_B^{(N)}$ and $D_M^{(N)}$, the delays in the two systems:

$$\frac{D_B^{(N)}}{D_M^{(N)}} \approx 1 + 2\,\frac{\sum_{k=0}^{N-1} \frac{1}{P_k} \left[\sum_{i=0}^{k} P_i(\mu - \lambda i)\right]^2}{\alpha\mu^3\mu_2}. \tag{40}$$

Letting $N$ go to infinity, $P_k$ goes to the Poisson distribution and (40) goes to

$$1 + 2\mu/\alpha\left(\frac{1}{1 + C^2}\right)$$

as promised. □

*Remark:* We are careful here not to write (40) as a limit but as an approximation. In order to strictly prove convergence in (40), two rather technical steps remain to be proved. The first is that the steady-state number in the queue (when scaled) converges to the steady-state value of the diffusion. The second is that the expected values of these steady states converge. Both are difficult issues in themselves and, aside from mathematical completeness, add little information to the approximation, which is the germaine issue of this paper.

## V. CONCLUDING REMARKS

We considered here a single-server queueing system with a nonhomogeneous Poisson arrival process whose rate is some constant $\lambda$ times the state of an independent M/M/$\infty$ system. In this paper, we derived limiting values for the mean delays as the traffic intensity goes to zero and to one.

This system was used to model the delay of packets at a packet switch. In this setting, an individual data customer generates packets at a constant rate $\lambda$ and the number of customers generating packets is given by the state of the M/M/$\infty$ system. This is, of course, not the only possible model for packet arrivals. Packets generated by an individual virtual call may be "smoother" or more "bursty" than packets generated by a Poisson process. It is also possible that restrictions could be imposed on the total number of simultaneous virtual calls, thereby making the infinite-server assumption unrealistic. Work is currently under way to extend the techniques of this paper to cover these and other more general models of bursty traffic.

Our ultimate goal in studying $D_B/D_M$ in both light and heavy traffic is to be able to derive simple, closed-form approximations for $D_B$ for all values of the traffic intensity $\rho$. One candidate approximation is to

linearly interpolate between the values obtained for $D_B/B_M$ when $\rho \to 0$ and when $\rho \to 1$ [see (4) and (5)]. This approach gives the following estimate for the mean delay:

$$D_B \sim \frac{\rho}{1-\rho}\frac{(1+C^2)}{2\mu}\left[1+\frac{\mu}{\alpha}\frac{1-(1-\rho)\phi(\beta)}{(1+C^2)}\right].$$

More sophisticated interpolations are possible (see Lauber and Smith[7] for applications of these techniques to estimating the delays in an M/G/c queue); however, in the absence of any test data, verification is difficult. One method of verification is to study the accuracy of these approximation techniques when applied to other models of bursty arrivals for which explicit expressions are known (see Yechiali and Naor[17]). This work is currently under way and will be reported on in the future.

## REFERENCES


1. H. Heffes, "A Class of Data Traffic Processes—Covariance Function Characterization and Related Queueing Results," B.S.T.J., *59*, No. 6 (July–August 1980), pp. 897–929.
2. A. Kuczura, "Queues with Mixed Renewal and Poisson Input," B.S.T.J., *51*, No. 6 (July–August 1972), pp. 1305–26.
3. R. V. Laue, unpublished work.
4. M. F. Neuts, "A Versatile Markovian Point Process," J. Appl. Prob., *16* (December 1979), pp. 764–79.
5. D. M. Lucantoni and M. F. Neuts, "Numerical Methods for a Class of Markov Chains Arising in Queueing Theory," Technical Report No. 78/10 (May 1978), Department of Statistics and Computer Science, University of Delaware.
6. D. Anick, D. Mitra, and M. M. Sondhi, unpublished work.
7. P. J. Lauber and D. R. Smith, unpublished work.
8. D. Mitra and D. Anick, unpublished work.
9. L. Kosten, "Stochastic Theory of a Multi-Entry Buffer (1)," Delft Progress Report, *1*, 10–18, 1974.
10. D. Y. Burman and D. R. Smith, "A Light Traffic Theorem for Multi-Server Queues," Math. Oper. Res., *8*, No. 1 (February 1983), pp. 15–25.
11. D. R. Smith, "Optimal Repairman Allocation—Asymptotic Results," Management Science, *24*, No. 6 (February 1978), pp. 665–74.
12. D. Y. Burman, unpublished work.
13. T. Kato, *Perturbation Theory for Linear Operators*, Berlin: Springer-Verlag, 1976.
14. D. W. Stroock and S. R. S. Varadhan, *Multidimensional Diffusion Processes*, Berlin: Springer-Verlag, 1979.
15. D. L. Igelhart and W. Whitt, "Multiple Channel Queues in Heavy Traffic," I. *Advances in Applied Probability*, 2, No. 1 (Spring 1970), pp. 150–77.
16. R. Cooper, *Introduction to Queueing Theory*, New York: North Holland, 1981.
17. V. Yechiali and P. Naor, "Queueing Problems with Heterogeneous Arrivals and Service," Oper. Res., *19*, No. 3 (May–June 1971), pp. 722–34.
18. J. Keilson, *Markov Chain Models—Rarity and Exponentiality*, New York: Springer-Verlag, 1979.


## APPENDIX A

### *Background on Phase-Type Distributions*

Appendix A summarizes results for phase-type distributions that are used in Section III. A service time is said to have phase-type distribution if it is distributed like the first exit time from a continu-

ous-time, finite-state Markov chain. Any distribution can be arbitrarily approximated in the sense of weak convergence by one of phase type. We assume that the states (also known as phases) are the nonnegative integers $i$, $i = 1, \cdots, m$. Let the rate of transition from phase $i$ to phase $j$ be $T_{ij}$ $(i \neq j)$, and the rate of exiting from phase $i$ be $\xi_i$. Define $T_{ii} = -\sum_{j \neq i} T_{ij} - \mathrm{E}_i$, i.e., minus the rate of leaving state $i$. We assume that $T$, the matrix of $T_{ij}$, is invertible; this is sufficient to imply that the real part of the spectrum of $T$ is strictly negative. A customer starts service in phase $i$ with probability $\omega_i$. Following Neuts,[4] we use the notation $(\omega, T)$ to describe this distribution, where $\omega$ is the vector of initial probabilities and $T$ is the $m$-dimensional matrix of rates.

The vector $\xi$ of ergodic probabilities for the service phase of the renewal process is determined by normalizing the solution to the equations

$$-\xi_i T_{ii} = \sum_j \xi_j T_{ji} + (\Sigma \xi_j \mathrm{E}_j)\omega_i, \tag{41}$$

or in matrix notation,

$$\xi T = -(\xi, \mathrm{E})\omega. \tag{42}$$

The service rate $\mu$ is defined as

$$\mu = (\xi, \mathrm{E}). \tag{43}$$

It is not difficult to show that $\mu$ is the reciprocal of the mean service time.

Let $\tau$ be the first exit time from the chain; let $\mathrm{M}_i(s) = E_i(e^{-s\tau})$, the Laplace Transform of $\tau$ given the initial state is $i$; and $M_i^n = E_i \tau^n$. The results of Proposition 1 are well known (see Kielson,[18] page 82, and Burman and Smith[12]).

*Proposition 1.* For $\tau$, $\mathrm{M}(s)$, and $\mu^n$ defined above,

$$\mathrm{M(s)} = (\mathrm{sI} - \mathrm{T})^{-1}\mathrm{E} \tag{44}$$

and

$$M^n = n!(-1)^n T^{-n} e, \tag{45}$$

where $e$ is the vector of all ones. In particular, the $n$th moment of the service-time distribution $(\tilde{M}^n)$ is

$$\tilde{M}^n = n!(-1)^n(\omega, T^{-n}e). \tag{46}$$

If $H^c(x)$ equals the tail of the service-time distribution, then the following corollary is immediate.

*Corollary 1.*

$$-\xi T^{-1}e = \int_0^\infty x\mu H^c(x)dx = \frac{\mu\mu_2}{2}, \tag{47}$$

and

$$-\xi T^{-1}(\beta I - T)^{-1}\mathrm{E} = \int_0^\infty e^{-\beta x}\int_x^\infty \mu H^c(s)dsdx. \tag{48}$$

*Proof:* To see (47), observe that $\mu H^c(x)$ is the density of the remaining service time when the process is sampled in equilibrium, and apply Proposition 1. The identity on the integral can be obtained by integration by parts. In a similar fashion,

$$\xi T^{-1}(T - \beta I)^{-1}\mathrm{E} = \frac{\xi[(T - \beta I)^{-1} - T^{-1}]\mathrm{E}}{\beta}$$

$$= \frac{\left[\int_0^\infty \mu H^c(x)dx - \int_0^\infty e^{-\beta x}\mu H^c(x)dx\right]}{\beta},$$

and integration by parts gives us (48). □

## APPENDIX B

### *Intuitive Derivation of the Results in Light Traffic*

The following argument is based on the intuitive notion that, in light traffic, almost all arriving packets arrive to an empty system and are served before the arrival of another packet. Furthermore, the times when there is exactly one packet in the system constitute almost all of the time in which an arriving packet might be subject to delay. Thus, in light traffic, it is easy to derive the proportion of time in which an arrival will be delayed, and to find the delay for such an arrival. A slight complication is introduced in the analysis by the fact that the arrival rate of packets is not constant, but this is easily overcome.

While the overall derivation is not rigorous, it is convincing and gives insight into the light-traffic behavior. The results, of course, are consistent with the rigorous results derived for phase-type distributions in Section III.

We will begin with traffic-independent results for the model described previously in the beginning of Section II, in which the instantaneous arrival rate equals $\lambda$ times the state of an M/M/$\infty$ queue with birth rate $\alpha$ and service rate equal to $\beta$. The equilibrium distribution of the M/M/$\infty$ queue governing the arrival process is easily seen to be Poisson with mean $\alpha/\beta$. However, this is not its distribution immediately after an arrival, since the rate of arrivals is proportional to the

state of the M/M/∞ queue. Thus, the conditional probability that the state of the M/M/∞ queue is $m$ (for $m \geq 1$) immediately after an arrival is

$$\frac{m(\alpha/\beta)^m \frac{1}{m!} e^{-\alpha/\beta}}{\sum_{k=0}^{\infty} k(\alpha/\beta)^k \frac{1}{k!} e^{-\alpha/\beta}} = \frac{1}{(m-1)!} (\alpha/\beta)^{m-1} e^{-\alpha/\beta}.$$

This may be thought of as the distribution of a Poisson random variable with mean $(\alpha/\beta)$ with support right-shifted one unit. This point of view is useful in computing the distribution of the state of the M/M/∞ queue $t$ time units after an arrival of a packet (denoted $N_t$), since the Poisson distribution is stationary, and the additional unit is still present with probability $e^{-\beta t}$. Thus,

$$E(z^{N_t}) = e^{(\alpha/\beta)(z-1)}[1 + e^{-\beta t}(z-1)]. \tag{49}$$

Now consider the system in light traffic. Denote the state of the system by $(N, M, T)$ where $N$ is the number of packets in the system, $M$ is the state of the M/M/∞ queue that modulates the arrival process, and $T$ is the remaining service time of the packet being served.

We first obtain an expression for $E(e^{-sT}z^M\delta_{1N})$ in light traffic, where $\delta_{1N} = 0$ for $N \neq 1$ and $\delta_{11} = 1$. We then show how to use this quantity to obtain the desired limits. In light traffic, almost every packet arrives to an empty system and is served before another arrival. By Little's Law

$$P(N = 1) = E\delta_{1N} \sim \lambda\alpha E(S)/\beta.$$

Again, making the assumption of the last sentence and looking at the system at only those times for which $N = 1$, we obtain by standard renewal theory arguments that

$$E(e^{-sT}z^M N = 1) = \frac{1}{ES} E\left[\int_0^S E(z^{N_t}) e^{-s(S-t)} dt\right] \stackrel{\text{def}}{=} \frac{1}{ES} G(z, s).$$

Multiplying, we obtain that

$$E(e^{-sT}z^M\delta_{1N}) \sim \lambda(\alpha/\beta)G(z, s). \tag{50}$$

By conditioning on $S$ and using (49), we obtain

$$G(z, s) = e^{(\alpha/\beta)(z-1)} \frac{1}{s} \left\{[1 - \tilde{H}(s)] + \left(\frac{z-1}{\beta - s}\right)[\tilde{H}(s) - \tilde{H}(\beta)\right\}, \tag{51}$$

where $\tilde{H}$ is the Laplace Transform of a service time.

Equations (50) and (51) can be used to find desired properties of the queue in light traffic. For example, since nearly all packets that

are delayed are generated when there is only one other packet in the system, and since the rate of generation of packets is $\lambda$ times the number of calls, we find the rate of generation of delayed packets, $\lambda_D$, is

$$\lambda_D \sim \lambda^2(\alpha/\beta)G_z(1, 0), \tag{52}$$

or

$$\lambda_D \sim \lambda^2 \left\{(\alpha/\beta)^2 \frac{1}{\mu} + \alpha/\beta[1 - \tilde{H}(\beta)]\right\}. \tag{53}$$

The Laplace Transform of the delay, given that the delay is greater than 0, defined to be $\phi(s)$, is also found easily since the delay of a packet equals the remaining service time on arrival. Thus,

$$\phi(s) \sim \frac{G_z(1, s)}{G_z(1, 0)},$$

or

$$\phi(s) \sim \frac{\alpha/s[1 - \tilde{H}(s)] + \dfrac{\beta}{\beta - s}[\tilde{H}(s) - \tilde{H}(\beta)]}{\alpha/\mu + 1 - \tilde{H}(\beta)}. \tag{54}$$

Inversion of the Laplace Transform gives (10) and also gives

$$E(D \mid D > 0) \approx \frac{\dfrac{\alpha\mu_2}{2} + \dfrac{1}{\mu} - \dfrac{1}{\beta}[1 - \tilde{H}(\beta)]}{\alpha/\mu + 1 - \tilde{H}(\beta)}, \tag{55}$$

where $\mu_2 = E(S^2)$. Of course, $P(D > 0) = \dfrac{\lambda_D}{\lambda(\alpha/\beta)}$ so that (52) and (54) give

$$E(D) \approx \frac{\lambda}{\beta}\left\{\frac{\alpha\mu_2}{2} + \frac{1}{\mu} - \frac{1}{\beta}[1 - \tilde{H}(\beta)]\right\}. \tag{56}$$

This may be combined with the Pollaczek-Khintchine formula to give (3).

## AUTHORS

**Donald R. Smith,** A.B. (Physics), 1969, Cornell University; M.S. (Operations Research), 1974, Columbia University; Ph.D. (Operations Research), 1975, University of California, Berkeley; Bell Laboratories, 1980—. Before joining Bell Laboratories, Mr. Smith was employed at Adaptive Technology, Inc. from 1970 to 1974, and was Assistant Professor in the Department of Industrial Engineering and Operations Research, Columbia University from 1975 to 1979. At Adaptive Technology, Mr. Smith developed mathematical models for

new techniques in statistical multiplexing. At Bell Laboratories he is in the Operations Research Department of the Network Analysis Center pursuing interests in applied stochastic processes.

**David Y. Burman,** B.S. (Mathematics), 1968, C.C.N.Y.; Ph.D. (Applied Mathematics), 1979, New York University; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Burman has worked on problems in network flows, scheduling, and stochastic processes.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

*Modernization of the Suburban ESS:*

# Overview: Evolution of the Suburban ESS

By T. E. GRASSMAN* and J. E. YATES*

(Manuscript received June 8, 1982)

This paper highlights the systematic evolution of the No. 2/2B Electronic Switching System (ESS), describing both hardware and software changes. It also introduces the subsequent articles that describe the modern interfaces and modern development environment enjoyed by the system. The No. 2 ESS was the first electronic switching system specifically designed for the suburban community. Subsequent incorporation of a more flexible and powerful processor provided a basis for a large increase in processing capacity and a continuing modernization of system structure. The evolved system, the No. 2B ESS, is now rich in features and offers modern administrative and maintenance capabilities.

## I. INTRODUCTION

The No. 2 Electronic Switching System (ESS),[1] designed in the 1960s to provide ESS capabilities to suburban communities, was developed under the same constraints as other large real-time control systems of that era. Memory was expensive and processors were not powerful enough to compensate for inefficiencies in software. As a

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

result, programs were written in assembly language and were tightly coupled. Compared with No. 1 ESS, then being deployed in urban areas, the No. 2 ESS faced a more stringent cost sensitivity due to the smaller number of subscribers supporting a given local switching system.

The original No. 2 Processor was replaced in 1976 with the 2B Processor, which has a more modern architecture based on improved semiconductor technology. This more powerful processor provided capacity for feature enhancement[2] and allowed utilization of more generalized and structured software techniques that have subsequently been incorporated into the system. This incorporation was accomplished by strictly controlling the portions of the software system that were allowed to change. Since that time the programs have continued to evolve and exploit the capabilities of the 2B Processor.

Most recently, the inclusion of modern software constructs in the 2BE3 generic has produced a design containing a modern data link interface that supports such capabilities as the 10A Remote Switching System (RSS), Automatic Message Accounting (AMA) data teleprocessing, and traffic data teleprocessing. Description of these features may be found in the companion articles of this series.

The evolutionary approach used to realize the present No. 2B ESS has resulted in a continuously viable product that remains economically attractive and modern in feature content. This paper will discuss the system evolution, and serve as an introduction to the other papers in this series.

## II. HARDWARE AND SYSTEM CHARACTERISTICS

### *2.1 No. 2 ESS*

The original No. 2 ESS central processor was designed using Transistor Resistor Logic (TRL) in the early 1960s. The architecture consisted of special purpose registers that were designed to accommodate a set of instructions tailored for use in telephone switching. For example, explicit instructions were included to support the software structure that used 8-word control blocks to process each telephone call. Its architecture and instruction set gave the No. 2 ESS the capability to efficiently switch telephone calls in a small to medium (2,000- to 20,000-line) office.

The hardware architecture used in the No. 2 ESS Processor supported two types of stores (memories). One of the stores was a permanent magnet twistor that contained the program as well as the data which specified the characteristics of lines and trunks. This 22-bit-wide memory was magnetically alterable off-line. The memory organization consisted of four 64K modules for a maximum of 256K

words. The selection of a module was independent of the normal addressing mode within a module. This required each program to independently administer the module selection for itself. The other store contained up to 32K 16-bit words and was on-line writable. It contained dynamic information related to the processing of calls.

An additional characteristic of No. 2 ESS was a wired logic unit that autonomously scanned for customer requests, and, under timed control, scanned for short-interval signals. Both of these logic control items deloaded the program and increased the efficiency of the overall system. Scanners and other peripheral units were controlled by a parallel bus system.

This processor was put into service in 1970 in the No. 2 ESS system and provided the desired service level, features, and cost benefits at the time of its initial cutover.

### *2.2 The No. 2B ESS*

#### *2.2.1 Background*

In 1971, a series of cost reduction studies led to the selection of the more powerful 3A Processor Control Complex (3A CC)[3] for incorporation into the No. 2 ESS. The 3A CC was designed for use in both No. 2B ESS and No. 3 ESS. The 3A CC was designed along general-purpose computer principles rather than on the specific switching logic instructions incorporated in the original No. 2 ESS Processor, and made extensive use of integrated circuits. The new processor could address up to one million words of relatively inexpensive semiconductor memory, provided a general-purpose instruction set and a serial bus access to peripheral units, and used a microprogram-based instruction decoder.

The principal asset of the 3A CC Processor was its microprogrammed instruction set. This feature allowed the processor to not only encompass the No. 3 ESS commands, but also to be extended in a relatively simple manner to emulate the original No. 2 ESS instruction set, and to add additional instructions that were parallel in nature to the No. 2 ESS set. This allowed the retention of most of the programs that had previously been developed to provide features and maintenance on the existing No. 2 ESS (except for the processor maintenance itself).

To minimize changes in the control of major peripheral equipment, a new interface [the 2B Input/Output (I/O) unit] was developed to connect the 3A CC to the parallel peripheral bus system. (The combination of the 3A CC, the 2B I/O unit, the increased memory, and the microcode for emulation is referred to as the 2B Processor.) In addition, programs already developed for the 3A CC, e.g., teletype-

writer (TTY) handlers and processor maintenance software, were adopted intact to avoid duplicate software development.

### *2.2.2 Instruction emulation*

The key element in replacing the No. 2 Processor was the emulation of the existing No. 2 ESS instruction set. The existing 22-bit data and instruction format of the No. 2 ESS did not fit in the 16-bit 3A CC memory. Converting each No. 2 ESS word to two 3A CC words would have resulted in an addressing incompatibility between the old and new systems, i.e., the same instruction would exist at different addresses in the two systems. If this were allowed to occur, object-level changes to correct field problems would have to be developed twice, once for the No. 2 ESS and again for the No. 2B ESS. By keeping the object-level addresses synchronized, this duplicate effort was avoided. The decision was made to expand the 3A CC store to 24 usable bits to allow exact emulation of No. 2 ESS instructions and data. Half-word instruction pairs could be accommodated in the same addressable word, and data packing was then identical in the two processors. A further decision was made to use the No. 2 ESS instruction manual as the requirements for the emulated instructions and to map the functional No. 2 ESS registers into the general-purpose registers of the 3A CC. Each register and instruction was fully emulated.

These restrictions gave several distinct advantages although they did not allow improvements in program techniques through use of the richer 3A CC instruction set. The advantages were as follows: a test program that previously tested the instruction logic on No. 2 ESS provided the acceptance test for the microprogrammed instruction set (i.e., no new program was required); an immediate check on correct program compilation was available by comparing raw program sizes, symbol values, and linkage characteristics between the existing and the emulated programs; debugging of these programs was required only on one machine (No. 2 or 2B Processor) since they could be automatically assembled from the same source for the other machine without human interference (this allowed nearly simultaneous program releases on the two different systems); and most existing documentation and procedures for the No. 2 ESS were usable with the 3A CC since most user interfaces were identical between the systems.

With these constraints, the bulk of No. 2 ESS programs (85 percent) could be used as they were currently written. Most of the remaining 15 percent were No. 2 ESS Processor maintenance programs, which were replaced in total by the 3A CC Processor maintenance programs. The programming effort of the 2B introduction then was completely restricted to the support programs (assemblers, loaders, etc.), six

interface programs, and new diagnostics and recovery programs for the wide store.

#### *2.2.3 Input/output emulation*

The No. 2 ESS wired logic I/O performed two functions. It looked for new customer service requests and collected the digits as they were dialed by the customer. In the new system, software was used to scan for service requests. The digit collection functions were exactly duplicated by an interrupt-level microcode sequence on the 3A CC. This provided an unchanged interface to existing programs that had depended on the wired logic. The only associated changes required were in recovery programs for handling digit collection failures.

The interface between the 3A CC and the No. 2 peripheral equipment was provided by the 2B I/O unit. This was implemented to be driven by the existing 3A CC serial channel interface. Driver programs were implemented in microcode so that controlling programs, issuing I/O instructions, were not required to change.

### *2.3 No. 2B ESS field introduction and evolution*

The No. 2B ESS was successfully put into service in 1976. Less than one year later the same program was used to provide service in an office requiring increased capacity through a retrofit of the processors. Since the 3A CC is much faster than the No. 2 ESS Processor, the No. 2B ESS call capacity (35,000 calls/hour engineered load) is much greater than that of the No. 2 ESS (16,500 calls/hour engineered load). By 1977, No. 2 ESS Processors were no longer being manufactured.

The No. 2/2B ESSs continued to evolve gradually—and in step—for several years while their programs were kept locked together by the restrictions discussed above. As the number of in-service No. 2 ESSs began to decline through processor retrofits, and the pressure to provide new and expanded features increased, a decision was made to discontinue new development on the No. 2 ESS, and to take further advantage of the 3A CC Processor's capabilities in No. 2B ESS.

The 2BE3 generic program, put into service in November of 1981, utilizes significant software restructuring in certain areas. The resulting advantages range from achieving expanded capacities for per-line features (e.g., call forwarding) to the development of several major new features and the communications interface that supports them.

### *2.4 A modern data communications interface*

The need for data communications between local switching offices and other systems (remote switches and various operation support

systems) has expanded in recent years. To meet this increasingly important need, the No. 2B ESS has incorporated, in the 2BE3 generic program, a modern data communications interface capability. This interface is based on the Serial Peripheral Unit Controller/Data Link (SPUC/DL), a cost-effective and flexible hardware element utilizing microprocessor control capable of supporting various communications protocols. The 3A CC serial channel I/O capability provides the high-speed communication path between the host and the SPUC/DL. An accompanying article, "A New Data Link Controller," describes the SPUC/DL and its architecture, design, and capabilities in considerable detail.

Firmware has been developed for the SPUC/DL to support levels 1 and 2 of the BX.25 data communications protocol. This capability is utilized in provision of interfaces to the Automatic Message Accounting Collection System (AMACS) and to the Engineering and Administrative Data Acquisition System (EADAS). Firmware has also been developed to support the RSS protocol, which predated the BX.25 standard, enabling the No. 2B ESS to provide control of the 10A RSS. The features associated with these interfaces are also described later in this series.

## III. NO. 2B ESS SOFTWARE EVOLUTION

Soon after the No. 2B ESS was successfully cut into service, an effort was begun to exploit the capabilities of the new processor. This effort, which had been planned from the beginning, aimed at eliminating data and program constraints. The success of these initial enhancements led to other incremental improvements. The program has matured to the point that enhancements driven by modern software practices, as well as the 2B Processor's architecture, are being pursued. These include operating-system-type primitives, data independence, and program decoupling.

This section will review some enhancements that demonstrate progress in a variety of areas. The techniques used are not new nor innovative by current standards. What is perhaps unique is the manner in which they have been applied to a large, complex, existing system.

The enhancements have been pursued in a systematic fashion. First, the software was extended to utilize the capabilities of the 2B Processor while still remaining functionally compatible with the original No. 2 ESS software. Commercial versions of these compatible programs were released in 1976 and 1977. Eventually the compatibility restriction was lifted and the software was extended to exploit noncompatible 2B capabilities. In about the same time frame, modifications to enhance functional and data independence were added.

### *3.1 2/2B compatible extensions*

Functional compatibility between the No. 2 ESS and the No. 2B ESS at the time of conversion was important not only to minimize risk, but also to ensure that future enhancements applied to both systems. The types of enhancements that could be introduced were severely limited by this restriction. Generally, they took the form of increasing the maximum quantity of a particular resource that the system could handle.

A prime example is the memory spectrum. Immediately following the conversion, the 2B Processor was limited to the 256K-word spectrum of the No. 2 Processor simply because the software was tailored to handling 18-bit addresses. The program modifications required to handle 20-bit data addresses and exercise the full megaword spectrum of the 2B Processor were scattered throughout the software. A pervasive change of that nature with its attendant risk is precisely the kind of complication that was intentionally avoided during the initial conversion process. With the converted system functioning well, exploiting the million-word address spectrum could be attacked as a separate problem. It was scheduled, designed, and implemented in 1976.

The implementation modified all affected programs to be able to handle 20-bit addresses. Since both the 2 and 2B versions were modified, they remained compatible and both were able to handle the larger addresses. The No. 2 version would, of course, never encounter an address that exceeded 256K.

Similar extensions followed in a well-defined and controlled manner. The maximum number of scanners, peripherals used to sense the states of circuits and/or lines, was increased from 12 to 31. The number of buffers utilized to control peripheral units was increased from 12 to 20. Each extension was scheduled and implemented separately, thereby avoiding the coincident introduction of complex changes.

### *3.2 Noncompatible extensions*

The number and nature of compatible extensions are limited. To make more fundamental structural changes to the software system required exploiting aspects of the 2B Processor for which there is no No. 2 counterpart. As an example, one could not begin to use the superior 2B instruction set since the No. 2 Processor did not contain those instructions. In like manner, the larger No. 2B ESS memory could not be utilized for program instructions. In 1977, the decision was made to decouple the new and the old systems, thereby eliminating the compatibility constraint on the program system.

This decision opened new vistas for improving the structure of the software system. The first step was to simply eliminate methodology

and operations that existed only because of the No. 2 Processor. For example, the No. 2 Processor cannot support relocatable programs. The 2B software system has now been converted to be mostly relocatable. Certain instruction sequences can be coded more efficiently using the 2B instruction set. Some sequences, in fact, were not needed at all and were simply eliminated. We were particularly fortunate to have available the excellent text processing facilities provided by the *UNIX** time-sharing system to aid in the search for these sequences.[4] These changes were collectively called program "clean up" and were performed as the first step in moving primary system development to the No. 2B ESS.

### *3.3 Modernizing the control structure*

#### *3.3.1 Isolating application software from the control programs*

One definition of an operating system is a set of standard functions that provide a hospitable environment for application programs. The call processing control mechanism in the No. 2B ESS did not supply this feature. Nevertheless, certain steps toward providing common, general functions for use by application programs were possible. Emphasis was on the introduction of these functions with a minimal perturbation to the overall system. In particular, designs that did not cause existing programs to become inoperative were selected for the implementation of these functions. Once a function is available, all new development can use it. On an independent schedule and subject to independent evaluation, a plan could be developed to convert existing application programs to use the new functions.

#### *3.3.2 Isolating application software from a resource manager*

This philosophy of gradual introduction has also been used to reduce the coupling between an existing resource manager and the application programs. The manager is responsible for finding and allocating paths through the switching network. To efficiently use the network, it attempts to share a portion of an existing path with the next path to be found for the same call. The shared portion is known as the A link and the operation is known as A-link sharing. As originally designed, the manager provided the mechanism, but the application programs were required to save the identity of any A link to be shared and to specifically request sharing when invoking the manager.

Consistent with this new philosophy, a universal mechanism for sharing A links has now been implemented. The path manager now has complete control over the sharing mechanism. Sharing is at-

* Trademark of Bell Laboratories.

tempted in every situation and not just at the discretion of the application program. The A-link sharing information has been eliminated from the application interface. The significant point is that once again this was accomplished in a fashion that did not require the wholesale change of existing programs. Unchanged application programs would still go through the motions of making A-link sharing decisions, but these would be ignored by the manager. In no way would the application program affect or compromise the control now centered in the path manager. All known A-link code was in fact stripped out of application programs for reasons of clarity and efficiency. Significantly, this was a conscious decision and not a foregone conclusion. In the event that some application A-link code was not detected, only the efficiency of the system and not its integrity was affected.

## IV. MODERNIZING THE DATA STRUCTURE

### *4.1 Isolating application software from data*

The original application interface to the No. 2 ESS database system (it was known as translations in those days) was reasonably well structured. All of the update routines were concentrated in one place, as were the access routines. Most application programs even used the access routines to retrieve the data. The fundamental problem was the transparency of the access routines. While they shielded the user from the gross structure of the data, returning the address of an individual record was not unusual. That practice, while promoting the ultimate in efficiency, resulted in the known record formats propagating throughout the application programs.

This data problem is more insidious than the corresponding control problem. It can be, and was, attacked with the same philosophy of installing the new without breaking the old. While exhibiting many of the same benefits as the new control interface, the database does not really pay dividends until the new interface is used to streamline the access and packing efficiency by allowing the records to be reformatted. All programs, including preexisting ones, had to be decoupled from the data by the new interface before any records containing that data could be modified.

Despite this observation on the general intractability of the problem, where it made sense to reorganize a portion of the database, a two-phased approach was used. The application programs were first decoupled from the data by a uniform and opaque interface supplied by the database programs. Since the underlying data structures did not change, converted, nonconverted, and "unaffected" data accesses continued to operate properly. The "unaffected" class contained some accesses for which the need for conversion was not predicted. The

existence of these oversights did not affect the operation of the system during the entire conversion process. Even assuming perfect analysis, the two-phased approach eliminated any coordination in the data access conversions. It was only necessary that all conversions occur before the database was modified.

The data were reformatted during the second phase. When the data changed, implicit and embedded data dependencies that were overlooked in the first phase surfaced and were dealt with. Certainly the disruption of the system was minimized by a two-phased approach. In the case of the No. 2B ESS conversion being reported herein, the entire decoupling of the application software from the data was not accomplished during Phase 1. Some converted code had to be reexamined and modified during Phase 2. Nevertheless, the final result was very successful based on the relative absence of conversion problems uncovered during testing. This method allowed the complete reformatting of the originating and terminating translators, which in turn allowed more features and provided a more regular data structure that could be enhanced gracefully.

### *4.2 Isolating application software from hardware*

Superimposing a logical structure and nomenclature on top of the physical one is a well-established mechanism for insulating application software from the details of the real world. Application programs typically direct output, not to a real device, but to a logical channel number. The operating system is charged with mapping this logical number to a real device. Unfortunately, the need for similar flexibility in identifying terminals of the No. 2 ESS switching network was not foreseen. There was but one type of network whose terminals were universally identified by their actual equipment location. With the introduction of a second type of network, the use of equipment location numbers to identify terminals became ambiguous. Rather than imbuing all application programs with the knowledge of two network types, a logical identification scheme similar to the one used for channels was created. The logical scheme identifies each terminal by a unique virtual equipment number. A mapping algorithm is used to go from real to virtual equipment number and vice versa.

The merits of a virtual numbering scheme are self-evident and will not be discussed. The feasibility of introducing a virtual numbering scheme into a large, complex software system developed almost two decades ago for a modest development effort is less evident. The 2BE3 generic program is an existence proof.

First, a few sensible limitations were imposed. The spectrum of the virtual equipment numbers was made to coincide with that of the original real equipment numbers. This, of course, means that the

maximum number of actual terminals of all types is still the same. The required performance of the No. 2B ESS is such that this is a reasonable restriction. Making the spectrums compatible means that all existing programs and data structures could at least handle the identifying number. The change would, therefore, be entirely transparent to all programs that used the number strictly as an identifier. The new number could still be passed as a parameter, stored in the same slot in the call control block, and used to index into data tables to retrieve per terminal information. Only the programs that actually required the real equipment number (equipment control programs and programs that print equipment locations for human use) needed to be modified. The subset of affected programs was relatively small. In addition, the required modifications were reasonably straightforward. Functions to do the mappings were added to the database interface. The modifications to other programs consisted mainly of inserting calls to these new functions.

The virtual equipment numbering scheme is now in place and functioning well. Because its introduction is thought to have saved development effort in other areas of the project, the consensus of opinion is that its overall development effort was less than other considered methods. It will, of course, continue to provide benefits in the future when, and if, additional network types are added to the No. 2B ESS.

## V. FEATURE EXTENSIONS

No. 2 ESS began on a small suburban switching machine. In the early 1970s, centrex features were added. With introduction of No. 2B ESS in 1976, real-time capacity was almost doubled, allowing even more business-type features.

The 2BE3 generic added compatibility with the 10A RSS (using the virtual terminal concept and the database modernization), EADAS, and AMACS, while allowing greater penetration of custom calling features (via the database modernization). Several of these features are described in accompanying articles.

## VI. CONCLUDING REMARKS

The No. 2/2B ESS experience demonstrates that a mature system can evolve its hardware and software systems to more modern technologies. The successful methodology included making only one major change at a time, and, where possible, introducing a modern software structure without, or at least before, breaking the existing structure. The evolution reported herein spans the 1970s. The No. 2B system was proposed in 1971. It was developed during the middle years of the

decade and was placed in commercial service in February 1976. The compatible software enhancements occurred between that time and the next release of the system in December 1977. The noncompatible enhancements have occurred since that time and continue at present.

The software changes as a group have simplified software administration and the introduction of code changes to support new telephone features. The development effort has been quite modest. Introducing standardization has introduced some inefficiencies. Certainly the general-purpose control and data functions require more memory and in many cases a small real-time overhead. These penalties are not a large price to pay for the advantages of improved software structure.

## VII. ACKNOWLEDGMENTS

Several people were instrumental in the No. 2B ESS concept and its original design. These included J. A. Herndon, C. E. Ishman, W. R. Nehrlich, P. D. Mandigo, and P. C. Richards. Beyond these few, the authors wish to thank the entire No. 2/2B ESS development organization that made the system a success. This paper is based in part on earlier works by Mandigo and Nehrlich.

## REFERENCES

1. Special issue on No. 2 ESS, B.S.T.J., *48*, No. 8 (October 1969).
2. P. D. Mandigo, "No. 2B ESS: New Features From a More Efficient Processor," Bell Lab. Rec., *54*, No. 11 (December 1976), pp. 304–9.
3. T. F. Storey, "Design of a Microprocessor Control for a Processor in an Electronic Switching System," B.S.T.J., *55*, No. 2 (February 1976), pp. 183–232.
4. L. E. McMahon, L. L. Cherry, and R. Morris, "UNIX™ Time-Sharing System: Statistical Text Processing," B.S.T.J., *57*, No. 6, Part 2 (July–August 1978), pp. 2137–54.

## AUTHORS

**John E. Yates,** S.B.E.E., 1960, S.M.E.E., 1962, Massachusetts Institute of Technology; Bell Laboratories, 1963—. Before coming to Bell Laboratories, Mr. Yates was a research assistant at the Massachusetts Institute of Technology. In 1962 he joined The Boeing Company. Since 1963, he has worked extensively on the No. 2 and No. 3 ESSs. Mr. Yates is currently responsible for development of application database change programs in No. 5 ESS.

## AUTHORS

**Thomas E. Grassman,** B.S.E.E., University of Arizona, 1961; M.E.E., New York University, 1963; Bell Laboratories, 1961—. Since coming to Bell Laboratories, Mr. Grassman has worked on a variety of 101 ESS and No. 2/2B ESS assignments and on the establishment of the BX.25 standard for data-link protocol. Presently, he is Supervisor of a No. 2B ESS call processing software group.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

## *Modernization of the Suburban ESS:*

# Adding Data Links to an Existing ESS

By C. E. ISHMAN,* R. B. SANDERSON,* L. M. TAFF,* D. P. TRUAX,* and C. T. TULLOSS*

(Manuscript received June 21, 1982)

As a major modernizing improvement data links for three important functions have been added to the 2B Electronic Switching System (ESS). The data-link system was designed with a layered architecture using an enhanced subset of the X.25 protocol. Levels 1 and 2 of the protocol are implemented in the hardware and software of a new device, the Serial Peripheral Unit Controller/Data Link (SPUC/DL). Within the SPUC/DL itself, check circuitry and software are used for error detection. A second SPUC/DL for each critical application provides redundancy on either a dedicated or dial-up basis. An elaborate diagnostic program within the 2B ESS can be invoked automatically or manually. This paper discusses some of the issues in retrofitting the BX.25 protocol to pre-X.25-speaking machines.

## I. INTRODUCTION

This paper describes the addition of real-time data-link capability to the No. 2B Electronic Switching System (ESS)†. The paper combines a technical description of the project with explanations of certain

* Bell Laboratories.

† Acronyms and abbreviations used in this paper are defined at the back of the Journal.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

key decisions. We attempt to be candid in assessing both the virtues and pitfalls of such decisions.

The No. 2B ESS is a medium-sized telephone switching system designed for rural and suburban offices with 5000 to 20,000 customer telephone lines. The addition of data links to the system was but one of several major enhancements in the development of a new generic program developed for the 2B—the 2B Extended feature generic #3 (2BE3)—described in accompanying papers.

### *1.1 Project goals*

Our major goal in this project was to create appropriate hardware/software systems to provide data-link capabilities from the 2B to each of three different remote Bell System *applications*. Each application had different characteristics and reliability requirements. An overview of the desired network capabilities is shown in Fig. 1. Note that other types of ESS machines as well as additional 2B ESS offices may interface to the Automatic Message Accounting Recording Center (AMARC) and Engineering and Administrative Data Acquisition System (EADAS) applications. This constrained the freedom to choose interface specifications freely. We wanted flexibility in our design so various combinations of these applications could be configured without trouble or extra cost.

A secondary goal was that as much as possible of the data-link subsystem resulting from our efforts be "portable;" the system should be adaptable to other applications on the 2B, and even other machines, after suitable hardware alteration.

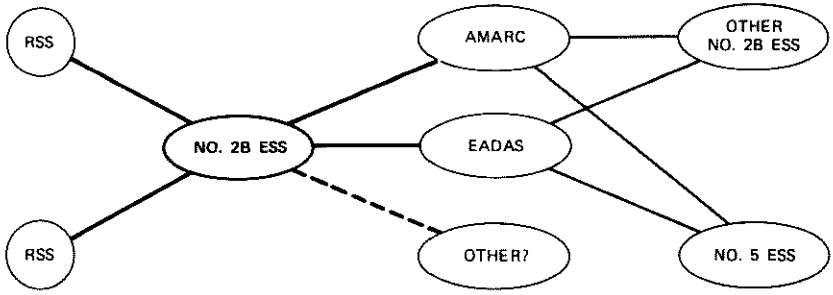
### *1.2 Boundary conditions*

#### *1.2.1 Use of BX.25 protocol*

The development of our project proceeded roughly in parallel with the developments of the AMARC and EADAS projects within Bell Laboratories, as well as with the No. 5 ESS project, which was also to interface with AMARC and EADAS. A general consensus was reached to adopt recommendation X.25 of the *Comite Consultatif International Telephonique et Telegraphique (CCITT)*. BX.25, an enhanced subset of X.25, has since become a standard within the Bell System. Its adoption had important consequences in the total design of a data communications system, as will become clear below.

#### *1.2.2 10A Remote Switching System*

The 10A Remote Switching System (RSS) is a small space-division system intended for rural communities with fewer than 2000 subscribers. It requires a larger *host* machine to switch interoffice calls. Its controlling program is resident in firmware, and has been developed




AMARC – AUTOMATIC MESSAGE ACCOUNTING RECORDING CENTER
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 1—Global view of planned No. 2B ESS.

over several years in conjunction with the No. 1/1A ESS, which provided host capability. The 10A RSS feature of the 2BE3 generic program will allow the 2B ESS to be a host as well. Up to 31 RSS machines can be supported by one 2B ESS.

The 10A RSS, having been developed before BX.25, constrained the generality of the data-link system. Nearly 100 units were in service as the 2BE3 generic was being completed. We were required to interface to the RSS with no changes to its firmware.

## II. GLOBAL SYSTEM ARCHITECTURE

We address in this section the architectural issues, both hardware and software, involved in the next level of detail from Fig. 1. For this purpose it may be useful for some readers to review briefly the structure of the BX.25 protocol.

### *2.1 Overview of BX.25*

BX.25 is a *layered* protocol consisting of independent layers or *levels* (see Fig. 2). Each level at one side of a data link communicates (in a logical sense) with the corresponding level on the other side. We use three levels, not counting the application program. A call-processing program (say) believes it is conversing with a billing program, for example, at the AMARC. To send a *message* to its counterpart the program simply calls a protocol level 3 interface routine. For its part, the conversation is complete.

Level 3 software will break the message, if necessary, into *packets* of length appropriate to the implementation (depending for practical reasons on the kind of transmission facility, available buffering capacity, etc.). It will then send the packets to the level 3 software at our destination, expecting—as one of its standard functions—a "message-

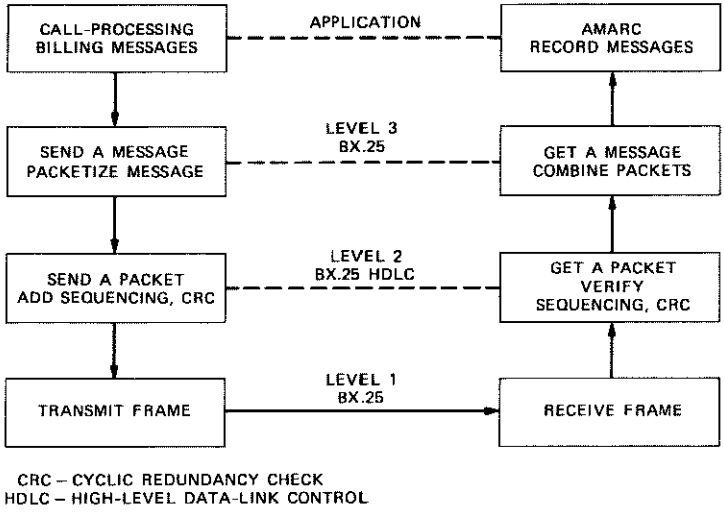



Fig. 2—Application layer and three layers of BX.25.

received" *acknowledgment*. It normally will not rest easy until such acknowledgment is forthcoming within a timed interval; it typically will retransmit the packets until a further interval expires, when it will begin recovery actions. Note, however, that "sending the packets" simply means handing them off to level 2 software.

Level 2 software is very analogous to level 3 in that suitable numbering is attached to the data, which are then delivered as *frames* to level 1.

Level 1 is implemented directly in hardware and is responsible for physically transmitting the data to the remote site. Level 2 frames are specified to contain a Cyclic Redundancy Check (CRC) at their trailing end, though in practice this is computed and inserted by an industry standard hardware chip. Thus there is a slight blurring of the line between levels 2 and 1.

While level 1 is supposed to transmit the data, the primary task of level 2 is to ensure that they arrive safely. Level 2 will retransmit unacknowledged frames, occasionally transmit *receiver-ready* messages on a temporarily unused link ("idle line assurance"), and handle initialization and disconnection functions. The numbering and acknowledgment functions of level 3 are primarily for flow control rather than verification purposes. A primary function of level 3 not yet mentioned is the routing of *logical channels* to their proper physical destinations.

Readers interested in more details about this protocol are referred to Refs. 1 through 3.

### 2.2 Architectures considered

One possible configuration would be to attach data-transmission hardware directly to the 2B ESS and write level 3 and level 2 software for the ESS. All the detail work of keeping track of unacknowledged frames, and time-consuming interrupts from transmission and reception, would be incurred by the switching machine itself. Since such overhead increases with the number of data links, real-time constraints made this an unrealistic option. In particular, we note that the link traffic increases just at least opportune times—when the machine is busy servicing telephone calls and has the least time available to handle link work. This configuration was never considered seriously. All realistic options use a separate autonomous processor to off-load the ESS.

We will discuss three such options given serious consideration. In the first option, level 3 and level 2 software are wedded together within a small auxiliary processor also containing the data-link hardware. In the second option, level 3 software resides with level 2 software capable of handling multiple data links, many of which are installed on the separate processor. The third option splits levels 3 and 2 into the ESS and attached processor, respectively, with level 2 servicing only one data link. This last option was eventually adopted.

#### 2.2.1 Monogamous levels 3 and 2 together

A configuration with levels 3 and 2 residing in a microprocessor was chosen by the designers of the AMARC side of our data network for their "AMARC Protocol Converter" (APC). This choice has the advantage that level 3 as well as level 2 is off-loaded from the main processor. The disadvantage of concern to us is that when it is necessary to change traffic to the standby link in a multi-link configuration, all level 3 current parameters as well as *all data currently queued at level 3* must be unloaded from one device and transferred to another. At the AMARC side of the link this is not as serious a matter as on the ESS side, where large volumes of data may be generated.

#### 2.2.2 Polygamous level 3 with multiple level 2's

A data link device for up to 16 independent links controlled by common level 3 and level 2 software is currently in use with the Bell System No. 1/1A ESS. This device, named the Peripheral Unit Controller (PUC), was the original model for our own system. Such a configuration has the advantage that links can be changed without intervention of the main processor. However, the PUC turned out not to be suitable for a machine the size of the 2B ESS; with enough power to drive 16 links there is a mismatch in the capacities and costs of the 2B ESS and the data-link controller. This consideration changed the

original direction of our project away from the PUC. Our device was originally to have been a PUC modified from parallel communication with the host to high-speed serial (6.67 Mhz); hence the name Serial Peripheral Unit Controller/Data Link, or SPUC/DL.

#### *2.2.3 Levels 3 and 2 living separately*

To answer the previous objection to high overhead cost for the office with few data links, we wished a small and inexpensive basic hardware unit. Vulnerability to failures could then be countered by duplicating units on applications requiring high reliability. An individual link need only be enough more reliable than the communications channel not to add significantly to the failure rate. Furthermore, difficulties in changing links could be reduced by keeping level 3 of the protocol and level 3 buffers within the ESS, which is fully duplicated, including all writable memory. In addition, where a controller implementing Level 3 would almost certainly need to be duplicated and have elaborate checking and matching circuitry to meet reliability requirements, the controller implementing Level 2 can use a simpler reliability check such as parity. Finally, of the three configurations discussed, this last seemed to us to be the most flexible, as well as the most fully portable to other applications and/or hosts. With this scheme substantial new code needed to be written for the 2B ESS—the level 3 protocol, and the maintenance software for the links. Furthermore, complete off-loading of the ESS processor does not occur. Though most of the work and interrupt processing is done in the SPUC/DL, there will be a small traffic-dependent load for managing level 3.

### *2.3 Configurations by application*

As we mentioned previously, each of the three initial applications for the SPUC/DL had its own requirements and the simple hardware could be configured separately for each.

#### *2.3.1 10A RSS*

Data links for each RSS are fully duplicated with hard-wired dedicated lines for both links, thus providing a "hot" spare. It is recommended that diverse transmission facilities be used for each data link. These rather strong requirements underscore the high reliability expected of this application. Call processing for lines connected to the 10A RSS is done on the host machine, in our case the 2B ESS. The data links are used to control the progress of calls. Thus, loss of data-link capability means loss of most service for RSS customers. (The RSS can be configured to provide local calls if the host connection becomes lost.) The 10A RSS machine is described at length in Ref. 4.

### 2.3.2 AMARC

Traffic on the AMARC data links does not carry information that affects customer service, and therefore does not have the intrinsically stringent reliability requirements imposed on RSS links. However, the billing information carried is valuable, and long outages could be costly. Short outages—some tens of seconds—would be tolerable if suitable buffering capacity were provided to avoid losing billing data. AMARC links were therefore configured with one data link on a dedicated hard-wired line, supplemented with a "cold" spare connected to the switched network. Changing to the cold spare after a dedicated primary link outage exceeding some minimum duration involves dialed telephone calls between the ESS and AMARC machines. Such a scheme saves the considerable cost of a second dedicated connection.

### 2.3.3 EADAS

Though data gathered on EADAS links are useful for planning telephone network growth, their temporary loss does not compromise either customer service or revenue. EADAS applications can therefore be configured with a single data link, equipped with a dedicated line.

### 2.4 Architectural summary

A diagram of a No. 2B ESS with its three types of configurations is shown in Fig. 3. We emphasize that the duplication on critical applications is for reliability; only one link is active at any time, and there is no load sharing.

## III. FUNCTIONAL REQUIREMENTS OF THE SYSTEM COMPONENTS

### 3.1 Hardware

The general requirements of the hardware are as follows:

1. Basic hardware. Hardware for the data-link controller must execute level 1 (the physical level) of BX.25, and provide enough memory space and processing power to implement level 2.
2. Link speeds. Initial applications were to use 2400 or 4800 bits/second (bps), but for possible new applications a goal was set for 56K-bps hardware capability.
3. Link Reliability. Our goal was to have data communication failures be caused predominantly by transmission facility failures rather than by data-link controller failures.
4. Repairability. The controller required self-test capability to isolate faults to a single-printed circuit board wherever possible.
5. Cost. The initial cost of the data links was required to be low for smaller offices with only a few links.
6. Expansion. Expansion should be modular, smooth, and trouble-free.

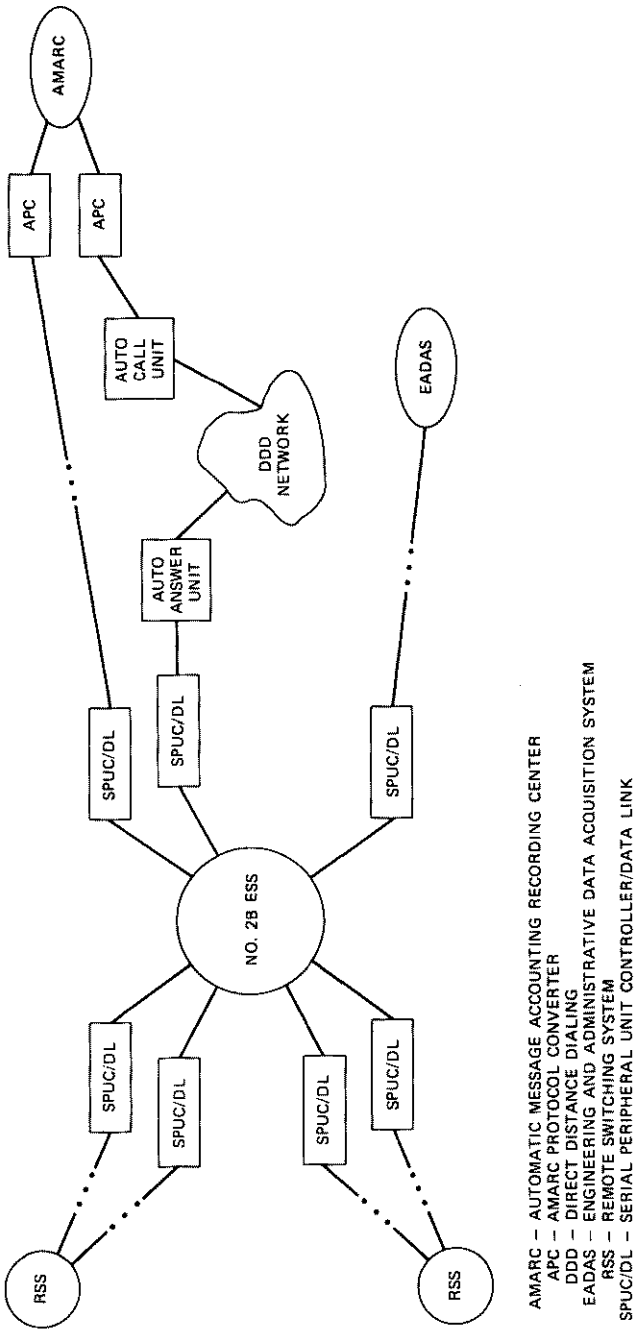



Fig. 3—The three data-link applications of the 2B ESS.

### 3.2 Software for the SPUC/DL

Stated simply, the SPUC/DL software requirement was to implement level 2 of the data-link protocol. Unfortunately, the three data-link applications for 2BE3 required two sets of protocol rules for level 2. The AMARC and EADAS applications use the procedures defined by the BX.25 specifications,[3] which were derived from the CCITT X.25 "Link Access Procedure B" or LAPB. The third application, RSS, uses the older version of the X.25 level 2 procedure "LAP." LAP and LAPB, though similar, are not compatible (LAP cannot communicate with LAPB). However, except for the initial handshake, some error recovery procedures, and a few differences in command formats, the core of information processing procedures is the same.

Initial SPUC/DL applications were to use link speeds of 2400 bps, although future applications of AMARC requiring 4800 bps were foreseen. These requirements can be easily met by a microprocessor device.

Data traveling over common carrier facilities are inherently susceptible to any noise disturbances afflicting such facilities. It is essential that data altered during transmission be detected and the correct data retransmitted. This is the *raison d'etre* for level 2 of X.25. To this end, X.25 employs a 16-bit Cyclic Redundancy Check (CRC) to detect frames altered by noise, and a sequence numbering scheme to facilitate retransmission of frames discarded because of bad CRC. Flow control procedures at level 2 also prevent data loss from buffer overflow.

An additional requirement for maintenance purposes was for some diagnostic capability "on board" the SPUC/DL. The on-board (resides in SPUC/DL memory) diagnostic was to be used by the ESS diagnostic to test hardware inaccessible to the ESS or requiring excessive ESS resources to test. When the ESS wished the SPUC/DL to begin its diagnostic routines, it was to instruct the SPUC/DL operating system to relinquish control to the on-board diagnostic programs. To go back to normal the ESS must issue a master reset of the SPUC/DL, since some diagnostic procedures would destroy the normal operating state.

Diagnostic requirements include a "loop around" test in which data packets would be sent out on the link, directly echoed back by the remote end, and received by the SPUC/DL and 2B ESS. This would normally be a protocol violation in the use of an *address* field at the frame level. We have enhanced the implementation to suspend normal protocol address checking when the diagnostic is running.

### 3.3 Software within the 2B ESS

We discuss here our requirements for both the protocol software and the maintenance, or fault-tolerant, software. Most of the protocol

requirements are general and serve to elaborate on our previous discussion of BX.25.

#### *3.3.1 Protocol-implementing software*

The 2BE3 generic program requires dedicated point-to-point (permanent virtual circuits), full-duplex, 2400 (or 4800) bits per second communication facilities. The No. 2B ESS software implements only those portions of the BX.25 specification needed to meet the 2BE3 generic requirements.

Level 3 of the BX.25 protocol controls the transfer of data between level 2 and the next higher level (the application layer in No. 2B ESS). Transferring includes several processes, namely: packetization, multiplexing, sequencing, and flow control.

***3.3.1.1 Packetization.*** Packetization is the process of a transmitter's dividing large messages into practical-sized packets, transmitting the packets, and having the receiver concatenate the packets to form the original messages. BX.25 does this by setting a "more-data" (M) flag bit in the level 3 header in all but the last packet of a message. The AMARC billing data messages have a maximum of 512 bytes; EADAS thirty-minute traffic data messages vary, with an average of 4,000 bytes depending on the central office. The packet size of 256 bytes was determined appropriate for the throughput desired and the expected error rate of the data transmission facility. If the packet size is large and the transmission facility error rate is high, many packets may receive errors and need to be retransmitted, which decreases the throughput.

***3.3.1.2 Multiplexing.*** Multiplexing is the process whereby a transmitter combines data packets with identical destinations but associated with various application functions (logical channels). The receiver separates (demultiplexes) the packets via the logical channel numbers and distributes them to the appropriate application function. AMARC has a billing-data logical channel, EADAS a traffic-data logical channel, and both applications have a time-of-day logical channel; thus, multiplexing is required with both the AMARC and EADAS applications.

***3.3.1.3 Sequencing.*** Sequencing of BX.25 level 3 data packets is required to assure packets are received in the identical sequential order of transmission. Each level 3 data packet header contains two modulo eight sequence numbers: a send sequence number P(S), which identifies the data packet itself, and a receive sequence number P(R), the expected number of the next data packet received.

BX.25 permits a level 3 transmitter to send a prescribed number of packets to level 2 without obtaining acknowledgment that earlier

packets were received correctly. These unacknowledged level 3 packets sent to level 2 are in the *window* until their reception has been acknowledged. The *window size* of each logical channel is the number of packets that can be sent to level 2 without being acknowledged; the AMARC window sizes are both five while EADAS windows are both two.

***3.3.1.4 Flow control.*** Flow control is provided by BX.25 to permit receivers a method to regulate the rate at which data packets are sent to it. A packet received with send sequence number $n$ is acknowledged with the next packet sent by setting the receive sequence number to $n + 1$. Any receiver having a data-capacity problem may delay the acknowledgment of received packets and avoid data loss.

***3.3.1.5 Requirements for the 10A RSS.*** RSS level 3 was originated with No. 1 ESS before the acceptance of X.25 and is not compatible with BX.25. RSS level 3 messages contain a unique "SYNC word" in the first two bytes. Received messages are discarded if they do not begin with a SYNC word. The SYNC word is manipulated entirely by the RSS level 3 program and is transparent to the No. 2B ESS application call-processing software.

A two-byte remote terminal header following each SYNC word contains the RSS message type in a five-bit "client identity" field. The level 3 software distributes (demultiplexes) received messages to the call-processing application programs by interpreting the client identity. An RSS level 3 header also contains an eight-bit word count field specifying the number of data words in the message.

Packetization of messages is also used in RSS level 3 communication. The maximum RSS packet size is sixteen bytes and the packet boundary is totally asynchronous with messages from an RSS. RSS-bound messages transmitted from the No. 2B ESS always begin a new packet but may require several packets, depending on their length.

Communication concerning only the No. 2B ESS and SPUC/DLs is passed in *control packets* with a unique packet descriptor. A *packet descriptor* is the first two bytes of *all* packets passed between the 2B ESS and a SPUC/DL, identifying the packet type (data or control) and specifying the packet length in bytes. Packet descriptors are not transmitted over the data links. Control packets are external to level 3 and pass directly between the SPUC/DL and its maintenance configuration and diagnostic programs.

***3.3.1.6 Timing requirements.*** We have mentioned previously that the decision to put level 3 in the 2B ESS imposes a penalty in traffic-dependent processing power. In requirements terms, level 3 had to be fast.

*3.3.1.7 Other requirements.* It was foreseen that part of the protocol-handling software package would include low-level routines for physical I/O to each SPUC/DL. Two requirements were imposed on these I/O routines: error routing and monitoring. Packets whose reception was not completely normal were to be routed to the data-link maintenance software. This includes nondata (control) packets, I/O errors, and packets from a nonactive SPUC/DL. A manually invoked monitoring utility was to report part or all the data sent to and read from an application. The monitor proved to be of value during system debugging and integration.

### *3.3.2 Maintenance (fault-tolerant) software*

There were several general requirements assigned to the data-link maintenance function.

1. Initialization and Loading. On certain rare events there may be occasion to reinitialize any or all links in the system. This may require a down-load of the SPUC/DL program from its magnetic-tape residence at the ESS. This would also be done when field-updating SPUC/DL software.

2. Handling Error Reports. Maintenance software is required to maintain records of such reports, which may include frequent loss of facility carrier, low efficiency of transmission, etc.

3. Error Recovery/Link Reconfiguration. Serious faults (e.g., loss of communication with a SPUC/DL) or slow degeneration of a link may require removal of the affected link from service and changing of traffic to the standby link, if available. Thus, data-link maintenance is charged with trying to reconfigure a failing system to maintain traffic.

4. Regulation of the Diagnostic Program. The diagnostic, controlled by the maintenance software, is to be executed automatically when a failure occurs, and routinely on a nightly basis to detect faults on standby links before they are needed in service.

5. Audits. Maintenance "audit" programs were required to check periodically the legitimacy of the control-block database of data-link maintenance, to correct and report single errors, and to reinitialize if multiple errors were found.

6. Console Operator Interface. The final requirement of the data-link maintenance software was to interface with the human operator of the system, providing keyboard input service and output of various responses, as well as spontaneous reports of normal and abnormal events.

### *3.3.3 The diagnostic program*

The diagnostic program had to detect and locate faults in the 2B

processor I/O interface, the SPUC/DL, and the component parts of the transmission facility. In the RSS application, it must also locate faults in the remote data-link control hardware. The diagnostic control structure was required to be sufficiently flexible to permit systematic testing during the installation of the SPUC/DL, the near- and far-end data facilities, and (for RSS) the remote data-link equipment. It had to be able to stress the overall data link enough to recreate faults observable only during heavy traffic.

## IV. DESIGN AND IMPLEMENTATION—SPUC/DL HARDWARE AND SOFTWARE

### *4.1 SPUC/DL hardware*

We can identify three key decisions in the evolution of our design. All three involved hardware/software trade-offs, the software being in the 2B ESS. The first decision involved the scope of the SPUC/DL, as we have discussed in Section II. The second decision was the use of periodic polling of the SPUC/DL rather than program-interrupt hardware. We will elaborate on this in Section V. The third decision was to use writable Random Access Memory (RAM) instead of Read-Only Memory (ROM).

#### *4.1.1 Use of writable memory*

The advantage of RAM is that updates of SPUC/DL software for circuits in the field are much less costly, requiring only software procedures rather than replacement of physical chips. The main disadvantage is that the memory is volatile; the program is destroyed (and must be reloaded by the 2B ESS) if power to the SPUC/DL fails; therefore, software within the 2B ESS had to be developed for downloading. In addition, the SPUC/DL software must have a nonvolatile residence from which it can be loaded. In our case this was to be the magnetic tape cartridge of the 2B ESS. The need for loading would cause concern only if long delays were incurred by the loading process. However, since 2B ESS-SPUC/DL communication proceeds nearly at memory speeds, the loading time mostly comprises the time it takes to retrieve the program from tape.

#### *4.1.2 Logical design*

The data-link controller itself consists of a single 16-bit microprocessor, its associated RAM, and two Direct-Memory Access (DMA) controllers, DMA1 and DMA2. Duplication within a single SPUC/DL is not necessary because of the relaxed reliability the distributed architecture allows. DMA controller 1 is a multiple-channel device allowing data transfers to take place between the 2B ESS and the

SPUC/DL independently of the microprocessor. Similarly, DMA controller 2 provides independent transfers between the SPUC/DL and the transmission facility (see Fig. 4). This architecture minimizes demands on the microprocessor while allowing high throughput. Level 1 protocol is incorporated in a Universal Synchronous-Asynchronous Receiver-Transmitter (USART) and standard interface chips.

### *4.1.3 Physical design*

To meet the requirements of modular growth and high multiple-

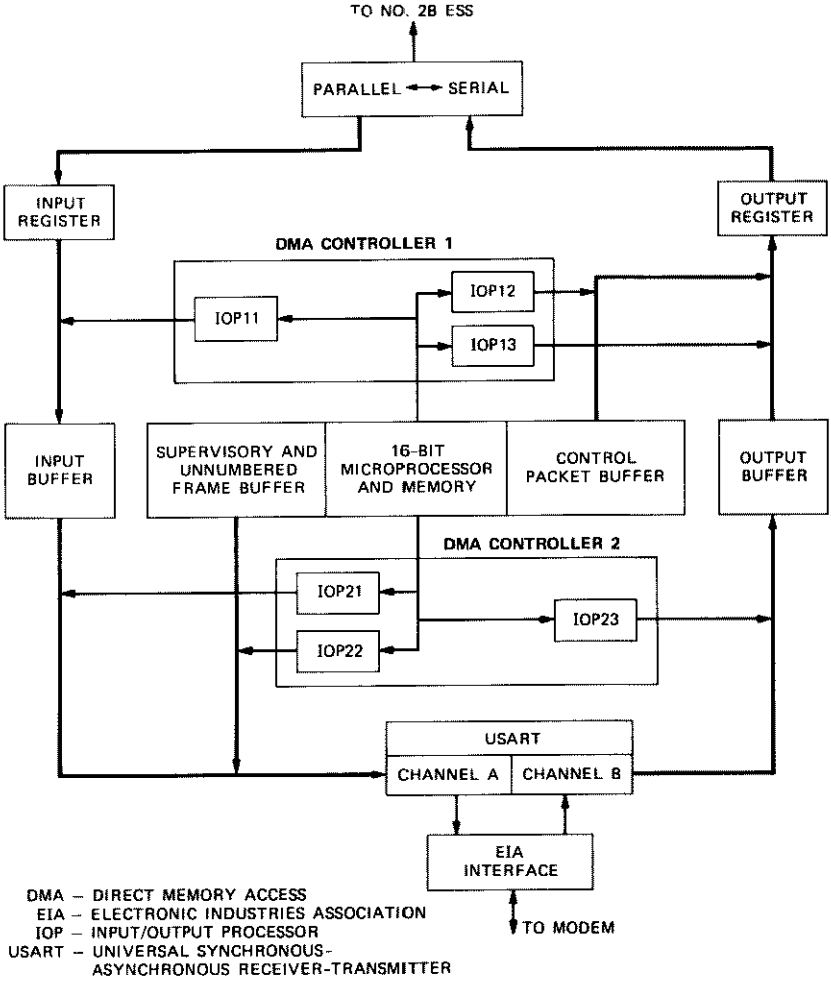



Fig. 4—SPUC/DL hardware components and architecture.

link reliability, each data-link controller was made a separate circuit. Pairs of controllers share a −48 volt power lead. Each SPUC/DL has its own power supply, transmission facility interface, microprocessor, memory, and communications channel to the 2B ESS. There are four printed circuit boards per SPUC/DL, containing the microprocessor and associated logic, memory, the data communications hardware, and the logic for I/O to the 2B ESS. Four complete SPUC/DL circuits, their associated power supplies, fuses, and connectors are housed in a single chassis.

A concentrated effort was made to provide a straightforward maintenance procedure. Each circuit has its own power fuse and modem cable. Each circuit is numbered, and this numbering is used on the circuit board enclosure, power supply housing, power fuse, and modem connector. In addition, each circuit is color coded, and the identifying color appears everywhere the circuit number is used. The 2B ESS SPUC/DL diagnostic program prints both the circuit name and color when it refers to a particular SPUC/DL.

#### *4.1.4 Reliability issues*

To provide sufficient reliability, the data-link controller's memory (program and data) is stored with parity and the parity is checked whenever data are read out or written. In addition, write-protection hardware is provided for the program memory. Any attempt to write into program memory will be prevented, and the 2B ESS will be informed. The status of the SPUC/DL hardware is summarized in a hardware status register that the 2B ESS can read.

The SPUC/DL on-board diagnostic program first computes a check sum over the program memory, tests the data memory, and then confirms proper operation of the microprocessor itself. Once the basic sanity of the microprocessor and correctness of its program is established, the diagnostic program goes on to test both the DMA controller that oversees data-link communication and the USART. The USART is tested by actually transmitting on the link. Additional loop-around circuits at the EIA interface, the associated modem, and the far-end modem allow faults to be isolated and identified easily. This philosophy of testing sections of the controller nearest to the central processor and then relying on those sections to test sections further removed allows economical and rapid fault isolation. Fault isolation is enhanced by partitioning the SPUC/DL into its four circuit boards. All interface circuitry to the 2B ESS is on one board. The second board contains DMA controller 1 and address decoding circuitry. The third board contains all data and program memory. The fourth board has the microprocessor itself, DMA controller 2, and the USART and modem

interface. Since the proper functioning of each board relies on the previous board but not on subsequent boards, most faults can be isolated to a single board.

### *4.2 Software within the SPUC/DL*

Because of the commonality between LAP and LAPB, it was decided to implement both of them in one software load; the ESS would specify one or the other with a parameter sent to the SPUC/DL on initialization. Though this decision increased the space used for code in the SPUC/DL, it eliminated keeping track of which code to download into which SPUC/DL and the need for extra space on the cartridge tape. Separate codes would also have caused complications with the diagnostic. Perhaps the largest factor in the single-program decision was the expense of having to design, code, test, and administer two separate programs.

First we look at the SPUC/DL software structure from a functional data flow viewpoint, and then at the overall structure of the operating system that implements these functions.

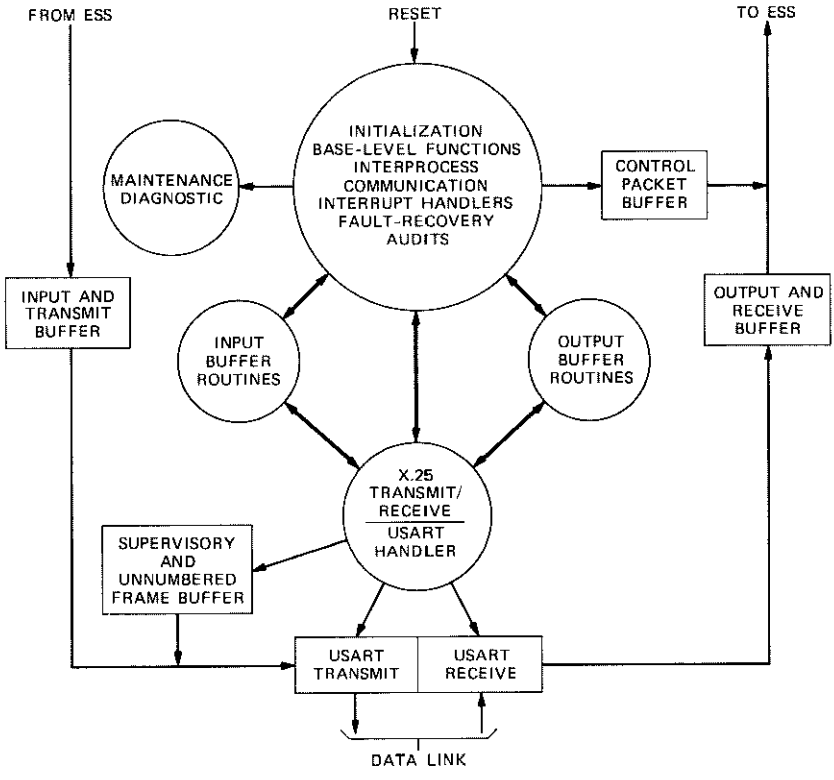
#### *4.2.1 Data flow within the SPUC/DL*

The hardware architecture of the SPUC/DL shown in Fig. 4 closely parallels the flow of data through the device. Internal to the SPUC/DL the send and receive processes run simultaneously. Data packets from the ESS enter the input buffer under control of DMA1 and are cataloged by the input routines, with *control* packets (messages for the SPUC/DL itself rather than for transmission) being processed immediately. Once packets have been cataloged, they are processed by the X.25 routines, which assign sequence numbers and set up the DMA2 transfer to the USART for output over the link as X.25 frames. Simultaneously with transmission, frames are coming in from the receive channel of the USART, are processed by the X.25 routines, and pointers to information frames are passed to the output routines. The output routines add packet descriptors to the information frames, and then set up the DMA transfer to the ESS.

This is a high-level view of the SPUC/DL and there are many other software processes involved in its operation, including maintenance routines, which play a large role in support of data handling.

#### *4.2.2 SPUC/DL Operating System*

The SPUC/DL Operating System may be visualized as a hierarchical multiprocessor operating system. In this system the 16-bit main processor acts as a master while the six individually programmable DMA channels are slave processors. Once a DMA channel has been




ESS – ELECTRONIC SWITCHING SYSTEM
USART – UNIVERSAL SYNCHRONOUS-ASYNCHRONOUS RECEIVER-TRANSMITTER

Fig. 5—SPUC/DL software processes; interprocess communication has been emphasized.

given a process, it will run that process to completion without further attention from the master. We denote these single channels as I/O processors, or IOPs. Most interprocessor communication is one way between the master and the IOP. Interprocess communication is handled using producer/consumer semaphores.

Figure 5 shows how the SPUC/DL software processes use the hardware to carry out the data communications function. Each I/O processor handles a specific I/O function. Communication between the SPUC/DL and ESS is handled by three channels of DMA1. In Fig. 4 they are labeled IOP11, IOP12, and IOP13. IOP11 runs the process of moving data from the input register to the input buffer. Once the SPUC/DL has been initialized by a master reset, the master processor starts this process running on IOP11. This process runs continuously and becomes inactive only when input buffer space is

exhausted or the process may be preempted by the microprocessor for status updates. This status information consists of "where will the next word be placed in the input buffer," or "deallocate *n* input buffer words." Unlike IOP11, IOP12 and IOP13 run only when there are processes scheduled for them (packets to be sent to ESS). IOP12 transfers control packets from scratch memory to the output register and IOP13 transfers data packets from the output buffer to the output register.

The DMA2 processors have almost mirror-image functions. IOP21 moves packets from the input buffer to the USART where they are serialized and sent out over the link. Packets are scheduled for IOP21 on a first-in/first-out (FIFO) basis with priority for packets being retransmitted. IOP22 moves supervisory and unnumbered frames from scratch memory to the USART. IOP23, which moves data in from the link, does not run continuously because of the intervention of the master processor to handle alignment of the start of incoming data. All other processes are run by the master processor, including the considerable primary task of processing X.25 frames.

#### *4.2.3 SPUC/DL software reliability*

In developing SPUC/DL programs considerable attention was given to reliable software design through modularity. Both the SPUC/DL Operating System and the level 2 procedures were coded in the C language and a top-down design structure was used wherever practical. Assembler-level code was used only where necessary for speed or to handle hardware-dependent functions. The high-level language had advantages both from an implementation and a testing standpoint. However, because both the hardware and software were new to us, during debugging it was occasionally necessary to look at the assembly code or binary output of the C compiler to determine the source of some problems.

The hardware of the SPUC/DL provided three features that also enhanced software reliability. First, the use of DMA hardware controllers for I/O eliminated the need for complex scheduling algorithms to move blocks of data. This is especially important in an I/O-intensive application such as a data-link controller. Second, write-protection hardware was provided. Since the program store of the SPUC/DL is in writable memory, without the write-protection feature of the hardware, the program would be vulnerable to bugs that alter the program itself. If a write-protect violation occurs, the SPUC/DL software is not altered but an interrupt routine attempts to save as much pertinent information as possible and then proceeds normally. The ESS is notified by a message from the SPUC/DL containing the hardware status register. The third hardware feature is a "sanity" timer, which

expires unless periodically serviced by the software. The timer guarantees that the microprocessor will not get stuck in any one place for an inordinate amount of time. If the sanity timer should expire, an interrupt is generated and is handled similarly to the write-protect error.

In addition to the hardware mechanisms mentioned, reliability is increased by periodic software audits. The audits check major global data structures, looking for inconsistent or out-of-range values. No attempt is made to correct audit-detected errors. The event is reported to the ESS, which prints a message.

### *4.3 System performance*

The SPUC/DL software has an average one-way delay of about 4.5 ms. It has been tested at 5 to 10 times the speed expected in initial field use.

## V. 2B ESS HOST SOFTWARE

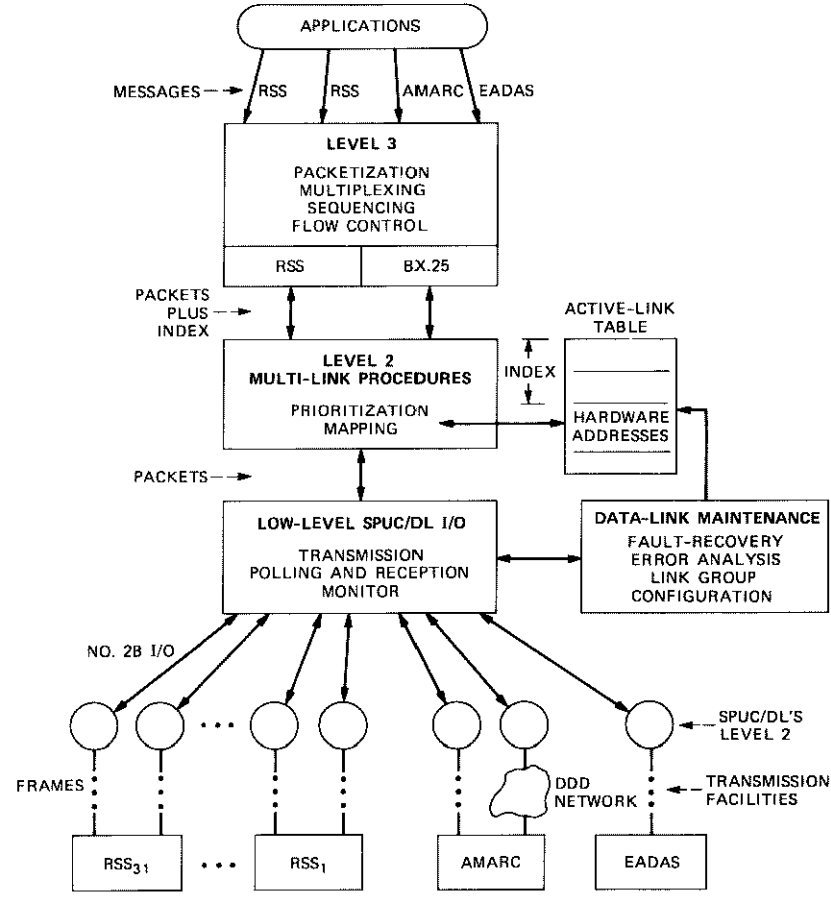
### *5.1 Architectural overview*

Figure 6 is a high-level diagram of the data-link software in the 2B ESS. When they have a message to transmit, call-processing routines, billing routines, and data-collection routines enter appropriate points in level 3. Level 3 converts the messages into packets, breaking large messages into multiple pieces if necessary. Multiple logical channels (logical destinations) are sorted for their physical destination (multiplexing). Appropriate sequence numbers are attached. The packets are transmitted immediately, if possible, or put on a queue if flow is currently under control.

When the packets are ready to be transmitted, the software passes them to the level 2 Multi-Link-Procedures. These routines sort the packets by priority, and then do a *mapping* to a definite hardware address (i.e., physical SPUC/DL). The mapping is done through a table managed by the data-link maintenance software. Thus, the routines doing transmission need have no knowledge of which link they are using.

The Multi-Link-Procedures pass the packets to the low-level transmission software, which physically outputs them to the SPUC/DL whose hardware address was passed as an argument. The link-traffic-monitoring utility is implemented at this level. Level 2 in the SPUC/DL ensures error-free transmission to the destination.

The inverse process, packet reception and assembly into messages, is similar. Note that on reception, not all packets get routed to the Multi-Link-Procedures. Control packets originating within the SPUC/DL, and information packets from links not currently active are given to the data-link maintenance software. The maintenance




AMARC – AUTOMATIC MESSAGE ACCOUNTING RECORDING CENTER
DDD – DIRECT DISTANCE DIALING
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
RSS – REMOTE SWITCHING SYSTEM
SPUC/DL – SERIAL PERIPHERAL UNIT CONTROLLER/DATA LINK

Fig. 6—Data-link software within the No. 2B ESS.

software also uses the low-level common I/O routines to transact its business with each SPUC/DL. In this way the monitoring utility will report *all* communication to any link or link group.

### *5.2 Device polling and timing considerations*

To have the SPUC/DL request attention from the 2B ESS, a polling method was adopted owing to complications in the 2B ESS demand-interrupt hardware. When it requires attention from the 2B ESS, the SPUC/DL sets a bit in its status register. The status register's bit controls a *ferrod*, a ferrite device whose magnetic state may be sensed

under program control by a *scanner*. These particular ferrods are arranged in rows of 16, so the status of 16 SPUC/DL "packet-present" ferrods may be read simultaneously. Once every base-level loop (100 milliseconds) the 2B ESS will poll all scanner rows containing SPUC/DL packet-present ferrods.

Clearly, such polling is inefficient; for polls on which no SPUC/DL has data, the polling time is wasted. The total time available for processing calls is reduced, and so therefore is the performance of the 2B ESS. The actual time penalty depends on how fast the software is. In our case the loss is around 0.6 percent with idle data links.

Of greater concern are the effects of measured-service billing (billing every telephone call) to the AMARC, and of RSS data-link traffic (several messages are required per call). In the AMARC case, the data links shorten the time spent on billing by about 30 percent relative to the previous use of magnetic tape. In the RSS case, only around 7 percent of the real-time of a typical call is due to the presence of the SPUC/DL. We can conclude that level 3 processing and programmed I/O to each SPUC/DL are not serious drains on the time available to the ESS, relative to its other tasks.

### *5.3 Maintenance software*

The link(s) from the 2B ESS to any single remote site make up a *link group* of one or two links. It is the task of data-link maintenance to ensure that a link is always active in every link group.

#### *5.3.1 Architecture*

The highest-level component of data-link maintenance is called *link-group control*, and has responsibility for keeping a link always active in every group. Because of the substantial differences in the three main applications, link group control was divided into three independent sections, one handling the EADAS link group, one handling the AMARC link group, and one handling all RSS link groups.

Link-group control depends on a collection of utility routines called *link-state control*, which performs tasks on single links, independent of the type of link group of which the link is a member. Conceptually, a given link may be thought of as controlled by, or belonging to, either link-group control or link-state control (but not both) at any one moment. Link-group control manages links that are "in service," that is, actively carrying data ("active") or available for such use if needed ("standby"). Link-state control operates on links that are "out of service" by virtue of having been "removed"—either automatically after detection of a fault, or manually through issuance of a console keyboard command by the operator of the system.

The link-state-control routines depend in turn on a *loader* for down-

loading the SPUC/DL software from its residence on the system magnetic tape cartridge, and on the diagnostic program, which is described in more detail below.

The software of the 2B ESS continuously executes a fundamental "base-level loop" of some 100 milliseconds. Within the loop various functions are given control with the understanding that no function will exceed its own allotment of time. That is, instead of being driven by a clock interrupt, context switches (task changes) are the responsibility of the base-level routines themselves. They must save and restore their own contexts if completion of their functions requires more than one allotment of loop time. For this purpose and for keeping longer-term records, data-link maintenance makes use of control blocks.

#### *5.3.2 Data structures*

Two main types of control areas are maintained by the software. The link-group control keeps a link-group control block for each link group. In addition to current-function context information for link-group control, records are maintained of the number of changeovers between links in the group. This can prevent thrashing back and forth during periods of trouble, such as a lightning storm. In general, this block is intended for those records associated with the entire group, rather than with one specific link.

The second main type of control block is maintained by the link-state control routines, and is called a link-state control block. One such block is reserved for every link defined in the 2B ESS database. Here is stored information on the current context of link-state control for each link. In addition, a collection of records is maintained about the recent history of each link. For example, it is remembered at what time the diagnostic program was last run (to prevent too frequent use of the diagnostic), as well as its result. Over 30 fields are reserved for counting each of the types of errors reported by the SPUC/DL or recognized by the 2B ESS. Cumulative counts are also kept for each of several classes of errors, a class being defined by its relative seriousness.

The other major data structures used by data-link maintenance are the permanent (read-only) office records defining the data links present and their attributes (modem type, hardware address, etc.) These structures are accessed through special database routines.

#### *5.3.3 Recovery strategies*

The primary strategy for recovering from data-link failures is to use the redundancy engineered into the system—that is, to move link traffic to the other link in a two-link group.

The 2B ESS can detect failures of various natures and degrees of severity. The most obvious and straightforward is an inability to communicate with the SPUC/DL because of failures of hardware input or output orders. Here it is first attempted to change, if possible, to the standby 2B ESS processor to see if the orders can then be executed. If not, the fault is classified as *hard*. This is the fault detected when the power supply of a SPUC/DL fails (or is turned off).

A wider range of problems can be discovered by the checksum mechanism. During initialization sequences and when returning an out-of-service link to service, the ESS resets the SPUC/DL by setting a bit in the status register. This action forces the SPUC/DL program counter to an absolute address where its initialization code begins. As part of this code, a checksum is computed on the entire program memory of the SPUC/DL. The computed checksum is returned to ESS along with a prestored value it is supposed to match. The return of a matching checksum shows that the ESS can access the SPUC/DL, that the processor within the SPUC/DL is functioning, that the program is loaded and is probably correct, and that the device is in a known state. A bad checksum probably means a memory fault. In an emergency, a link can be "forced" into service even with a bad checksum comparison, so long as some checksum is returned.

Some symptoms are generally associated with the carrier facility rather than the data link. As mentioned previously, when troubles such as carrier loss are encountered, they are recorded. Noise on the facility will cause low link *efficiency*, defined as the ratio of bytes acknowledged properly to the total number transmitted. These troubles and a variety of others are individually counted within the data-link maintenance software. When the cumulative counts of such errors exceed some threshold value, recovery action begins. The counts of each type of event are weighted by their relative seriousness or impact on the system. The recovery strategy for these (individually) "nonfatal" errors is to compare the recent history of such errors for both links in an RSS link group, changing links if the standby SPUC/DL seems better. In an AMARC link group, since there is no guarantee that a standby link is available at the AMARC site, the 2B ESS must depend on the AMARC to detect facility problems and to change links. Only a severe problem will cause the 2B ESS to remove an AMARC link.

#### *5.3.4 Changing links*

The operation of changing traffic from one link to another in a link group is fundamental in the data-link recovery strategy, and required considerable development effort. The two applications for which the

operation is required, AMARC and RSS, are quite different. We will give a brief overview of each.

*5.3.4.1 Changing AMARC links.* The change operation on an AMARC link group attempts to follow the BX.25 protocol in most respects. Carrier is detected on the standby link after automatic dialing and answering hardware completes the connection via the DDD network. The ESS sees that the AMARC machine issued a "disconnect" on the primary link within its protocol level 2. Next, level 2 of the protocol becomes active on the standby link. Suitable security precautions are now taken to ensure the connection is genuine. Finally, level 3 of the protocol begins, and the operation is complete.

In the other direction the procedure is simpler. After the protocol has initialized at level 2, receipt of any information frame on the primary link will cause the 2B ESS simply to disconnect (hang up) the standby link.

*5.3.4.2 Changing RSS links.* For historical reasons, changing traffic between RSS links is complicated. The procedure was originally developed for the 1/A ESS. As we described in Section II, in the 1/1A environment both RSS links are connected to a single processor, the PUC/DL. Since there is no level 3 protocol for the RSS, resetting protocol level 2 (such as might be expected when changing links) has serious repercussions. Level 2 resets cause loss of data because they clear frame buffers. In the PUC/DL a changeover procedure was designed so no level 2 reset would occur and no data would be lost. The PUC/DL transfers the outstanding frames and current values of the protocol parameters (sequence numbering) to the new hardware by changing a pointer in its database. Thus, after changing links the old level 2 is running on the new link. Similar actions occur in the RSS; no reset of level 2 takes place, and no data are lost.

Since it was part of our design goal to leave RSS code unaltered, it was necessary for the 2B ESS and SPUC/DL to emulate the actions of the PUC/DL during the changing of the links. This was done by the following sequence: (1) protocol level 2 is brought up on the standby link, though transmission of data is disabled; (2) transmission stops on the active link while it is emptied of received data; (3) untransmitted and transmitted but unacknowledged data are read out of the active link and written into the standby link; (4) the protocol state of the active link is transferred to the standby to preserve the sequence numbering; (5) a "change links" message is transmitted to the RSS on the standby link, which turns the standby into the active link, and causes the RSS to change also.

### *5.4 Diagnostic software for the SPUC/DL*

Fault location for the SPUC/DL data links can be best characterized

by its diversity. The diagnostic must contend with three different link configurations—AMARC, RSS, and EADAS—each having its own testing requirements. There are three execution environments: the 2B ESS, the SPUC/DL microprocessor, and for RSS, the remote terminal microprocessor. In addition, the diagnostic faces unusual conditions within the 2B ESS. It must be executable in the off-line processor, and must accommodate several instances of itself executing in parallel.

We have added two features to the diagnostic that are new to the 2B ESS. First, the SPUC/DL diagnostic specifies explicit replaceable units in the TTY output messages, rather than coded trouble location information. Second, to simplify installation and growth of a SPUC/DL configuration, we have implemented *partial* diagnostic capabilities—the ability, for example, to run only the first few tests so uninstalled hardware will not cause error messages.

To achieve flexibility in dealing with different link configurations, the diagnostic was organized for table-driven execution of independent tests, the table being specified either by the application type, as defined in the 2B ESS database, or by an operator-generated input message. This permitted maximum use of common diagnostic code, yet flexibility in tailoring specific tests for specific applications. The table also provided a simple means of providing special test functions not normally run—for example, an exhaustive SPUC/DL program memory test rather than the checksum test normally used by the routine diagnostic.

The diagnostic proceeds with the 2B ESS executing directly initial interface tests to the SPUC/DL, then requesting the SPUC/DL to run tests of its internal hardware. It then uses the SPUC/DL to do analog loop-back testing to the local modem, followed by digital loop-back to the remote modem, and finally (for RSS) requesting remote terminal hardware tests to be run. For those tests not executed by the 2B ESS, the diagnostic acts as a control program and analyzes data to convert the cryptic test failure information into human-readable messages specifying the replaceable unit.

While most of the diagnostic tests for the SPUC/DL data links use conventional diagnostic testing techniques, the digital (remote) loop attempts to find faults normally discovered under heavy-traffic conditions. This test sends 1000 sixteen-byte packets from the 2B ESS through the SPUC/DL, over the link, through the remote loop-back, and back to the ESS through the SPUC/DL. The SPUC/DL uses its operational link efficiency algorithms to quantify the quality of the link. A link is reported as a good link, a failing link, or a "degraded" link—usable but performing below what is considered optimum.

After a fault has been repaired, it is necessary to test the 2B-SPUC/DL interface from both the on-line and off-line central processors.

This ensures not only that faults in the off-line interface are located but that repair of an interface board has not introduced a fault in the standby processor to SPUC/DL interconnection.

The SPUC/DL diagnostic is the first program to take full advantage of a newly enhanced 2B ESS software subsystem for maintenance, which provides a degree of multitasking convenience. Use of the "Multi (base-level) Scan Function" (MSF) program allows for multiple simultaneous executions of the diagnostic on different data links with a single copy of the diagnostic and several instances of its data memory, one for each execution. The only significant complication is message routing from each SPUC/DL to the appropriate data memory, which is done through interfaces with the I/O and the data-link maintenance programs.

## VI. PROJECT POSTMORTEM

### *6.1 Problem areas during development*

Though the difficult problems of the project integration phase are freshest in our minds as this paper is written, there were problems of various kinds throughout the project. The most difficult *technical* problems in the data-link area were met in the integration and final debugging phase of the project. The problems were usually lost words of data, or protocol failures that stopped communication. They occurred randomly in time, usually when data-link traffic was high, and were not easy to reproduce. At least three factors interacted to increase the difficulty in solving the problems.

First, the SPUC/DL test facility did not have enough capacity to stress the links. Only the 2B ESS laboratories could drive the links at the acceptance-test levels, and time in these laboratories was in great demand by people working on all aspects of the 2BE3 generic program. The SPUC/DL test facility was improved, but only after it was clear that serious but infrequent problems were present in the SPUC/DL. The test-facility shortcomings meant that the SPUC/DL had more problems than it should have during the integration phase. The service-affecting problems had been mostly resolved by the time the test facility became really adequate.

The second factor lies in the distributed nature of our systems: with three asynchronous processors, it was often difficult to isolate where any particular problem arose—the 2B ESS, the SPUC/DL, or the application (AMARC, EADAS, or RSS).

Third, debugging tools for the system as a whole lacked some desirable features. The 2B ESS laboratories had powerful debugging hardware for halting the processor, dumping memory locations, tracing execution paths, etc. These facilities were not available on the SPUC/DL, particularly in the late stages of the project when factory-made

units were installed in the laboratory to find any remaining troubles in the production hardware. There were several occasions where it would have been very desirable, for example, to have an event within the ESS halt the microprocessor within the SPUC/DL. We mention in passing that a commercially obtained data communications analyzer proved to be a key instrument in finding certain protocol problems. Ours could record data traffic onto a cartridge tape, stopping after an event of interest; it could also recognize X.25 frame and packet types.

The last few insidious problems in the SPUC/DL turned up in both the hardware and software. The hardware problems were typically sensitivity to "glitches," causing, for example, an erroneous extra DMA cycle. The software problems were usually those of process synchronization and shared resources. The internal architecture of the SPUC/DL, both hardware and software, supports a high degree of concurrency. This increases throughput and minimizes delay, but leads to these internal process synchronization problems. Perhaps more rigorous monitors for shared resources and synchronization semaphores would have paid dividends during the integration phases.

### *6.2 Particularly successful areas*

The problems outlined above led to intense activity as the project neared completion. We were relieved, however, to experience a happy ending; the underlying problems were found and corrected for all observed symptoms.

The climax of our development effort came with the acceptance test for the first 2B ESS—10A RSS field installation. During the 24-hour test, some 60,000 telephone calls were originated on the RSS, processed by the 2B ESS through the SPUC/DL, and properly completed back to the RSS. There were no indications of any problems in the data-link area.

It seems to us that one factor contributing to this success was the attitude that the problems should be *understood* and solved. It might be possible, for example, to stop printing an error message if the message announces some event that seems not to have a harmful effect on the system. We have chosen to explore such events as deeply as resources permit, with gratifying results.

Thus, we feel we have a data-link system essentially free from operational errors. It remains to be seen if our reliability goals will be met.

## VII. ACKNOWLEDGMENTS

We acknowledge the considerable contributions of several people to various phases of the project. G. Dobrowski was involved with the

preliminary analysis leading to the architecture of the system. Most of the original SPUC/DL hardware design is by T. Peterson. T. Brinkman assisted with the initial versions of the SPUC/DL microprocessor code. W. McCalla was responsible for most of the original design and strategy decisions in the data-link maintenance area. Much of the data-link maintenance coding was done by R. Atkins, E. Bily, and H. Williams. J. Kent and D. Sikora produced the program for the 2B ESS to diagnose the SPUC/DL. We thank D. K. Ford and S. C. Yuan for their contributions to troubleshooting the hardware, G. Mannon for assistance with modem issues, and D. R. Bierma for continuous attention to testing and administering new issues of both hardware and software.

## REFERENCES

1. M. S. Sloman, "X.25 Explained," Computer Commun., *1*, No. 6 (December 1978), pp. 310–27.
2. D. W. Davies, D. L. A. Barber, W. L. Price, and C. M. Solomonides, *Computer Networks and Their Protocols*, New York: John Wiley and Sons, 1979.
3. American Telephone and Telegraph Company, Bell System Technical Reference "Operations Systems Network Protocol Specification: BX.25, Issue 3," Publication 54001, June 1982.
4. Special issue on No. 10A Remote Switching System, B.S.T.J., *61*, No. 4 (April 1982).

## AUTHORS

**C. E. Ishman,** Argonne National Laboratory 1961–1967; Bell Laboratories, 1967—. At Argonne National Laboratory, Mr. Ishman worked on hardware for the zero-gradient synchrotron. Since joining Bell Laboratories he has contributed to many system- and maintenance-software areas on the No. 2 ESS, including call-processing features, both diagnostics and fault-recovery for peripheral units, and conversion of the generic program to the 3ACC processor for the 2B ESS. He is presently a member of the No. 5 ESS System Test Department.

**Richard B. Sanderson,** B.S. (Electrical Engineering), 1971, Northwestern University; M.S. (Electrical Engineering), 1972, Stanford University; Bell Laboratories, 1972—. Mr. Sanderson has worked on peripheral circuit design for the number 1, 1A, and 2B ESS machines. He is presently Supervisor of the International Circuit Design Group. Member, Tau Beta Pi, Eta Kappa Nu, IEEE.

**Louis M. Taff,** S.B. (Physics), 1963, Massachusetts Institute of Technology; Ph.D. (Nuclear Physics), 1969, Iowa State University; University of Groningen, Netherlands, 1969–1976; Fermi National Accelerator Laboratory, 1976–1980; Bell Laboratories, 1980—. At the University of Groningen, Mr. Taff engaged in nuclear physics research and became extensively involved with real-time data acquisition systems. At Fermilab, he was a principal in the production of a packaged software system for data acquisition and analysis, now in wide use in high-energy physics. The data-link portion of the 2BE3 generic program was his first project at Bell Laboratories. Mr. Taff is currently working on performance-measurement instrumentation for the number 1A ESS. Member ACM, APS, Sigma Xi.

**Donald P. Truax,** B.S. (Electrical Engineering), 1966, Michigan State University; M.S.E. (Computer Technology), 1967, University of Michigan; Bell Laboratories, 1966—. Mr. Truax has been involved in the development of fault-recovery, diagnostic, and system integrity programs for No. 1 ESS. Projects have included EADAS and ACD data-link maintenance, and HILO and EPSCS trunk maintenance. Mr. Truax also worked on the PUC and PUC-DL maintenance for No. 1 ESS. He has supervised maintenance projects for both No. 1 ESS and No. 2 ESS. No. 1 ESS projects included PUC-DL maintenance for RSS, ETS, and CCIS and diagnostic improvements for the Remreed networks. Mr. Truax currently supervises the MTSO Maintenance Group for the Advanced Mobile Phone Service project. Member, IEEE, Eta Kappa Nu, Phi Kappa Phi.

**Charles T. Tulloss,** B.S.E.E., 1974, Tennessee State University; M.S.E.E., 1975, Ohio State University; Bell Laboratories, 1974—. Prior to his work on the 2BE3 data link, Mr. Tulloss has worked in the area of processor-to-processor communications and protocols on the Enhanced Private Switched Communication Service (EPSCS) and the Electronic Tandem Service (ETS) projects. Currently Mr. Tulloss is involved in hardware test and evaluation for No. 5 ESS. Member, Eta Kappa Nu, IEEE Computer Society.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

## *Modernization of the Suburban ESS:*

# Hosting the No. 10A Remote Switching System

By D. W. BROWN,* J. J. DRISCOLL,* F. M. LAX,* M. W. SAAD,* and J. G. WHITEMYER*

(Manuscript received August 31, 1982)

The No. 10A Remote Switching System (RSS) has brought a new dimension to telephone switching in the rural area. The capability to host a 10A RSS was first made available on the metropolitan switches, the No. 1 and No. 1A ESS. The capability has now been extended to the suburban switch, the No. 2B ESS. This article describes the additions and modifications made to the No. 2B ESS to allow it to host the 10A RSS. The discussion also covers the administrative and maintenance features provided and their associated host software implementation.

## I. INTRODUCTION

### *1.1 Definition and basic description*

A local Electronic Switching System (ESS) provides the call control for a 10A Remote Switching System (RSS). The 10A RSS acts as a slave executing orders sent to it from the host ESS and reports events, such as line originations, to the host. A major advantage of this type of distributed control is that the complex tasks of call processing can use existing host software and share host equipment and trunking facilities. This sharing of host ESS software makes it possible to easily

* Bell Laboratories.

©Copyright 1983, American Telephone, & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

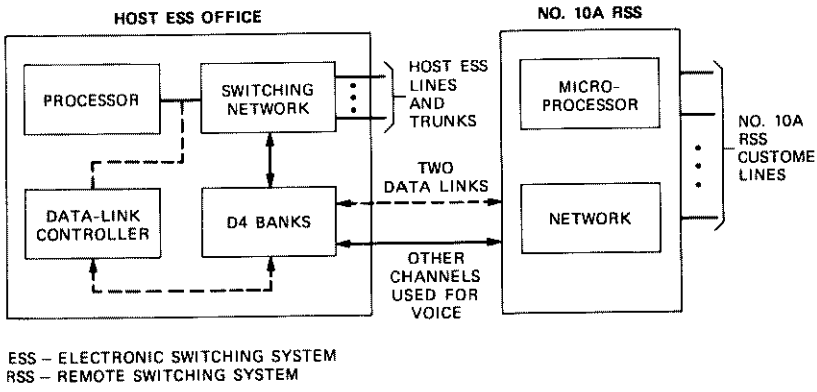



Fig. 1—RSS configuration using digital carrier.

provide RSS lines with the sophisticated features that are offered to host ESS lines.

The major components required for 10A RSS operation include a host ESS, one or more 10A RSS frames, a data-link controller, interconnecting voice channels, and data links. Figure 1 shows this configuration for an application using digital connectivity between the host ESS and a 10A RSS (remote terminal). At this time, the host function has been developed for the Western Electric No. 1 ESS, No. 1A ESS, and No. 2B ESS machines. The data links are used for communication between the host ESS and the 10A RSS and provide the means by which orders from the host are transmitted to the 10A RSS and acknowledgments are returned to the host. Voice channels are used to provide the RSS lines with access to the host network and are selected dynamically. The data link employed for the No. 2B ESS-RSS communication function is a new design utilizing an intelligent Serial Peripheral Unit Controller Data Link (SPUC/DL).* The 10A RSS data-link communication makes use of portions of the X.25 protocol.

The 10A RSS basic frame can serve up to 1024 lines. A companion frame may be added to allow up to 2048 lines to be served by a single RSS entity. The design is such that the basic element of growth can be as small as eight lines. Voice and control communications between the remote and the host can be made over either digital or analog carrier facilities and the range may be as great as 280 miles, depending primarily on the type of transmission facility.

In the event of total carrier system or data-link outage, the 10A

* Acronyms and abbreviations used in this paper are defined at the back of this Journal.

RSS is arranged to automatically transfer to a stand-alone mode of operation, which provides basic telephone service between stations connected to that RSS unit. In the stand-alone mode, special provisions can be made to handle emergency types of traffic such as "911". Details regarding stand-alone operation may be found in an earlier BSTJ article, "Remote Terminal Firmware," by D. A. Anderson et al. in the April 1982 issue on the 10A RSS.

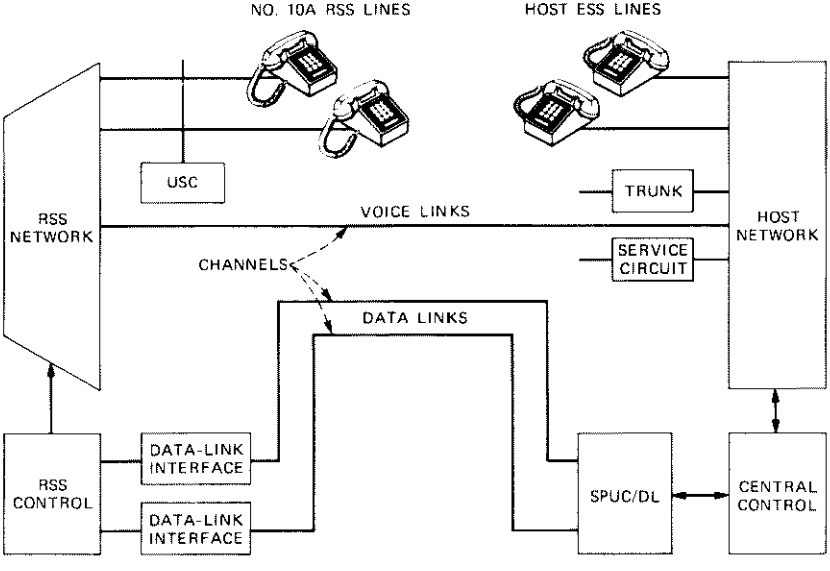
All major units of the 10A RSS are duplicated, and a continuous dialogue is exchanged between the host and remote site concerning the overall health of the system. The basic system philosophy is that the remote unit is a complete slave to the host and merely reports events to the host; the RSS then receives a stream of explicit orders from the host concerning that event. All maintenance procedures must be controlled from the host. Initiation of diagnostics and other functions may be further remoted to a Switching Control Center (SCC).

## II. CALL PROCESSING

### *2.1 Data communications*

#### *2.1.1 Hardware overview*

Figure 2 shows the 10A RSS—No. 2B ESS host interface and the




ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM
SPUC/DL – SERIAL PERIPHERAL UNIT CONTROLLER/DATA LINK
USC – UNIVERSAL SERVICE CIRCUITS

Fig. 2—10A RSS—No. 2B ESS host interface.

hardware components involved in the transmission of data between the remote terminal and the host. Communication between the two machines takes place over a pair of low-speed data links that share the same transmission facilities as the voice channels that interconnect the remote terminal with the host. Each data link is placed on a separate transmission facility for reliability. Where carrier facilities are used, the links are assigned to a dedicated voice channel on the carrier system, with each link being assigned to a separate carrier terminal.

Both RSS links are 2400 bps synchronous links. The on-line link is active and carries the entire data traffic between the host and 10A RSS, while the off-line link is maintained in a standby state as a spare. The on-line, off-line status of the links is determined by the host office and is based on error information accumulated by the software responsible for driving the links. At the remote terminal the link is interfaced to the RSS microprocessor through a Data-Link Interface (DLI) circuit. The DLI provides a small amount of data buffering and performs a number of control functions essential to implementing the synchronous link protocol.

At the host, the link interfaces with a functionally similar line interface unit that is part of the SPUC/DL. The function of the SPUC/DL is to provide the control for physically transmitting and receiving data on the links and to provide data buffering for the host office. The data being transmitted and received by the SPUC/DL are buffered on a per-link basis within the terminal. Sufficient data buffering is provided to allow the host to efficiently exchange large blocks of data with the terminal on a schedule that is efficient to the host.

#### *2.1.2 Software overview*

The routines that control the data transmission between the remote terminal and host are located in the remote terminal, the SPUC/DL, and the host office. There are two basic functions to be performed: data must be transferred reliably over the link and an interprocess communication system must be provided to allow software processes in the host to communicate with processes in the remote terminal. These two functions are provided by the data-link protocol software and a set of message-routing routines. The protocol provides virtually error-free transmission of data over the link by executing a set of error-detection and error-correction procedures. The message routines allow a process in one machine to direct a message to a process in the other. These two systems are largely independent and bear a hierarchical relationship to one another in the sense that the message routing routines rely on the link protocol routines to accurately transmit data from one end of the link to the other.

### 2.1.3 Data-link protocol

The protocol routines are executed in the remote terminal and the SPUC/DL. The 10A RSS application uses the link-level portion of the X.25 protocol to control the link. It is a bit-oriented protocol designed for synchronous link operation. To provide for error detection and correction, the data to be transmitted are segmented into numbered blocks termed frames. As frames are transmitted they are sequentially numbered and a cyclic check code is computed over the data in the frame. The frame numbering scheme makes it possible to identify frames for retransmission and to detect missing frames in the received data. As frames are processed at the receiving end of the link, the cyclic check code is recomputed and compared to the one transmitted with the frame. A mismatch indicates that a transmission error has occurred. A positive acknowledgment is returned to the data transmitter for all frames received correctly, and a retransmission request is returned if a frame is received in error.

The protocol software at the SPUC/DL has the additional function of providing link status reports to the host machine. Error conditions such as high transmission error rate, frame acknowledgment timeouts, and loss of data carrier are monitored by the protocol and reported to the host. The transmission error rate is determined from the number of retransmission commands received and sent by the SPUC/DL protocol program. From this data the host data-link state control can take action to remove a link from service if it becomes inoperative or if its throughput is restricted because of excessive data errors.

### 2.1.4 Message-routing routines

The routines that are responsible for routing data between individual processes in the two machines are executed by the remote terminal and the host processor. These programs assume that data received from the link protocol programs are error free and that any additional error control procedures for detecting transmission errors are unnecessary. These programs are designed to transmit data between buffers associated with client programs in the two machines. A program having data to transmit will load the data into its associated buffer. The buffer will be activated for the message-routing routines and the data will then be transferred to a buffer associated with the destination program in the other machine. The message format is shown in Fig. 3.

### 2.1.5 Host data transmission

The buffering technique adopted for the transmission of peripheral orders to the remote terminal was designed to be compatible with the structure of the existing host software. The order buffering and mes-

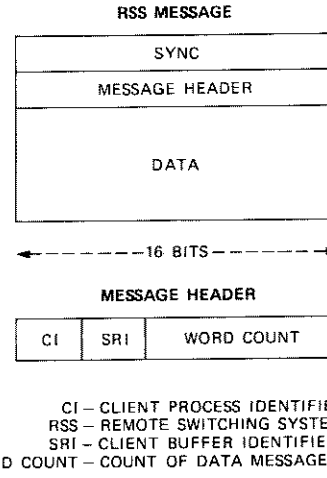



Fig. 3—RSS message structure.

sage transmission must be tailored to the host software structure if the existing call-processing and maintenance routines are to be preserved. A few remarks on the nature of the call-processing programs are necessary to understand the interface requirements.

The host call-processing programs are divided into call segments that process an input from a subscriber or a peripheral circuit to completion in one real-time segment. All the peripheral orders required to process the input are generated by the separate set of input/output (I/O) programs. Peripheral operations in the host or remote terminal take tens or hundreds of milliseconds to execute, which precludes their direct execution within the call segment. During execution, a typical program segment may generate several remote terminal peripheral messages that are destined for different RSSs. In most cases it will also generate a number of orders to be executed in the host periphery in conjunction with the remote terminal actions. The execution routines must be able to coordinate the transmission of the remote terminal orders to the different RSSs. Frequently, it is necessary to execute the remote terminal-host peripheral actions in a predefined sequence where either the host or the remote terminal action must occur first.

The call-processing and maintenance programs are coupled to the I/O programs through a set of I/O buffers that are loaded with the peripheral orders to be executed. All host peripheral orders are buffered in Peripheral Order Buffers (POBs), where they are executed by the POB execution programs designed to handle the timing requirements presented by the host periphery.

A set of Remote Order Buffers (ROBs) in the host machine buffer

the peripheral orders being transmitted to the remote terminals. They are loaded and administered by the call-processing programs in the same manner as the peripheral order buffers. The orders to be sent to the remote terminal are loaded in the ROB via a set of order macros that provide a high-level interface with call processing. When order loading is complete, the ROB is activated and the I/O routines transmit the orders to the remote terminal. An individual ROB may be used to send orders to any RSS. The host can have any number of ROBs pending to send orders to an RSS; however, each remote terminal has a fixed set of eight ROB records that are used to store orders received from a ROB at the host. The ROB records are buffers associated with the peripheral order execution program in the remote terminal that executes the orders transmitted from a ROB.

#### *2.1.6 ROB execution protocol*

A rudimentary protocol has been established to coordinate the activities of the call-processing routines at the host with the execution of ROB orders at the remote terminal. Several orders will be grouped together into a single message at the host to be transmitted to the remote terminal. The orders in the message are executed at the remote terminal and upon completion an acknowledgment is returned to the host. The acknowledgment will indicate whether all the orders in the message were successfully executed and must be received by the host before any further actions will be permitted on this call. If the remote terminal encounters a failure in executing an order, the acknowledgment message will specify the failed order to the host and the remote terminal will suspend execution of all remaining orders in the ROB record.

It is essential to receive a positive confirmation on the status of the orders for several reasons. If an order failure occurs, the host fault-recovery routines can be scheduled to clear the call from the system. In addition, the acknowledgment allows the host to correctly sequence any other peripheral actions with the remote terminal orders. When the acknowledgment is received, the host can activate an associated POB or return to a call-processing program to implement the next action on the call. The restriction that additional RSS orders will not be transmitted until the acknowledgment is received also prevents the host from transmitting multiple sets of orders for the same call that would be executed in an arbitrary sequence at the remote terminal.

### *2.2 RSS network associated data*

The processing of RSS calls requires the allocation and management of resources that are physically located at the 10A RSS or shared between the host ESS and the 10A RSS. These resources include the channels that interconnect the host and 10A RSS, receiver off-hook

(ROH) tone circuits located at the 10A RSS, and the 10A RSS network crosspoints. Also, the status of RSS lines is maintained at the host. Several factors were considered in deciding whether to place these functions in the 10A RSS or in the host ESS. These factors are:

1. The effect on service because of the additional time delay if the 10A RSS has to be interrogated to determine line status and to hunt voice channels and network paths.
2. The additional software development required if the host ESS programs have to take a real-time break to interrogate the 10A RSS to obtain a line's status. Host software is structured around data that can be accessed without taking a real-time break.
3. The duplication of development effort that is required to provide the same functions in several host ESS systems.

Factor 3 indicates that the overall development effort would be reduced by placing the line, channel, and 10A RSS network path administration in the 10A RSS. However, the service criteria and the effect on the existing host software were judged to be more important. Therefore, these functions are allocated to the host ESS.

The overall development effort is reduced by making the program and data structure designs independent of the host ESS. Thus, they are highly portable between ESS machines. This section describes the data structures required to administer the RSS resources.

#### *2.2.1 Data structures*

Data structures are required in the host ESS memory (call store) to record the status of the RSS facilities and to provide for their administration. To simplify the engineering of the office, these data structures are provided in sizes that correspond to the three basic network sizes of the 10A RSS. Each of these structures is described below.

***2.2.1.1 Network block.*** One network block is provided for every 10A RSS. Its size is fixed regardless of RSS equipage. Among the subblocks contained in the network block are the network map, the channel status blocks, and the remote miscellaneous scan-point status map.

Scan points are provided at the 10A RSS for uses such as alarms, make-busy keys, and stop-hunt keys. The remote unit periodically scans these scan points and, via the data link, reports any changes to the host ESS, which updates the bits in its map to indicate the present state of the scan point.

***2.2.1.2 Path memory remote record.*** A Path Memory Remote (PMR) record is provided for each possible line and channel network appearance. Since the 10A RSS network can be equipped in three different sizes, considerable ESS memory is saved by also providing PMRs in block sizes corresponding to the network size. PMRs contain information about the state of the terminal and a pointer that is used to

point to another memory block (call register or path memory for junctor) involved in the call.

*2.2.1.3 Path memory for junctor record.* A Path Memory for Junctor (PMJ) record is a block of call store that is associated with a junctor in the 10A RSS network. It is used to store path and terminal information when the junctor is in a network path. A PMJ also contains a state and pointer, which are used to link to another PMJ or to a call register. A PMJ is provided for each equipped junctor. Blocks of PMJs are allocated based on the RSS network size.

### *2.3 Call-processing strategy*

The RSS host call-processing software provides an ESS central office with the capability to supply ESS features to lines served by the remote switching system. Since most of the call-processing functions for RSS lines are performed by the host ESS office, a full family of ESS features can be provided to the remote subscribers. The RSS call-processing software resident in the host ESS provides the means of controlling a remotely located switching system by taking advantage of existing equipment and control capability in the ESS. Firmware in the remote terminal supplements the host call-processing software appropriately. All call-processing control resides in the host ESS and any required actions at the RSS are requested via data-link messages to the RSS. This permits the host to exercise total call control.

#### *2.3.1 Originating call*

A line originating in the RSS is first recognized during line scanning performed by the RSS microprocessor. The RSS line-scanning program in the remote terminal recognizes the line off-hook, performs hit timing, and sends an origination request data-link message to the host ESS. If service is allowed, the host marks the RSS line busy and hunts an idle voice channel between the RSS and ESS. It also hunts a path through the RSS network from the originating line to the selected voice channel, and selects a customer digit receiver in the host along with a host network path from the voice channel to the receiver. A Remote Order Buffer (ROB) is then executed to send appropriate data-link messages to the remote terminal to set up the RSS network path and set the line supervision mode to repeat supervision of the originating line over the channel in the dialing (fast repeat) mode. To minimize the dial-tone delay interval, when the above ROB is initiated the host executes orders to its periphery (via a POB mechanism) to set up the host network path between the voice channel and receiver to provide dial tone. That is, the ROB and POB are executed in parallel.

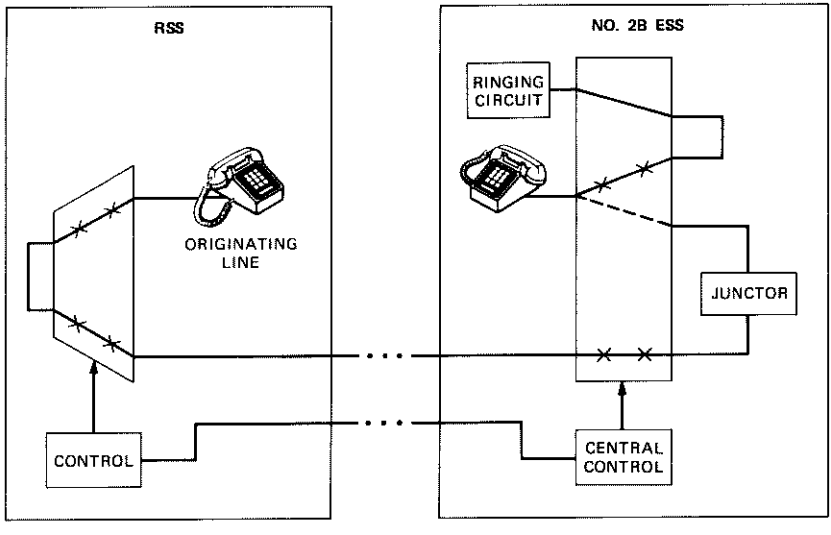
Processing of the call from this time on proceeds basically the same

way as an origination by a host line. Digits are collected and analyzed by the same host software used to process host calls. At the completion of dialing and digit collection, a data-link message is sent from the host to the remote terminal to set the line supervision mode to repeat the supervision of the originating line over the channel in the talking (slow repeat) mode. This mode conserves remote terminal microprocessor real-time capacity.

The RSS originating call, from this point on, is routed and completed normally (excluding terminations to RSS lines) just as non-RSS line origination processing. This originating call configuration is depicted in Fig. 4. Upon answer by the called party or completion of outpulsing, the talking connection is established from the voice channel through the host network. RSS answer timing, billing, traffic, and other administrative functions are all applied and performed by the host just as for non-RSS calls. If the call terminates to an RSS line, special terminating RSS functions are performed, as discussed in the following sections. When either the calling or called parties disconnect, disconnect functions are performed, as discussed in Section 2.3.5.

### *2.3.2 Terminating call*

An RSS terminating call is recognized when the host ESS performs the called number [terminating Directory Number (DN)] translation on digits collected from an originating host line or trunk. RSS lines




ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 4—RSS originating call.

are distinguished from host lines by special RSS indicators in the terminating line translation output. After the translation is completed, special actions, as with the RSS originating call, are required to set up ringing. The host hunts an idle voice channel to the RSS, hunts a path in the RSS network between the voice channel and the terminating line, and seizes an idle host ringer and audible service circuit with associated host network paths to the voice channel and originating line or trunk, respectively. In addition, a talking path through the ESS network (between the voice channel and the originating line or trunk) is reserved.

A ROB is activated to send data-link messages to the remote terminal to connect the terminating line to the voice channel and apply ringing to the line. Upon receipt of the data-link orders, the remote terminal selects an idle universal service circuit along with a metallic bus and time slot to provide the type of ringing specified in the data-link message. Supervision of the line is transferred across the voice channel to the host in the fast repeat mode.

Upon successful execution of the ROB data-link orders in the remote terminal, the host executes a POB to set up paths in the host network from the voice channel and the originating line or trunk to its associated service circuit. Power cross and low-line resistance tests are done on the voice channel from the host ringing circuit. The host ringing circuit is then left in a state to monitor ring trip sent by the remote terminal over the voice channel to the host. Actual ringing is applied to the line by the Universal Service Circuit (USC) at the remote terminal; the ESS host ringing circuit does not apply ringing voltage to the voice channel, but is only used to monitor for ring trip. This call configuration, as depicted in Fig. 5, maximizes use of the existing terminating call sequences in the host.

When the called party answers, the remote terminal automatically releases and idles the ringing facilities (USC, metallic access bus, and time slot), relays the ring trip report (off-hook signal) of the line across the voice channel, and sets the supervisory mode to slow repeat.

The host ESS detects answer over the voice channel at the ringing service circuit, tears down the ringing and audible circuit connections in the host, and sets up the talking path that was previously reserved between the voice channel and the originating line or trunk. If the originating line in the RSS terminating call description is actually another voice channel to the same RSS as the terminating line, the call is considered an intra-RSS call and special actions are invoked as described in Section 2.3.4.

Disconnect actions are identical to those for the RSS originating call except for the disconnect timing associated with the terminating versus the originating party.

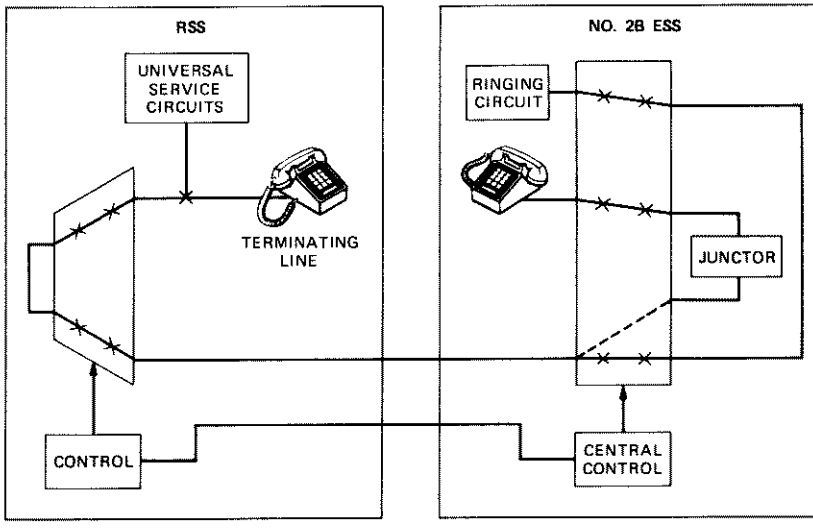



Fig. 5—RSS terminating call.

### *2.3.3 RSS reverting calls*

RSS reverting calls involving a call between two parties on a party line are handled in a manner similar to host reverting calls. However, ringing is provided in a way similar to how it is applied on RSS terminating calls with the exception that two time slots are needed in the RSS remote terminal so that ringing can be applied to both customers. This RSS ringing option, which can be either ac/dc or superimposed, is completely independent of the host ESS office ringing option, or any other RSS served by the same host. The RSS universal service circuit has the capability to provide either ringing option under firmware control.

### *2.3.4 Intra-RSS call*

An intra-RSS call, where both the originating and terminating parties are served by the same RSS, is handled initially as a combination of an RSS originating call and an RSS terminating call. In the No. 1 ESS and No. 1A ESS applications, after answer, the host initially establishes a talking path within its network between the two voice channels. Immediately following the establishment of this talking connection, certain RSS actions are invoked to reswitch-down the call so that the talking connection resides entirely within the RSS network. This releases the ESS network path and voice channels for use on other calls. In the No. 2B ESS application, when answer is received

and both parties are served by the same RSS, the call is reswitched-down immediately without first establishing a talking connection through the host. This sequence will not result in a momentary open interval after the initial talking connection has been established.

The reswitch-down action is initiated when the host hunts an RSS network path within the RSS between the originating and terminating lines. A ROB is activated to send data-link orders to the remote terminal to disconnect both line-to-channel network paths and connect the two lines through the RSS network. The supervisory mode of the RSS lines is set to scan for either a disconnect or switchhook flash, depending on the features associated with each line. Since the intra-RSS connection is entirely within the RSS, a change in supervisory state of the line must be reported over the data link to the host. The sequence of intra-RSS call configurations including reswitch-down is illustrated in Fig. 6.

If a network path in the RSS is not available or if one of the lines is an RSS coin line, the intra-RSS call is not reswitched-down and is connected through the host ESS network. Intra-RSS calls involving coin lines are not reswitched-down in order to utilize host coin-disconnect routines and thus simplify disconnect actions.

The use of various custom calling services or other special services requires a reswitch-up of an intra-RSS call to establish a talking path between the two parties via the host network using two voice channels. This allows existing host software and equipment to be utilized to provide these customer services. The following operations require a reswitch-up operation on an intra-RSS call:

1. A flash by an RSS customer to add on a third party
2. A terminating call to one of the two parties of an intra-RSS call that has the call waiting/terminating feature
3. A busy verification test by an operator of one of the two parties of an intra-RSS call.

When the host determines that a reswitch-up function must be performed, for any of the reasons given above, the host seizes two idle voice channels to the RSS and hunts a path between them in the host network; the host also hunts a path between the two voice channels and both lines in the remote terminal. A POB is executed in the host to set up a talking connection between the two voice channels followed by a ROB to send data-link messages to the remote terminal to disconnect the intra-RSS talking connection, connect each line to a voice channel, and set the supervision state of each line to the talking mode (repeat supervision of the line over its respective voice channel). Once the intra-RSS call is reswitched-up to a talking connection via the host ESS network, the original service requested can be provided just as it would to a normal line-to-line connection of two host lines.

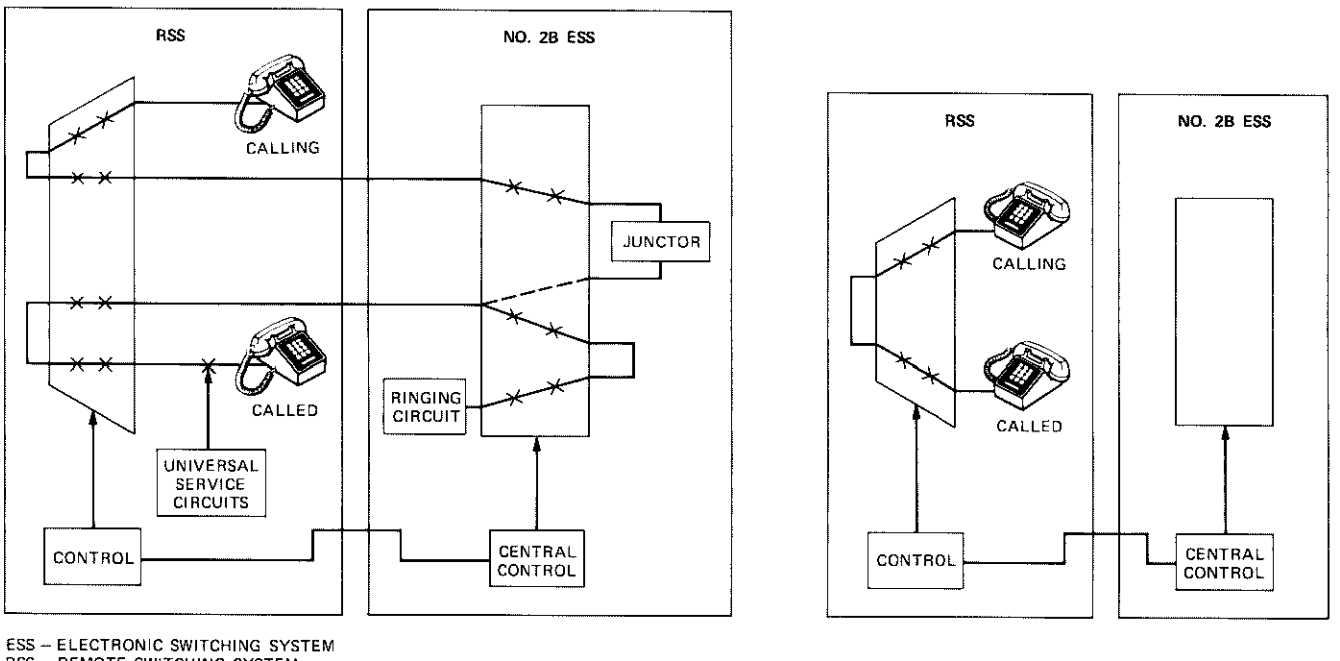



Fig. 6—Intra-RSS connection sequence. (a) Ringing connection. (b) Talking connection.

### 2.3.5 Disconnect functions

Disconnect actions for RSS calls are a function of the particular call configuration involved, i.e., intra-RSS calls or RSS calls through the host network. For call configurations involving both RSS and ESS paths, the ESS disconnect programs control the call-disconnect actions. The same disconnect sequence that is performed on ESS host lines is used on RSS channels that terminate on the host line network. This disconnect control strategy provides the ability to centrally recognize an RSS channel during normal host-disconnect processing. This recognition occurs when ESS programs perform a restore-verify action on the RSS channel. At this point in the host-disconnect processing, unique host RSS disconnect modules are invoked to disconnect the network path of the RSSs. The normal host-disconnect program and RSS-disconnect module then autonomously complete their respective disconnect actions. The only common resource between the two control programs is the RSS channel. The ESS host-disconnect program administers the host end of the channel on its network, and the RSS host-disconnect program administers the RSS end of the channel on its network.

In the case of an intra-RSS call, where the call configuration involves a connection totally within the RSS network and no channels or ESS network paths are involved, the RSS lines are supervised in the 10A RSS. Hits, flashes, and on-hooks are detected by the RSS; flashes and on-hooks are reported to the ESS host via data-link message. These messages are routed to unique RSS disconnect control modules in the host for proper processing. In the case of an on-hook, these programs perform the proper disconnect timing and then execute ROBs to disconnect the intra-RSS network paths and idle the lines involved. The intra-RSS call is reswitched-up on receipt of a flash message (as discussed in Section 2.3.4).

## 2.4 Database integrity

In an electronic switching system, the status of resources and telephone calls is recorded in temporary memory. This data can become mutilated because of program bugs, hardware errors, or program design errors, resulting in the loss of resources or system degradation. The problem is further complicated by RSS because parts of the new data structures in the host ESS are duplicated in the remote terminal. The new structures are:

1. PMRs for lines
2. PMRs for channels
3. Network map
4. Remote order buffers (ROBs)
5. Remote miscellaneous scan point map.

The actual data stored in the two copies of these structures are not identical in all cases since the host and remote terminal do not perform identical functions. For example, the state stored in a line PMR at the remote terminal represents the supervisory state of the line (origination, repeat supervision onto a channel, high and wet, etc.), whereas the state in the host PMR represents both the status of the line (idle, busy, maintenance) and the type of path memory configuration, as discussed previously. However, there is a mapping between the two sets of states in that the host-line state implies the possible supervisory states of the line. Deviations from this mapping should only be due to the time lag of the data link.

To ensure the integrity of the data structures, the first step is to prevent as many errors as possible. RSS software applies many of the techniques that have been successfully used in other ESS projects to avoid potential causes of errors. Some of these are good documentation, standardized program interfaces, structured design, structured programming, a high-level programmer's language, and a standardized data definition language. In addition, access to the new data structures introduced into the host is limited to the administrative programs that have the responsibility for that particular database.

The second step is to make the programs as error tolerant as possible since data errors will still occur. The main technique for this is defensive programming. The degree to which a data error is propagated through the system depends upon how the programs use the data. In order to have a minimal effect upon the system, programs should account for bad data. Some specific types of defensive coding techniques are:

1. Range checks on data to prevent overindexing tables
2. Accounting for all possible subroutine return code values
3. Use of symbolic definitions for data values
4. Accounting for all possible program inputs
5. Invalid data value checks.

Despite the preventive and defensive techniques that are employed, errors can still occur in the data. Programs are required to detect these errors and restore the facilities to the proper condition to avoid system degradation. These audit programs are responsible for detecting and correcting data errors and the initialization programs are responsible for restoring system facilities when the degradation is severe enough to cause major system degradation.

#### *2.4.1 Audit programs*

The integrity of the data structures is checked and corrected by a set of audit programs. Each audit program is individually tailored to a specific data structure or group of data structures and determines if

the data items follow certain established rules. If the checks fail, the audit programs make appropriate corrections to all resources (both software and hardware) associated with the particular error and print error messages on the teletypewriter. The audit programs also aid in the initialization of the data structures.

The audit philosophy adopted for RSS is that each entity will maintain the integrity of its databases independently. If the host finds a discrepancy in its database, it corrects the problem by resetting the state of all host resources involved and instructs the remote terminal to put its facilities into a known (usually idle) state. If the remote terminal finds a discrepancy in its database, it sends a message to the host and the host audit programs initiate the actions given above.

Thus, there are three classes of audits associated with the RSS system:

1. Host audits that maintain the internal integrity of the data structures in the host
2. Remote terminal audits that maintain the internal integrity of the data structures in the remote terminal
3. Audits that guarantee that the host and remote terminal data structures are consistent.

Audit classes 1 and 3 are discussed below.

***2.4.1.1 Host audits.*** Existing host audits are extended to include the new data structures and new data values introduced with RSS. The main audit modifications are for the RSS path memory, the RSS network map, channels, and ROBs.

The RSS path memory and network map are audited by making the following checks:

1. Point-to-point-back checks are performed between PMRs and PMJs.
2. Point-to-point-back from a PMR or PMJ to a call register are performed if linkage to a call register is indicated.
3. The junctor busy-idle bit in the network map is checked to ensure that it is idle if and only if the PMJ is idle.
4. Each of the other network map bits that is marked busy is checked to guarantee that it is in a valid path.

ROBs are audited by periodically rebuilding the idle link list and by timing ROBs associated with transient call records. If a ROB remains busy for an extensive length of time, the ROB and any associated call register are idled. All paths and circuits are also idled.

The corrective action taken by the host audits is to idle facilities (hardware and software) in the host and to send orders to the remote terminal to cause the facilities at that end to be idled.

***2.4.1.2 Synchronization between host and remote terminal.*** The problem of maintaining the data structures in the host and remote terminal in

synchronization is greatly simplified by taking advantage of the normal system operation. The remote terminal updates its data in response to orders from the host. Bits in the remote copy of the network map are marked busy or idle in response to orders to set up or tear down network paths. Thus, no network map audit is required between the host and remote terminal since the host does the hunting and idling of paths and the remote copy will tend towards the proper state even if it does temporarily get out of step.

Similarly, ROBs are controlled from the host end and normal operation will result in the remote copy being brought back into step with the host.

The remote terminal audits check that all equipped lines have supervision turned on. Any equipped lines that are unsupervised are reported to the host via a data-link message. If the host audits determine that the line state really calls for supervision to be on, the line and any associated resources are idled.

Periodically, the host sends a copy of its version of the remote scan point map to the remote terminal, which overwrites its map with the host's data. If the hardware state of the scan point differs from the map, the normal scan program will detect this as a change and report it to the host, resulting in both copies being brought back into step.

#### *2.4.2 Remote terminal initialization*

System initialization programs are responsible for correcting errors that prevent the system programs from cycling correctly. The initialization programs are usually executed as a consequence of errors being detected by the processor check circuits that monitor the sanity of the system operation. In the remote terminal, the primary checks for monitoring proper program operation are the system sanity timer, which monitors the main program cycle time; the write protect circuitry, which prevents illegal writes into program store; and certain peripheral error checks, which detect attempts to access unequipped areas of the periphery. If any of these errors are detected, it is indicative of an error in the system database. The method of recovery is to initialize a segment of the data and then return to the normal program cycle.

Since the initialization process inherently destroys a portion of the call-processing data, a corresponding set of calls will be lost, and it becomes a requirement for the initialization program to release the peripheral circuits associated with these calls. Any network links, channels, or service circuits employed on these calls must be idled by the initialization program before a return to normal system operation is begun. This is accomplished by releasing all the circuits that are marked idle in the initialized database.

Whenever an error is detected by a fault-detection circuit, a processor interrupt is generated that executes the fault-recovery programs. Various fault-recovery actions are taken, depending on the type of errors detected and their frequency.

The amount of data that is initialized on the first error is small. As successive errors are detected, the severity of the initialization is increased with the consequent loss of progressively greater numbers of calls. The ultimate action is to initialize the entire database and restore the system to an idle state. The goal of handling the fault recovery in stages, with increasingly more severe initializations, is to restore the system to a working mode with the loss of a minimal number of calls.

At either the remote terminal or the host, there are three fundamental levels of initialization: a minimal clear, a transient clear, and a stable clear. A minimal clear involves the initialization of the variable data associated with the active processes in the system and has the potential of disrupting, at most, several calls. A transient clear will initialize all the data in the system that is related to any call in progress. All calls in the process of being established or disconnected will be lost; however, calls in a stable talking state will be preserved. The final state of initialization is a stable clear where the entire database is reinitialized and all calls are lost.

For the RSS system, the initialization process is somewhat more involved than normal because the host and remote terminal databases are interrelated and an initialization level (phase) in one machine affects the database of the other system. Although the host machine is responsible for the control of the remote terminal on a call-related basis, the operation of the processors in the two machines is fairly autonomous with respect to their instantaneous activities. This is the situation for fault detection where the two systems are entirely independent. Each machine is responsible for initializing and carrying out its own fault-recovery actions and the level of the accompanying initialization will be solely determined by the conditions within the machine that detected the error. In effect, either machine is able to initiate any level of initialization on its own database independently of the other. However, as part of the initialization procedure, the hardware and software within the two machines must be synchronized so that the databases reflect a consistent set of calls. The calls that were destroyed in one machine must be reported to the other so they can be cleared from that system also. For example, a host-transient clear will destroy a number of calls that involve remote terminal lines. These calls must be cleared in the remote terminal so that periphery and call records are in agreement with those in the host.

The exact procedure for synchronizing the two machines will depend

on which system has initiated the phase. Since the host is in charge of call control, its call records are regarded as the master copy and the remote terminal state is brought into agreement with its set of records. The host is therefore in charge of synchronizing the two machines.

Whenever a phase occurs in the host, the synchronization procedure is straightforward. The host will initialize its database and periphery and will then send initialization orders to the remote terminal to bring it into agreement with its updated records. When the remote terminal undergoes an initialization, it reports the level of the initialization to the host. In some instances, on a stable clear, for example, this is sufficient information to allow the host to initialize its database. In the case of a transient clear, it is also necessary to transmit a map of the lines that are in a transient state in the remote terminal. From the data specifying the initialization level and the map of transient lines, the host is able to update its call records and periphery. Once this is accomplished, it will conduct an initialization of the remote terminal in the same manner as for a host-initiated phase.

During a high-level initialization, interactions between the host and the remote terminal must be modified. For example, during the resynchronization of the host and remote path memory transient databases, it is necessary to prevent these databases from being accessed by call-processing routines. To implement this capability, the RSS is modeled as a finite-state machine with ten possible states. The state of each remote terminal together with a list of activities allowed in that state is maintained by the host in a database. Software, which functions differently based on the state of the remote terminal, checks this database to determine what action is appropriate in the given RSS state.

## III. MAINTENANCE ENHANCEMENTS

Extension of telephone communication service through the 10A RSS brings with it an associated extension of maintenance capabilities. An obvious maintenance need relates to the SPUC/DL and data-channel facility between the No. 2B ESS and No. 10A RSS processors. Integrity and maintenance issues for this subsystem are dealt with in the companion article of this series, "Adding Data Links to an Existing ESS," by C. E. Ishman et. al.

Another area requiring maintenance enhancement is related to the addition of voice channels to the network architecture of the combined systems. Unlike other "links" of the No. 2B ESS network, the voice channels are nonmetallic, largely external to the central office environment, and are maintained by a separate craft force. These factors require the design to deal with issues of availability, transmission and

noise performance, and trouble sectionalization. The channel maintenance software package, developed to address these issues, is described in the following sections.

Other maintenance additions, for telephone customer loop testing, are necessitated by the remoteness of the RSS and the nonmetallic network upon which these loops terminate. Testing capabilities, similar to those available for host-terminated lines, must be provided for RSS customer loop trouble detection and resolution. Such capabilities are described in the following sections on line maintenance. Finally, as a means of achieving test hardware economies in the RSS, the host diagnostic capabilities were expanded to test Dual-Tone Multifrequency (DTMF) receivers at the RSS as well.

### *3.1 Channel maintenance*

Basic needs for the effective maintenance of RSS voice channels may be divided as follows:

1. The system must provide routine, automatically initiated tests that periodically diagnose individual channels to detect performance-affecting faults—particularly faults characterized by progressive and marginal degradation.
2. The above set of tests should also be initiated by call-processing programs in instances where in-process integrity checks indicate errors that could be caused by faulty channels.
3. The resolution of some hardware faults will require manual intervention. This is accomplished by: (a) allowing the above diagnostic tests to be executed upon demand, and (b) providing dc and ac test access to each channel for voltmeter-type testing. This latter facility has been provided through an enhancement of the existing central office Trunk Test Panel (TTP).

#### *3.1.1 Channel diagnostics*

The channel diagnostics, which play a major role in fulfilling all three needs, have been designed with a "start-small" philosophy; the sequence of tests is selected so as to confirm basic functions first, and then to build upon this knowledge in carrying out subsequent evaluation. The following sequence of tests is terminated when a failure is detected at any step:

1. Power cross test
2. False cross or ground test
3. Supervision test
4. Restore-verify test
5. Low line resistance test
6. Dial pulse test

7. ac far-to-near test
8. ac near-to-far test.

The power cross test, shown schematically in Fig. 7a, uses the same detecting circuit as do customer lines and is designed to detect crosses to central office battery or commercial power in the cabling between the No. 2B ESS switching network and the carrier terminal. By performing this test first, potential damage to other test circuits is avoided. The False Cross or Ground (FCG) test cannot use the existing test method since it is overly sensitive to the capacitative load seen at the input of the carrier's channel unit. Instead, the existing continuity and polarity test circuit is connected, as shown in Fig. 7b, to detect the existence of conductor-to-conductor and conductor-to-ground shorts (up to approximately 2000 ohms) in the same cabling.

The supervision test verifies the ability of the channel to pass on-hook and off-hook signals from the RSS remote terminal to the No. 2B ESS host. With the channel idle, on-hook supervision is first confirmed by applying loop battery toward the channel with the continuity test circuit (as diagrammed in Fig. 7c). After sending an off-hook order to the RSS (via the data channel), the central office current detector is scanned for the expected off-hook.

The restore-verify test, in a manner similar to that used with customer lines, verifies that the supervisory attending element can be reconnected to the channel and can detect the off-hook signal associated with a channel "origination." This is accomplished, as shown in Fig. 7d, by causing the RSS to transmit an off-hook signal toward the host when the host has its line attending element connected to the channel.

The low line resistance test is still another carry-over from host-line testing. Here, the objective is to detect the existence of high-resistance (up to approximately 15,000 ohms) crosses from conductor to conductor and from conductor to ground. While the intent of making such a test on a customer line is to prevent false ring tripping, the benefit to channel testing is to better characterize the nature of a fault. This test connection is shown in Fig. 7e.

Since originating RSS customers make use of central office digit receivers, the voice channels must be capable of conveying dialed digits, which are transmitted on the channel as a series of supervisory transitions. The dial-pulse test performs this function by: (1) connecting the central office end of the channel to a digit receiver, and (2) requesting the RSS to transmit a digit "0" (ten dial pulses). The test passes if the digit receiver shows that ten dial pulses have been received. The associated connections are shown in Fig. 7f.

The final two tests on the channel form a gross verification of its two-way transmission capability. The ac far-to-near test, shown in Fig. 7g, connects a source of milliwatt level 1-kHz tone at the remote

end and a tone detector at the central office end. The complementary near-to-far test, shown in Fig. 7h, verifies transmission in the opposite direction and employs a similar technique. Since the tone detectors used in these tests are sensitive down to −30 dbm, the diagnostic results cannot confirm adherence to rigid (±1 dB) limits. The Remote Office Test Line (ROTL) feature has therefore been enhanced to perform accurate transmission measurements on voice channels as well as on trunks. This capability is discussed in the following section.

#### *3.1.2 Automatic channel transmission testing*

The ROTL feature, first implemented in the No. 2-EF-1 generic, was designed to allow automatic transmission testing of trunks by the Centralized Automatic Reporting on Trunks (CAROT) processor. Since the RSS channels have a functional role similar to that of local interoffice trunks, enhancement of the ROTL feature was selected as an efficient method for channel transmission testing. The resulting design, as implemented in the 2BE3 generic, proceeds along the following typical sequence in the testing of a channel:

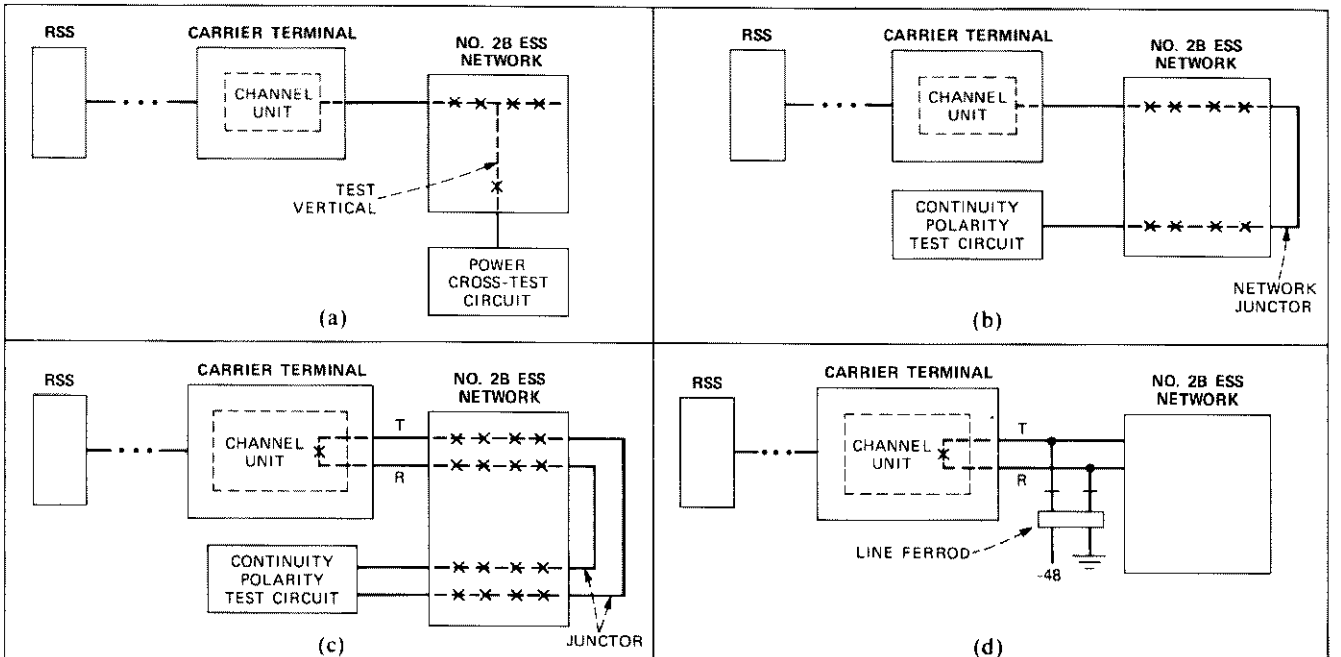
1. The CAROT center connects to the ROTL access (incoming) port via the standard Direct Distance Dialing (DDD) network.

2. Priming data specifying the test parameters is sent from the CAROT, through the ROTL access port, to a Multifrequency (MF) receiver. This digit receiver is accessed via a normal central office connection from the ROTL test port, as shown in Fig. 8.

3. Upon receipt of the channel identity, the central office releases connections to the MF receiver, establishes a connection between the ROTL test port and the selected channel, and requests the RSS to connect a transponder to the remote end of the selected channel.

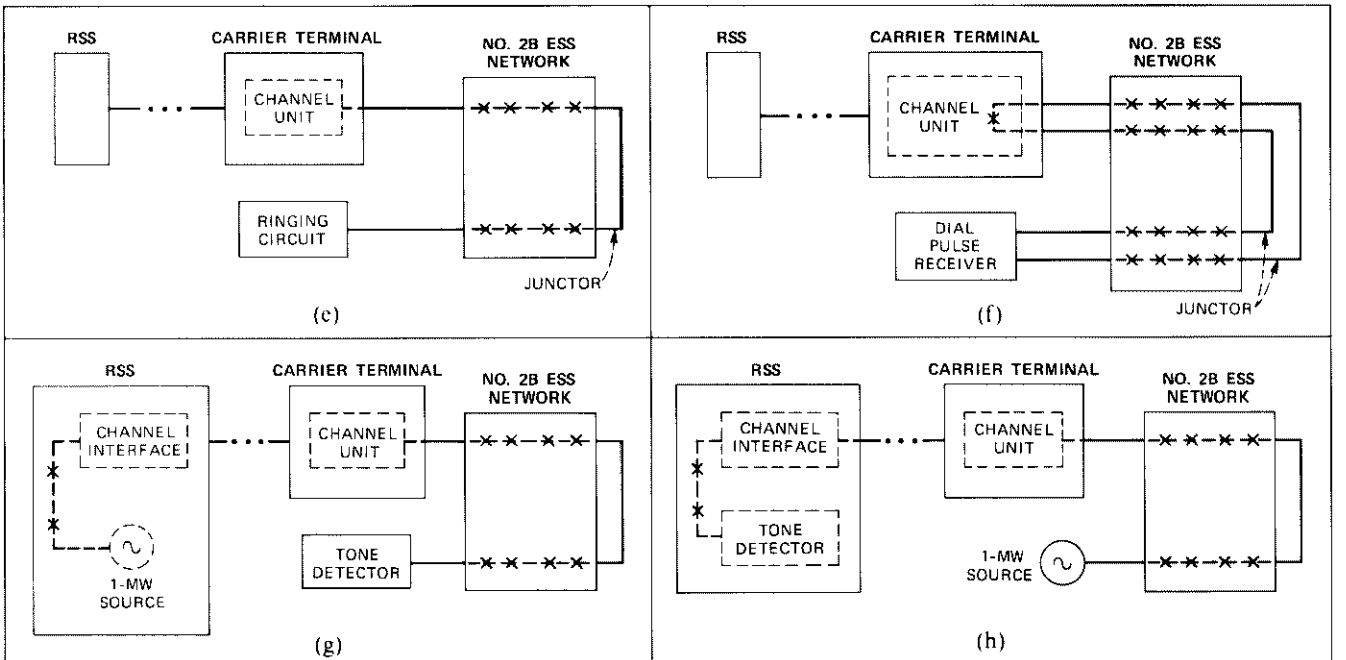
During this sequence of events, considerable data "handshaking" takes place between the CAROT and the No. 2B ESS—much of this communication in the form of bursts of "test progress tone" sent by the ROTL circuitry to the CAROT. The end result is a composite connection, as shown in Fig. 9, which permits the RSS transmission measuring transponder (termed a miniresponder) to interact with the CAROT in performing transmission measurements in either direction on the channel. The CAROT may then restore or remove channels from service based on the test results.

Since the CAROT-ROTL system can make loss and noise measurements accurate to 0.1 dB, this design provides a low-cost means of maintaining voice channel transmission characteristics within acceptable limits. Also, because the RSS miniresponder can be accessed in a similar fashion using a portable ROTL System Test Set, the same high-resolution measurements are available to the maintenance craft on a demand basis.




ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 7—Test configurations of channel diagnostic: (a) Power cross test. (b) False cross/ground test. (c) Supervision test. (d) Restore-verify test.




ESS – ELECTRONIC SWITCHING SYSTEM
RSS – REMOTE SWITCHING SYSTEM

Fig. 7 (Continued)—Test configurations of channel diagnostic: (e) Low-line resistance test. (f) Dial pulse test. (g) AC far-to-near test. (h) AC near-to-far test.

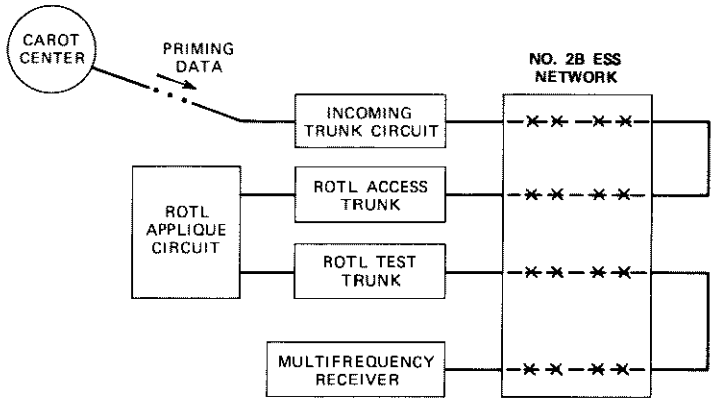



Fig. 8—Initial CAROT-ROTL connection.

### *3.1.3 In-process error detection*

While routine testing is an essential ingredient in the channel maintenance package, means must also be provided to deal with channel-related failures encountered during the processing of a telephone call. Such a facility will properly dispose of channels with transient faults (which are not detected during periodic exercise) and channels with newly occurring faults.

Once a channel is implicated by a failure in call processing, automatic routines dispose of the channel as follows:

1. A diagnostic test is run on the channel and, if the test fails, the channel is removed from service (subject to previously specified numerical limits).

2. If the diagnostic passes or cannot be run because of resource blockage, a report of the incident is made to error analysis programs running at the RSS.

3. The error analysis routines, which accept failure input from the remote terminal as well as the host, evaluate the failure history of each channel. This evaluation is carried out using two algorithms: a peer group comparison and a "quick check."

4. If the peer group analysis indicates that a given channel's error rate is significantly higher than that of its peers, that channel will be removed from service (subject to the specified maintenance limits).

5. If the error analysis indicates a "quick check" failure (three successive channel errors occurring on the same channel), channel diagnostics are run on the suspected channel. Depending on the

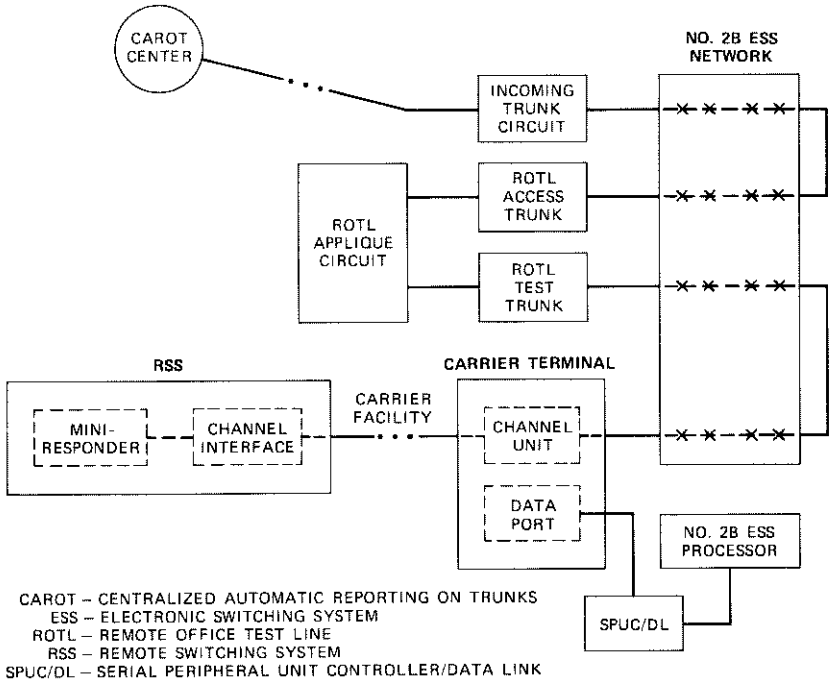



Fig. 9—Channel testing with CAROT and ROTL.

success or failure of the diagnostic exercise, the action taken is the same as that listed in (1) and (2) above.

### *3.1.4 Manually controlled channel testing*

As previously mentioned, the ROTL capability for channel (a) testing may be used automatically (via CAROT) or manually. In addition, to provide direct metallic access to the channel and control of its associated circuit states, the central office trunk test panel programs have been modified to provide test functions similar to those available for trunks.

A typical sequence, which permits a loop-around connection of two channels, is itemized below. Such a connection is used in two-way transmission loss measurements on channels.

1. Using the panel-mounted telephone set at the trunk test panel, digits are dialed that direct a specified channel (a) to be connected to a source of milliwatt tone (i.e., a 102-type test line) at the RSS. This connection, shown in Fig. 10, permits the TTP transmission measuring set to determine the loss of the channel in the far-to-near direction.

While this channel is accessed, existing keys at the TTP may be used to change the circuit state of the channel at the RSS. (These

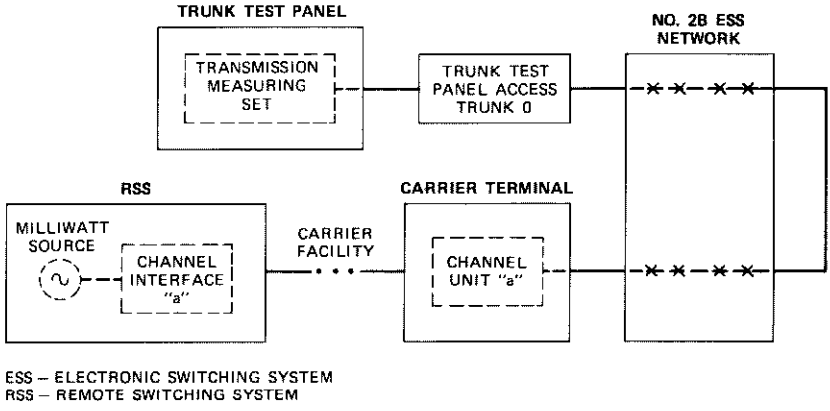



Fig. 10—Measurement of channel far-to-near loss.

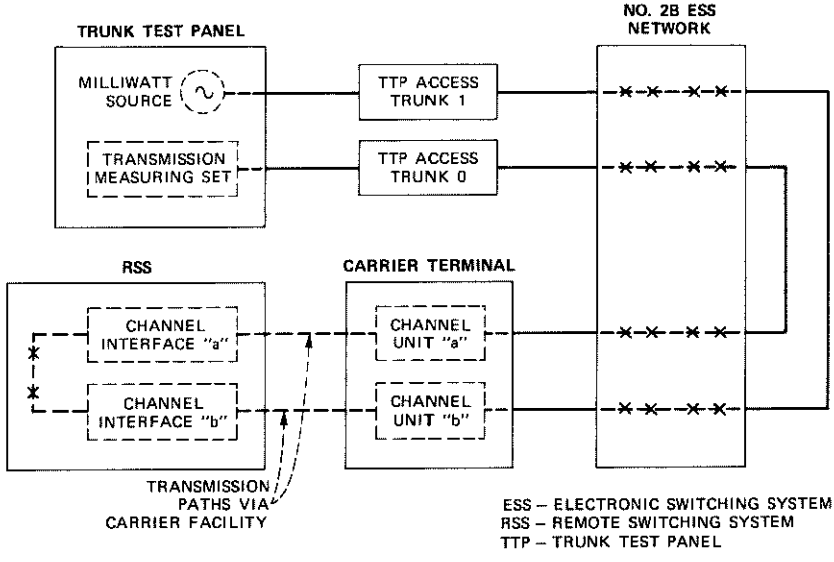



Fig. 11—Measurement of channel near-to-far loss.

states correspond to supervisory conditions, loss pads in and out of circuit, and the connecting of a balanced terminating network.)

2. Dialing a different test code and channel identity will cause a second channel (b) to be connected between another TTP access trunk and a loop-around connection at the RSS. This connection, shown in Fig. 11, has produced a configuration in which both ends of a two-channel transmission path are connected to the TTP. Now with a source of 0 dbM connected to the second channel and a transmission-

measuring set connected to the first, the near-to-far loss on the second channel may be determined.

3. When any channel is released from the TTP, several options are possible, depending upon the channel's previous status and the method of TTP disconnect. For example, if the channel had initially been out of service, TTP release will initiate a diagnostic test of the channel, restoring it to service if the diagnostic passes. If an initially idle (in-service) channel is released with the "make busy" key operated, the channel is unconditionally removed from service.

### *3.2 Line maintenance*

Like channel maintenance, line maintenance may be subdivided into capabilities that: (1) by periodic, automatically initiated testing, attempt to detect progressive hardware degradation before service is affected; and (2) by allowing demand-type human-controlled testing, enable the maintenance craft to resolve the nature and location of faults. Automatic Line Insulation Testing (ALIT) occupies the first of these categories and, like its counterpart for host-terminated lines, detects high-resistance crosses and grounds on metallic pairs between the RSS and the customer premises. The second category of features includes an interface to the Local Test Desk (located in a centralized repair service bureau) and the station ringer test, which verifies the correct operation of the subscriber's station set.

#### *3.2.1 Line insulation testing*

If it were not for the nonmetallic network in the RSS remote terminal, line insulation testing could be performed by the host No. 2B ESS. Instead, the remote terminal is equipped with its own insulation testing circuit and the means to make a metallic connection between this circuit and all customer line terminations.

As shown in Fig. 12, the RSS ALIT circuit accesses the customer line via the Metallic Access Bus (MAB), the Line Test Access Bus (LTAB), and the Universal Service Circuit (USC), the latter in the "bypass" state. RSS firmware has exclusive control of this connection and testing sequence once a line and test parameters are specified via data order. That is, after the RSS receives a line test request from the host, the remote terminal selects an idle USC, connects the MAB, performs the test, disconnects from the line, and returns the test results in an acknowledgment message over the data channel. The test parameters received in the requesting message are similar to those specified for host line testing and permit testing sensitivities ranging from 80 kΩ to 5 MΩ. In addition to the line identity, the requesting order also includes a specification of the test mode. The mode, an

operating company and craft option, selects the extent of the testing performed on an individual line. The choices are:

1. A foreign EMF (electromotive force) test, which looks for false power crosses to either conductor,
2. The TRG (tip-ring to ground) test, which measures resistance between the conductors (tied together) and ground,
3. Leakage test, which measures resistance between the conductors,
4. A "general" test, consisting of a sequential application of all the preceding tests.

RSS line insulation testing is included in the automatically initiated sequence and follows, RSS by RSS, the insulation testing of host lines. While the testing of RSS lines could have been done concurrently with the testing of host lines (since separate and dedicated hardware is involved), such a design would have increased the complexity of teletypewriter output messages, which identify line failures. By completing the testing of all lines in one entity (i.e., the host, or one of the RSSs) before proceeding to test lines in the next, a single "header" message serves to identify the entity for all line failures that follow. This has the added benefit of grouping those reports corresponding to their geographic locations.

#### *3.2.2 Local test desk*

Demand-type line testing of host and RSS lines may be performed from a test cabinet, located within the No. 2B ESS central office, and from a Local Test Desk (LTD) at some remote location. To accommodate the variation of metallic and carrier facilities between the LTD and the host and between the host and an RSS, a number of options have been provided. The simplest case, from a design point of view, exists where there is a low resistance (less than 2600 ohms) metallic path from the LTD to the RSS customer's telephone. As shown in Fig. 13, the connection from the LTD to the host uses the standard No. 2B ESS—LTD incoming trunk circuit and a dedicated metallic facility between the RSS and the host. This metallic connection is completed in the RSS by connection through the Metallic Line Access Port (MLAP), the USC, and the MAB. LTD testing, which consists of various voltage and resistance measurements, may then proceed over the established connection.

Since such low-resistance facilities are expensive and rarely found in the rural and suburban environment, an alternative implementation is provided. This arrangement takes advantage of the Remote Testing System (RTS), which was previously provided to work over nonmetallic facilities between the LTD and the central office. The resulting design employs a multifrequency telemetry Remote Line Test (RLT) unit at the RSS. As shown in Fig. 14, the microprocessor-controlled

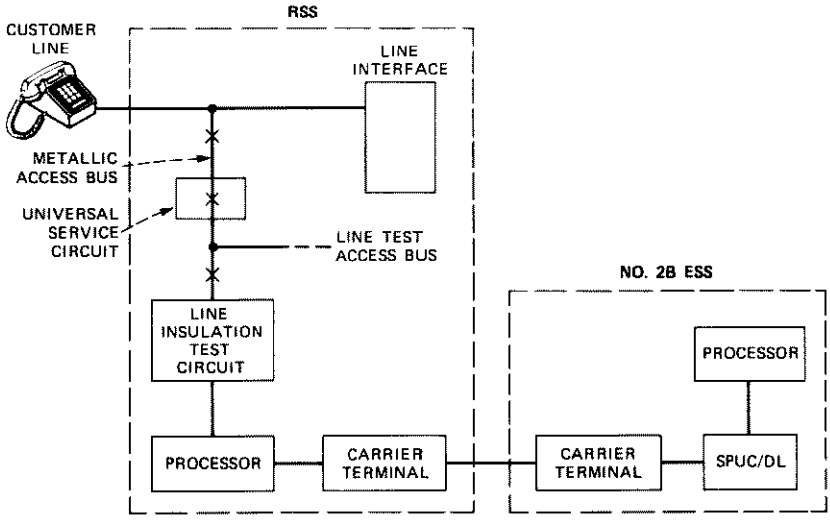



Fig. 12—RSS line insulation testing.

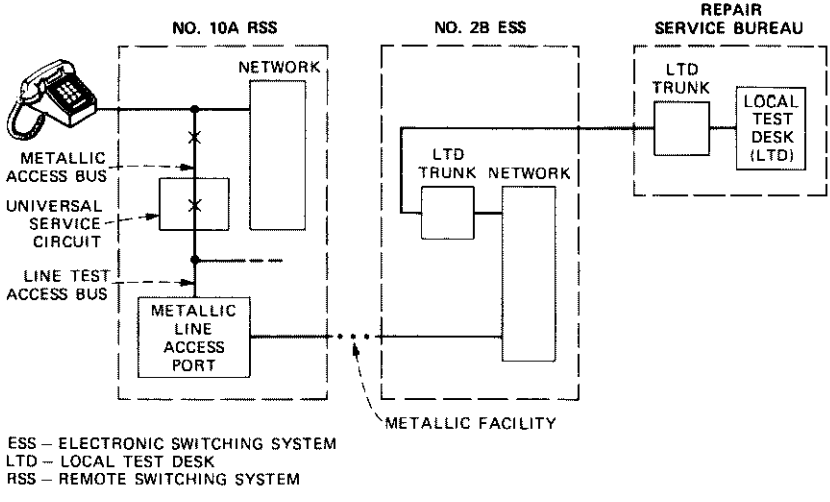



Fig. 13—Metallic option for the local test desk.

RLT has metallic access to the customer line via the MLAP, USC, and MAB. The connection of the LTD and RLT provides for test requests being sent to the RLT and for test results being returned to the LTD. Supervisory signals from the LTD (e.g., disconnect) are

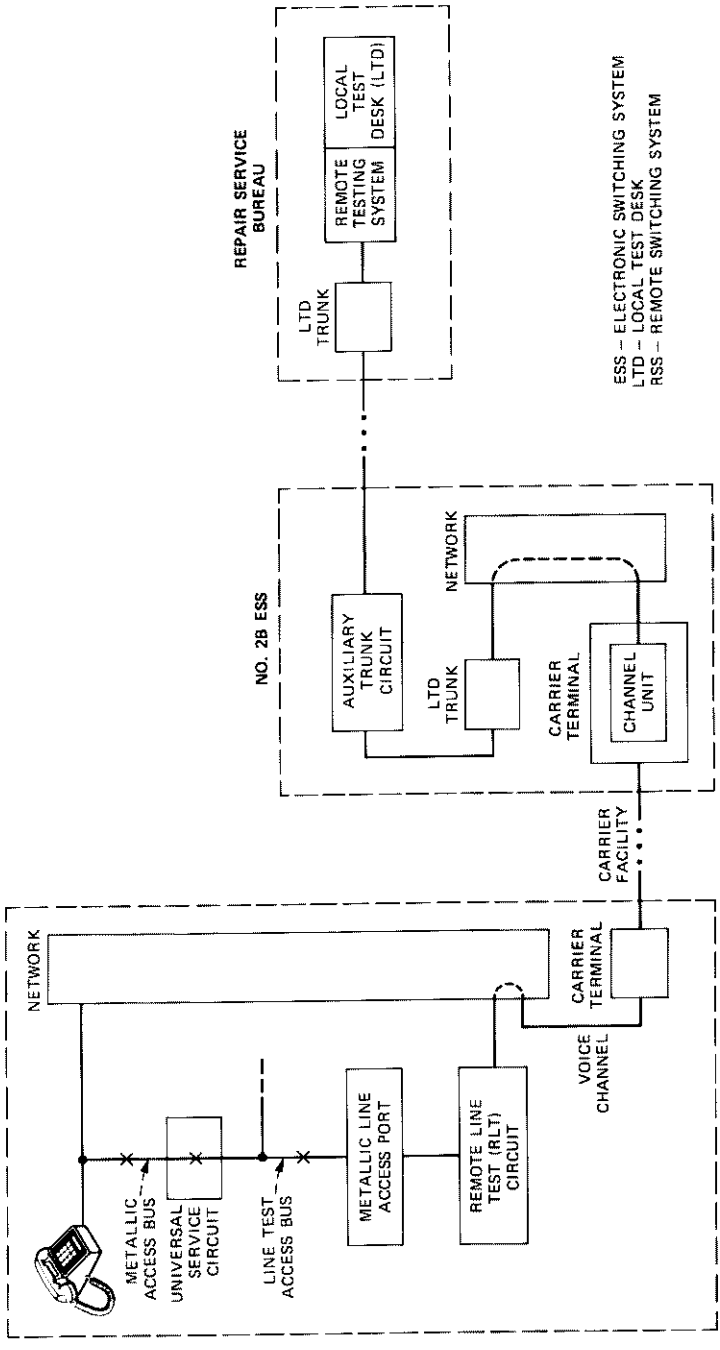



Fig. 14—RLT option for the local test desk.

intercepted by the auxiliary trunk circuit in the No. 2B ESS, since the host is responsible for maintaining and tearing down the entire connection. The RLT and the auxiliary trunk circuit at the host have therefore functionally replaced the central office equipment associated with the remote testing system.

### *3.2.3 Differences between host and RSS LTD line testing*

Since the two testing options (RLT or MLAP) require different trunk circuit hardware at the central office, the incoming call from the LTD may appear on either type of trunk. This requires the LTD operator to know the testing option associated with a particular RSS, and it requires the host No. 2B ESS to verify that the incoming LTD trunk type agrees with the testing option of that RSS. This latter confirmation is performed when the line's directory number is translated into an RSS identity.

The Local Test Cabinet (LTC), normally located near the distributing frame in the central office, is not equipped with the remote testing system. Thus, the LTC is incompatible with RLT-equipped RSSs and can test RSS lines only if that RSS is equipped with the MLAP option.

Another difference between LTD/LTC testing of host versus RSS lines concerns the action taken if the line is found busy when the initial test connection is made. When a host line is being tested, a suitably marked incoming LTD trunk can be connected to the busy line by means of the no-test facility of the No. 2B ESS switching network. If the host line becomes idle during the LTD connection interval, the line is marked busy (again) and the LTD is reconnected using a normal network path. In the analogous RSS case, the LTD is bridged onto the existing line connection in the RSS network. Since the software line status may not be changed, a camp-on request is registered at the RSS. When the line becomes idle, the pending LTD request causes the line to be rebusied, but no path reconfiguration is required.

One of the LTD functions is to verify whether a given line can originate. This is done on host lines by requesting the line attending element (ferrod) to be reconnected to the customer loop and then placing a resistive bridge across the loop. When the off-hook condition is sensed, dial tone is returned to the tester via the LTD trunk circuit, as shown in Fig. 15. Such operation is not possible with RLT testing of an RSS line since returning dial tone through the LTD trunk would open the LTD-RLT path and prevent further communication between them. Instead, when a line origination test is to be performed on an RSS line, that line is connected to a digit receiver in the host (shown in Fig. 16). The digit receiver then performs the functions of: (1)

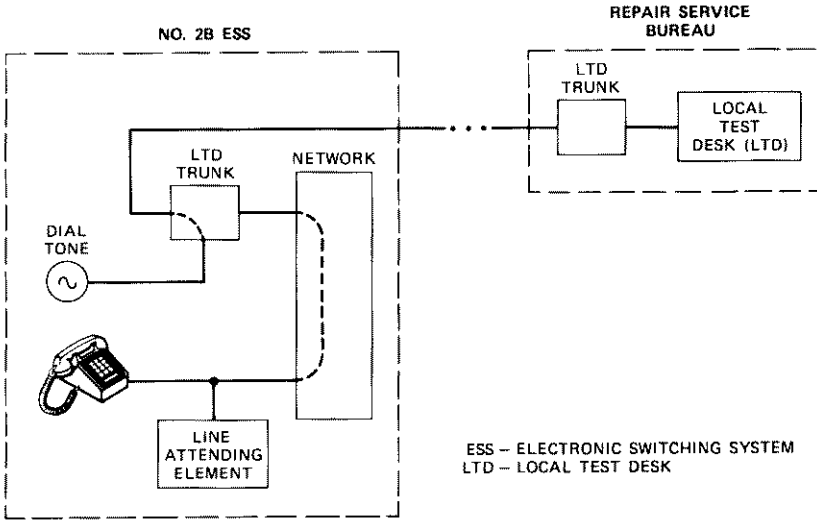



Fig. 15—Local test desk line origination test for host lines.

detecting supervisory changes on the line, and (2) returning dial tone when the off-hook signal is detected. Dial tone attachment and removal can then be monitored by the LTD, which is connected to the customer's line.

#### *3.2.4 Station ringer testing*

The station ringer test is under the control of the customer station and permits the verification of:

1. Station dialing capability
2. Party-identifying ground (in the off-hook state)
3. On-hook leakage, and
4. Ringing code and ring trip.

Since the sequence of actions required to perform these tests is well known to craft accustomed to testing host lines, a basic requirement for the service was that the craft interface remain unchanged for RSS lines.

The procedure, and its implementation for RSS lines, is diagrammed in Fig. 17. The test sequence begins by dialing a special code, normally a unique NNX, followed by the last four digits of the line's directory number. The RSS line is then connected to the host's station ringer test circuit via the same channel that had previously been connected to the customer digit receiver. The first step, the off-hook resistance test, is initiated with a switchhook flash by the tested line. A data-link order sent to the RSS causes the resistance test to be made by a

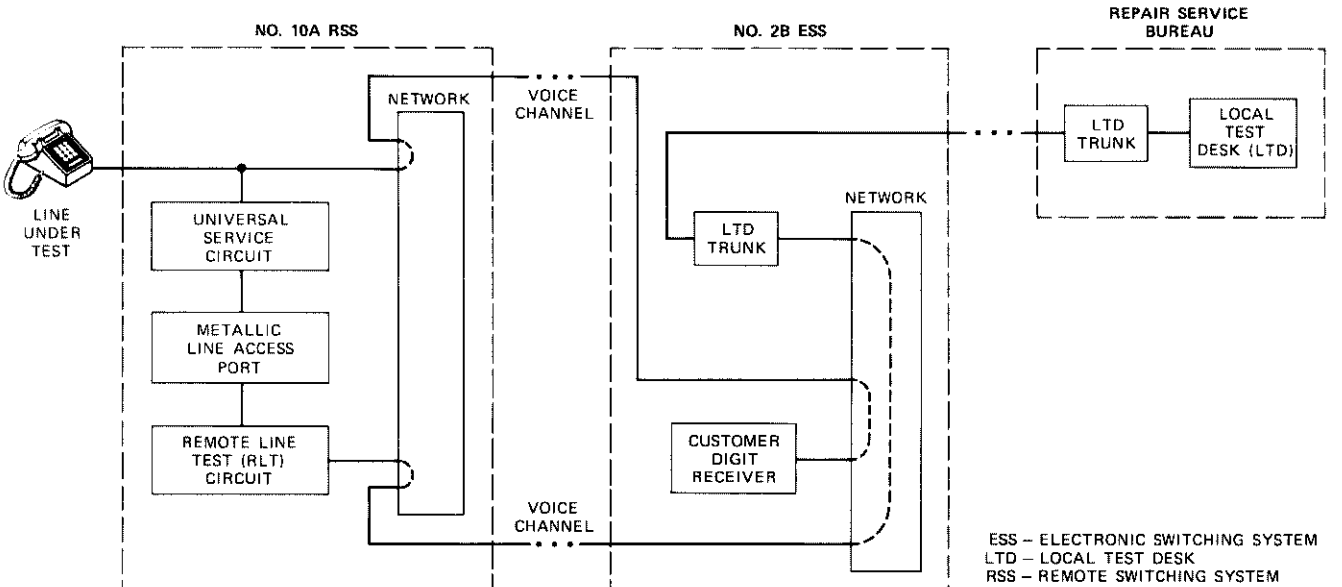



Fig. 16—Local test desk line origination test for RSS lines.

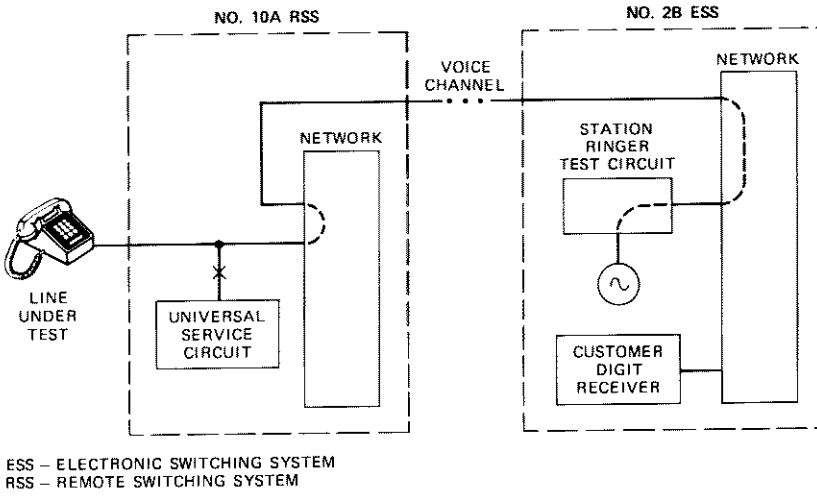



Fig. 17—Station ringer testing for RSS lines.

USC. The results of this test are then indicated with a steady vs. interrupted tone returned by the station ringer test circuit.

The second test on a (noncoin) customer line is an on-hook leakage test and is initiated by on-hook supervision. Again, the USC at the RSS is directed to perform the resistance test; its results are returned to the host via data-link message. If the resulting leakage measurement is acceptable, terminating ringing is applied to the line by the USC. Ringing indicates the result of the leakage test as well as confirming that the station set ringer is functional and that the correct ringing code is applied. The final test, ring trip, is made if the station set is taken off-hook within the next 3 minutes. As in the previous instances, since the ringing is applied by the RSS's USC, the tripping of ringing must be reported to the host via a data message.

At this point, the station ringer test circuit is connected to the line under test. A station set on-hook would idle all connected resources or, if a repeat test is desired, a switchhook flash causes the off-hook resistance test to be made, starting another test cycle.

### *3.3 DTMF receiver testing*

As we mentioned previously, the RSS is capable of "stand-alone" operation should its interface with the host be lost. This feature therefore required that DTMF receivers be equipped in the RSS so that customers utilizing *Touch-Tone** dialing may be served. To

* Trademark of AT&T.

achieve design economies in the RSS, a DTMF test circuit (like that in the host) was not provided. The alternative to manual routine testing was to provide an automatic diagnostic using the host's DTMF test circuit.

Such operation is possible because the stand-alone circuits (including the DTMF receivers) have access to the RSS voice channels via a link "multiple" arrangement shown in Fig. 18. This network design, originally intended for mutually exclusive connections of RSS lines to either channels or stand-alone circuits, permits a voice channel connection to a DTMF receiver without use of the RSS network junctors. The resulting configuration during the execution of the diagnostic involves the host's DTMF receiver test circuit, as shown in Fig. 19.

Maintenance software is furnished to diagnose one or more receivers on a demand or automatic basis. For the automatically initiated case, diagnostics of the RSS receivers are carried out one RSS at a time following the corresponding diagnostics for host digit receivers.

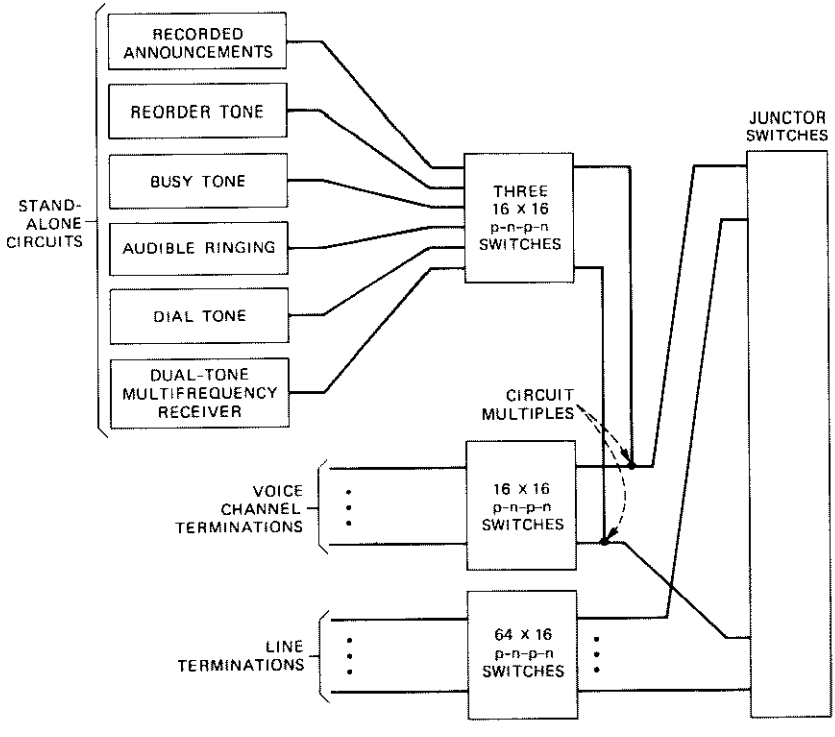



Fig. 18—Stand-alone circuit network connections.

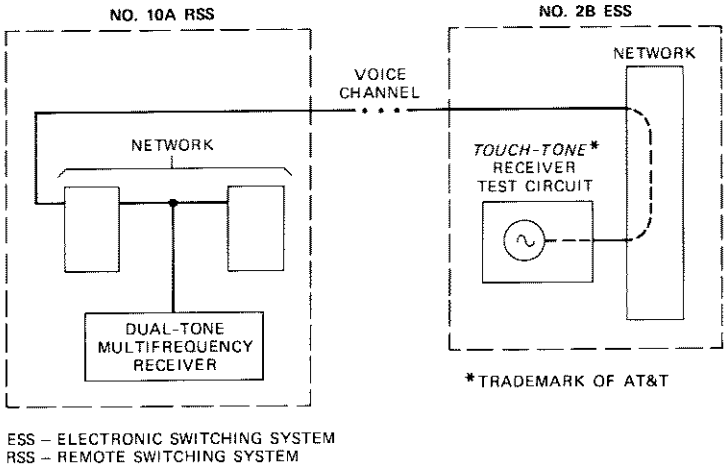



Fig. 19—*Touch-Tone* receiver diagnostic connections.

## IV. REAL-TIME PERFORMANCE

This section discusses two aspects of the RSS feature: the real time required to process an RSS call and the effect of RSS on the call capacity of an office.

The many components of an RSS call were determined by performing extensive measurements in the system laboratory. The results of this study indicated that an ESS line-to-RSS line call requires about 33 percent more processor real time than a host line-to-line call. Similarly, an RSS line-to-ESS line call requires about 49 percent more real time and an intra-RSS call (reswitched-down) requires about 113 percent more processor real time than a host line-to-line call.

Because of overhead associated with each RSS hosted by a No. 2B ESS and real-time factors associated with each RSS call, the No. 2B ESS call capacity is reduced with an increasing proportion of RSS traffic. The amount of this reduction is a function of the number of RSSs hosted and the RSS traffic.

When this project was still in the initial planning stages, it was evident that providing RSS capabilities would affect the host office capacity. Throughout the entire development of the feature, considerable emphasis was placed on performance issues. This included such items as: planning effective communication strategies between the host and SPUC, minimizing RSS-related work in non-RSS segments of code, and optimizing and fine tuning frequently executed functions. The result of these activities is a relatively small penalty in host capacity due to RSS.

## V. ACKNOWLEDGMENTS

The authors wish to acknowledge the outstanding efforts of the many individuals at Bell Telephone Laboratories, American Telephone and Telgraph, and Western Electric who have contributed to this work.

## AUTHORS

**Donald W. Brown,** B.S.E.E., 1962, University of Louisville; M.S.E.E., 1964, New York University; Bell Laboratories, 1962—. Mr. Brown has been engaged in the system design of proposed Signal Transfer Points for the Common Channel Interoffice Signaling network. Since 1976, he has worked on system and program design for the Remote Switching System. He was past Supervisor of the RSS Remote Terminal Maintenance Group and is now Supervisor of the Cell Maintenance Group for the Advanced Mobile Phone Service system.

**John J. Driscoll,** B.S.E.E., 1964, Clarkson College of Technology; M.S.E.E., 1966, Stevens Institute of Technology; Bell Laboratories, 1964—. Mr. Driscoll has been engaged in various aspects of electronic switching system development. Initially, he worked on maintenance and call-processing software development for No. 2 ESS, and later he worked on exploratory studies and call-processing software development for the Advanced Mobile Phone Service. He was involved with call-processing and administrative software development on the No. 1 ESS and the No. 2B ESS RSS systems. Presently, he is working on No. 5 ESS performance evaluation and capacity improvement.

**Frederick M. Lax,** B.S.E.E., 1974, University of Notre Dame; M.S.E.E., 1975, Massachusetts Institute of Technology; Bell Laboratories, 1974—. From 1975 to 1979, Mr. Lax was engaged in software development for No. 1/1A ESS call-processing and No. 10A Remote Switching System (RSS) feature development. In 1979, he was named Supervisor of the RSS First Application/General Availability Features Group and, subsequently, Supervisor of the No. 5 ESS Audit Systems Group. He is presently Head of the No. 5 ESS Software Recovery Department.

**Michael W. Saad,** B.S.E.E., 1964, M.S.E.E., 1965, University of Illinois; Bell Laboratories, 1966—. M. Saad first worked in the maintenance software area for the Automatic Voice Network (AUTOVON) system. His No. 4 ESS experience includes call-processing design, peripheral error-analysis strategies and design, peripheral-unit-configuration program design, and No. 4 ESS translations for international switching. Mr. Saad was supervising the 2B ESS RSS Call-Processing Group and is currently supervising the Administrative Services Group for No. 5 ESS.

**J. G. Whitemyer,** B.S.E.E., 1963, University of Akron; M.S.E.E., 1965, Ohio State University; Bell Laboratories, 1963—. Mr. Whitemyer joined Bell Laboratories in 1963 and worked in the Local Crossbar Switching Laboratory where he was involved in the development of Custom Calling Services and *Picturephone** Service on the No. 5 Crossbar System. He was promoted to

* Registered service mark of AT&T.

Supervisor in 1970 and, shortly thereafter, took responsibility for the system design of No. 3 ESS. Subsequent assignments included responsibility for circuit and microcode design of the 3A Central Control, field support for the No. 3 ESS and No. 2B ESS, and software design for the No. 2B ESS. He is presently Supervisor of the 2BE4 Project Management and Software Group. Mr. Whitemyer holds patents relating to the implementation of telephone switching features and telemetry.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

## *Modernization of the Suburban ESS:*

# Billing and Measurements Modernization

By J. P. LODWIG* and D. A. WARD*

(Manuscript received May 28, 1982)

With the introduction of the 2B Extended Feature Generic #3 (2BE3), No. 2B Electronic Switching System has modernized its arrangements for billing and traffic measurements. High-speed, synchronous data links are used to teleprocess billing information to the No. 1A Automatic Message Accounting Recording Center and to forward traffic data to the No. 1A Engineering and Administrative Data Acquisition System Center. This article describes the implementation of these features in the 2BE3 generic and how these new interfaces have resulted in more precise and more reliable data.

## I. INTRODUCTION

The billing and traffic measurements capabilities available in the No. 2B Electronic Switching System (ESS)† today are the culmination of an evolutionary process spanning more than a decade. During that period, the ESS environment was changing at a very rapid pace, and as a result the billing and traffic measurements processes were continually being upgraded to meet the demands for each generic. Repre-

* Bell Laboratories.

† Acronyms and abbreviations are listed at the back of this issue of the *Journal*.

©Copyright 1983, American Telephone, & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

senting the latest capabilities for billing and traffic measurements in the No. 2B ESS are the No. 1A Automatic Message Accounting Recording Center (AMARC) Interface feature and the No. 1A Engineering and Administrative Data Acquisition System (EADAS) feature, respectively. Both the AMARC and EADAS features require near-real-time data collection at the No. 2B ESS. The data are then passed to I/O message buffers for subsequent transmission to an AMARC or EADAS data center. The AMARC formats the billing information for a Revenue Accounting Office (RAO) to use in preparing customer bills, while EADAS provides real-time surveillance and disperses traffic data to various Operation Support Systems (OSSs) for use in maintaining the switching system and planning future growth. Data to the AMARC and EADAS systems are transmitted via synchronous, 2.4-kb data links using the BX.25 protocol. The AMARC feature also uses 4.8-kb data links for those offices presenting too large a load for the 2.4-kb link.

### *1.1 No. 1A AMARC*

Although several hardware and software versions of an AMARC exist today, the No. 2B ESS interfaces only with the No. 1A AMARC equipped with the 1AAM4 generic. The AMARC provides service to various types of switching systems today with several different interfaces giving flexible format, protocol, and data-link speeds between the switching systems offices and AMARC. The AMARC combines the data received from the switching systems into a format acceptable by an RAO. The data are then recorded on magnetic tape, which is sent to the RAO for processing into telephone bills for the customers. Figure 1 represents the interface of a No. 2B ESS with an AMARC.

### *1.2 EADAS*

An EADAS data center (with the 1AED4 generic) connected to the No. 2B ESS (with the 2BE3 generic) receives traffic data and plant data from the No. 2B ESS. The traffic data are load measurements such as peg and usage counts for office totals, and peg, usage, and overflow counts for trunk and service circuit groups. The plant data are measurements that indicate the health of the system. Examples are control unit and peripheral unit diagnostic all-tests-passes and faults. The traffic and plant data sent to EADAS and processed by the EADAS data center are then used by other associated OSSs. The Network Operations Report Generator (NORGEN), part of EADAS, accesses the processed EADAS data directly for its near-real-time reports. Downstream OSSs process EADAS data after they have been sent to them via magnetic tape. Two such downstream OSSs to access the data via the Traffic Data Administration System (TDAS) are

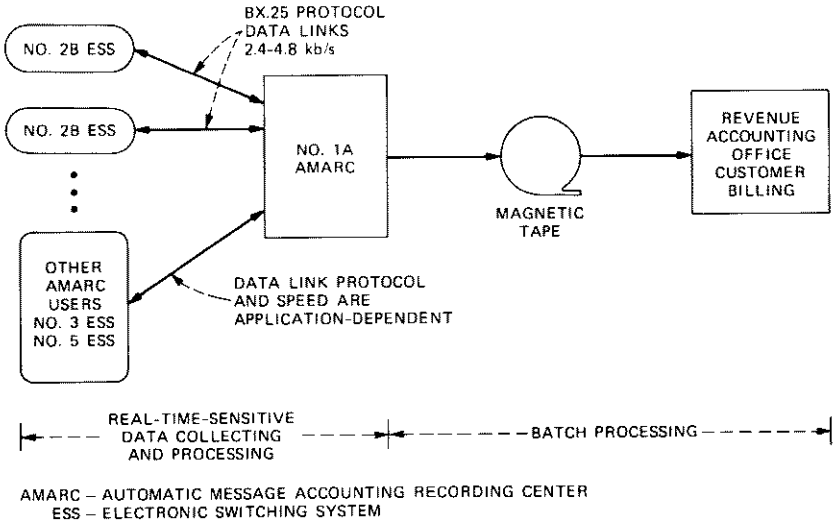


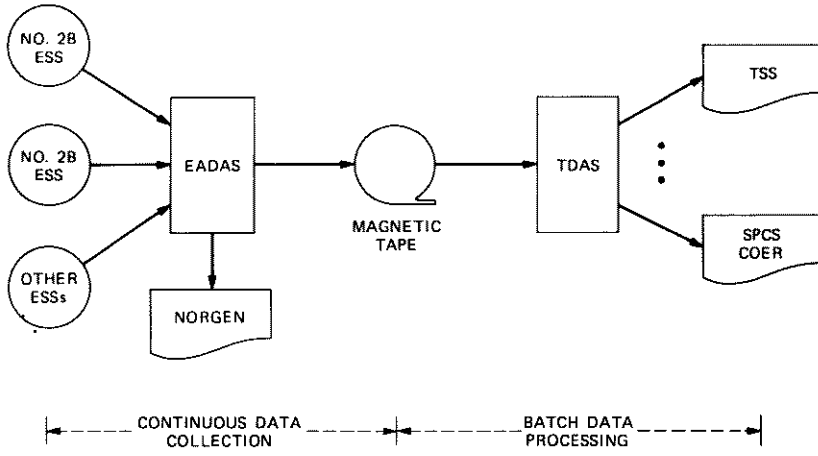

Fig. 1—Interface of a No. 2B ESS with an AMARC.

SPCS COER (Stored Program Control Systems Central Office Equipment Reports) and TSS (Trunk Servicing System). The reports generated by SPCS COER and TSS are needed by network administrators to monitor switching system services, measure utilization, and calculate capacity of the switches, and by trunk administrators to compute trunk group traffic load and current trunk requirements as part of their day-to-day jobs. The capability of providing timely data to SPCS COER and TSS is advantageous (see Fig. 2 for layout). The OSSs can now develop additional needed reports (e.g., summarizing centrex group counts). Also, with the addition of RSS host capability in 2BE3, the traffic data for an RSS is collected by the No. 2B ESS. Extreme Value Engineering (EVE) techniques are used for the engineering of an RSS. The EVE selection is done by EADAS and the engineering calculations are done by SPCS COER.

## II. No. 2B ESS HISTORY AND LIMITS

### *2.1 Billing data collection*

Prior to the installation of the 2BE3 generic, the primary method of billing calls was through Local Automatic Message Accounting (LAMA). Other methods used to a lesser degree were Centralized Automatic Message Accounting (CAMA) and in some instances message registers. The choice of billing arrangement was based on applicable tariffs (flat rate or local measured service) and resulting call volumes. Flat rate areas used message registers or CAMA while local measured service areas used LAMA.




COER – CENTRAL OFFICE EQUIPMENT REPORTS
EADAS – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM
NORGEN – NETWORK OPERATIONS REPORT GENERATOR
SPCS – STORED PROGRAM CONTROL SYSTEMS
TDAS – TRAFFIC DATA ADMINISTRATION SYSTEM
TSS – TRUNK SERVICING SYSTEM

Fig. 2—No. 2B ESS and Operation Support Systems.

The LAMA version of No. 2B ESS billing provides the most complete and versatile billing process. However, in some cases daily collection of LAMA tapes could be costly. One possible solution to this first problem was to enhance the software and upgrade the tape recorder to accommodate full measured service billing. However, the collection of the tapes on a daily basis still presents an unattractive cost factor.

### *2.2 Traffic data collection*

In a No. 2B ESS office, each teletypewriter (TTY) will serve as a primary output device for one function. A 110-baud data link could be connected from the traffic TTY controller circuit to an EADAS monitored interface. Once it is output to any device, the data no longer exist in the corresponding register, and subsequent output to any other device is not possible. In addition, a large office could not transmit all of the data to EADAS because of the limited data-link speed. Thus, only a subset of the total data was sent (i.e., one or two hours worth each day, assigning different data to the schedules). Another problem was the data skew associated with these output processes. The TTY print rate was 10 characters per second. At this rate it took a minimum of 5 minutes to print a traffic schedule of 600 registers. (Most No. 2B offices have more than 600 registers on a schedule and some require

15 minutes print time for even larger schedules.) The last line of data on a printed schedule is therefore skewed from the first line by the print time. To solve the need for all the data all the time and for accurate data (i.e., prevent data skew) holding registers are added to the 2BE3 generic.

## III. SOLUTION AND NEW PROBLEMS

The inclusion of AMARC and EADAS in the 2BE3 generic provides the increased billing capacity required by the switch and improved traffic information for engineering the switch. This does not preclude the use of billing arrangements available prior to 2BE3 (e.g., LAMA), where increased billing capacity is not required. Also, most of the traffic information capabilities available prior to 2BE3 have been retained. However, as with any new approach, new problems appeared for both AMARC and EADAS. In the early planning phase, one of the more basic problems was establishing interface requirements acceptable to Bell Laboratories systems engineering, AMARC development, EADAS development, and the No. 2B ESS development organization. The selection of a data-link protocol that could be used for both EADAS and AMARC was a highly desirable consideration. Also, during this same time frame, No. 5 ESS committed to AMARC and EADAS for its first application. This further complicated the situation in that now there was another system with a dissimilar architecture and software environment that could affect the development schedule and interface requirements.

## IV. NEW PROTOCOL

Selection of the communications protocol was a major consideration in establishing AMARC and EADAS requirements. A protocol represents a formal agreement on the exchange of information between two or more entities. It provides a multilayered set of rules that govern the interconnecting electrical signals (level one), packetized data (level two), complete message (level 3), and user application data transfer procedures (level 4 and above). Several protocols were considered, e.g., DDCMP, BYSYNC, X.25, etc., but only X.25 had the recommendation of the Comite Consultatif International Telephonique et Telegraphique (CCITT). Eventually a subset of X.25, now referred to as BX.25, was proposed and accepted as the AMARC feature protocol. Subsequent investigation of this proposal revealed that the proposed BX.25 protocol was also suitable for the EADAS feature. A more detailed presentation of this capability in the No. 2B ESS 2BE3 generic is provided in this issue of the *Journal* in an article entited "Adding Data Links to an Existing ESS."

## V. INTEGRATION OF AMARC

The AMARC feature encompassed many areas of call processing as well as data integrity and transmittal. Some areas were related to AMARC alone, while others were a consequence of a retrofit constraint that required simultaneous recording by AMARC and LAMA. Rather than attempting to cover the entire scope of AMARC, we will discuss only the basic operational capability and selected concerns.

### *5.1 Basic operation*

The AMARC feature collects billing data in real time in the No. 2B ESS and then transmits this data using either a dedicated 2.4- or 4.8-kb/s full-duplex transmission facility or an automatic-dialed backup data-link facility of the same transmission speed in case of primary link failure. The No. 2B ESS/AMARC interface uses a double-entry billing system consisting of an initial/answer entry and a disconnect entry. LAMA recording, however, uses a triple-entry billing system consisting of initial, answer, and disconnect entries. Data for each LAMA entry are collected in real time and loaded into a dedicated LAMA buffer prior to being directed to the tape recorder via I/O control at interrupt level. The use of special AMA registers allows the basic billing structure to remain as it had been for LAMA and still achieve the double-entry billing required by AMARC. Each AMA register is linked with a specific call by direct extension of the Transient Call Record (TCR) memory already designed into the No. 2B ESS architecture. These TCR extensions are then referred to as TCRX registers. The initial billing data are collected and placed into the TCRX registers until answer time and then, upon verification of Minimum Chargeable Duration (MCD), the data are moved into an AMARC data buffer. At disconnect, the data are again collected in the TCRX. When the entry is assembled, it is then loaded into the AMARC data buffer.

### *5.2 Entry association*

An obvious requirement for assembling the double entry into one call record by AMARC is the need to associate the initial/answer entry with the corresponding disconnect entry. The method provided by the No. 2B ESS is to map an equipment number (the physical location of a call on the No. 2B ESS or RSS network) into a 15-bit Virtual Equipment Number (VEN) and record the VEN as the low-order 16 bits of a 24-bit Call Assembly Index (CAI) that accompanies each billing entry. The CAI high 8 bits are zeroes except for the case of call-forwarded calls. The special use of the high 8 CAI bits for call-forwarded calls will be described later in this article.

The CAI counterpart in LAMA recording is the Call Identity Index

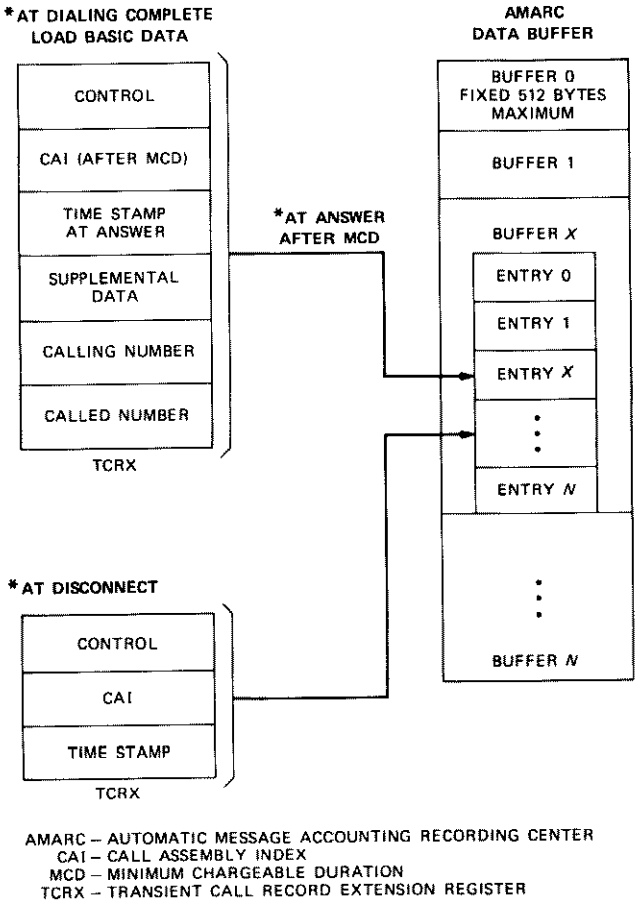



Fig. 3—Basic billing information flow between internal AMARC registers and buffers.

(CII). LAMA recording guaranteed that CIIs appearing on the tape would be unique for the duration of a call by maintaining a table of the active CIIs derived using a time-consuming hashing scheme based upon the VEN. Since the method of call sequencing associated with the CAI no longer requires the ESS to provide a hashing algorithm, significant real-time savings are achieved for the No. 2B ESS call-processing execution environment. Figure 3 represents the basic billing information flow between internal AMARC registers and buffers.

### *5.3 AMARC data buffer*

The No. 2B ESS AMARC data buffer is an engineerable area of read/write memory. The maximum size is designed to provide up to two minutes of billing data storage during a data-link failure. Auto-

matic recovery to the dialed backup data link can be obtained within this two-minute period. The buffer is segmented into message size entities of 512 bytes each. When a buffer segment has insufficient room for the next data entry, or a fixed time limit has expired, the data are either sent immediately or queued for later transmission through the I/O program. The decision depends on available buffer space on the Serial Peripheral Unit Controller for Data Links (SPUC/DLs). Further information on this subject can be found in the article "Adding Data Links to an Existing ESS," also in this issue of the *Journal.*

### *5.4 Real-time benefit*

The I/O process that moves the data buffer to a SPUC/DL is accomplished at a significant real-time savings compared with the I/O interrupt process for LAMA. Whereas LAMA had a data acceptance limitation (primarily related to hardware design) of three bytes of data within a 20-millisecond interval, the hardware structure of the SPUC/DL allows software definition of the maximum number of data bytes in a message. The I/O design in 2BE3 is flexible but currently allows 256 bytes of data to be sent to the SPUC/DL during one 100-millisecond cycle of the operating systems main loop. Thus, the repetitive I/O overhead associated with small segments of information pertaining to each LAMA recorded call is very significant compared to the AMARC overhead for sending 256 bytes of data for 15 to 20 calls. The overall real-time savings due to the AMARC billing feature, including I/O changes, was 2.8 milliseconds per call (3.6-ms AMARC versus 6.4-ms LAMA). This has the effect of increasing the overall call-billing capacity of the No. 2B ESS when full measured service is offered.

### *5.5 Auxiliary registers*

For some cases where the data required for a specific call type cannot be contained in a single TCRX register, auxiliary registers are provided to hold the overflow data and are referred to as auxiliary overflow registers. The auxiliary registers are unique to the AMARC feature, while the TCRX registers may be used for other features not related to call processing. The auxiliary registers are associated with the call by placing an index pointer into the TCRX used for the call. Figure 4 shows the connectivity of auxiliary overflow registers plus an extension into auxiliary call-forwarding registers.

### *5.6 Multiple calls associated with one CAI*

Each leg of a call-forwarded call requires separate billing to each of the parties involved with the call. Since separate billing is required,

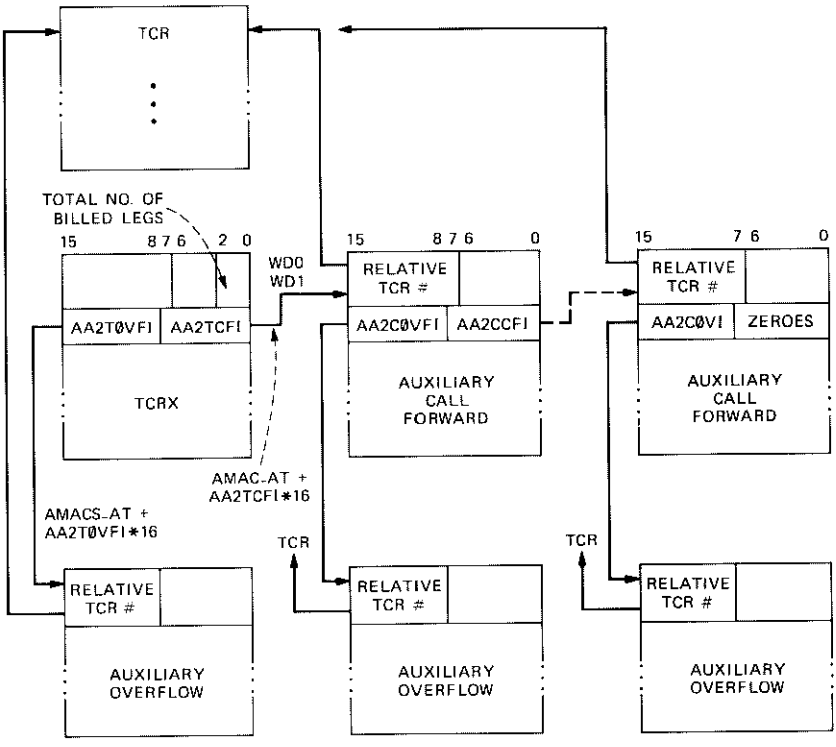



Fig. 4—AMARC message register interrelation.

the simplest method of administration is to provide a separate initial/answer entry followed by a corresponding disconnect entry for each leg of the call. This requires a different CAI for each billed leg and thus presented a problem in that the only VEN available at disconnect is the originating party VEN. This occurred since the No. 2B ESS call-processing structure prior to the 2BE3 generic did not provide a record of the intervening call legs once the call was made stable. The LAMA billing relied on the CII record table to provide the correct billing entry association. In the 2BE3 generic a record of the number of billed legs of the call is kept in a Stable Information Entry (SIE) during the stable state of the call. The SIE itself is accessed based upon the originating party VEN and codes defining the specific use for that SIE.

Correct billing entry association for AMARC call-forwarded billing records is achieved by prefixing an incremental count to the CAI at answer time for each call-forwarded billing entry except for the first

billed leg of the call. The CAI can then be expressed as the CAI = (N-1)*2**16 + VEN, where VEN represents the 16 low-order bits of the CAI and N equals the billed leg number (1–3) for call-forwarded calls. At disconnect, a billing entry is made for each billed leg of the call by once again prefixing a count to the CAI for each additional billed leg of a call exceeding the first leg. The correct number of disconnect entries is determined by the data available in the SIE.

The initial billing information for call forwarding is collected in the associated TCRX and in supplemental auxiliary registers as required. As in the case of auxiliary overflow registers, call-forward registers are associated with the call by loading an index point to the current register into the register servicing the previous leg of the call. An audit capability is provided by placing the TCR number in each register. Thus, random audits may test a register for activity and, by going to the specific TCR, test for the expected connectivity back to the auxiliary registers originally interrogated. This connectivity is illustrated in Fig. 4.

### *5.7 Retrofit considerations*

Several items of concern exist for the case of an AMARC retrofit into an office previously served by LAMA. One concern relating to the LAMA/AMARC simultaneous recording requirement during retrofits is the administration of call types, e.g., measured service, toll, etc. LAMA entry codes (equivalent to AMARC call types) relating to various calling conditions do not map directly with AMARC call types. This required careful preparation of administrative documentation to allow in some cases a temporary association of call-type categories for LAMA and AMARC. A significant consequence of the call-type association is the ability of the telephone operating companies to validate system operation by comparing the billing records of LAMA and AMARC during the retrofit mode of operation. Another consideration for the retrofit case is the recovery strategy. Since LAMA was the prime billing medium prior to retrofit, all calls billed during the retrofit are obtained from LAMA data tapes; thus all failure modes default to promoting successful LAMA billing.

## VI. INTEGRATION OF EADAS

The EADAS feature modified the traffic and plant-collecting and printing routines. With the BX.25 protocol as a common base, developing the interface requirements became a matter of resolving application-level interfaces between No. 2B ESS, No. 5 ESS, and No. 1A EADAS. There are interface differences between the No. 2B ESS to No. 1A EADAS and the No. 5 ESS to No. 1A EADAS due to the

different architecture structures of No. 2B ESS/generic 2BE3 and No. 5 ESS. Having the No. 2B ESS conform to the identical interface between No. 5 ESS and No. 1A EADAS would cause a real-time penalty to No. 2B ESS. The resulting operational interface between the No. 2B ESS and No. 1A EADAS is described next.

### *6.1 Basic operation*

The high-speed EADAS interface feature provides for the collection and transmittal of both traffic and plant measurement data to No. 1A EADAS. With this implementation the traffic and plant measurement data are collected on two schedules—a 30-minute schedule and a 24-hour schedule. The schedules start collecting 30 seconds before the clock 00 minute and 30 minutes and finish by one minute after 00 or 30. The data are collected and stored in holding registers within 90 seconds, resulting in a maximum of 90 seconds of data skew. The data for most offices will be collected within 30 seconds. The resulting data skew of 30 to 90 seconds is significantly more accurate than the previous skew of 5 to 15 minutes. Schedules are not transmitted until a poll is received from No. 1A EADAS requesting a schedule. The No. 2B ESS and No. 5 ESS communicate with the 1AED4 generic of No. 1A EADAS using the BX.25 protocol.

The interface between the No. 2B ESS and the No. 1A EADAS is a poll-and-answer mechanism. The No. 1A EADAS polls for four types of data:

1. Time of day
2. 30-minute data
3. 24-hour data
4. Record base (working member) data.

The No. 2B ESS answers with the:

1. Time of day
2. Data from the H (specified busy hour), C (specified nonbusy hour), W (specified weekly), and PLT (specified plant) schedules
3. D (daily or division of revenue) schedule
4. Working member count of trunk and service circuits for each poll.

The No. 1A EADAS also sends a message when it is planning to discontinue operation temporarily for such things as maintenance. The No. 2B ESS accepts the message and prints out a message to the TTY to inform the craft. The No. 2B ESS can send error response messages to No. 1A EADAS: (1) when the data for a message have been overwritten because current polling from the No. 1A EADAS is still continuing when the next collection is started, or (2) if a poll comes in when the 2B is collecting data for a message.

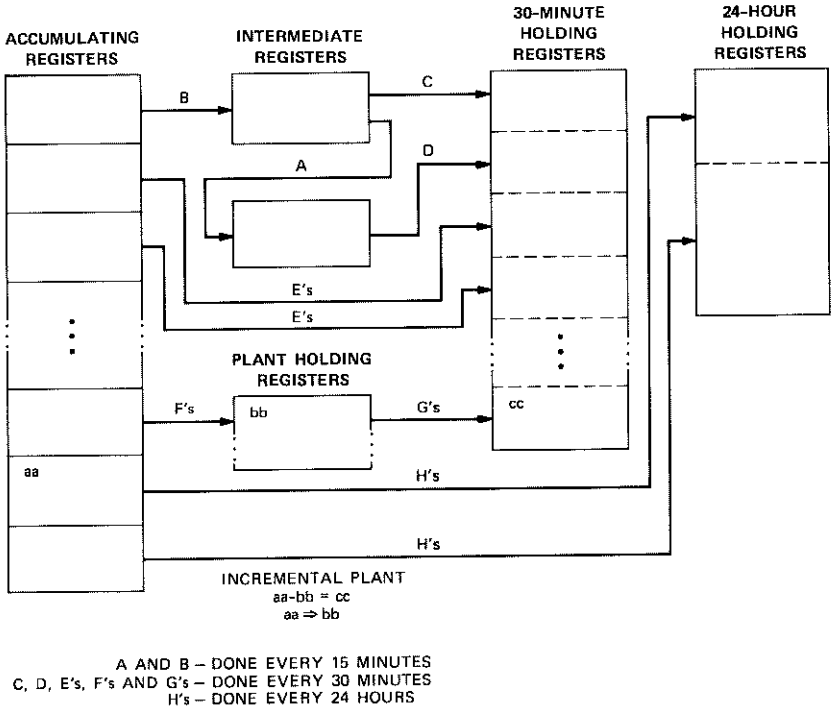



Fig. 5—Data collection.

### *6.2 Data collection*

Data collection is started every quarter hour to collect the intermediate counts and every half hour to collect the 30-minute counts. Also, the 24-hour counts are collected at midnight before the 30-minute counts. In Fig. 5 the lines A through H match the description of each type of count. The quarter-hour counts, load service measurements, are moved to the intermediate registers (A and B). The intermediate registers contain two sets of these registers, the previous quarter hour's and the current quarter hour's. The two sets of counts are kept until the half hour when both sets are moved to the 30-minute holding registers (C and D). Also, every half hour the counts such as office totals, service circuit, trunk circuit, multiline hunt, and simulated groups, junctor usage, centrex, and network usage are moved to the 30-minute holding registers (Es). Then the half-hour increment of the plant counts are moved to the 30-minute holding registers (Fs and Gs). This is done by subtracting from the current plant accumulating counts the intermediate plant holding registers, storing the difference in the 30-minute holding registers, and moving the accumulating plant

counts to the intermediate plant holding registers. The 24-hour counts, division of revenue, are moved to the 24-hour holding registers (Hs).

### *6.3 Override No. 1A EADAS*

If necessary, the 2BE3 generic can revert to the standard H, C, W, and D schedules through human intervention. When an office has EADAS active, the 30-minute data are from all the data that could have been on the H, C, and W schedules and the 24-hour data are from the D schedule. An office could revert to the standard schedule output if desired or if the No. 1A EADAS center would go off-line for an extended period of time. The No. 2B ESS office would then provide the data on hard copy and paper tape punch. In the No. 2B ESS, the standard schedules are printed automatically under the control of the Traffic Work Table (TWT) from the holding registers. This arrangement (non-EADAS) does not support RSS EVE engineering needs for SPCS COER. This mechanism is overridden when EADAS is active and it can be reenabled with a single TTY input message to inhibit EADAS.

### *6.4 Interface differences*

Because of the differences in No. 2B and No. 5 ESS architecture, the interface with No. 1A EADAS does differ on three points:

1. The No. 2B ESS keeps time differently than the No. 5 ESS.
2. The No. 2B ESS does not indicate overflow of registers through the special register values as does No. 5 ESS. The values 65,526 through 65,535 are used as special register values, i.e., for overflow, bad data, and unequipped register.
3. The 2B transmits two-byte data low byte first instead of the reverse.

Therefore, the No. 1A EADAS does handle these differences. For the No. 2B ESS to have followed the same interface as between No. 5 ESS and No. 1A EADAS on these three points would have cost the No. 2B ESS a real-time penalty.

## VII. SUMMARY

The addition of the AMARC and EADAS features to the 2BE3 generic enhanced the No. 2B ESS capability. AMARC achieves the billing systems goals of reduced operating expense while providing full measured service billing capability for the No. 2B ESS. Also, owing to a redistribution of tasks and restructuring of the programs, AMARC allows higher call capacity in the No. 2B ESS compared with LAMA billing.

All traffic and plant data are sent to EADAS, which allows for the efficient integration of the local switching systems into Total Network

Operation Plan (TNOP). Also, owing to the addition of holding registers there is virtually no data skew. The EADAS feature is using a communication interface that could support network management messages and controls in a future generic.

## VIII. ACKNOWLEDGMENTS

The authors extend their appreciation to all of the 2BE3 development team for their cooperation in providing the AMARC and EADAS capabilities for the No. 2B ESS. In particular, we wish to thank Iris Dowden and Eric Kampmeier for their contributions towards AMARC and Wendell Savino for his participation with EADAS.

## AUTHORS

**Deborah A. Ward,** A.A.S. (Computer Technology), Purdue University Calumet Campus; Bell Laboratories, 1973—. After joining Bell Labs in 1973, Mrs. Ward worked on No. 3 ESS and No. 2B ESS projects. During this time, she was involved in the translations design and the program design, coding and testing of the translations access routines for No. 3 ESS. She also was involved in maintaining the administrative traffic program for No. 3 ESS. Her work on No. 2B ESS has also been in the administrative traffic area. Mrs. Ward has been a member of the No. 5 ESS Data Base Design Department at Bell Laboratories in Naperville, Illinois, and since early 1982 has been working on the No. 5 ESS project.

**John Lodwig,** B.S.E.E., 1971, Illinois Institute of Technology; Bell Laboratories, 1967—. Mr. Lodwig's initial work at Bell Laboratories involved circuit design with the No. 101 ESS, and later the 3A Common Controller, which has been used as a central processor for several electronic switching systems, e.g., No. 2B ESS, No. 3 ESS, etc. In 1976, he made the transition from circuit design to software design and became involved with automatic message accounting billing. It was during the period from 1976 through 1980 that he became involved with the Automatic Message Accounting Recording Center (AMARC) billing feature for the No. 2B ESS. More recently, Mr. Lodwig has been working with common channel interoffice signaling and domestic local planning for the No. 5 ESS.

# ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| 2BE3 | 2B extended feature generic #3 |
| 3A CC | 3A Processor Control Complex |
| ALIT | automatic line insulation testing |
| AMA | automatic message accounting |
| AMACS | automatic message accounting collection system |
| AMARC | automatic message accounting recording center |
| APC | AMARC protocol converter |
| CAI | call assembly index |
| CAMA | centralized automatic message accounting |
| CAROT | centralized automatic reporting on trunks |
| CCITT | Comite Consultatif International Telephonique et Telegraphique |
| COER | central office equipment reports |
| CII | call identity index |
| CRC | cyclic redundancy check |
| DDD | direct distance dialing |
| DL | data link |
| DLI | data-link interface |
| DMA | direct-memory access |
| DN | directory number |
| DTMF | dual-tone multifrequency |
| EADAS | Engineering and Administrative Data Acquisition System |
| EMF | electromotive force |
| ESS | electronic switching system |
| EVE | extreme value engineering |
| FCG | false cross or ground |
| FIFO | first-in/first-out |
| I/O | input/output |
| LAMA | local automatic message accounting |
| LAP | link access procedure |
| LAPB | link access procedure B |
| LTAB | line test access bus |
| LTC | local test cabinet |
| LTD | local test desk |
| MAB | metallic access bus |
| MCD | minimum chargeable duration |
| MF | multifrequency |
| MLAP | metallic line access port |
| MSF | multi-scan function |
| NORGEN | network operations report generator |
| OSS | operation support system |

| | |
|---|---|
| P(R) | receive sequence number |
| P(S) | send sequence number |
| PMJ | path memory for junctor |
| PMR | path memory remote |
| POB | peripheral order buffer |
| PUC | peripheral unit controller |
| RAM | random access memory |
| RAO | revenue accounting office |
| RLT | remote line test |
| ROB | remote order buffer |
| ROH | receiver off-hook |
| ROM | read-only memory |
| ROTL | remote office test line |
| RSS | remote switching system |
| RTS | remote testing system |
| SCC | switching control center |
| SIE | stable information entry |
| SPCS | Stored Program Control Systems |
| SPUC/DL | serial peripheral unit controller/data link |
| TCR | transient call register |
| TDAS | Traffic Data Administration System |
| TNOP | Total Network Operation Plan |
| TRG | tip-ring to ground |
| TRL | transistor resistor logic |
| TSS | Trunk Servicing System |
| TTP | trunk test panel |
| TTY | teletypewriter |
| TWT | traffic work table |
| USART | universal synchronous-asynchronous receiver-transmitter |
| USC | universal service circuit |
| VEN | virtual equipment number |

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

David A. J., Meyer G. G. L., **Unstructured Mean Iterative Processes in Reflexive Banach-Spaces.** Siam J Con 21(1):140–152, 1983.

Gehani N. H., **High-Level Form Definition in Office Information Systems.** Computer J 26(1):52–59, 1983.

Odlyzko A. M., **Minima of Cosine Sums and Maxima of Polynomials on the Unit-Circle.** J Lond Math 26(Dec):412–420, 1982.

## ENGINEERING

Alferness R. C., Buhl L. L., **Long-Wavelength Ti:$LiNbO_3$ Waveguide Electrooptic TE↔TM Converter.** Electr Lett 19(2):40–41, 1983.

Aspnes D. E., Theeten J. B., **(Fr) Dielectric Semiconductor Interfaces Analysis Using Spectroscopic Ellipsometry.** Act Electr 24(3):217–227, 1982.

Bronez T. P., Cadzow J. A., **An Algebraic Approach to Superresolution Array-Processing.** IEEE Aer El 19(1):123–133, 1983.

Candy J. C., Wooley B. A., **Precise Biasing of Analog-to-Digital Converters by Means of Auto-Zero Feedback.** IEEE J Soli 17(6):1220–1225, 1982.

Cooper J. A., Capasso F., Thornber K. K., **Semiconductor Structures for Repeated Velocity Overshoot.** Elec Dev L 3(12):407–408, 1982.

Fichtner W., Watts R. K., Fraser D. B., Johnston R. L., Sze S. M., **0.15 μm Channel-Length MOSFETs Fabricated Using E-Beam Lithography.** Elec Dev L 3(12):412–414, 1982.

Forrest S. R., Kohl P. A., Panock R., Dewinter J. C., Nahory R. E., Yanowski E., **A Long-Wavelength, Annular $In_{0.53}Ga_{0.47}As$ P-I-N Photodetector.** Elec Dev L 3(12):415–417, 1982.

Liu S., Nagel L. W., **Small-Signal MOSFET Models for Analog Circuit-Design.** IEEE J Soli 17(6):983–998, 1982.

McCaughan L., Murphy E. J., **Influence of Temperature and Initial Titanium Dimensions on Fiber-TI-$LiNbO_3$ Wave-Guide Insertion Loss at Lambda = 1.3 μm.** IEEE J Q El 19(2):131–136, 1983.

Perry A., **The Two-Level, Single-Period Network Layout Optimization Problem and Its Solution.** Infor 20(4):336–356, 1982.

Tabatabaiealavi K., Choudhur A. N., Alavi K., Vlcek J., Slater N. J., Fonstad C. G., Cho A. Y., **Ion-Implanted $In_{0.53}Ga_{0.47}As/In_{0.52}Al_{0.48}As$ Lateral pnp Transistors.** Elec Dev L 3(12):379–381, 1982.

Tillman F. A., Kuo W., Hwang C. L., Grosh D. L., **Bayesian Reliability and Availability—A Review (Review or Bibliog.).** IEEE Reliab 31(4):362–372, 1982.

Vanderziel J. P., Logan R. A., **Dispersion of the Group-Velocity Refractive-Index in GaAs Double Heterostructure Lasers.** IEEE J Q El 19(2):164–169, 1983.

Vanderziel J. P., Temkin H., Logan R. A., **Quaternary 1.5 μm (InGaAsP/InP) Buried Crescent Lasers With Separate Optical Confinement.** Electr Lett 19(3):113–115, 1983.

## MANAGEMENT/ECONOMICS

Stiglitz J., Weiss A., **Alternative Approaches to Analyzing Markets With Asymmetric Information—Reply (Letter).** Am Econ Rev 73(1):246–249, 1983.

## PHYSICAL SCIENCES

Aspnes D. E., Studna A. A., **Dielectric Functions and Optical-Parameters of Si, Ge, GaP, GaAs, GaSb, InP, InAs, and InSb from 1.5 to 6.0 eV (Review or Bibliog.).** Phys Rev B 27(2):985–1009, 1983.

Baraff G. A., Schluter M., Allan G., **Cluster-Extended Greens-Function for Electronic-Structure of Localized Defects in Solids.** Phys Rev B 27(2):1010–1016, 1983.

Bokor J., Bucksbaum P. H., Auston D. H., **Measurement of Picosecond Ultraviolet-Laser Pulse-Widths Using an Electrical Auto-correlator.** Appl Phys L 42(4):342–344, 1983.

Bondybey V. E., English J. H., **Electronic-Structure and Vibrational Frequency of Cr-2.** Chem P Lett 94(5):443–447, 1983.

Boyle N. G., Mcbrierty V. J., Douglass D. C., **A Study of the Behavior of Water in Nafion Membranes.** Macromolec 16(1):75–80, 1983.

Cais R. E., Sloane N. J. A., **A Statistical-Theory of Directional Isomerism in Polymer-Chains and its Application to Polyvinylidene Fluoride.** Polymer 24(2):179–187, 1983.

Carroll P. J., Lannin J. S., **Vibrational Properties of Crystalline Group-Vi Solids—Te, Se, S.** Phys Rev B 27(2):1028–1036, 1983.

Chang R. P. H., Darack S., **Plasma Enhanced Beam Deposition of Thin Dielectric Films.** Appl Phys L 42(3):272–274, 1983.

Chin A. K., Zydzik G., Singh S., Vanuitert L. G., Minneci G., **$Al_2O_3$ as an Anti-Reflection Coating for InP/InGaAsP LEDs.** J Vac Sci B 1(1):72–73, 1983.

Cieplak M., Banavar J. R., **Sensitivity to Boundary-Conditions of Ising Spin-Glasses.** Phys Rev B 27(1):293–296, 1983.

Denbroeder F. J. A., Klerk M., Vandenberg J. M., Hamm R. A., **A Comparative-Study of Diffusion Induced Grain-Boundary Migration, Recrystallization and Volume Diffusion During the Low-Temperature Diffusion of Al Into Cu and Au.** Act Metall 31(2):285–291, 1983.

Downey P. M., Auston D. H., Smith P. R., **Picosecond Correlation-Measurements of Indium-Phosphide Photoconductors.** Appl Phys L 42(3):215–217, 1983.

Dubois L. H., Nuzzo R. G., **Small-Molecule Chemisorption of $NiSi_2$—Implications for Heterogeneous Catalysis.** J Am Chem S 105(3):365–369, 1983.

Dutt B. V., Brasen D., **TEM Observation of Precipitates of Cd-Phosphides in Cd-Diffused InP—A Correlation With the Previously Proposed Diffusion-Models.** J Elchem So 130(1):207–214, 1983.

Elhanany U., Brennert G. F., Warren W. W., **Enhanced Paramagnetism and Spin Fluctuations in Expanded Liquid Cesium.** Phys Rev L 50(7):540–544, 1983.

Flamm D. L., Donnelly V. M., Ibbotson D. E., **Basic Chemistry and Mechanisms of Plasma-Etching.** J Vac Sci B 1(1):23–30, 1983.

Fleury P. A., Lyons K. B., Katiyar R. S., **Acoustic Anomalies in $Tb_2(MoO_4)_3$ and the "Missing" $A_1$ Optic Mode.** Phys Rev B 26(12):6397–6407, 1982.

Fuerst C. O., Fischer J. E., Axe J. D., Hastings J. B., McWhan D. B., **Pressure-Induced Staging Transitions in $KC_8$: Observation of a Fractional Stage.** Phys Rev L 50(5):357–360, 1983.

Ginsberg A. P., Osborne J. H., Sprinkle C. R., **Electronic-Structure and Bonding in the Disulfur and Diselenium Complexes $[M(X_2)(PH_3)_4]^+$(M = Rh, Ir; X = S, Se)** Inorg Chem 22(2):254–266, 1983.

Hockberger P., Connor J. A., **Intracellular Calcium Measurements with Arsenazo-III During Cyclic-AMP Injections Into Molluscan Neurons.** Science 219(4586):869–871, 1983.

Huggins H. A., Gurvitch M. **Magnetron Sputtering System Equipped With a Versatile Substrate Table.** J Vac Sci A 1(1):77–80, 1983.

Hwang J. C. M., Brennan T. M., Cho A. Y., **Initial Results of a High Throughput MBE System for Device Fabrication.** J Elchem So 130(2):493–496, 1983.

Jackson C. M., Zettl A., Gruner G., Disalvo F. J., **Frequency-Dependent Conductivity in $HfTe_5$ and $ZrTe_5$.** Sol St Comm 45(3):247–249, 1983.

Kash K., Wolff P. A., Bonner W. A., **Non-Linear Optical Studies of Picosecond Relaxation-Times of Electrons in n-GaAs and n-GaSb.** Appl Phys L 42(2):173–175, 1983.

Kevan S. D., **Evidence for a New Broadening Mechanism in Angle-Resolved Photoemission from Cu(111).** Phys Rev L 50(7):526–529, 1983.

Klauder J. R., **Wiener Measures for Quantum-Mechanical Path-Integrals.** Lect N Phys 173:245–247, 1982.

Kuk Y., Feldman L. C., Silverman P. J., **Transition from the Pseudomorphic State to the Nonregistered State in Epitaxial-Growth of Au on Pd(111).** Phys Rev L 50(7):511–514, 1983.

Kurkjian C. R., Paek U. C., **Single-Valued Strength of Perfect Silica Fibers.** Appl Phys L 42(3):251–253, 1983.

Lee T. P., Burrus C. A., Linke R. A., Nelson R. J., **Short-Cavity Single-Frequency InGaAsP Buried-Heterostructure Lasers.** Electr Lett 19(3):82–83, 1983.

Levin R. M., **The Step Coverage of Undoped and Phosphorus-Doped $SiO_2$ Glass-Films.** J Vac Sci B 1(1):54–61, 1983.

Levine B. F., Logan R. A., Tsang W. T., Bethea C. G., Merritt F. R., **Optically Integrated Coherently Coupled $Al_x Ga_{1-x}$ As Lasers.** Appl Phys L 42(4):339–341, 1983.

Liminga R., Abrahams S. C., Glass A. M., Kvick A., **Temperature-Dependence of the Nuclear Positions and Spontaneous Polarization in Pyroelectric $Ba(NO_2)_2 \cdot H_2O$.** Phys Rev B 26(12):6896–6900, 1982.

Marshall J. H., **The Nickel Metal-Catalyzed Decomposition of Aqueous Hydrophosphite Solutions.** J Elchem So 130(2):369–372, 1983.

Menezes S., Miller B., Bachmann K. J., **Electrodissolution and Passivation Phenomena in III-V Semiconducting Compounds.** J Vac Sci B 1(1):48–53, 1983.

Menezes S., Miller B., **Surface and Redox Reactions at GaAs in Various Electrolytes.** J Elchem So 130(2):517–523, 1983.

OBryan H. M., Thomson J., **$Ba_2Ti_9O_2O$ Phase-Equilibria.** J Am Ceram 66(1):66–68, 1983.

Petroff P. M., **Device Degradation and Recombination Enhanced Defect Processes in III-V Semiconductors.** Sem Insul 5(3–4):307–319, 1983.

Phillips J. C., **Can Glass Flow—Reply (Letter).** Phys Today 36(2):95, 1983.

Piancastelli M. N., et al., **Pyridine Chemisorption on Silicon and Germanium(111) Surfaces.** Sol St Comm 45(3):219–221, 1983.

Rousseau D. L., Ondrias M. R., Lamar G. N., Kong S. B., Smith K. M., **Resonance Raman-Spectra of the Heme in Leghemoglobin—Evidence for the Absence of Ruffling and the Influence of the Vinyl Groups.** J Biol Chem 258(3):1740–1746, 1983.

Schwartz G. P., Gualtier G. J., **Evaluation of Aluminum-GaAs Schottky Barriers Using Norde Modified Current-Voltage Analysis.** Appl Phys L 42(3):265–267, 1983.

Seliskar C. J., Heaven M., Leugers M. A., **The Free Jet Spectrum of the Toluene 2668-A Origin Band.** J Mol Spect 97(1):186–193, 1983.

Sermage B., Eichler H. J., Heritage J. P., Nelson R. J., Dutta N. K., **Photo-Excited Carrier Lifetime and Auger Recombination in 1.3-$\mu$-M InGaAsP.** Appl Phys L 42(3):259–261, 1983.

Shank C. V., **Measurement of Ultrafast Phenomena in the Femtosecond Time Domain.** Science 219(4588):1027–1031, 1983.

Simpson A. M., Jericho M. H., Disalvo F. J., **Elasticity and Thermal-Expansion of 2H-$TaSe_2$ Between 4K and 130K** Sol St Comm 44(12):1543–1546, 1982.

Tam S., Hsu F. C., Ko P. K., Hu C., Muller R. S., **Hot-Electron Induced Excess Carriers in MOSFETs.** Elec Dev L 3(12):376–378, 1982.

Tersoff J., Bayer D., **Quantum Statistics for Distinguishable Particles.** Phys Rev L 50(8):553–554, 1983.

Vaidya S., Sinha A. K., Andrews J. M., **Electromigration Induced Shallow Junction Leakage with Al Poly-Si Metallization.** J Elchem So 130(2):496–501, 1983.

Williams J. H., et al., **Quantitative Geometric Characterization of Two-Dimensional Flaws via Liquid-Crystals Thermography.** Mater Eval 41(2):190+, 1983.

## SOCIAL AND LIFE SCIENCES

Desarbo W. S., **Gennclus—New Models for General Nonhierarchical Clustering Analysis.** Psychometri 47(4):449–475, 1982.

Fishburn P. C., **Nontransitive Measurable Utility.** J Math Psyc 26(1):31–67, 1982.

Foley M. A., Johnson M. K., Raye C. L., **Age-Related-Changes in Confusion Between Memories for Thoughts and Memories for Speech.** Child Dev 54(1):51–60, 1983.

Hanley M. J., Wilkening G. M., **Nonionizing Radiations—Current Issues and Controversies—A Minisymposium (Editorial).** J Occup Med 25(2):95, 1983.

Petersen R. C., **Bioeffects of Microwaves—A Review of Current Knowledge.** J Occup Med 25(2):103–110, 1983.

Rosenbaum D. A., et al., **Hierarchical Control of Rapid Movement Sequences.** J Exp Psy P 9(1):86–102, 1983.

Stack L. C., Lannon P. B., Miley A. D., **Accuracy of Clinicians Expectancies for Psychiatric Rehospitalization.** Am J Comm P 11(1):99–113, 1983.

Starr S. J., **A Study of Video Display Terminal Workers.** J Occup Med 25(2):95–98, 1983.

Weiss M. M., **The Video Display Terminals—Is There a Radiation Hazard.** J Occup Med 25(2):98–100, 1983.

## SPEECH/ACOUSTICS

Hall J. L., **A Procedure for Detecting Variability of Psychophysical Thresholds.** J Acoust So 73(2):663–667, 1983.

Pelech I., Zipfel G. G., Holford R. L., **A Wake-Scattering Experiment in Thermally Stratified Water.** J Acoust So 73(2):528–538, 1983.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

## *Human Factors and Behavioral Science:*

# Introduction

By E. E. SUMNER*

(Manuscript received March 15, 1983)

Human factors and its parent discipline, behavioral science, help us understand human capacities, interests, and needs. This understanding is especially important now, because modern technology makes unusual demands on its users. Knowledge and techniques derived from the behavioral sciences can help identify and design new products. This paper discusses the background and roles of the behavioral sciences, and it summarizes the contents of this issue.

This special issue of *The Bell System Technical Journal* presents a sample of recent activities of the behavioral science and human factors community at Bell Laboratories. Articles were drawn from several areas to give the reader an idea of the ways behavioral science and human factors contribute to the work at Bell Laboratories and, now, American Bell, a new subsidiary of AT&T.

This issue is intended for three groups of readers. First, behavioral scientists should gain a perspective of how their discipline is used in our Company. Second, members of the engineering and technical community who read this journal will see how behavioral science and human factors can contribute to the best use of technical innovation. The third audience this volume serves is the growing population of students who might find work in the field to be as interesting and challenging as we have.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

## I. HUMAN USES FOR NEW TECHNOLOGY

We are now in a qualitatively new era of invention. Throughout history, imaginative people have known what they would like to do, but they have not had the means to do it. They wanted to fly, to communicate with each other from afar, to be entertained easily, and to have knowledge at their fingertips. Until now, reality has lagged behind imagination. Today, inventions such as the transistor, the laser, and large-scale circuit integration have reduced the gap between what we can imagine and what we can do. The future seems unlimited; but we face a new problem—how to make wise choices among unlimited opportunities.

Choices between alternative designs of a product have always reflected trade-offs among technical feasibility, cost, and utility to the user. But new technology has increased engineering flexibility and lowered the cost of components until the balance among these trade-offs has shifted dramatically toward the needs and desires of the user. Ways of designing a product to best fit users' interests can now be addressed as never before.

Word processing is a vivid example of this shift. When Christopher Sholes and his associates designed the typewriter a little over 100 years ago, even a slow operator would tie up the keys. Since faster machine action wasn't possible, Sholes' answer was to spread the high-use keys far apart in the key basket, which in turn forced a keyboard arrangement that is less than optimal for the user.[1] In contrast, it is not surprising to anyone now that a computer-based word processing system can keep up with a dextrous typist entering text from a keyboard. The machine can even operate on the text between key strokes. Such processing systems can instruct new users, provide page- or line-oriented editors, and format output according to users' needs. Computer-based text handling can thus be "user friendly"; it can allow the novice typist to substitute rudimentary knowledge of a set of computer programs for the psychomotor skills of an advanced typist. The balance is seen to have shifted even further toward the user when we consider the computer language analysis techniques described later in this issue (see the articles by Frase,[2] Macdonald,[3] and Gingrich[4]).

As human issues become more central to design, pressure increases to provide a technology of invention, design, and evaluation for human use. This technology is the focus of human factors or human performance engineering. It is an application of behavioral science concerned with the design of machines and procedures for human use. Bell Laboratories established the first industrial human factors laboratory in the late 1940s under the direction of John E. Karlin. Examples of early work include all-number-dialing studies, design of better dial layout, and development of objective preference methods. (Hanson[5]

provides a history of human factors work at Bell Laboratories in this issue.) The need for this work has increased, and participation of human factors specialists has grown explosively in recent years. For example, the number of people with primary work assignments in human factors or behavioral science has tripled in the last decade, from a starting point of about 100 workers. Many of these people recently joined American Bell, a new subsidiary of AT&T, which designs and markets enhanced services and equipment such as telephones and PBXs. Human factors specialists are now involved in virtually every Bell Laboratories and American Bell development having an important human performance component.

Even so, the potential for the applied behavioral sciences has only begun to be realized, partially because technology has just begun to challenge our ability to imagine new applications, and partially because the contribution of behavioral and related sciences has been difficult to foresee. This volume shows, by example, the contribution of behavioral science to an understanding of human perceptual, cognitive, and motor behavior, and how human factors applies this understanding to the design of new products.

Today, we are not only faced with choices between alternative designs, we must also decide what *should* be designed. This is the hardest question. Currently, we have technologies searching for applications, and new computer applications increasingly resemble intelligent, almost human, actions. Hence, an active role for behavioral science in planning these applications is crucial.

## II. BEHAVIORAL SCIENCE RESOURCES

Most of the people doing applied and basic behavioral science research at Bell Laboratories and American Bell have been trained as experimental psychologists. Not many have degrees in human factors, because until recently there have been few programs in the field. In the past fifteen years, psychological research on learning and perception has broadened to include cognitive psychology, which studies thinking, language, and other aspects of information processing. Psychologists are well acquainted with sensory and central nervous system limits on human information handling, through studies of thought processes, linguistic and problem-solving skills, learning and forgetting, social communication, perceptual motor skills, and strength and endurance. Psychologists are also trained in experimental design, statistics, and the measurement of human performance, attitudes, preferences, and motivation.

Behavioral science work in Bell Laboratories has been differentiated mainly by its intent and the clientele that uses it. Basic research has been directed more at understanding the processes underlying behav-

ior, while applied research, emanating from human factors groups, has been associated with designing specific products to meet the needs, interests, and performance capabilities of its users. If research and applications specialists were to collaborate on a study, the measures taken of a person's performance—for instance, reaction time to red or green lights—would be of equal interest to both, but would have been collected for different reasons. For research, we would interpret these measures as a reflection of some underlying process, while for applications we would see the behavior as a fact to be reckoned with in product design. For example, Pierce and Karlin[6] found that reading rate is independent of the number of alternative words on a list of fixed length. The researcher might interpret this result as a reflection of channel capacity of humans as information processors. The applications designer might find the invariance of reading rates important for determining transmission requirements for a speech communication system.

In basic research, we attempt to structure work to provide generality beyond the experimental conditions under which data were collected. Researchers routinely publish work, which is read by other researchers, academics, students, and their colleagues. Applied research, on the other hand, often requires team cooperation, and people are responsible to others for providing information needed to complete a design. Although information gathered is often private and is generally collected to answer questions about the product being designed, data collected in one application often find good use in other applications (see McCormick[7] and Woodson[8]). Applied work has sometimes been seen by some as just the use, refinement, and dissemination of existing methods, rather than the invention and development of tools that generalize across tasks. This is wrong. Many behavioral concepts and techniques, important to theory as well as application, have arisen from practical problems. Aptitude testing and the measurement of human channel capacity are only two examples.

The role of basic research in Bell Laboratories in the study of psychological processes is well known to the outside world through research published over the last 20 years. Hanson[5] briefly summarizes highlights of that work in this volume. An example of applied work completes this section.

The human factors specialist contributes to product design at each stage in product development. One trade-off of particular concern is the extent to which the operation of a product can be made self-evident and require only minimal user training. Training can represent a substantial investment for a new electronic telephone system being installed in a large business having hundreds or thousands of users. The number and range of features available to each user mean that even a well-designed system will have to be carefully explained.

Dooling and Klemmer[9] have described work conducted on training procedures for business telephone users. The problem was that, although customers were happy with their systems, they wanted more training. But telephone companies found that two-hour training sessions, run with small groups of users, were too expensive. The results of several studies led to a more efficient training package that required only one hour of training time. The highlights of these studies are as follows.

Originally, participants in a session were trained in the detailed operation of as many as a dozen features in the two-hour session. More information was being presented than many participants could absorb. Testing showed that a card summarizing the operation of the features was a powerful performance aid. Consequently, training people to use the card replaced much of the detailed explanation and contributed to a reduction in training time. Learning to use telephone features is a cognitive task with only a minor motor component. If it is not a difficult manual task is it necessary to give hands-on experience? A set of laboratory and field studies showed that working the features on the telephone in the session did not significantly aid user performance. Here was a way to further shorten the training session, by eliminating unnecessary activities. Finally, self-instruction was found to be as effective as the group sessions both in laboratory experiments and in field studies. Thus, it was not crucial for all users to attend the training sessions.

Human factors specialists contributed here by proposing alternative training schemes based on what they knew about training and the ways people learn efficiently. They then devised studies and gathered data on the efficiency of those alternatives. Providing recommendations from theory alone would not have been enough to convince either the human factors people or telephone company management to use new procedures.

The skills of behavioral science personnel are critical now. Throughout the computer industry, much is said about "user-friendly" systems, yet in few places are the tools of behavioral science well understood, much less effectively applied. Computers are no longer just science and engineering tools; they are used every day at work, at school, and at home. Computers are for everyone, and for the Bell System this means finding new ways to relate technology to human needs and abilities.

## III. STRUCTURE AND THEMES OF THIS SPECIAL ISSUE

This issue is divided into five sections. The "History and Methods" section describes the growth of human factors work at Bell Laboratories and techniques that have been used to study human response

to communications products. Hanson's paper reviews human factors history, while Eigen's paper describes field test methods. Many techniques are used by behavioral scientists, of which field testing is only one. However, Eigen's paper gives an overview of one traditional use for human factors work—i.e., to evaluate products created by communications engineers.

The second section, "Characteristics of Human Performance," describes fundamental properties of human perception and performance. Julesz' and Bergen's paper describes mechanisms of visual perception, concentrating on the perception of texture. The paper by Rosenbaum explains how people control the timing of finger movements, and this may, in the future, help us understand how better to design keyboards and other control devices and tools. The final two papers in this section address higher-level cognitive processes. The paper by Cleveland, Harris, and McGill describes studies of how people derive information from graphic displays, while the paper by Egan explores how people reason when solving verbal problems. In particular, Egan shows that people can be trained to think in new ways, using strategies they normally do not use. These papers provide context for understanding possible limits and potentials for human adaptation to new technology.

The third section, "New Technological Demands," includes papers that deal with the unusual requirements that modern technology places on us. The paper by Donegan and Koppes introduces the three papers that follow it; their introduction emphasizes the wide variety of skills and characteristics that are needed by the craft personnel who install, maintain, and repair telecommunications equipment. The paper by Kohl describes work on the design of shapes to improve handling physical objects, the paper by Paul describes improved designs for splicing connectors, and the final paper in this trilogy, by Flamm, describes connecting-block design changes that reduce errors often made with complex apparatus. Following these papers, the article by Cohen moves on to the design of keyboards and people's ability to use new technology, such as membrane keyboards. Thus, papers in this section progress from simple to more complex performance.

The fourth section, "Interface Design," expands the theme of verbal interactions demanded by new technology. The paper by Furnas, Landauer, Gomez, and Dumais explores alternate ways to design key words to ease people's interactions with computers. The paper by Streeter, Ackroff, and Taylor extends this concern to the design of abbreviations for command names. The theme of human interactions involving verbal information is carried forward in the paper by Karhan, Riley, and Schoeffler, as they describe the design of telephone cards to provide easy access to dialing information people need when phoning. This theme is extended in the paper by Coke and Koether, which

explores the match between people's ability to read and the reading demands placed on them by the language of telecommunications documents. The final paper, by Holmgren, shows that technology, too, has limits, and that proper adaptation of humans to the limited speech of machines can make it possible for machines to understand what humans say.

The fifth and final section of this issue, "New Functions for Technology," is an example of bringing behavioral science to bear on the design of new products. The papers in this section, by Frase, Macdonald, and Gingrich, describe the *UNIX* Writer's Workbench software, a set of programs that do many things editors do when they proofread and comment on various features of written material. This section exemplifies the issue raised at the start of this introduction; we have the technology, we understand the physical principles needed to design new electronic tools; we have the engineering skills to build those tools, and we have the computing techniques to program them. But we need a behavioral perspective to know where human minds falter and need help, and where they excel and are best left alone. Human factors—behavioral science, generally—is an essential resource in this new era of communication design.

## IV. ACKNOWLEDGMENTS

This special issue was created through the efforts of many people. Paul A. Turner proposed this special issue to the Editorial Committee of *The Bell System Technical Journal*, with the encouragement of Thomas H. Crowley, a member of that committee. V. G. Stetter, followed by Eric Wolman, served as administrative coordinator for the issue. A working-level group, chaired by Barry L. Lively (now of American Bell) and including Lawrence T. Frase, Thomas K. Landauer, and Christine A. Riley (American Bell) solicited, edited, and organized the papers in the volume. Special recognition should be given to Lawrence T. Frase and Barry L. Lively for their contributions. A panel of more than 30 behavioral scientists and human factors specialists within Bell Laboratories and American Bell read and commented on the papers. The ground rules for that procedure discourage acknowledgment of those reviewers by name, but the reader should know that their contributions were vital to the issue.

## REFERENCES

1. C. Leckberg, "The Tyranny of Qwerty," Saturday Review, *55*, No. 40 (September 1972), pp. 37–40.
2. L. T. Frase, "*UNIX*™ Writer's Workbench Software: Philosophy," B.S.T.J., this issue.
3. N. H. Macdonald, "The *UNIX*™ Writer's Workbench Software: Rationale and Design," B.S.T.J., this issue.

4. P. S. Gingrich, "The *UNIX*™ Writer's Workbench Software: Results of a Field Study," B.S.T.J., this issue.
5. B. L. Hanson, "A Brief History of Applied Behavioral Science at Bell Laboratories," B.S.T.J., this issue.
6. J. R. Pierce and J. E. Karlin, "Reading Rates and the Information Rate of a Human Channel," B.S.T.J., *36* (March 1957), pp. 497–516.
7. E. J. McCormick, *Human Factors in Engineering and Design*, 4th ed., New York: McGraw-Hill, 1976.
8. W. E. Woodson, *Human Factors Design Handbook*, New York: McGraw-Hill, 1981.
9. D. J. Dooling and E. T. Klemmer, "New Technology for Business Telephone Users: Some Findings from Human Factors Studies," in *Information Technology and Psychology: Prospects for the Future* (edited by R. A. Kasschau, et al.), New York: Praeger Publishers, 1982, pp. 148–65.

## AUTHOR

**Eric E. Sumner,** B.M.E., Cooper Union, 1948; M.S. (Physics), 1953 and professional degree (Electrical Engineering), 1960, Columbia University; Bell Laboratories, 1948—. Upon joining Bell Laboratories, Mr. Sumner worked on wire spring relays, trouble recording apparatus, and ESS circuits. He was responsible for the first commercial PCM system for exchange trunks. He was promoted to Director and served in positions responsible for various transmissions systems. He was Executive Director of the Loop Transmission Division from 1967 to 1979, when he was appointed Executive Director of the Customer Network Operations Division. He assumed his present position as Vice President, Software and Processor Technologies, in 1981. Mr. Sumner is a member of the Advisory Board to Georgia Institute of Technology's School of Electrical Engineering; Chairman of the Advisory Council to Cooper Union's School of Engineering; President of IEEE Communications Society; Chairman of IEEE Policy Board; and recipient of the Alexander Graham Bell Medal of the IEEE for pioneering contributions to the field of digital communications. Mr. Sumner is author of a number of technical articles and talks and has been granted 11 patents. Fellow, IEEE; member, Tau Beta Pi and Pi Tau Sigma.

# History and Methods

This special issue begins with a brief history of human factors and behavioral science at Bell Laboratories. Bruce Hanson traces the expansion of applied and basic behavioral science at Bell Laboratories over more than three decades. His account shows the important role of laboratory studies and field tests in helping us understand how people interact with new telecommunication systems. The second paper in this section, by Daryl Eigen, discusses field test methods. Laboratory studies are often conducted early in a product or system development, while field tests occur later and involve realistic simulations. Eigen shows how field tests help to ensure that the human interface to complex telephone services meets the user's needs.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# A Brief History of Applied Behavioral Science at Bell Laboratories

By B. L. HANSON*

(Manuscript received March 16, 1982)

The history of applied behavioral sciences at Bell Laboratories follows two main paths. The first, the primarily customer-oriented "human factors" tradition, began in the late 1940s and has been characterized by an empirical approach, relying heavily on laboratory and field simulation. The second, the employee-oriented "human performance technology" tradition, had its roots in Bell Laboratories behavioral research organization formed in the late 1950s. It has been characterized by a more rule-oriented approach to the integration of human users and operators into large, computerized systems. This paper traces the evolution of these two applied traditions and the behavioral research organization and examines the people and the events that influenced their growth and success.

## I. INTRODUCTION

From the earliest days of telephony when Alexander Graham Bell and Thomas Watson worked to perfect Bell's invention into a useful and convenient communication device, designing for users has been an important goal of the Bell System. As the number of Bell System

* American Bell.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

employees has grown to more than a million, there has been an equal concern for using people efficiently in satisfying assignments. But it was not until the 1940s that specialists began to be hired to address customers' and employees' needs, and today Bell Laboratories employs hundreds of such specialists working in the behavioral sciences.

This article traces the major lines of evolution of applied behavioral science at Bell Laboratories and examines some of the forces that have shaped and continue to shape its growth, particularly the behavioral research organization. While there were some early applications elsewhere in the Bell System (the Western Electric Hawthorne studies conducted in the late 1920s and early 1930s, for example[1]), these are not treated here, except as they influenced events at Bell Laboratories.

## II. SETTING THE STAGE

Human factors, human performance technology, and engineering psychology are a few of the many names used to describe the application of the behavioral sciences (psychology, sociology, anthropology, etc.) to the design of systems that involve people. While each of these names has its own connotations, human factors will be used here for simplicity to denote the entire discipline. Human factors traces back to World War II when American and British psychologists worked to match complex new weapons systems to the people who would employ them. Most human factors specialists, then and now, have been trained in psychology as specialists in learning, human performance, visual and auditory perception, motivation, social behavior, or decision making. Many have received training in systems analysis, industrial engineering or specifically in human factors engineering.

Human factors at Bell Laboratories has evolved along two paths which have only recently begun to come together. The first arose from the needs of telephone customers. The second arose in response to the needs and skills of employees, and led to a "pure" behavioral research organization as well as applied activities. This article describes the two applied paths and the path of behavioral research. To prevent getting lost in digressions, a chronological road map is provided in Fig. 1. References point to sources of additional information.

## III. TELEPHONES AND CUSTOMERS

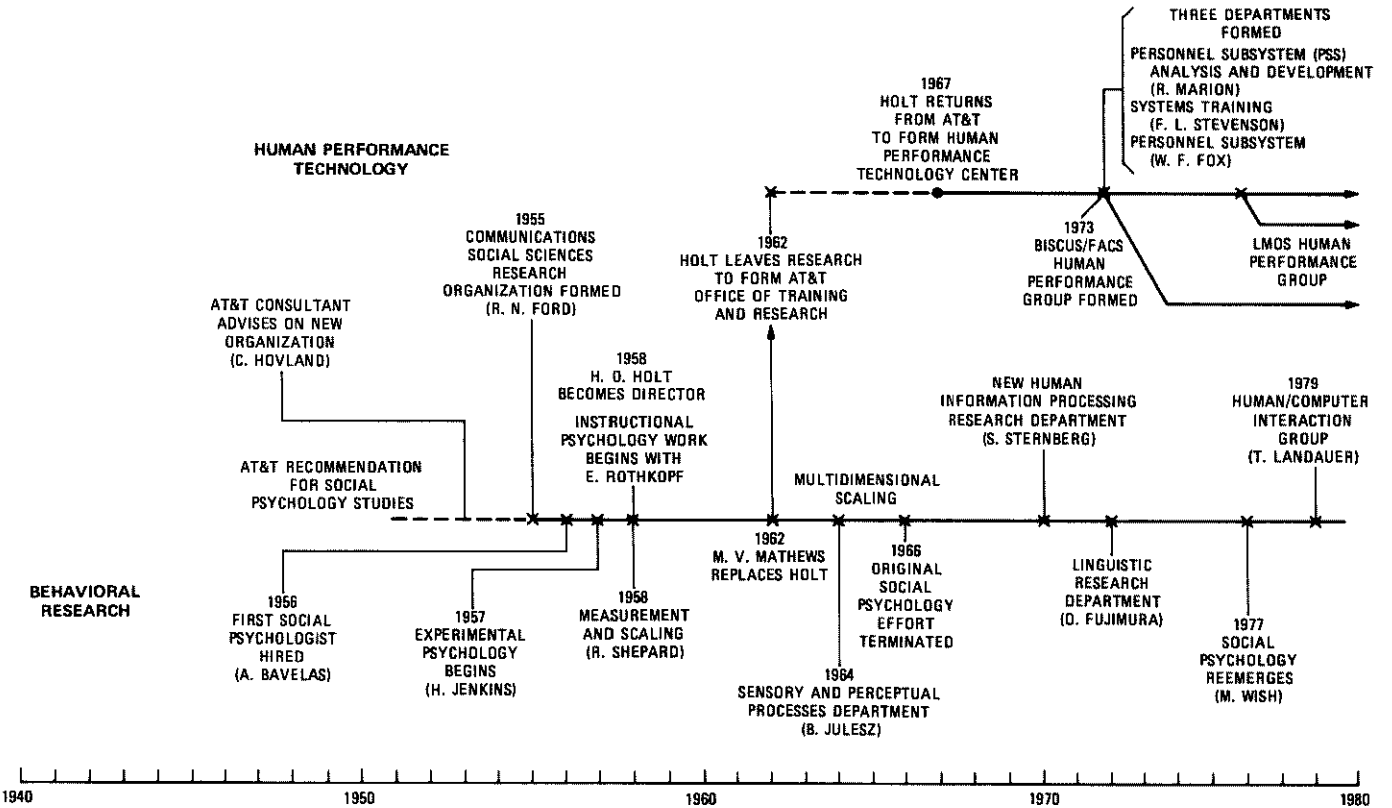
From the beginning of telephony, human factors decisions were being made about telephones. The first concern was to improve the quality and intelligibility of transmitted speech. Signalling was also important, especially as the number of users started to grow. Other, more subtle innovations also helped to make the telephone more useful

and convenient. The invention of the switchhook meant users no longer had to remember to throw a switch after each call. More calls went through and many batteries were saved because when the receiver was "hung up" on the switchhook (it had to go somewhere), the job was done automatically. The one-piece telephone handset added portability and convenience, but only because designers calculated the dimensions of the heads of potential users to ensure that when the receiver was placed to the ear the microphone would be the right distance from the lips. At Bell Laboratories in the 1920s and 1930s, work was focused on designing better telephone sets by considering the physical dimensions of customers' heads and hands[2] and on understanding the properties of the human ear and voice so that electrical transducers and circuits could be improved.[3] It was in this second area, known as psychoacoustics, that behavioral science was formally instituted at Bell Laboratories.

The pioneering work in this field was conducted at Bell Laboratories under the leadership of Harvey Fletcher, Wilden A. Munson, and others,[4] but, by the 1940s, laboratories at other institutions were doing similar work for communications systems for the World War II effort. One such laboratory was managed by S. Smith Stevens, an experimental psychologist at Harvard. Fletcher was aware of the work in Stevens' lab and, in 1945, hired John E. Karlin from the list of notable psychologists who worked there.

Karlin worked only briefly in psychoacoustics before realizing that there were many other opportunities for valuable behavioral sciences work. On the basis of his observations, he made a proposal to his management for a broad program of customer studies. Shortly thereafter, the User Preference Research Department was formed, headed by Walter A. Shewhart (widely known for his work in quality control), and staffed by Karlin, Robert R. Riesz, and others. In 1951, Karlin succeeded Shewhart as head of the department.

Karlin's department continued to expand, performing a growing range of studies on user preference, telephone design, and network applications. Notable among these was a series of studies confirming people's inability to make reliable preference judgments about things they have never experienced. In one of these studies, customers were asked to judge their preference for handsets lighter than those already on their telephones. They all preferred the existing 18-ounce weight. But when they had the opportunity to handle handsets of various weights, they preferred ones that were much lighter. No one showed a preference for handsets as heavy as the standard one. Even 12-ounce handsets were heavier than people preferred. The customers could not predict their own preferences without having actual experience with the alternative choices. This rejection of armchair opinions set the


HUMAN PERFORMANCE TECHNOLOGY
BEHAVIORAL RESEARCH
AT&T CONSULTANT ADVISES ON NEW ORGANIZATION (C. HOVLAND)
AT&T RECOMMENDATION FOR SOCIAL PSYCHOLOGY STUDIES
1955 COMMUNICATIONS SOCIAL SCIENCES RESEARCH ORGANIZATION FORMED (R. N. FORD)
1956 FIRST SOCIAL PSYCHOLOGIST HIRED (A. BAVELAS)
1957 EXPERIMENTAL PSYCHOLOGY BEGINS (H. JENKINS)
1958 H. O. HOLT BECOMES DIRECTOR
INSTRUCTIONAL PSYCHOLOGY WORK BEGINS WITH E. ROTHKOPF
1958 MEASUREMENT AND SCALING (R. SHEPARD)
1962 HOLT LEAVES RESEARCH TO FORM AT&T OFFICE OF TRAINING AND RESEARCH
1962 M. V. MATHEWS REPLACES HOLT
MULTIDIMENSIONAL SCALING
1964 SENSORY AND PERCEPTUAL PROCESSES DEPARTMENT (B. JULESZ)
1966 ORIGINAL SOCIAL PSYCHOLOGY EFFORT TERMINATED
1967 HOLT RETURNS FROM AT&T TO FORM HUMAN PERFORMANCE TECHNOLOGY CENTER
THREE DEPARTMENTS FORMED
PERSONNEL SUBSYSTEM (PSS) ANALYSIS AND DEVELOPMENT (R. MARION)
SYSTEMS TRAINING (F. L. STEVENSON)
PERSONNEL SUBSYSTEM (W. F. FOX)
1973 BISCUS/FACS HUMAN PERFORMANCE GROUP FORMED
LMOS HUMAN PERFORMANCE GROUP
NEW HUMAN INFORMATION PROCESSING RESEARCH DEPARTMENT (S. STERNBERG)
LINGUISTIC RESEARCH DEPARTMENT (O. FUJIMURA)
1977 SOCIAL PSYCHOLOGY REEMERGES (M. WISH)
1979 HUMAN/COMPUTER INTERACTION GROUP (T. LANDAUER)
1940
1950
1960
1970
1980

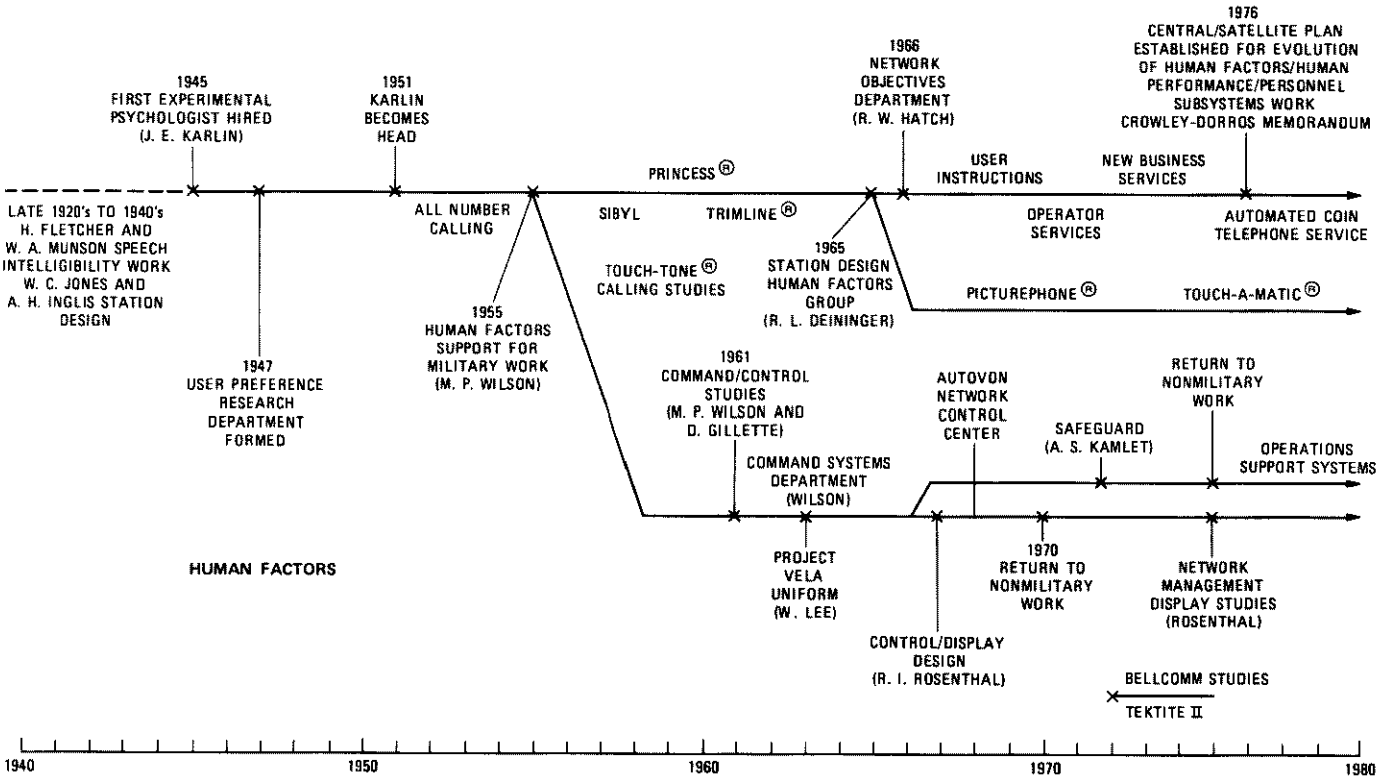



Fig. 1—Development of applied behavioral sciences organizations at Bell Laboratories.

tone for the activities in Karlin's department. The work was empirically based and it depended on providing people with product experience through laboratory simulation of alternatives.[5] At the root of this approach were the research methods and empirical orientation of experimental psychology.

The activities of Karlin's department in the early 1950s included studies comparing the use of seven-digit telephone numbers, termed "All Number Calling," with the two-letter, five-digit numbers in use at the time.[6] Both the telephone companies and customers were concerned that this change, designed to increase the number of available telephone numbers, would make dialing more of a problem. Laboratory studies showed that dialing performance was slightly faster with all digits, although long-term memory for numbers appeared to be slightly reduced. The small size of these effects led to the overall prediction, later confirmed, that All Number Calling would not impair customer dialing.[7] Other projects included design of dials for the 500-type telephone set (the little dots in the dial make dialing significantly faster, because they make it easier to see when the dial has stopped rotating from the previous number dialed). There were some futuristic studies demonstrating the feasibility, from the customers' viewpoint, of machine recognition of spoken digits (people with speech difficulties actually did better when they thought they were talking to a machine), and effects on customer perceptions of transmission quality of Time-Assigned Speech Interpolation (TASI)—a procedure for using the natural pauses in a telephone conversation to send bits of another conversation over the same line.

### *3.1 Military work*

By the mid-1950s, word of the success of Karlin's enterprise had reached Bell Laboratories military development organization at Whippany, New Jersey, which was dealing with some complex control systems with significant human factors implications. At that organization's request, M. Paul Wilson was transferred from Karlin's department to organize a military human factors effort.

In complex systems, the division of responsibility between human operators and hardware called for some creative new analytic techniques. In a series of projects, each characterized by complex command-control systems—such as the XM3-H tactical radar: the Nike-Ajax, Nike-Hercules, and Nike Zeus missile systems; and the SAGE air defense system—Wilson and his colleagues developed this new approach to command/control systems. Two major characteristics were:

1. The emphasis on the human's role as decision maker in the system; and

2. The importance of information displays in conveying critical information required for correct and timely decisions.

Shortly thereafter, Wilson became head of the newly formed Command Systems Department, with a single human factors group supervised by W. L. Lee, another former Karlin employee. In the mid-1960s, this group was supervised by Robert I. Rosenthal, who extended the Wilson formulations to the design of control panels and display systems.

One of the interesting developments from this organization was the design of a network control center for the AUTOVON military communications system.[8] It was significant for two reasons. First, the human factors specialists conceived and designed the system rather than serving as consultants. Second, it introduced the concept of "exception reporting"—limiting displayed information to unusual or abnormal conditions that the system operator needs to know. As a result, system operators could focus on analyzing and solving problems, the most critical job and the one they could do best, while the computer's ability could be used to sort through large quantities of data to find abnormal conditions. This work exemplified the kinds of things the military human factors people could do for the Bell System. In fact, Rosenthal's group later designed the display system for the No. 4 Electronic Switching System (ESS) electronic toll switching machine using the principles established in military work.

Human factors studies had a significant part in notable government-sponsored projects that included the Safeguard Antiballistic Missile System and the Bellcomm company that was organized to provide technical support for NASA's Apollo project. Work on the antiballistic missile system began in 1967 with a human factors system evaluation at White-Sands, New Mexico, and continued with Arthur S. Kamlet's Command and Control Test Facility.[9] Human factors at Bellcomm was led by A. N. Kontaratos and focused on basic studies of problems of extended isolation and the dynamics of small technical groups. Some of this work, led by Nicolas Zill, used Tektite II, a small undersea laboratory where groups of scientists worked for extended periods. The Bellcomm effort was quite independent of other Bell System human factors activities.

By the mid-1970s most of the military work had been phased out at Bell Laboratories and the people had been reassigned to telecommunications projects, bringing with them the command/control system approach and experience in broader aspects of system design.

### *3.2 Sibyl, station equipment, and network applications*

About the time Wilson left in the mid-1950s, the Karlin organization began development of a new, complex system for conducting simula-

tion studies. This new system, called Sibyl, could be programmed to insert a wide range of network transmission and switching impairments into real telephone calls, record user-calling behavior, and record user-preference judgments.[10] Up to 100 Bell Laboratories volunteers could be connected to the system.

Sibyl's first use was in the evaluation of push-button vs. rotary dialing. It recorded dialing times, intervals between digits, and errors. In later applications, Sibyl varied factors such as dial tone delay, network blocking, or delay after dialing. Following each test call, Sibyl would call the user back to get a subjective evaluation. Users would dial one of several digits to express their opinions. Knowledge of customer opinions of various levels of service enabled designers to engineer the telephone network to satisfy customers without spending money on improvements that customers did not care about.

In 1966, Sibyl and other transmission quality activities were consolidated in a separate department under Richard W. Hatch. In 1970, psychologist Herman R. Silbiger was chosen to supervise a new human factors group in Hatch's department, specializing in subjective evaluation of network performance. Although it has undergone much modification Sibyl is still in use.[11]

Karlin's department also continued its work in customer instructions (design of flowchart dialing instructions for centrex users), transmission (defining the optimal parameters for trading off echo suppression with the listener's ability to interrupt, especially on long-delay satellite connections) and telephone set design. Particularly notable was Richard L. Deininger's work on the design of the *Touch-Tone** telephone. Most of the telephone set design work was carried out in cooperation with the Station Instrumentation Department (headed up by Alfred H. Inglis, and then Harris F. Hopkins), and the *Touch-Tone* telephone design was no exception. Deininger's responsibility was to determine the optimal parameters of the push-button dial, particularly the arrangement of the 10 buttons.

Deininger's studies led to the selection of the now ubiquitous *Touch-Tone* telephone dialing arrangement. His studies showed that it was strongly preferred over the now equally familiar calculator arrangement.[12]

Karlin's department also worked jointly with the Station Instrumentation people to design the *Trimline** telephone.[13] Early efforts to design a dial-in-handset telephone had been unsuccessful because the sets were uncomfortably bulky and unattractive. Invention of the space-saver dial by Charles F. Mattke enabled Lionel W. Mosing of Karlin's department to fashion a comfortable and attractive telephone

* Trademark of AT&T.

of angular design, which he called the *Trimline* telephone. The final version of this set is now installed in millions of homes and appears on display in the design collection of New York's Museum of Modern Art. Much additional customer work was conducted primarily by engineers in the Station Instrumentation department. In 1964 this work was inherited by Leo Schenker's Telephone Station Studies Department, and Deininger was named to supervise the human factors work.

### *3.3 Station studies*

For several years, Deininger's group was heavily involved in design and evaluation of the *Picturephone** visual telephone service, particularly aspects of image and quality and camera placement.[14] In 1970, after Deininger's untimely death, the group was managed by Gaber P. Torok, who worked to bring human factors work closer to the product-development activity, speeding up the development cycle by reducing the need for lengthy field testing and subsequent modifications.

In 1973, Murray J. Katz replaced Torok as supervisor of this group. Advancing technology, increasing customer demand for new products and the Bell System's entry into a more competitive environment all led to steadily increasing demand for the services of Katz's group. As a result of Katz's human factors evaluations, modifications were made to many new station products including the *Touch-a-matic*† repertory dialer (changes in labels, button characteristics, handset location), *Design Line*† decorator telephones (modifications to improve transmission characteristics, balance, and comfort), and *Dataphone*† II data communications terminal (design of maintenance panels and instructions). They also conducted extensive studies of the effects of mobile telephone usage on driving behavior[15] and worked jointly with Karlin's department on design of calling procedures for the Advanced Mobile Phone Service (AMPS).

Throughout the 1970s, station design work was changing from largely physical design ("knobs and dials" as it is sometimes called) to more procedural or feature-oriented work. Human factors specialists were addressing questions of how to implement new and complex features for residence and business customers. There was also a growing awareness of the role of human factors in identifying and satisfying customer needs. As a result, human factors was playing a larger role in the product design process. In the late 1970s, increasing demand for human factors support led to the formation of new groups in both station and business terminal development.

---

* Registered service mark of AT&T.
† Trademark of AT&T.

### 3.4 Repositioning of human factors

In the mid-1960s, Karlin's department addressed a wide variety of customer issues such as the potentially disruptive effects of satellite delay on telephone conversations,[16] design of flowchart instructions, and diagnosis of network impairments. The problem with diagnosing network troubles was to correlate customer descriptions of problems with the problems, themselves. The solution was to create the same network impairments in the laboratory and ask test customers to describe the problems. Once the customer descriptions were tied to the difficulties, customer complaints could be used to pinpoint and repair the troubles.

Feasibility studies were also conducted to see whether business customers, if given computerized switching systems, could successfully program their own rearrangements and changes, e.g. reassignment of telephone numbers and features when employees moved offices. Such changes tend to occur frequently, and can be costly to both the customer and the telephone company if an installer must be dispatched to make them. It was found that, with the right user interface, customers could make their own changes with little difficulty, so long as they had a way to trace and correct their errors. An added benefit of this capability, which now exists in a number of Bell System customer switching systems, is that customers can make changes instantly, without having to wait for a telephone company employee to be dispatched.

The Human Factors Department had started in the research area and still viewed itself largely as an applied research department, but by 1970 it found itself in a development organization which was interested in increasing the direct, near-term payoff of human factors. Where the department had previously been organized around psychological topics (visual studies, interpersonal communication, etc.) it was asked to realign along Bell System lines (e.g., operator services, loop and outside plant). Along with this realignment, groups were given the responsibility for identifying and addressing existing human factors opportunities in their assigned areas and for contacting the appropriate organizations to discuss them.

During this same period there was a growing concern throughout Bell Laboratories about the impact of technology on customers and employees. While there is question as to cause and effect, the growing sensitivity to the impact of new products on customers and employees and the refocusing of human factors research were followed by remarkable growth in the human factors area. Within two years, Bell Laboratories "consumers" of human factors work were negotiating for long-term commitments of human factors support and were exploring the option of starting their own human factors efforts to guarantee

adequate staffing of their work. By 1976, new human factors efforts had begun in both operator services and loop operations, and a director-level committee, conceived and led by Herbert M. Zydney, was assembled to plan an orderly expansion strategy.

The resultant plan, known as the Crowley-Dorros plan after the two executive directors who commissioned it—Thomas H. Crowley and Irwin Dorros—outlined a strategy for all human factors work at Bell Laboratories, including both customer and employee issues. The plan called for Karlin's department to focus on customer issues and to serve as a centralized resource to a growing number of satellite human factors groups that would be closely tied to specific development activities. The plan also viewed Karlin's department as a training ground for individuals to start such satellite groups.

As a case in point, an entire group, led by Edmund T. Klemmer, was transferred from Karlin's department to continue its work on business customer telephone systems. The group started with five people and an emphasis on procedures and instructions, and has since grown to two groups working on all aspects of business services, from customer needs studies to field introduction.[17]

In 1977, Karlin retired from Bell Laboratories after 32 years of service. (See Ref. 18 for Karlin's entertaining and insightful parting comments on the state of human factors at Bell Laboratories.) He was replaced by Charles B. Rubinstein, an electrical engineer from the Research Area who had done work in the psychophysics of color perception and visual thresholds. Shortly after Rubinstein arrived, Robert I. Rosenthal, who had supervised military control-display work under M. Paul Wilson and later worked on network management display systems, joined the department. In the late 1970s, the department's work included design of customer dialing procedures for AMPS,[19] and other new services that relied on recorded instructions to customers, such as the Voice Storage System,[20] the Automated Credit Card Service,[21] and Automated Coin Telephone Service (ACTS).[22] Each of these services depended on recorded instructions that could explain to customers what they were to do next. Computer-controlled voice recording and playback equipment enabled human factors people to conduct extensive laboratory simulations of these services, identifying deficiencies in procedures or instructions that led users to make mistakes. ACTS, for example, uses recorded messages to tell coin telephone customers how much money to deposit in the telephone. Laboratory and field studies were required to determine how long to wait for customers to deposit coins, and how often to prompt them for the remainder. Time limits had to be established to ensure that a human operator could come on the line to assist customers having difficulty. Customer satisfaction was also monitored to

ensure that people would be comfortable with the less personal approach to coin service. As a result of the attention to human factors, ACTS has met with complete success. Customers have few difficulties with the service, and customer acceptance has been outstanding.

Recently a new focus has emerged. Bell Laboratories is developing new, complex products whose success will depend on the quality of the user interface. As a result, Rubinstein's department and other customer-oriented human factors groups are directing their efforts to human/computer interface issues.

## IV. THE RESEARCH CONNECTION

### *4.1 Behavioral research beginnings*

The second path in the evolution of applied behavioral sciences at Bell Laboratories began in the early 1950s at AT&T. The Personnel Relations department had been doing applied field work for some time, but had become frustrated by lack of knowledge of the social processes which influenced organizational success. Bolstered by a 1953 commitment by the AT&T Board of Directors to attracting and developing capable employees and first-rate leaders, Robert K. Greenleaf, Director of Personnel Research at AT&T, asked Bell Laboratories to start a new research group, the Communications Social Science Research Department.

The new department was led by Robert N. Ford who transferred from AT&T Personnel Relations, and its charter was to focus on problems of communication, organization, and leadership in small groups. The department was established at Bell Laboratories because BTL had experience in managing basic research and because of the stimulation other Bell Laboratories research activities could provide. Ford was to report directly to William O. Baker, then Vice President and Director of Research. To help get the effort under way, and to help recruit good research people, psychologist Carl I. Hovland was brought in from Yale as a consultant.[23]

The first person hired was Alex Bavelas, whose specialty was human motivation. Bavelas left after only a short time to join the Stanford faculty. Morton Deutsch, another social psychologist, was the second to arrive, and stayed for several years. Both Bavelas and Deutsch went on to establish major reputations in the academic world.

In addition to this venture in social psychology, efforts were begun in experimental and educational psychology. The experimental psychology activity began in 1957 with Herbert Jenkins (a disciple of Harvard behaviorist B. F. Skinner), who established a lab to study learning in pigeons. He was followed by Ernest Z. Rothkopf, an educational psychologist specializing in learning and instruction, and

Roger Shepard, another Harvard experimentalist recruited by Hovland and specializing in information processing.

In 1958, H. O. (Ollie) Holt, an educational psychologist who had been Deputy Director of George Washington University's Human Resources Research Office (HumRRO), took over operation of the center from Ford, who returned to AT&T to pursue his interest in job enrichment. Holt immediately began to develop a plan for entering the new field of programmed instruction. He worked primarily with consultants such as Hovland, Tom Gilbert from the University of Tennessee, and Skinner, himself. Given the massive sums of money spent by the Bell System in training its employees, the payoff from such work was potentially enormous.

This work showed such promise that Holt was transferred to AT&T in 1962 to head the Office of Training Research which had the job of implementing individualized instructional technology throughout the Bell System.[24] It is Holt who provides the ultimate connection to applied behavioral sciences, so the path of applied work follows him. But the Behavioral Research organization, with its worldwide reputation for excellence, deserves a short history of its own.

#### *4.2 Basic research in psychology*

Research on the social psychology of group interaction was the initial focus of the Behavioral Research Center. The problems ranged over issues of cooperation, influence, and social perception and the communicative behavior entailed. The amount of work in these areas gradually diminished, and the original social psychologists had all left by 1966. However, social psychology reemerged from 1978 to 1982 with a new central concern in interpersonal communications. The issues now involved the reasons why people choose one communication modality over another (e.g., voice-only versus face-to-face) and the consequences. This work was headed by Myron Wish, whose involvement in a study of the use of *Picturephone* visual telephone service sparked renewed interest in such matters.

There were other topics that, like group psychology, were pursued only for limited periods. Most notable of these were efforts in auditory neurophysiology between 1964 and 1968, and the psycholinguistics of grammar from 1963 to 1971.

By contrast, the research in human learning and instructional technology, which was under way by 1958, has continued, expanded, and diversified up to the present. For example, Ernst Z. Rothkopf and Lawrence T. Frase conducted a large number of experiments on the effect of adjunct questions on learning from text. These not only showed practical means for greatly improving instruction, but also deepened understanding of the active role of learners in studying.

Pioneering studies of the role of organization (e.g., the placement of repetitions of information) and surface structure (e.g., word choice) in learning from written prose led, among other things, to the development of computer methods for determining text difficulty (see Ref. 25). The fundamental work on instructional materials contributed to the development of widely used editorial aids such as the *UNIX** Writer's Workbench software (see Refs. 26, 27, and 28), and AT&T's Training Development Standards, a guide used in course development throughout the Bell System.

Two other areas that have continued since their establishment in the 1950s are human information processing and psychological measurement. Human information processing research at Bell Laboratories goes at least as far back as the work of John R. Pierce and John Karlin on estimating human channel capacities[29]; but as a continuous institutional activity, it can be dated from George Sperling's discovery and description of a visual memory that stores, for about a second, considerably more information than can be read out intact.[30] Saul Sternberg's elegant work on rapid scanning of active symbolic memories followed shortly.[31] He and his collaborators also made important advances in the study of temporal order judgments, and most recently, the control of rapid action sequences in speech and typing.

The Human Information-Processing Research Department, headed by Sternberg since 1970, has also been the home of pioneering work and major contributions in many other parts of cognitive psychology: verbal learning, picture memory, semantic memory, word recognition, visual perception, motor control (see article by Rosenbaum,[32] this issue) and adaptation, attention, and reasoning. In 1979 a new group was formed under Tom K. Landauer's leadership to explore cognitive problems in human use of computers. Its work has centered on issues in interactive language design and the proper representation of human knowledge to assure mutual understanding of the messages passed between machines and users (see article by Furnas et al., this issue).[33]

Measurement and scaling were among Roger Shepard's many interests when he first joined the Labs in 1958. Along with Joseph B. Kruskal, he developed the first effective methods for nonmetric multidimensional scaling of human similarity judgments.[34] There has been an almost continuous development of new theory, computational methods, and applications ever since. For example, in recent years J. Douglas Carroll and his collaborators have developed the method of individual differences multidimensional scaling, a method that identifies psychological dimensions by their orderly variations in perceptual importance as reflected in judgments of object similarity by

* Trademark of Bell Laboratories.

different people. These methods have found extensive use in such diverse areas as marketing research and telephone tone-ringer design.[35]

Over the last 20 years, the Research division has also sponsored a variety of other efforts with behavioral science content. There has been much noteworthy work in vision, color vision, and visual perception under the leadership of Bela Julesz. One example is his own invention and use of random-dot stereograms to demonstrate the independence of depth perception from object recognition and his explorations of fundamental mechanisms of perceptual processing. (See Ref. 36.) Speech analysis and evaluation research was conducted in a department headed by Peter B. Denes, and elsewhere. Linguistic research has been under way since 1971 in a department led by Osamu Fujimura, who himself has studied the production of motor action sequences, an interest shared recently by several other investigators. Finally, mention should be made of the important fundamental analyses of speech acoustics and perception carried out by James F. Flanagan's Acoustics Research Department and Manfred R. Schroeder's Hearing and Speech Synthesis Research Department.

Over the years there has been fairly regular communication between the behavioral research groups and the applied behavioral scientists. There have been many seminar series attended by both, frequent consultations and visits, and since 1977 a Bell Laboratories-wide convention of all behavioral scientists every 18 months. There are occasional cross-assignments of months or a year's duration, and a steady trickle—averaging perhaps one per year—of people moving from one kind of work to the other; most movement has been from the smaller population of research to the larger applied areas. There has been a great deal of useful stimulation in both directions.

## 5. HUMAN PERFORMANCE TECHNOLOGY

### *5.1 Human performance and employee systems*

The Bell operating telephone companies, with their huge billing and accounting systems, were among the first businesses to make widespread use of computers in their operations. During the late 1950s and early 1960s, computers became an integral part of their accounting operations. While the development of computerized systems had resulted in substantial savings and better service, there had also been some difficulties, largely because of a mismatch between such systems and the employees who worked with them. The initial solution to these problems was thought to be in better training, so Ollie Holt and Harry A. Shoemaker of AT&T's Office of Training Research were asked to get involved.

At about the same time, it became clear that centralized planning

and development of such systems would be more efficient and would ensure system-wide standards. Finally, in 1967, largely through the efforts of AT&T Assistant Vice President Bruce Warner, the Business Information Systems Programs (BISP) organization was created from parts of the AT&T Planning Department. Holt was asked to join the new organization as Director of the Human Performance Technology Center. After several months Holt moved, along with the whole BISP organization, to Bell Laboratories.

Holt brought with him several of his former staff members including Fred L. Stevenson, a training specialist originally with Pacific Telephone. The group's early experience showed that training was not going to be a universal cure for design deficiencies, so they set about to develop an organization that could influence system design.

One of the first additions to Holt's staff was Bill F. Fox, a psychologist with human factors training and experience. Fox had worked with HumRRo and Lockheed Aircraft. Fox's assignments were to adapt military experience with large hardware systems to the development of large software systems and to build a team of psychologists to do the work.

Holt also recruited Ralph Marion, a psychologist and personnel specialist from Southern Bell. Marion's assignment was to develop documentation that would enable system designers to do as much of the job as possible, themselves. Together, they continued to recruit both operations people from the Telephone Companies and behavioral scientists. By the end of 1970 they numbered over 40.

They called their work personnel subsystems (PSS), a phrase with military roots, rather than human factors roots, a term which they felt referred narrowly to control-display design rather than systems concerns. In later years, the name was changed to human performance engineering to more clearly characterize the major objective. In 1970 the Human Performance Center was organized into three departments. Fox's Personnel Subsystems Department provided direct consultation and support to projects and worked to develop PSS technology; Marion's Personnel Subsystem Analysis and Development Department refined the technology and documented it; and Stevenson's Systems Training Department trained others in the use of the technology.[37] This organization worked well, and the technology, documentation, and training it developed are in use in information systems organizations throughout the Bell System.[38]

In many ways Fox's department was the BISP homologue of John Karlin's Human Factors Department. Both were staffed primarily by behavioral science professionals who provided consultation to system designers and developers, and both experienced significant successes in the 1970s. The primary difference between them was in their way

of approaching the job. The Karlin approach was to solve each design problem empirically; the Fox approach was to assist the project in utilizing the PSS technology to solve its own problems. In each case, the approach was appropriate to the problem—Karlin dealt with a number of specific problems, each with significant implications for many telephone users, while Fox was faced with vast needs for job requirements, training, etc., which could not have been efficiently done by behavioral scientists alone.

Typically, Fox's people would be assigned to projects as technical advisors to the project people who had design responsibility. For the most part, these project people were telephone company people on rotational assignment who had no prior experience with PSS. These people used the documentation produced by Marion's department and were trained in Stevenson's. The success of this work and the desire of the project organizations to have complete responsibility and guaranteed support led to the spin-off of separate PSS groups. In 1973, George A. Schweickert and Mort H. Kahn left Holt's center to form groups in Business Information System Customer Service Facilities Assignment and Control System (BISCUS/FACS).[39] Shortly after, the organization involved in the Trunks Integrated Record Keeping System (TIRKS) formed its own group under Roy J. Porterfield.[40] Barry K. Schwartz also left to work on systems engineering for network administration systems and later formed a group to work on the Total Network Data System (TNDS). By 1976, most of the projects which had started as part of BISP had their own groups of specialists, and few of Fox's people were assigned to specific BISP projects. Instead, many of them were working on user aspects of Operations Support Systems (OSSs) developed elsewhere in Bell Laboratories to aid in the operation and maintenance of various aspects of the Bell System network. The first of these to come to Fox's attention was the Loop Maintenance Operations System (LMOS).[41] By 1976, one of Fox's groups, led by Grace H. Leonard, had transferred to the LMOS organization.

In late 1976, the Crowley-Dorros plan—which had called for Karlin's department to serve as a centralized customer human factors resource—named Fox's department as the resource for employee human performance activities, serving as both consultants and the source of trained people to start satellite groups. Since that time, a number of new groups have been created in LMOS and other OSS projects, employing the methods and procedures developed in the late 1960s by Holt and Fox and their colleagues.

The growth of groups and activities has continued unabated through the late 1970s and early 1980s. Again the increasing complexity resulting from new technology has placed new burdens on employees

and provided new opportunities and challenges for behavioral scientists.

## VI. SUCCESSES. . .AND CHALLENGES

Measured by its impact and its growth, human factors has had notable success at Bell Laboratories, particularly since the early 1970s. Both the empirical orientation of Karlin and the technological approach of Holt and Fox have had significant impact. Why has human factors been a success at Bell Laboratories when external efforts in both industry and the military have fared less well? There are several possible explanations.

First, the philosophy and structure of the Bell System have been important. Its strong tradition of good customer service implies a need to give explicit consideration in the design process to the customers. The vertical integration of the system encouraged Bell Laboratories to design products and systems which are not only attractive at time of purchase, but which also continue to perform cost-effectively. In such an environment, design goals such as minimizing customer errors or increasing employee efficiency and decreasing turnover become more important. The numbers of employees and customers potentially affected by new designs also multiply the benefits that result from human factors.

The growing complexity of technology has also played a significant role. While technological advances have provided many new products and services, these products and services have become more complicated for the customers who use them and the employees who install and maintain them. Computerization of internal operations has led to substantial complications in some aspects of employee jobs. Human factors specialists have the responsibility to minimize the problems imposed by this complexity and to make new products and services "friendlier," more useful, and more attractive.

In addition to the technical challenges of the information age, the major challenge to human factors in the 1980s is to continue to expand its role in the design/development process. This implies both applying existing human factors skills in new areas and developing new skills which complement existing ones. For example, extension of human factors methods to the definition of customer needs for new products will lead to earlier and more influential involvement in product design. Development of systems engineering skills will result in improved communication with engineers and greater influence over design decisions. Specialized knowledge in electrical engineering and computer science will enable evaluation of the trade-offs that must often be made between human factors and hardware and software constraints. With a broader view of its responsibilities, human factors should continue to grow and contribute at Bell Laboratories.

## VII. ACKNOWLEDGMENTS

The author would like to thank the many people who provided information and insights for this history. I am particularly indebted to H. O. Holt for his encouragement and perspective and to W. F. Fox and T. K. Landauer for their generous assistance. Thanks are also due to J. E. Karlin, M. J. Katz, and R. I. Rosenthal for the significant background material they provided. Special thanks go to B. L. Lively for resolving the many thorny issues generated by a project of this nature.

## REFERENCES

1. G. Homans, "Group Factors in Worker Productivity," in E. Maccoby, T. Newcomb, and E. Hartley, eds., *Readings in Social Psychology*, 3rd ed., New York: Holt, Reinhart and Winston, 1958, pp. 583–95.
2. W. C. Jones and A. H. Inglis, "The Development of a Handset for Telephone Stations," B.S.T.J., *11*, No. 2 (April 1932), pp. 245–63.
3. H. Fletcher, "The Nature of Speech and Its Interpretation," B.S.T.J. *1*, No. 1 (July 1922), pp. 129–44.
4. H. Fletcher and W. A. Munson, "Loudness, Its Definition, Measurement, and Calculation," B.S.T.J., *12*, No. 5 (October 1933), pp. 377–430.
5. R. R. Riesz and H. D. Irvin, "Simulation in Engineering," Bell Lab. Rec., *36*, No. 7 (July 1958), pp. 238–41.
6. J. E. Karlin and R. K. Potter, "Preference Research," Bell Lab. Rec., *32*, No. 5 (May 1954), pp. 161–6.
7. J. E. Karlin, "All Numeral Dialing, Would Users Like It?" Bell Lab. Rec., *36*, No. 8 (August 1958), pp. 284–8.
8. J. W. Gorgas, "AUTOVON, Switching Network for Global Defense," Bell Lab. Rec., *46*, No. 4 (April 1968), pp. 106–11.
9. A. S. Kamlet and J. D. Musa, "Human Factors Testing of Computerized Systems," Bell Lab. Rec., *52*, No. 8 (September 1974), pp. 245–51.
10. "Simulating Speech Through Space," Bell Lab. Rec., *38*, No. 8 (August 1960), pp. 296–8.
11. J. L. Sullivan, "Is Transmission Satisfactory? Telephone Customers Help Us Decide," Bell Lab. Rec., *52*, No. 3 (March 1974), pp. 90–8.
12. R. L. Deininger, "Human Factors Engineering Studies of the Design and Use of Pushbutton Telephone Sets," B.S.T.J., *34*, No. 4 (July 1960), pp. 995–1012.
13. C. L. Krumreich and L. W. Mosing, "The Evolution of a Telephone," Bell Lab. Rec., *44*, No. 1 (January 1966), pp. 8–14.
14. R. R. Stokes, "Human Factors and Appearance Design Considerations of the Mod II PICTUREPHONE Station Set," IEEE Trans. Communic. Tech., *COM-17*, No. 2 (April 1969), pp. 318–23.
15. A. J. Kames, "A Study of the Effects of Mobile Telephone Use and Control Unit Design on Driving Performance," IEEE Trans. Vehicular Tech., *VT-27*, No. 4 (November 1978), pp. 282–7.
16. E. T. Klemmer, "Human Factor Problems in Satellite Telephoning," Human Factors, *8* (December 1966), pp. 475–80.
17. S. H. Ellis and R. A. Coskren, "A New Approach to Customer Training," Bell Lab. Rec., *57*, No. 2 (February 1979), pp. 60–6.
18. J. E. Karlin, "The Changing and Expanding Role of Human Factors in Telecommunications Engineering at Bell Laboratories," Eighth International Symposium on Human Factors in Telecommunications, Churchill College, Cambridge, England, September 1977. Reprinted in J. P. Duncanson, ed., *Getting It Together: Research and Applications in Human Factors*, Proceedings of a Symposium sponsored by the Metropolitan Chapter of the Human Factors Society, Stevens Institute of Technology, Hoboken, New Jersey, November 17, 1977.
19. B. L. Hanson and C. E. Bronell, "Human Factors Evaluation of Calling Procedures for the Advanced Mobile Phone System (AMPS)," IEEE Trans. Vehicular Tech., *VT-28*, No. 2 (May 1979), pp. 126–31.
20. B. L. Hanson, R. J. Nacon, and D. P. Worrall, "New Custom Calling Services," Bell Lab. Rec., *58*, No. 6 (June 1980), pp. 174–80.

21. J. O. Hardy, D. G. Raj-karne, and K. A. Raschke, "Handling Coin Toll Calls—Automatically," Bell Lab. Rec., *58*, No. 8 (September 1980), pp. 256–62.
22. M. R. Allyn, T. M. Bauer, and D. J. Eigen, "Human factors in the design of a new service," Bell Lab. Rec., *58*, No. 5 (May 1980), pp. 155–61.
23. C. I. Hovland, "Two New Social Science Research Units in Industrial Settings," American Psychologist, *16* No. 2 (February 1961), pp. 87–91.
24. H. O. Holt, "Programmed Self Instruction," Bell Telephone Magazine, *42* (Spring 1963), pp. 24–7.
25. E. U. Coke and M. E. Koether, "A Study of the Match Between the Stylistic Difficulty of Technical Documents and the Reading Skills of Technical Personnel," B.S.T.J., this issue.
26. L. T. Frase, "The *UNIX*™ Writer's Workbench Software: Philosophy," B.S.T.J., this issue.
27. N. H. Macdonald, "The *UNIX*™ Writer's Workbench Software: Rationale and Design," B.S.T.J., this issue.
28. P. S. Gingrich, "The *UNIX*™ Writer's Workbench Software: Results of a Field Study," B.S.T.J., this issue.
29. J. R. Pierce and J. E. Karlin, "Reading Rates and the Information Rate of a Human Channel," B.S.T.J., *36*, No. 2 (March 1957), pp. 497–516.
30. G. Sperling, "Successive Approximations to a Model for Short Term Memory," Acta Psychologica, *27* (1967), pp. 285–92.
31. S. Sternberg, "Memory Scanning: Mental Processes Revealed by Reaction-Time Experiments," American Scientist, *53*, No. 4 (1969), pp. 421–57.
32. D. A. Rosenbaum, "Central Control of Movement Timing," B.S.T.J., this issue.
33. G. W. Furnas et al., "Statistical Semantics: Analysis of the Potential Performance of Key-Word Information Systems," B.S.T.J., this issue.
34. R. N. Shepard, "The Analysis of Proximities: Multidimensional Scaling With an Unknown Distance Function," Psychometrica, *27* (1962), pp. 219–46.
35. J. D. Carroll and M. Wish, "Measuring Preference and Perception with Mathematical Models," Bell Lab. Rec. *49*, No. 5 (May 1971), pp. 146–54.
36. B. Julesz and J. R. Bergen, "Textons, the Fundamental Elements in Preattentive Vision and Perception of Textures," B.S.T.J., this issue.
37. F. L. Stevenson, "An Individualized Approach to BISP Training," Bell Lab. Rec. *52*, No. 9 (October 1974), 271–78.
38. H. O. Holt and F. L. Stevenson, "Human Performance Considerations in Complex Systems," Science, *195* (March 18, 1977), pp. 1205–209.
39. J. J. Yostpille, "BISCUS/FACS Processes Service Orders Automatically," Bell Lab. Rec. *55*, No. 4 (April 1977), pp. 96–102.
40. J. W. Timko, "TIRKS Takes the Paperwork Out of Trunk Recordkeeping," Bell Lab. Rec. *53*, No. 9 (October 1975), pp. 368–75.
41. J. J. Appel and J. L. Woodruff, "Streamlining Loop Operations," Bell Lab. Rec., *56*, No. 6 (June 1978), pp. 147–54.

## AUTHOR

**Bruce L. Hanson,** B.A. (Psychology), Johns Hopkins University; M.S., Ph.D. (Psychology), Pennsylvania State University; Bell Laboratories, 1972–1982; American Bell, 1983—. After joining Bell Laboratories in 1972, Mr. Hanson worked on the human factors aspects of central office craft jobs. He also has helped evaluate hardware, instructions, and procedures for outside plant craftspeople, engineers, and managers, and he has studied human factors in new services such as the Advanced Mobile Phone Service. In 1977, he became Supervisor of the Network Systems group, designing user interfaces for voice storage services. Before he assumed his present position, he also supervised studies of customer aspects of billing, PhoneCenter® and Business Office service. Mr. Hanson is currently Supervisor of the Customer Interfaces Group in the Product Family Services Planning Department and ABI. This group is engaged in the design and planning of user-computer interfaces for ABI products. Member, Human Factors Society, the American Psychological Association, and Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Methods for Field Testing New Telephone Services

By D. J. EIGEN*

(Manuscript received December 29, 1981)

Telephone services are increasingly complex and diverse, and they require more human-machine interaction than ever before. A field test can help improve a new service by ensuring that it is easy to use with little chance for error. This paper discusses the methodology of field testing. Specially tailored telephone service evaluation methods, based on field test experience with the Calling Card Service, are presented in detail.

## I. INTRODUCTION

New telephone services involve more customer-system interaction than ever before, and making the use of these services easy and error-free is a major goal of their development. Properly designed dialing plans, announcements, timings, tones, and instructions increase customer acceptance, minimize customer errors, and promote use of the service. The design of new services can be evaluated by coordinated studies that include:

- Analysis of present services,
- Interviews with customers,

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

- Laboratory studies of proposed protocols for customer-system interactions,
- Field tests of services, and
- Product follow-up studies.

Field testing is the largest and most costly step in this coordinated set of studies. This paper discusses the methodology of field testing, using a field test of the Calling Card Service as an example.

## II. CALLING CARD SERVICE FIELD TEST OVERVIEW

An analysis of operator-handled credit card service indicated that a reasonable proportion of credit card calls could be automated. Interviews with credit-card, bill-to-third-number, and collect customers verified their interest in and need for an automated Calling Card Service, and laboratory studies provided evidence that customers could use the Calling Card Service successfully.

The Calling Card Service field test was conducted in Milwaukee from November 1977, to June 1978.[1,2] Regular telephone credit card numbers could be dialed in order to place Calling Card calls from about 3000 noncoin phones in the Milwaukee area and from 70 coin phones at Milwaukee's airport, two downtown hotels, and a few local restaurants. Bright orange placards on the trial coin phones instructed customers on how to use their telephone credit card number. In addition, operators were trained to assist callers and answer questions. (Calls from unequipped stations were handled as usual.)

To use the trial Calling Card Service, customers first dialed zero plus the number they wished to call. Special programs in the Traffic Service Position System (TSPS) routed incoming "0+" calls from trial stations to a small team of specially trained operators who helped simulate Calling Card Service—in actual service no operators are used. Besides the TSPS console, these operators had a video display terminal linked to a minicomputer (see Fig. 1).

When a call arrived from a specially equipped station, the trial operator notified the minicomputer, which then delivered a tone to prompt the customer to dial a Calling Card number. Detectors received the dialed digits and sent them to the minicomputer over a data link for verification. Calls with valid Calling Card or credit card numbers were allowed to proceed and were billed appropriately.

Depending on the protocol being tested, the minicomputer displayed step-by-step instructions on a terminal screen to guide the operator in handling each call. For example, to encourage customers to redial after making errors, the minicomputer might display the instruction, "Please hang up and dial zero plus the number you are calling," which was to be read to the customer. By making simple changes in the minicomputer program, the operator's treatment of calls could be

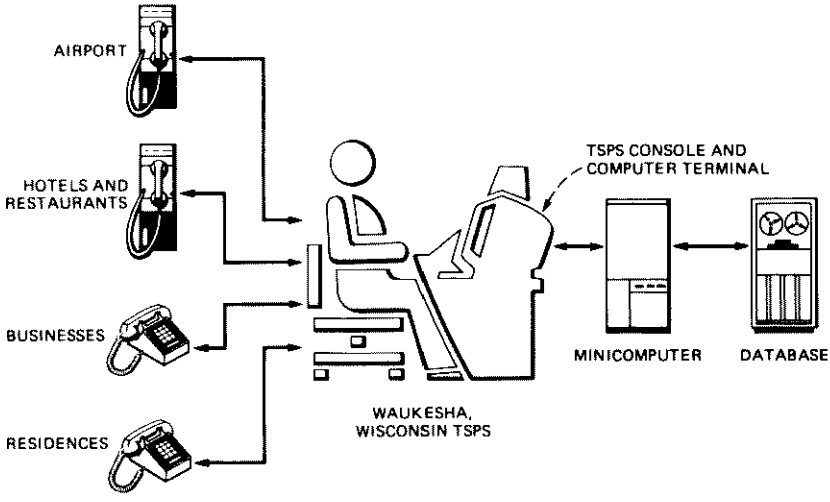



Fig. 1—Test setup.

altered, often without additional training. This flexible arrangement allowed easy testing of many different protocols and rapid changes among them.

The minicomputer recorded the time and details of each call. These records were analyzed rapidly to determine how the protocol could be improved. Throughout the trial, protocols were varied by changing announcements, timings, access to operators, error-correction procedures, and other aspects of the caller interface. In all, 24 variations of the protocol were tested at equipped coin phones, 14 at noncoin phones. Over 10,000 customers dialed more than 30,000 automated Calling Card calls during the trial and more than 5,000 customer interviews were obtained.

## III. FIELD TEST ACTIVITIES

### *3.1 Field tests*

In a field test such as that for Calling Card Service, a simulation of the proposed service is actually offered to a limited, but representative, set of customers on a trial basis. In some circumstances, when there is sufficient confidence in the form of the service, the actual product can be used as a test vehicle.

The field test can be used to adjust the technical and operational aspects of the service. It can be used to determine whether the service fits a customer need, and it can also be used to improve estimates of willingness to pay. And finally, it can be used to evaluate customer performance, satisfaction, and usage in the effort to provide a service

that optimizes the customer-system interaction on which new services rely.

The greater the fidelity of the test to the real service situation, the greater one's confidence in the final success of the service being tested. Field tests provide increased fidelity over more indirect techniques for evaluating services.

### *3.2 Study plan*

Developing a study plan is the first task in planning a field test. A study plan must include the following five steps:

1. State the objectives of the study and delineate issues to be resolved by it. The primary objectives of the evaluation of the human-machine aspects of telephone services are to:
   a. Determine if usage, satisfaction, and performance are at acceptable levels.
   b. Predict the levels of usage, satisfaction, and performance in the final service.
   c. Refine the service to improve usage, satisfaction, and performance.

Many other detailed issues for a particular service may require resolution.

2. Determine constraints on study service and resources necessary to accomplish the test. Decisions on test methodology necessarily involve practical choices. For example, the marketplace often imposes serious time constraints on the development, deployment, and evaluation of a telephone service.

3. Design and refine the service and the human-machine interfaces involved. Some initial human-machine telephone service interface must be defined before the evaluation process can be initiated. Continued analyses, interviews, and laboratory studies are best used to generate and refine the service alternatives prior to the field test itself.

4. Delineate variables that may influence the results and hypothesize their interactions. Three categories of variables need to be specified for the field test:
   a. Independent variables—those variables that are to be deliberately manipulated or held constant. Among the possible variables of this type (with some examples) are the following:
      (1) Service protocol—announcements, tones, timings, error-handling strategies, and digit strings.
      (2) Capabilities of the service—billing, routing, and screening.
      (3) Availability—geographic constraints, time-of-day constraints, and station-type constraints.
      (4) Type of instruction for customers—media and format.
      (5) Marketing effort—promotion.
      (6) Rate—price structure.

b. Dependent variables—those variables whose values are affected by changes in the independent variables. Some examples are given below.
   (1) Subscription—initial interest and sign-up rate.
   (2) Usage—rate of first and repeated use.
   (3) Acceptance—judged worth and satisfaction.
   (4) Customer performance—speed, error, and abandonment rate.
c. Parameters—those identifiable variables that are free to vary. Among these are:
   (1) A priori condition—predisposition toward service, demographic mix of customer population, etc.
   (2) Internal characteristics—the test design or method used, intrinsic characteristics of the customer in the test, etc.
   (3) External characteristics—the geographic, environmental, and temporal setting of the test.

These variables are not necessarily mutually exclusive. Some parameters might be fixed or systematically varied, and, thus, be made independent variables.

It helps to have hypotheses about how these variables will interact. Figure 2 demonstrates one model of several possible models of the relationships among variables for telephone services. A different model might assume that instruction may affect acceptance and subscription more directly than is shown in Fig. 2.

5. Define methods for increasing confidence in the results of the study. Some of the most important aspects of a good study plan are motivated by the need to counteract rival explanations of the results

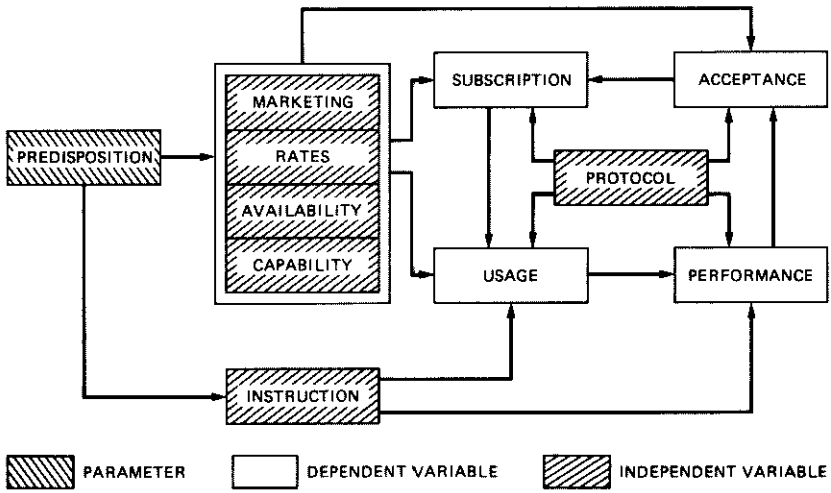



Fig. 2—Variable block model.

and, thereby, increase confidence in the validity of the field test as an indicator of the success of the service.[3–5] Section IV is devoted to this component of the study plan.

### *3.3 Test development*

After the study is designed, the following tasks must be done.

1. Define data gathering tools.
2. Define data analysis techniques.
3. Define test service delivery vehicle.
   a. Define test service requirements.
   b. Design test service delivery vehicle.
   c. Implement service delivery vehicle (hardware/software development).
   d. Integrate trial system and test.
4. Acquire and prepare customers.
5. Acquire and prepare site(s).
6. Define test operations and procedures for utilizing results.

Figure 3 is an illustration of the interrelationships among these tasks.

#### *3.3.1 Data collection*

The following data collection methods were used in the Calling Card Service field trial.

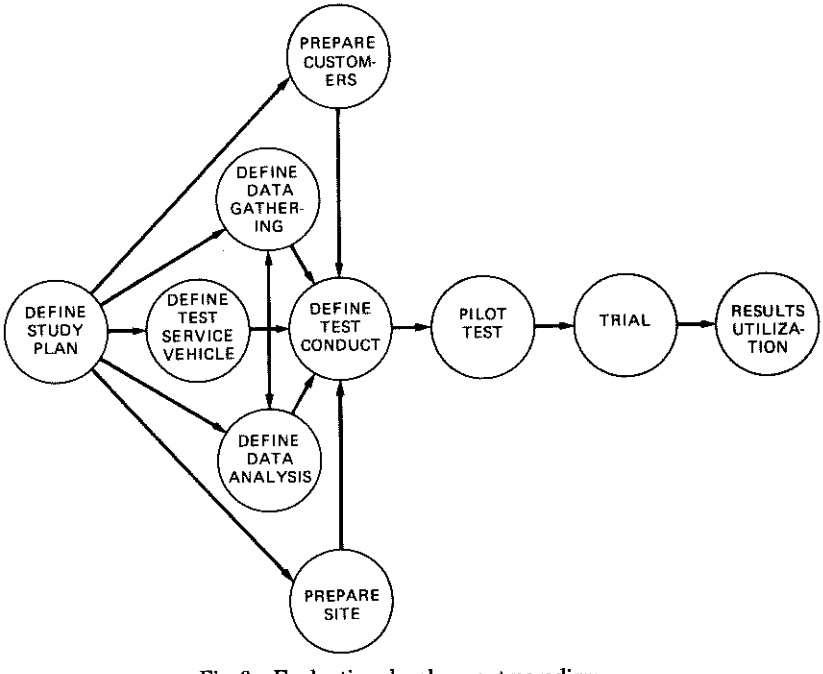



Fig. 3—Evaluation development paradigm.

*3.3.1.1 Interviews/questionnaires.* Interviews and questionnaires provide measures of attitude, as well as corroborative measures of usage, and can be given before, during, and after the field test. In-person, face-to-face interviews with customers who had used the Calling Card Service provided some of the more compelling evidence of the success of the service. The following spontaneous comments illustrate the range of such data.

Example of positive response:

1. "It's excellent. I compliment the phone company for coming through with this. Makes it a lot easier."

Example of negative response:

1. "In the past, you dialed Helen (operator) and said, 'I want to talk to Joe,' and right off Joe was on the line—it seems you are trying a lot of new services to get rid of Helen."

*3.3.1.2 On-line measurements.* These measurements are the machine recording of customer and system events, such as picking up a receiver or placing it back on the switchhook (hanging up), dialing digits, tones or announcements, error detection, and call progress states. By recording, time stamping, and storing these events, the sequence of customer-machine interactions can be reconstructed.*

*3.3.1.3 Billing.* Billing records routinely provide various data on a call, such as calling and called number, call duration, date, time of day, class of charge (e.g., collect, DDD, and so forth).

*3.3.1.4 Observation.* Observation provides a crude but valuable technique for gathering data that cannot be obtained by other means.

*3.3.1.5 Customer feedback.* Feedback mechanisms for customer complaints and suggestions are maintained by the telephone companies offering the service.

*3.3.1.6 Records.* Records and logs of equipment malfunction and events recorded in newspapers are important for detecting or counteracting erroneous or misleading results. Data in such records can be used to measure dependent variables.

These methods of collecting data are all relatively insensitive to one another in that a specific error in one measurement is very unlikely to affect another measurement. Thus, the methods can be used to corroborate each other.

---

* The Bell System as a common carrier is entitled to certain privileges to monitor the quality of its services. Bell Laboratories as an agent of the Bell System is extended these privileges. Only data necessary to ensure good service are gathered, and they are kept secure, in strict confidence, and statistically anonymous. To guarantee customer privacy, data are never gathered after a call is placed, i.e., after called party answers.

### 3.3.2 Analytic tools and techniques

The practice of feeding back the results as input to the experimental design process, while providing an efficient data gathering technique, places an additional burden on analysis. Results must be provided in a timely manner to be of use in formulating the next service improvements to be tested. Figure 4 illustrates a data processing stream used to field test the Calling Card Service.

Standard statistical routines must be used with caution since the assumptions on which they are based may be violated. For example, sampling is often nonrandom.

One heuristic is to leave the service unperturbed for some reasonable period of time (two or more weeks depending on the call volume). The extent of variations or noise in the data is measured and later used as a benchmark to assess meaningful variations potentially due to service manipulations.

### 3.3.3 Test service provisioning

A system to deliver a voice-prompted test service should include the ability to deliver tones and announcements, to route to an operator, and to receive dialed digits. Connection to the network, billing, and data processing and storage also need to be considered.

### 3.3.4 Acquire and prepare sample

Obtaining sufficient numbers of representative customers for the field test is important. The field test approach usually requires thou-

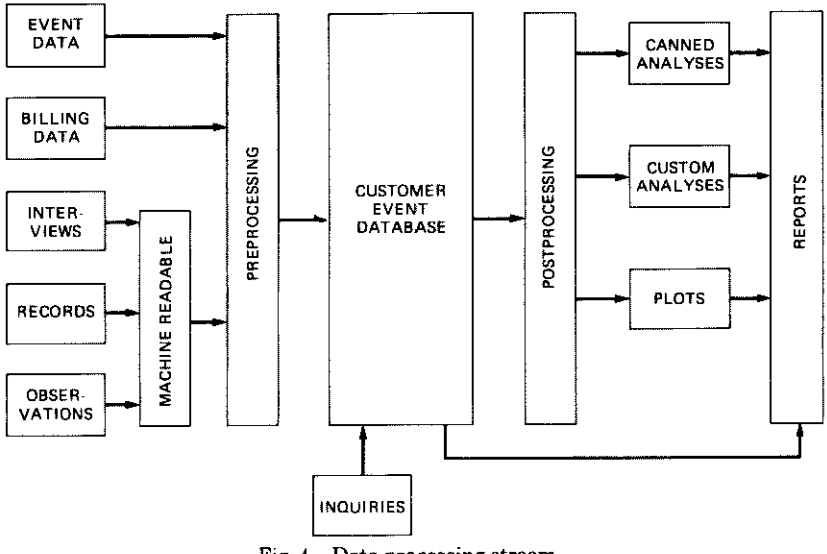



Fig. 4—Data processing stream.

sands of calls to provide adequate data. Inability to obtain sufficient customers, if not attributable to test limitations alone, may be a key indicator of the potential failure of a new service. The means for selecting and acquiring customers has to be designed with test validity in mind.

The process of obtaining participants and assigning them to groups can be the source of some of the strongest rival hypotheses. Random selection is the principal counteraction to ensure the representativeness of the population sample and group equivalence. But, randomization is not always possible. Moreover, even when it can be used, aspects of the customer acquisition and assignment process can still introduce biases that will limit the ability to generalize.

Selection processes are outlined below:

1. Determine target population using interviews and market studies.
2. Characterize the history, predisposition, environmental influences, etc., of target population.
3. Draw customers randomly from within the target population and from a similar group outside the target population. Sampling customers outside the target population will help to validate the use of the target groups for tests. If random selection is not possible, select customers who are matched to characterization of the target population.
4. Solicit participation for the test in a representative way. Preferably, contact customers in the same way they would be contacted for the final service.
5. Check participants for degree of similarity to target population characterization.
6. Interview random samples of those who agree and disagree to participate to determine the reasons for participation or nonparticipation. For example, the fact that it is a test rather than an actual service may have influenced either decision. Also, use a priori factors to account for any differences between those who agreed and those who refused.
7. Randomly assign participants to control groups and treatment groups. If random assignment is not possible, use self-selection of treatments.
8. Use measurements taken before the test to help determine the nature of group differences.

#### *3.3.5 Results utilization*

One of the largest pitfalls in the practice of implementing an evaluation of telephone service can be the lack of assurances and mechanisms for integrating evaluation results into the final product. Results from the test must be timed to allow development of changes

in the final product. A responsible organization must be named for coordinating the integration of results and the final product itself must be designed to allow changes. Appropriate flexibility in the initial design of the product can reduce delays in product introduction.

## IV. DESIGNING TO COUNTERACT OTHER EXPLANATIONS

Steps taken in designing the test—which decrease confidence in rival hypotheses and increase confidence in the working hypotheses (the one we wish to prove)—are called counteractions. In laboratory research, some typical counteractions are:

- Orthogonality and counterbalancing of independent variables
- Control and preclusion of extraneous variables
- Random collection of subjects in the attempt to eliminate selection bias.

Field settings require adapting such measures to real-world constraints.

The counteractions useful in field tests can be divided into those required for evaluating a static service (i.e., determining if a service meets a priori acceptability standards) and those required for evaluating attempts to improve a service. Counteractions can be adapted to preclude, disconfirm, or control rival hypotheses.

### *4.1 Service evaluation*

One of the simplest counteractions is to take multiple measurements of the same service, such as measurements of usage, satisfaction, and performance. Taking these measurements in more than one way (multiple methods) is also useful. Confidence in a particular result is increased if different sources point to the same conclusion.

One fear (rival hypothesis) that occasionally strikes is that the data are somehow mutilated to spuriously inflate results. This fear became real in the Calling Card Service field trial when the service began performing better than expected. Interviews, observation, and billing data corroborated the computer-collected data so strongly as to explode this rival hypothesis. Test calls removed any remaining shards of doubt.

Many rival hypotheses are based on test time. For example, the results may be due to some unusual circumstance or event, or the test may give rise to a trend in usage or performance that may be transitory (e.g., novelty effects or start-up effects) or the test may give rise to cyclic or delayed effects. One simple counteraction used in the Calling Card Service field trial to account for these problems was to repeat a test or take continued measurement of the service over time. If a unique circumstantial or historical event affected the outcome of a

test, then subsequent measures could provide evidence to support or negate this rival hypothesis.

Before, during, and after usage measurements of all coin stations (trial and nontrial) at the airport were inspected to determine if any events, trends, or cyclic phenomena confounded the data. Specifically, the Christmas holiday season and an airline strike had to be taken into account.

While multiple measurements and time-spaced testing counteract many problems, they give rise to problems of their own. One measurement may affect another. For example, repeated interviewing may cause customers to change their behavior more as a function of the interviewing process than the service itself.[6] Fortunately, the bulk of the measurements typically taken in a telephone service field trial are made on-line and are unknown to the telephone caller.* The customer is not usually aware in these tests that digits dialed, on-hooks and off-hooks are recorded and time stamped. These measures are presumably nonreactive, and thus there is no plausible explanation of measurement affecting customer behavior and attitude.

One could avoid the problem of reactivity by using only nonreactive measures. But interviews and questionnaires provide invaluable data. We handled this problem in the Calling Card Service field trial by interviewing some customers once, others twice, and still others not at all. Comparisons of these groups in terms of nonreactive measures and subsequent interviews, however, did not substantiate a reactivity effect.

Establishing a control group is another strategy that can be used to help determine if the results were due to historical events or arbitrary circumstances. The assumption is that those extraneous events or circumstances that affect the treatment group also affect the control group. To the extent that the precision of the measures allow, differences between the control group and the treatment group can, therefore, be attributed to the presence of the treatment, that is, the service.

People at nontrial coin stations at the airport were interviewed as a control for interviews taken at trial coin stations. This procedure allowed us to determine if there were any changes in customer attitudes that might have been attributable to events alone and not to service. The interview control group thus served as a counteraction to the rival hypothesis of history. For example, those people interviewed some days were angry because of plane delays. If this irritation had an effect, it would show up in both treatment and control group scores.

---

* The customer generally is aware that a test is being conducted and that service is being measured; thus, the problem of test reactivity cannot be completely discounted.

Randomization in the selection process assures equivalence between the treatment and control groups. If randomization is not possible, a plausible rival hypothesis is that the people assigned to one group differ, on the average, from those in the others and that these intrinsic differences are the real cause of any observed difference among the groups. A method for determining whether such rival hypotheses are correct is to make measurements of the groups before the treatment (service) is administered. Comparing the pretreatment and post-treatment observations of the group given the trial service with simultaneous observations of the control group (not given the trial service) tests the rival hypotheses that these two groups are inherently different, or in other words, that results are due to the history of the customers or their maturation or learning during testing.

Customers were interviewed and billing data were tracked before and after the Calling Card Service field trial. These data were used to determine the differences among customers and between trial and nontrial stations.

### *4.2 Improving service*

In the preceding discussion, counteractions for a static service evaluation were discussed. A new service can be said to consist of a set of variables that may be manipulated to possibly improve service.

The key question is: How can we be sure that changes in service outcome are due to our service manipulations? Establishing causal connections for what improves or degrades service increases understanding. Increased understanding will tend to increase confidence in predictions of how the final service will fare.

#### *4.2.1 Varying the service*

A simple model of service variation is depicted in Fig. 5. The knobs in the figure represent the independent variables; the meters, the dependent variables. The object of varying the service is to manipulate the knobs in such a way as to cause the meters to register a beneficial change in service. If the meters are considered to show customer satisfaction, usage, and performance, the object then is to find the combination of knob positions that maximizes the readings on each meter (metaphorically speaking).

With every knob manipulation, knowledge and understanding is increased, whether the meters change positively, negatively, or not at all. Knowledge gained from the previous adjustment permits efficient and effective design of subsequent service adjustments. However, such a test-adjust-test method requires a system that can provide very quick analysis of results.

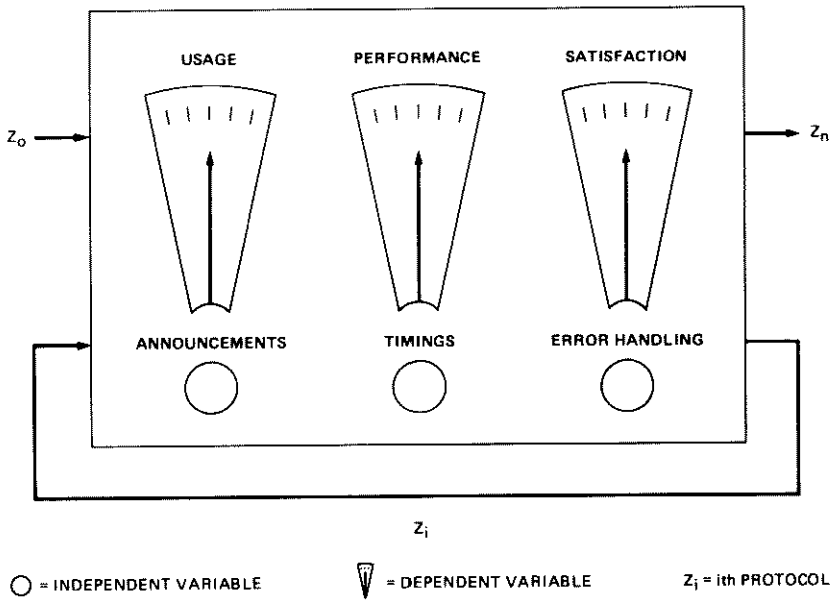



Fig. 5—Field trial metaphor—with results fed back as input into the field trial design.

#### *4.2.2 Increasing confidence in service manipulations*

Improving service can be thought of as repetitions of static service evaluation. The same counteractions discussed earlier can be used and augmented for service improvement. To account for the possibility of other events interposing effects on the dependent variables coincident with the adjustment of a service, the concept of the control group is again appropriate.

One new problem that is introduced in improving service is that multiple treatments (changes) may interact with each other. That is to say, customers' responses to a subsequent service may not be the same as those to the first service they experienced. Further, the very act of changing the service may impact attitudes and customer performance.

Adding a new group of customers for every new protocol iteration counteracts this threat to validity. (Appendix A contains notations for representing this design consideration and the others discussed here.)

The rudiments of these design principles were used in the Calling Card Service field trial. Service was manipulated in the trial in an attempt to improve service. A particular service configuration is here called a service protocol. Because many of the trial phones were in the airport, there were always a great number of new customers trying the service, more than enough to support the manipulations.

When customers dialed 0+NPA NXX XXXX at trial stations a tone prompt was given (sometimes followed by an announcement prompt) and the following things (in most protocols) could happen:

1. The customer could dial a credit card number.
2. The customer could time out without dialing and be connected to an operator.
3. The customer could dial 0 after the prompt and be connected to an operator.
4. The customer could abandon the call.

Ideally, all customers with credit card numbers would dial them. All others would dial 0 after the tone prompt to reach an operator. Practically speaking, this was not possible. Rather, different protocols were tested to attempt to increase the proportion of dialed credit card calls, increase the proportion of customers who needed an operator to dial 0, or decrease abandons.

Figure 6 summarizes the service manipulations for the placarded coin stations. Figure 7 shows the proportion of the four classes of call dispositions: (1) dial credit cards, (2) dial 0, (3) abandons, and (4) time-outs for each protocol for the placarded coin stations. The area under each curve represents the incremental proportion of 0+ calls.

While these curves are revealing, they cannot be used alone to determine the best protocol, i.e., the one with the highest satisfaction, performance, and usage. However, unacceptable protocols can readily be identified: they are those with abandonment rates in excess of 25 percent and those with dialed credit card call rates less than 25 percent.

Fortunately, the first coin placarded service protocol had low abandonment rates and high credit card dialing rates. The subsequent

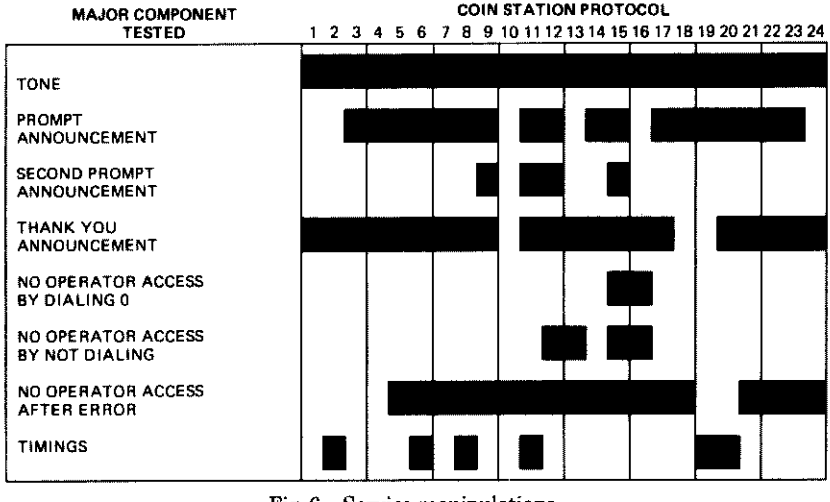
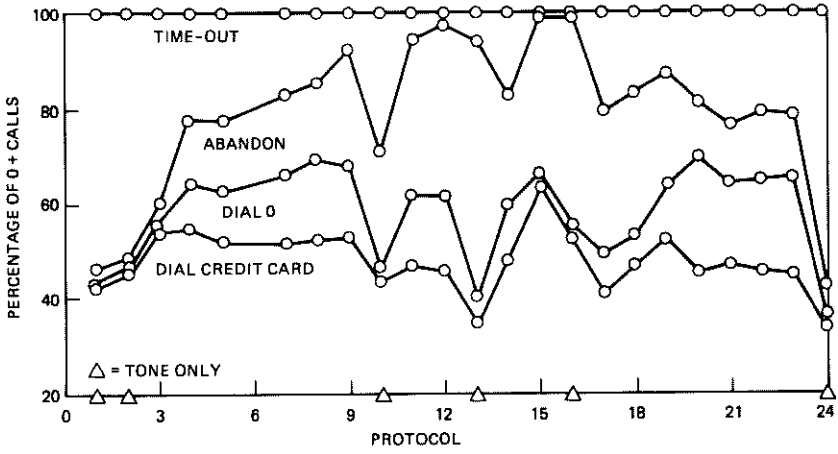



Fig. 6—Service manipulations.




NOTE: PROTOCOLS WITHOUT A TONE-ONLY INDICATOR HAVE A TONE AND AN ANNOUNCEMENT. PROTOCOL 17 INCLUDES A SPECIAL ANNOUNCEMENT DESIGNED TO DISCOURAGE CALLS FROM UNENABLED ROTARY STATIONS.

NOTE: THESE CURVES REPRESENT INCREMENTAL PERCENTAGES.

Fig. 7—Dialing performance at placarded coin stations by protocol.

manipulations were primarily aimed at improving service, although clearly the opposite effect was sometimes achieved.

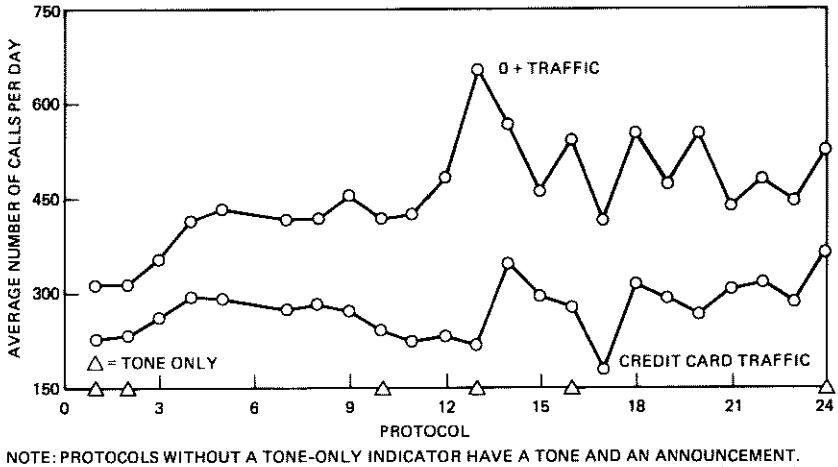
A changing volume of 0+ calls and a changing call mix were plausible explanations for one service protocol doing marginally better than another. As Figure 8 shows this was sometimes a consideration.

Another rival hypothesis was related to assumptions about those who abandoned. If all or a large number of those who abandoned were credit card customers, estimates of the proportion of credit card dialers could vary significantly. Figure 9 shows the percentage of dialed credit card calls (of all credit card calls) with and without abandons counted as credit card calls. Including and excluding abandons in this way provides lower and upper bounds, respectively, of the percent of credit card calls dialed.

Still another rival hypothesis was that usage (or attempts) may be higher, but performance lower when comparing one protocol to another. Figure 10 illustrates success rates by protocol.

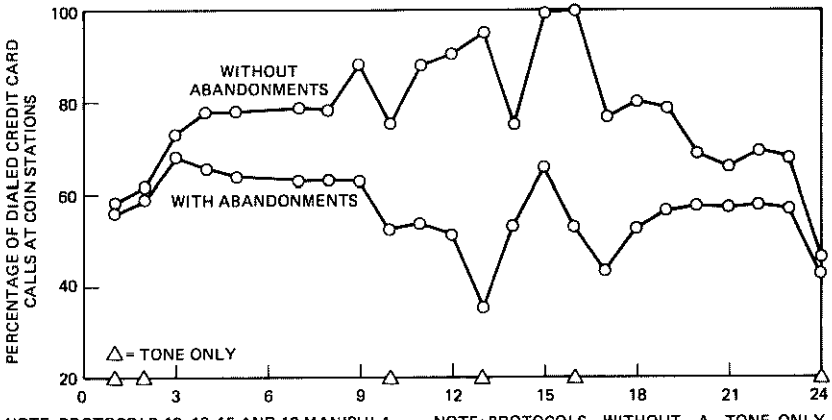
Rival hypotheses due to repeated usage (familiarity and learning) were accounted for by sorting out and examining results from repeat users.

Service protocols were compared to determine the effect of a particular manipulation. Figures 11 and 12 compare two protocols in terms of abandonments and percent of 0+ dialing. These figures lead to the conclusion that repeating a prompt announcement in this service does not increase dialing but does increase abandons. Thus, repeating announcements is a less desirable alternative.




NOTE: PROTOCOLS WITHOUT A TONE-ONLY INDICATOR HAVE A TONE AND AN ANNOUNCEMENT. PROTOCAL 17 INCLUDES A SPECIAL ANNOUNCEMENT DESIGNED TO DISCOURAGE CALLS FROM UNENABLED ROTARY STATIONS.

Fig. 8—Volume of 0+ traffic at placarded coin stations.




NOTE: PROTOCOLS 12, 13, 15 AND 16 MANIPULATIONS INVOLVED LIMITING OPERATOR ACCESS WHICH REDUCES 0 VOLUME AND CAUSED 0 + CUSTOMERS TO EITHER ABANDON OR DIAL A CREDIT CARD NUMBER.

NOTE: PROTOCOLS WITHOUT A TONE-ONLY INDICATOR HAVE A TONE AND AN ANNOUNCEMENT. PROTOCOL 17 INCLUDES A SPECIAL ANNOUNCEMENT DESIGNED TO DISCOURAGE CALLS FROM UNENABLED ROTARY STATIONS.

Fig. 9—Dialed credit card calls with and without abandonments by protocol.

### *4.2.3 Comparison groups*

There are two ways of making service comparisons in the test design just discussed, which consisted of staggered subjects and measurements of responses to service refinements. First, customer comparisons can be made across service changes. This consists of multiple measurements or time-spaced measurements of the same customer. The

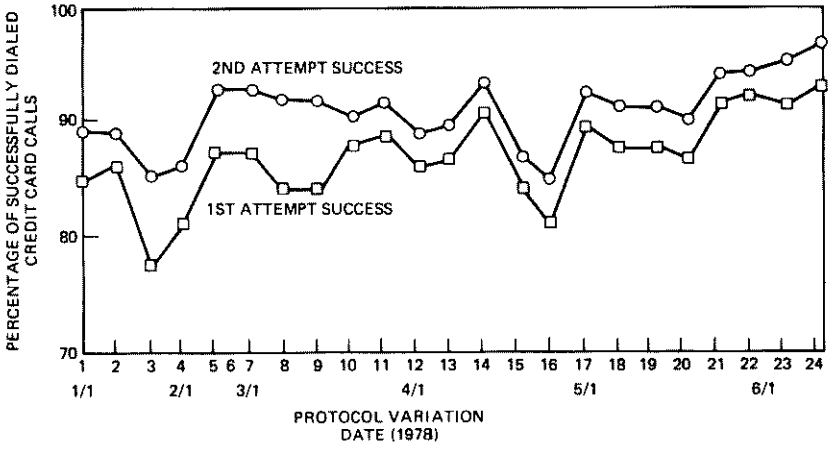



Fig. 10—Customer dialing success rate.

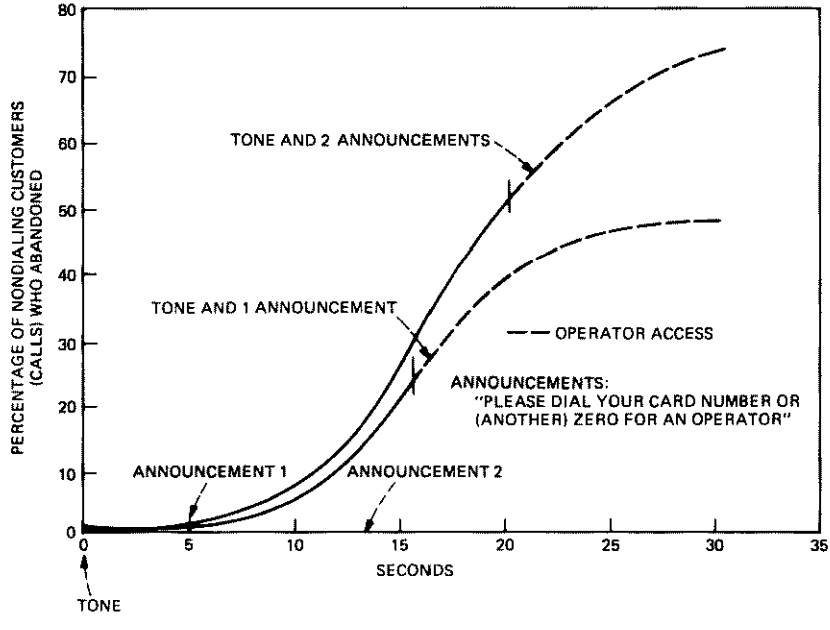



Fig. 11—Abandonments as a function of repeated announcements at coin placarded stations.

error variance is small because there are no intersubject differences, but there is multiple treatment interference. Second, service comparisons can be made across new subject groups, but treatment effects are confounded with history and by any bias introduced by the selection of groups.

Comparison groups can be arranged to preclude history as a con-

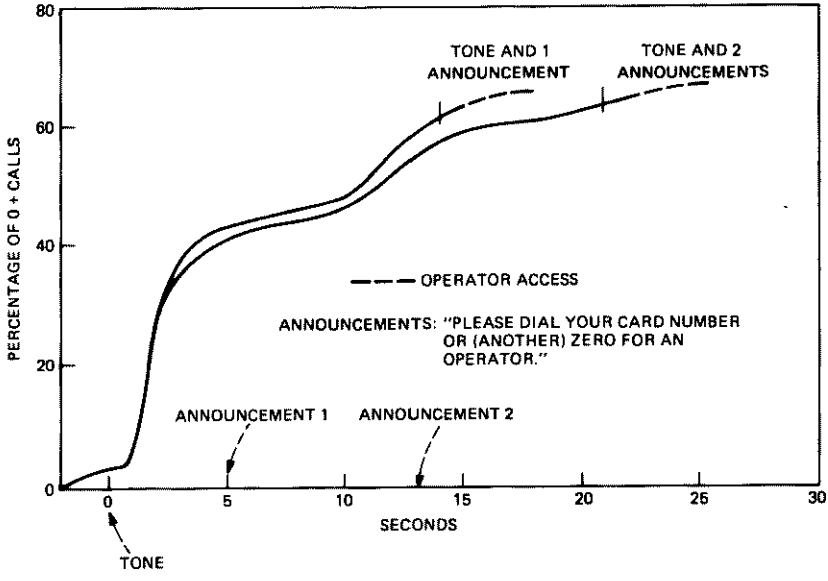



Fig. 12—Dialing as a function of repeated announcements at coin placarded stations.

founding variable by offering different service arrangements simultaneously (at different telephones, for example). For best results, each of these comparison groups should be as similar as possible except for any planned differences in treatment (service).

In the Calling Card Service field trial, we devised three comparison groups: noncoin stations, placarded coin stations (with special bright orange instruction cards), and placardless coin stations. These comparison groups were not mutually exclusive. Crossovers occurred, especially between placarded coin and placardless coin groups. But for every comparison group, and, as discussed, for every manipulation, there were new customers. Service changes were replicated across these comparison groups further increasing confidence and providing information about the effects of the station type.

#### *4.2.4 Refinement*

Because the number of variables and variable states in a field test of this kind is usually large, only a small subset of the number of possible treatment combinations can be used. Moreover, the protocol is evolving and a delay is often necessary in the testing of different combinations. Unfortunately, this delay means the manipulation is confounded with the history of the customer population. However, prudent selection of changes and repetitions within a treatment group and across comparison groups can yield a more complete understanding of the effects of the service components and their interactions.

In the Calling Card Service, the primary objective of the service manipulations was to make reasonable changes which were hypothesized to increase usage, satisfaction, and performance. This objective was sometimes subjugated to the purpose of increasing confidence in the belief that the changes in the independent variables (and not the parameters) caused the observed changes in the dependent variables. One such manipulation was to deliberately make it progressively more difficult for the customer to get to an operator, i.e., protocols 12, 13, 15, and 16. It was hypothesized that more customers would use their credit card if it were made relatively more difficult to obtain an operator. However, this hypothesis was not borne out; the proportion of credit card calls dialed was either unchanging or decreasing, and abandons increased as one would suspect. Clearly, other manipulations were more successful.

As the treatment change chart shows, changes in the state of more than one independent variable were made more often than not. The large number of variables prohibited single sequential changes in each variable. For example, the error-leg protocol was manipulated independently and simultaneously with the access-leg protocol many times. Evidence gathered from protocols prior to this practice showed little or no interaction from changes in the access leg and the error leg, except that the number of people making errors fluctuates.

Inferences made about the protocol change were made in terms of the multiple change because the independent variables were confounded. Inferences based on effect separation and independence assumptions were made cautiously and verified, when possible, in later manipulations.

#### *4.2.5 Replication*

Any changes in the dependent variables caused by a change in the protocol (or service) and not by history should be present in successive identical tests (within the bounds of measurement error). Replication of effects increases confidence in the hypothesis that a change in protocol actually causes a change in the dependent variable being measured. As a practical matter, the utility of repeated replications of a single treatment is sharply reduced by the need to manipulate as many variables as possible in the time allowed. However, intermittent replications can serve to benchmark drifting or trending data.

On several occasions, we repeated protocols within and across comparison groups in the Calling Card Service field trial. Replications afforded an increase in confidence by allowing an assessment of changes in dependent variables due to confounding of external and internal parameters with the independent variables. The replications, in effect, provided a basis for comparison. And the replications in-

creased confidence in the assertion that the observed changes in the dependent variable were caused by the deliberate change in the independent variables. For example, protocol 14 amidst protocols 12, 13, 15, and 16 (coin placarded stations) was a replication of an earlier protocol and supported the hypothesis that the poor performance of the other protocols was real and not an artifact of internal or external conditions.

Some replications were designed to provide more information about the contributing effects of certain service components. Most notably, major service changes were tested with both the tone-only and tone-and-announcement conditions. For example, protocol 13 (coin placard station) was the same as protocol 12, but without the prompt announcement. This is similarly true for protocols 15 and 16.

#### *4.2.6 Partial counterbalancing*

Arranging the treatment order in special ways—e.g., the Latin Square design—across comparison groups provides for pairwise comparisons of different treatment orderings. Such arrangements, however, are often not practical in field trials of telephone services because of the large number of variables and variable states (treatment conditions) that should be tested. The large number of treatment combinations required of typical counterbalancing techniques preclude their use. Manipulation of protocols on the basis of data gathered from prior manipulations also precludes planned counterbalancing.

But it may be possible partially to counterbalance treatment pairs across comparison groups in order to assess some multiple treatment order effects. If the manipulations evolve, the application of counterbalancing schemes have to be delayed by at least one time frame to have knowledge of the treatment pair.

#### *4.2.7 Post-manipulation static study*

After service manipulations are finished, it is useful to study the final service at rest. In this period in the Calling Card Service field trial (protocol 24), transient effects due to changes in the service were not present, thus improving predictions of the final service.

## V. CONCLUSION

Many service issues were resolved by the trial of Calling Card Service. Most important, the customers who tried the service continued to use it because they felt it was faster and more convenient than operator-assisted credit card calling. Moreover, the design of the service was critically dependent on the results of the field trial manipulations. Both announcements and instruction placards were found to be very effective in stimulating customers to dial. The field trial data

provided objective measurements of optimum customer usage and acceptance and lower rates of error and abandonment in the service offered. Table I lists the criteria found most useful in selecting the attributes of the final service.

When Calling Card Service was first offered in Buffalo in July 1980, a product follow-up evaluation study was conducted to monitor customer use, performance, and acceptance levels. The measurements of actual service conformed closely to estimates made from field trial data and, thus, support the utility of test methodology in guiding service design choices. The methodology developed for evaluating Calling Card Service is now being applied to the next generation of services, such as teleconferencing.

## VI. METHODOLOGY SUMMARY

The pre-field trial activities of analysis, interviews, and laboratory studies are undertaken to refine the service definition and assess the potential of its success. A study plan is constructed which defines objectives, resources and constraints, study variables, acceptance criteria, and study design. Field test requirements are developed that specify the data collection, analysis, and customer acquisition and preparation procedures as well as specify the hardware, software, and network aspects of a telephone service test vehicle.

Given adequate requirements the test is developed and implemented. During the test, the service is manipulated and data are collected and analyzed to optimize criteria of usage, performance, and satisfaction. Trade-offs are made relative to available resources, data validity, confidence in results, and time constraints. If success is predicted for the service being tested, results are then integrated into the design and the service is offered and monitored. Figure 13 summarizes the total process of conducting a field test of a telephone service.

Table I—Criteria for selecting service attributes

| Independent Variables | Dependent Variables |
|---|---|
| 1. Announcement Presence | a. Percentage Dialed<br>b. Abandons |
| 2. Announcement Wording | a. Percentage Dialed<br>b. Confusion (interviews)<br>c. Abandons |
| 3. Number of Attempts Allowed | a. Success Rate<br>b. Percentage of Subsequent Attempts |
| 4. Operator Access | a. Percentage Dialed<br>b. Abandons<br>c. Satisfaction (interviews) |
| 5. Interevent Timing | a. Distribution of Times to First Digit |
| 6. Interdigit Timing | a. Distribution of Pauses Between Digits |
| 7. Protocol (service in general) | a. Percentage Dialed<br>b. Satisfaction (interviews)<br>c. Success Rate |
| 8. Delays | a. Abandons |

Fig. 13—Design and evaluation process for human-machine telephone service protocols.

## VII. ACKNOWLEDGMENTS

I would like to thank E. C. T. Walker and C. A. Riley for their valuable comments on this manuscript. K. R. Hickey, R. J. Jaeger, N. S. Pearson, and J. L. Santee are thanked for their input as well. The Calling Card Service test depended heavily on the contributions of T. M. Bauer and E. A. Youngs.

## REFERENCES

1. M. R. Allyn, T. M. Bauer, and D. J. Eigen, "Planning for People: Human Factors in the Design of a New Service," Bell Lab. Rec., *58*, No. 5 (May 1980), pp. 55–161.
2. D. J. Eigen and E. A. Youngs, "Calling Card Service—Human Factors Studies," B.S.T.J., *61*, No. 7 (September 1982), pp. 1715–35.
3. D. T. Campbell and J. C. Stanley, *Experimental and Quasi-Experimental Design for Research*, Chicago: Rand McNally, 1966.
4. D. T. Cook and D. T. Campbell, *Quasi-Experimentation Design and Analysis Issues for Field Settings*, Chicago: Rand McNally, 1979.
5. H. M. Parsons, *Man-Machine System Experiments*, Baltimore: Johns Hopkins, 1972.
6. E. J. Webb et al., *Unobtrusive Measures: Nonreactive Research in the Social Sciences*, Chicago: Rand McNally, 1966.

## APPENDIX A

### *Field Evaluation Design Notation*

To provide another perspective on the field design methods, the following notation is provided.

#### *A.1 Service*

A service (or treatment) can be defined by set of dependent variables with specific values, represented here as a vector $\dot{Z}$.

$$\dot{Z} = \begin{vmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_n \end{vmatrix}$$

#### *A.2 Measurements*

Let 0 stand for observation. A matrix of $n$ kinds of measurement taken in $m$ different ways is represented by $\ddot{0}$.

$$\ddot{0} = \begin{vmatrix} 0_{11} & 0_{12} & \cdots & 0_{1m} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ 0_{n1} & 0_{n2} & \cdots & 0_{nm} \end{vmatrix}.$$

When multiple measures (taken multiple ways) are repeated, the notation is:

$$\ddot{O}^m = \ddot{O}_1 \cdots \ddot{O}_m .$$

### A.3 Improving service

Two aspects of the field study, static service evaluation and improving service, could be represented as:*

| Static Service Evaluation | | Improving Service |
|---|---|---|
| $\dot{O}_0^m \dot{Z}_0 \ddot{O}_1^m$ | $\dot{Z}_1$ | $\ddot{O}_2^m \cdots \ddot{O}_n^m \dot{Z}_n \ddot{O}_{n+1}^m$ |

This design effectively constitutes repetitions of the static service evaluation.

### A.4 Control group

| | |
|---|---|
| ${}_{A'}\ddot{O}_0^m \dot{Z}_0 {}_A\ddot{O}_1^m \dot{Z}_1 {}_A\ddot{O}_2^m \cdots {}_A\ddot{O}_n^m \dot{Z}_n {}_A\ddot{O}_{n+1}^m$ | Treatment Group |
| ${}_{A'}\ddot{O}_0^m {}_{A'}\ddot{O}_1^m {}_{A'}\ddot{O}_2^m \cdots {}_{A'}\ddot{O}_n^m {}_{A'}\ddot{O}_{n+1}^m$ | Control Group |

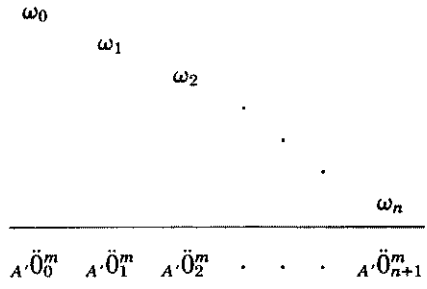
### A.5 Staggered treatment groups

New subjects are added (or are available) at each treatment change. Limited resources usually necessitate the continued manipulation of the variables within treatment groups. The line offsets the control group from the treatment groups and denotes unequivalence of the groups.

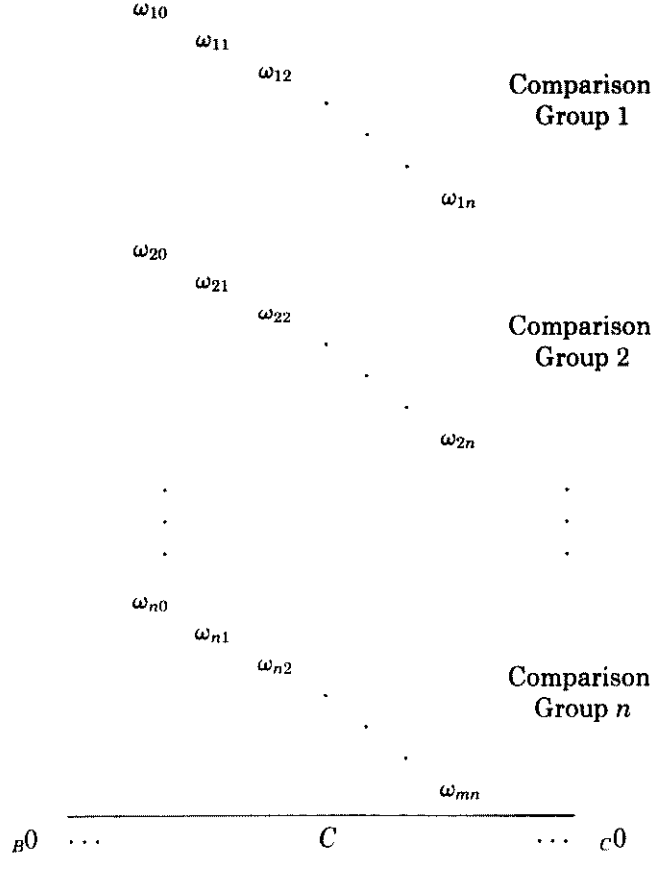
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ${}_{A'}\ddot{O}_{10}^m$ | $\dot{Z}_0$ | ${}_{A'}\ddot{O}_{11}^m$ | $\dot{Z}_1$ | ${}_A\ddot{O}_{12}^m$ | $\cdots$ | ${}_A\ddot{O}_{1n}^m$ | $\dot{Z}_n$ | ${}_A\ddot{O}_{1n+1}^m$ | Treatment Group 1 |
| | | ${}_{A'}\ddot{O}_{21}^m$ | $\dot{Z}_1$ | ${}_A\ddot{O}_{22}^m$ | $\cdots$ | ${}_A\ddot{O}_{2n}^m$ | $\dot{Z}_n$ | ${}_A\ddot{O}_{2n+1}^m$ | Treatment Group 2 |
| | | | | | | ${}_{A'}\ddot{O}_{nn}^m$ | $\dot{Z}_n$ | ${}_A\ddot{O}_{nn+1}^m$ | Treatment Group $n$ |
| ${}_{A'}\ddot{O}_{n+10}^m$ | | ${}_{A'}\ddot{O}_{n+11}^m$ | | ${}_{A'}\ddot{O}_{n+12}^m$ | $\cdots$ | | | ${}_A\ddot{O}_{n+1n+1}^m$ | Control Group |

* $\dot{Z}_i$ denotes the vector of service-independent values for the $i$th service change (treatment). ${}_A\ddot{O}_i^m$ denotes the matrix of multiple measurements by multiple methods measured $m$ times for the $i$th service (treatment) change. The left-hand $A'$ subscript (versus $A$ subscript) denotes that some measurements present in the trial may not be available in the pretrial measures (or in the control group).

To reduce the notational burden, let $\omega_i$ denote the $i$th staggered treatment group.




### *A.6 Staggered treatment groups with comparison groups*




Note that the Control Group, $C$, is augmented with historical and follow-up measurements.

### A.7 Replications

Replications are denoted by the appropriate repetition of protocol time frame subscripts on the independent variable vector:

$$ {}_{A'}\ddot{O}_0^m \boxed{\dot{Z}_0} {}_{A'}\ddot{O}_1^m \dot{Z}_1 {}_{A}\ddot{O}_2^m \boxed{\dot{Z}_0} {}_{A}\ddot{O}_3^m \dot{Z}_2 {}_{A}\ddot{O}_4^m \cdots \dot{Z}_n {}_{A}\ddot{O}_p^m. $$

### A.8 Partial counterbalancing

$$ \begin{array}{llll} \boxed{{}_{A'}\ddot{O}_{101}^m \dot{Z}_0 {}_{A}\ddot{O}_{111}^m \dot{Z}_1} & \ddot{O}_{121}^m \dot{Z}_2 \cdots & \Big\} & \text{Comparison Group 1} \\ {}_{A'}\ddot{O}_{211}^m \dot{Z}_1 & {}_{A}\ddot{O}_{221}^m \dot{Z}_2 \cdots & & \\ & {}_{A'}\ddot{O}_{321}^m \dot{Z}_2 \cdots & & \end{array} $$

$$ \begin{array}{llll} {}_{A'}\ddot{O}_{102}^m \dot{Z}_0 \; {}_{A'}\ddot{O}_{112}^m \dot{Z}_1 & {}_{A}\ddot{O}_{122}^m \dot{Z}_0 \cdots & \Big\} & \text{Comparison Group 2} \\ \boxed{{}_{A'}\ddot{O}_{212}^m \dot{Z}_1} & {}_{A}\ddot{O}_{222}^m \dot{Z}_0 \cdots & & \\ & {}_{A'}\ddot{O}_{322}^m \dot{Z}_0 \cdots & & \end{array} $$

The two boxed groups are the time-delayed counterbalanced treatment pairs $\dot{Z}_0$ and $\dot{Z}_1$.

## AUTHOR

**Daryl J. Eigen,** B.A. (Psychology), 1972, M.S. (Electrical Engineering), 1973, University of Wisconsin; Ph.D. (Industrial Engineering), 1981, Northwestern University; Bell Laboratories, 1973—. Mr. Eigen initially worked in the Human Performance Technology Center. He then was involved in feature and service planning for the Traffic Service Position System and, later, the No. 4 ESS. He is currently Supervisor of the System Analysis and Human Factors Group for No. 4 ESS. Member, IEEE, APA, Human Factors Society, and Tau Beta Pi.

# Characteristics of Human Performance

Vigorous efforts to improve understanding of underlying processes in human behavior have been made at Bell Laboratories. Current interests include most major aspects of human information processing and performance, including perception, thinking, movement control, and learning. This section includes samples of recent work on fundamental mechanisms in visual pattern recognition (Julesz and Bergen), the timing of repetitive actions (Rosenbaum), the perceptual interpretation of graphical displays (Cleveland, Harris, and McGill), and individual variations in modes of mental problem solving (Egan). While these four papers by no means exhaust the problem areas under investigation, they do give a fair picture of the wide range of issues, methods, theory, and data involved, and the diverse domains of knowledge being enriched.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# Textons, The Fundamental Elements in Preattentive Vision and Perception of Textures

By B. JULESZ* and J. R. BERGEN*

(Manuscript received September 23, 1981)

Recent research in texture discrimination has revealed the existence of a separate "preattentive visual system" that cannot process complex forms, yet can, almost instantaneously, without effort or scrutiny, detect differences in a few local conspicuous features, regardless of where they occur. These features, called "textons", are elongated blobs (e.g., rectangles, ellipses, or line segments) with specific properties, including color, angular orientation, width, length, binocular and movement disparity, and flicker rate. The ends-of-lines (terminators) and crossings of line segments are also textons. Only differences in the textons or in their density (or number) can be preattentively detected while the positional relationship between neighboring textons passes unnoticed. This kind of positional information is the essence of form perception, and can be extracted only by a time-consuming and spatially restricted process that we call "focal attention". The aperture of focal attention can be very narrow, even restricted to a minute portion of the fovea, and shifting its locus requires about 50 ms. Thus preattentive vision serves as an "early warning system" by pointing out those loci of texton differences that should be attended to. According to this theory, at any given instant the visual information intake is relatively modest.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

## I. INTRODUCTION

In this article we give an overview of some insights into the workings of the human visual system gained during two decades of research at Bell Laboratories, and culminating in the discovery of a few local conspicuous features that we call textons. Textons appear to be the basic units of preattentive texture perception,[1] when textures are viewed in a quick glance with no further effort or analysis. Although this article goes beyond texture perception into preattentive vision in general, studies of texture discrimination led to the basic insights presented here and provide excellent demonstrations of the main findings. Based on our findings we propose a novel theory of vision in which the preattentive visual system inspects a large portion of the visual field in parallel and detects only density differences in textons. It then directs focal attention to these loci of texton differences for detailed scrutiny.

Now, after 20 years of research, when we know what textons are and their role in vision is clarified, we can save the reader from following the rather difficult steps that led to their discovery. [The reader interested in the history of these developments, and in the sophisticated mathematics necessary to generate textures with certain stochastic constraints, should turn to the original articles referred to in a recent review by one of us[1] and to the Appendix.] Here we follow an axiomatic treatment. The main findings are presented in Section II as heuristics (similar to axioms, but not necessarily totally independent), immediately followed by many demonstrations. The reader can test the power of these newly acquired heuristics by being able to predict and then verify which texture pairs will be perceived to be different, and which will appear as a single texture. The reader can thus understand the new theory of vision without mathematical knowledge.

Section III emphasizes the essentially local nature of texture perception. Section IV relates the psychologically identified textons to some neurophysiological results concerning local feature analyzers in primate cortex. Section V extends the texton theory from texture perception to the discrimination of briefly presented patterns. In Section VI a model of human vision is proposed that postulates two different modes of visual system function. Section VII discusses some implications of this model.

## II. HEURISTICS: DEFINITION OF TEXTONS AND THEIR INTERACTIONS IN PREATTENTIVE VISION

Visual textures are defined as aggregates of many small elements. The elements can be either dots of certain colors (e.g., black, white, grey, red) or simple patterns. For purposes of this article, we consider

only elements that do not overlap, and are placed at either regular or random positions, in identical or in random angular orientations.

Usually in our demonstrations two textures (composed of two different elements) are placed side-by-side, or one is embedded in the other, as shown in Fig. 1. When the reader cursorily inspects Fig. 1, an area made up of +'s will appear to stand out from the surrounding texture composed of L's. Indeed, without scrutiny, that is without detailed element-by-element inspection, the reader might not notice that a third area composed of T-shaped elements is also embedded in the texture of L's. We call this effortless perceptual segregation of the texture composed of +'s from the surrounding texture of L's *preattentive texture perception.* On the other hand, if texture discrimination requires element-by-element scrutiny, as is the case of finding the T's in the L's, we call this way of looking with scrutiny *focal attention.* We will show many other preattentively indiscriminable texture pairs (e.g., Figs. 3c and 6b), which, because they do not segregate, often are not even perceived as containing different elements until this is pointed out.

Although in all texture perception the preattentive system is dominant, the role of focal attention can be even further reduced by brief presentation. The reader who is not convinced by the qualitative difference between preattentive and attentive texture discrimination

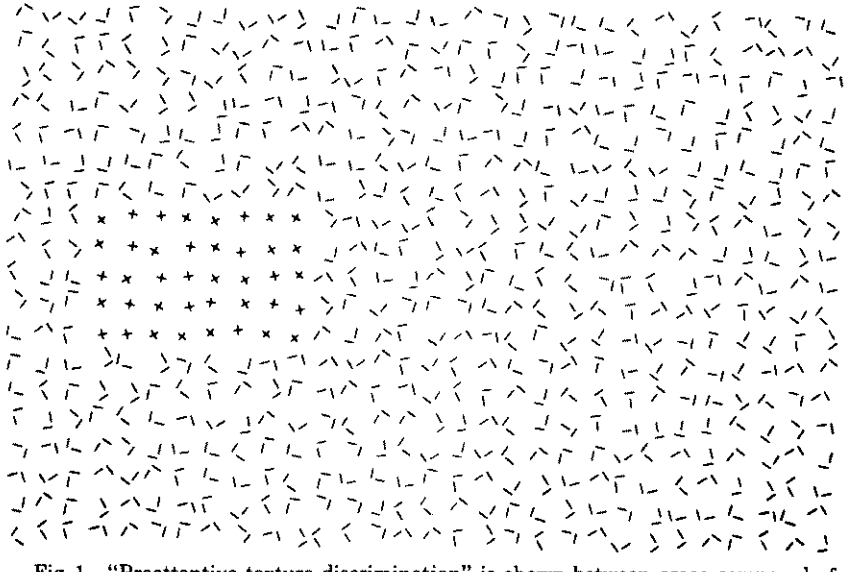


Fig. 1—"Preattentive texture discrimination" is shown between areas composed of +'s and L's, while element-by-element scrutiny, called "focal attention" is required to find the T's embedded in the L's.

might inspect Fig. 1 through a camera shutter set at 1/10 second exposure time.

*Heuristic 1: Human vision operates in two distinct modes*

1. Preattentive vision—parallel, instantaneous, without scrutiny, independent of the number of patterns, covering a large visual field, as in texture discrimination.
2. Attentive vision—serial search by focal attention in 50-ms steps limited to a small aperture, as in form recognition.

*Heuristic 2: Textons*

1. Elongated blobs—e.g., rectangles, ellipses, line segments with specific colors, angular orientations, widths, and lengths.
2. Terminators—ends-of-line segments
3. Crossings of line segments

*Heuristic 3: Preattentive vision directs attentive vision to the locations where differences in the density (number) of textons occur, but ignores the positional relationships between textons.*

Before we discuss the implications of these heuristics, let us apply them to a few pairs of elements and predict whether the texture pairs formed from these elements will yield preattentive texture discrimination or not. This application of the rules also helps to clarify them. For instance, elongated blobs of different widths or lengths are different textons, as Fig. 2a demonstrates. The larger sized R's containing longer and wider line segments form a texture that segregates (i.e., is preattentively discriminable) from its surround, which is composed of smaller R's with shorter and narrower line segments.

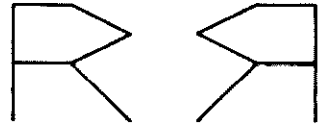
Similarly, elongated blobs of different orientations are different

(a) (b)

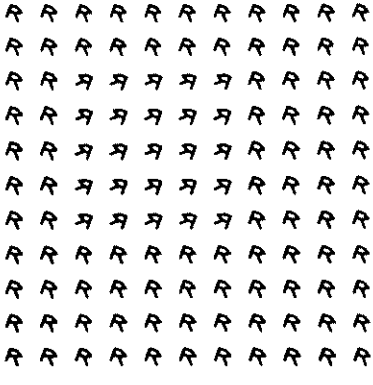
Fig. 2—Preattentive texture discrimination based on texton differences between line segments of (a) length and width and (b) angular orientation. (*Nature*, March 12, 1981[1])

textons. Indeed, in Fig. 2b the texture pair composed of the same sized R's having two different orientations in the two textures, yields preattentive discrimination. Obviously, the same elongated blob shape with the same orientation yields different textons if the colors (e.g., black, gray, white, red, green, etc.) are different.

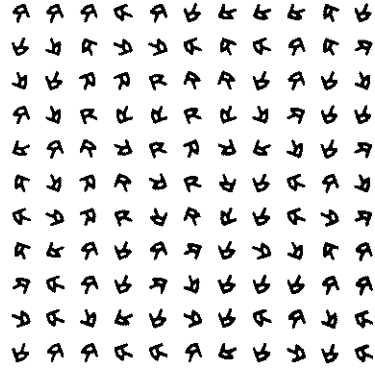
Now, let us predict what would happen if we took an R and a mirror-image R, as shown in Fig. 3a, and formed a texture pair by throwing them in random orientations. Obviously, without randomizing the orientations, the two textures would yield texture discrimination, since even though their widths and lengths agree, some of the line segment textons have different orientations in the R and in its mirror image, though the widths and lengths agree, as shown in Fig. 3b. However, if the two elements are thrown at random orientations, then the two textures formed have the same average density of textons (i.e., in some area of integration the number of line segments with the same color, width, length, and orientation is identical). Therefore, the preattentive visual system should not be able to direct focal attention to loci of texton differences that form the boundary between the two regions. Indeed, an inspection of Fig. 3c yields a single, uniform texture. It requires laborious element-by-element inspection for several seconds



(a)



(b)



(c)

Fig. 3—Demonstration of how the heuristics given in text predict why (a) R and its mirror image in aggregates yield texture discrimination (b), or are indistinguishable (c). (*Perception*, 1973[3])

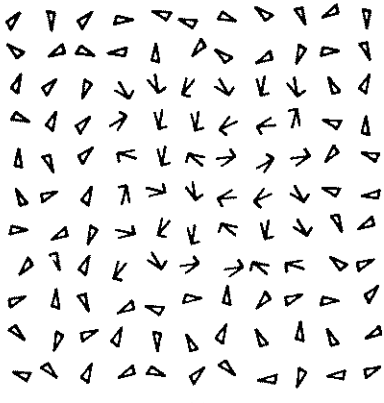
to find the boundary between the array of R's and mirror-image R's. Obviously, in a 100-ms presentation discrimination of these textures is impossible.

Let us note that if one were to select a pair of elements without knowing the rules given above, most probably the resulting texture pair would be discriminable. Only through the joint effort of our colleagues (D. Slepian, M. Rosenblatt, E. Gilbert, L. Shepp, H. Frisch, T. Caelli, and J. Victor) from 1962 to 1978 were some elegant methods found that yielded indistinguishable textures, even though their elements appeared very different.

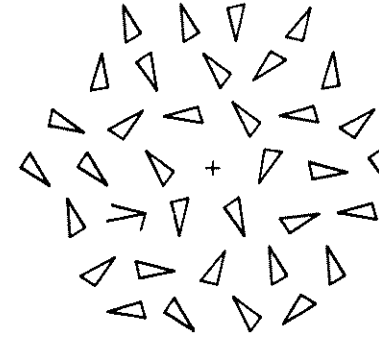
In the next examples we stress the importance of terminator textons. For instance, in Fig. 4a the two elements are composed of three identical line segments (i.e., same orientation, width, and length). The only difference is in the number of their ends-of-lines (terminators). The triangle-shaped element has no open ends, while the "dual" element has three ends-of-lines. One should expect texture segregation, given such a large difference in terminator number, and as Fig. 4b demonstrates, this is the case.

As a matter of fact, discrimination is so strong that a single element can be preattentively detected among 35 dual elements, as shown in Fig. 4c. This arrangement is now routinely used by us in studying

(a)



(b)

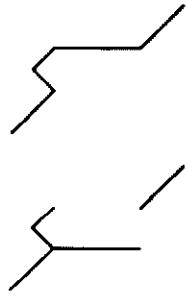


(c)

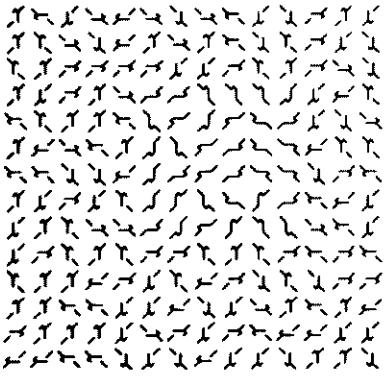
Fig. 4—Demonstration of how the heuristics given in text predict preattentive texture discrimination (b) and even discrimination of a single element among many (c), based on terminator number difference (zero versus three) between elements (a). (*Nature*, March 12, 1981[1])

pattern discrimination in preattentive vision, as discussed in Section V. Here we note only that when there is a texton difference (as in Fig. 4c) detecting one element in the midst of 35 other elements is almost as easy as detecting the difference between two elements (shown in Fig. 4a) for presentation times as brief as 100 ms.

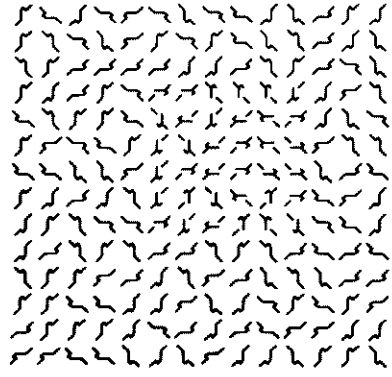
In the next example, both members of the element pair of Fig. 5a are again composed of the same five line segments (each corresponding line segment in the two elements has identical width, length, and orientation, respectively) but one element contains only two ends-of-lines, whereas the other contains five. This large difference in terminator numbers should yield texture segregation, and inspection of Fig. 5b demonstrates that it does. Figure 5c consists of the same texture pair as Fig. 5b, except that the texture containing the five terminators is now the surround. Although, as predicted, the large difference in terminator numbers again yields texture segregation, the appearance of the boundary between the two regions is different for Fig. 5b and 5c.



(a)



(b)



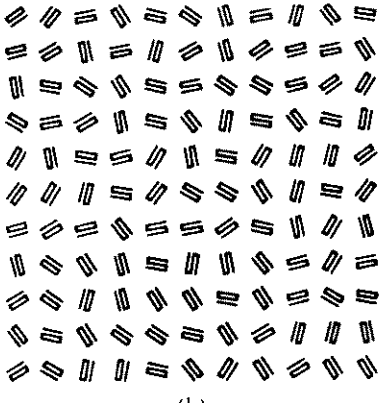
(c)

Fig. 5—Similar to Fig. 4 except the terminator number between elements is two versus five.

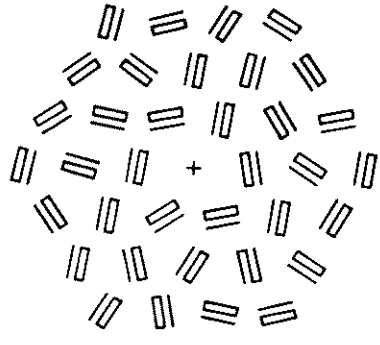
The next example, shown in Fig. 6a, consists of the "S"- and "10"-shaped elements, that in isolation appear quite different. However, the two contain the same number of line segment textons (three identical horizontal and two identical vertical line segments) and both contain two ends-of-lines. The fact that the positional relationship between these textons is different (as it is in Fig. 3b) can be perceived only by the attentive visual system (yielding the percept of an S versus a 10). However, according to Heuristic 3 the preattentive system can count only the density (number) of textons and ignores their relative positions. So, according to our rules, a texture pair composed of these elements contains the same average density (number) of textons, and thus should be indistinguishable. Surprising as it may seem, the texture pair is indeed preattentively indistinguishable as demonstrated by Fig. 6b. [Readers who find this demonstration of the distinction between preattentive and focal vision not adequately convincing without brief presentation should note the contrast between the attentively different percepts of Fig. 6a, and the texture pair in Fig. 6b, which remains difficult to distinguish even with element-by-element scrutiny.]

Finally, let us demonstrate the third texton, the crossing of elongated blobs (line segments). Figure 7a shows the conspicuous difference between a texture pair that segregates based on the presence or absence of elements having crossing versus not-crossing line segments.

(a)



(b)



(c)

Fig. 6—Demonstration of how the heuristics given in text predict why (a) the differently appearing S- and 10- shaped elements in aggregates (b) and one S in 10's (c) are indistinguishable. (*Nature*, March 12, 1981[1])

If the elements have identical textons, including crossing (or not-crossing line segments) the texture pairs become preattentively indistinguishable. The positional relationship between the line segment textons is unnoticed by the preattentive system. The difference in gap size between the L-shaped elements in Fig. 7b yields a preattentively indistinguishable texture pair. Particularly interesting is the demonstration in Fig. 7c where T- versus L-shaped elements yield an indistinguishable texture pair. Although we have kept a small gap between the perpendicular line segments that make up the L's and T's, preattentive discrimination of texture pairs composed of these elements is impossible even when the gaps are not resolvable. Apparently, the difference of a single end-of-line terminator is not adequate to yield texture segregation. Finally, Fig. 7d depicts a preattentively indistinguishable texture pair, where, with scrutiny, it is obvious that the

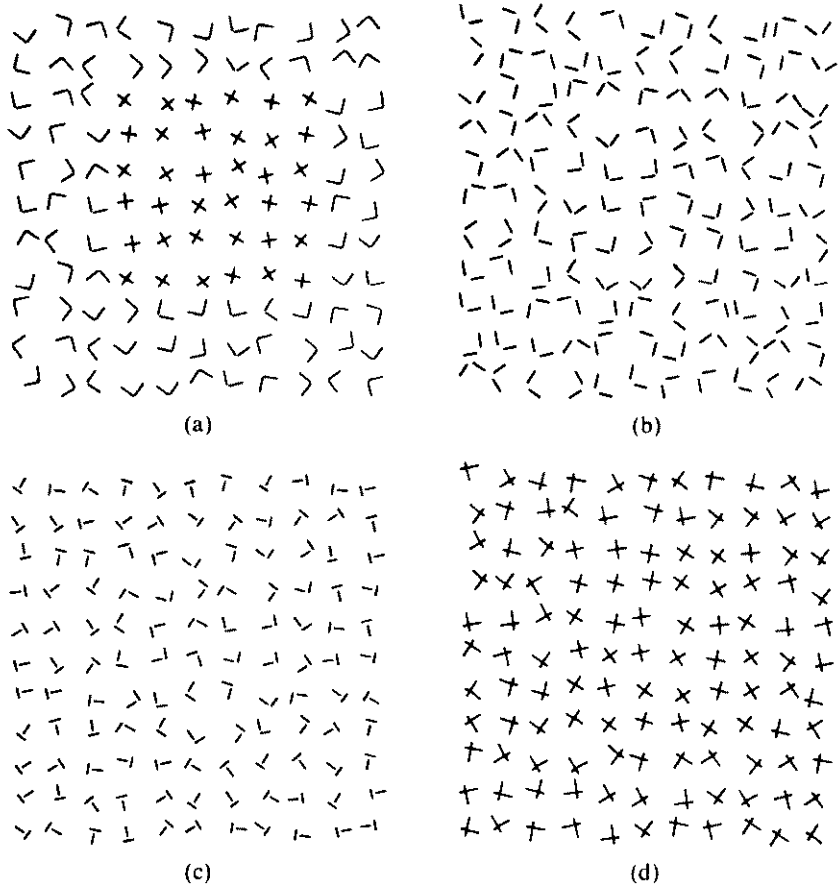



Fig. 7—Demonstration that crossing of line segments is a texton.

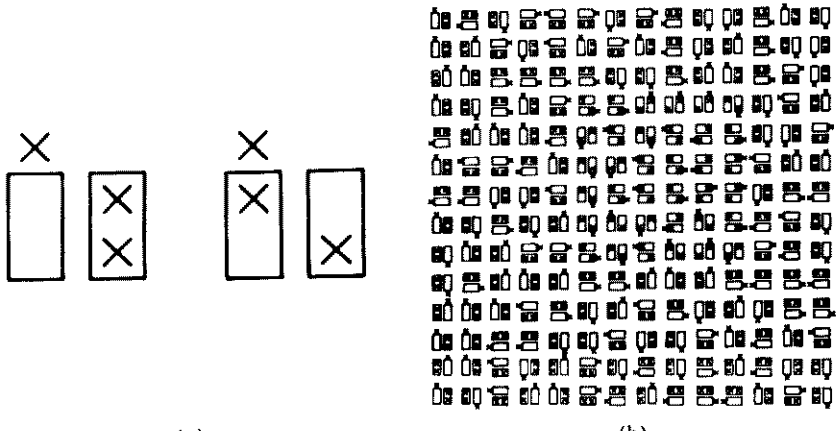
elements contain line segments that either cross at midpoint or cross far from the midpoint.

The last two examples are given in Figs. 8a and b and Figs. 9a, b, and c. From the element pairs containing the same textons, the reader can predict that although their elements in isolation appear very different, the resulting texture pairs will be indistinguishable.

In all these demonstrations the texture elements consisted of line segments. For line segments the definition of terminators (ends-of-lines) and their crossings are straightforward. For elongated bars with substantial width these definitions are less direct. Particularly difficult is the notion of terminators, because instead of terminators some combination of white elongated bars in a black surround with black elongated bars in white surround might suffice. So, we are not certain whether terminators are independent textons. Nevertheless, as a first approximation these three heuristics work remarkably well.

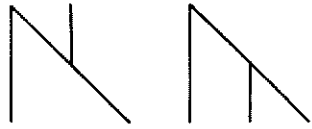
## III. PREATTENTIVE TEXTURE PERCEPTION IS ESSENTIALLY A LOCAL PROCESS

The essence of all the findings reported in the previous section can be summed up as follows: In texture perception the preattentive visual system utilizes only local conspicuous features, textons, and these textons are not coupled to each other (i.e., a vertical and horizontal line segment do not cohere to form an L or T). The preattentive system utilizes globally only the textons in the simplest possible way by counting their numbers (densities). This might surprise many of our readers who assume that texture perception utilizes complex global statistical interactions between textural elements.

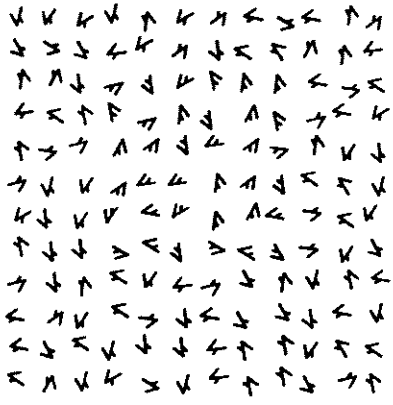


(a) (b)

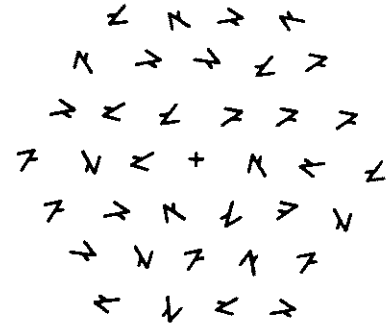
Fig. 8—Since the element pair in (a) is composed of the same textons, the texture pair (b) composed of these elements is preattentively indistinguishable. (*Philosophical Transactions*, 1980[25])



(a)



(b)

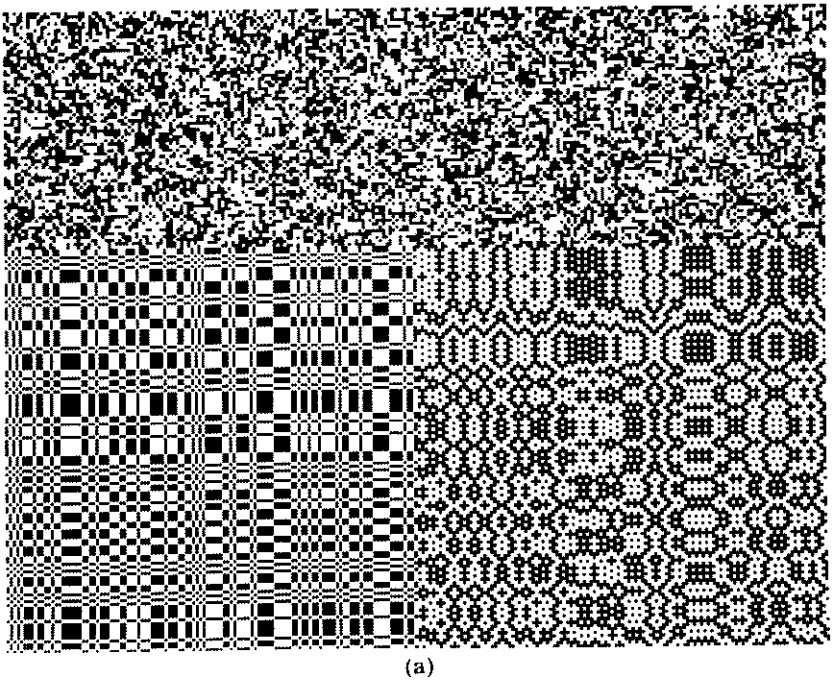


(c)

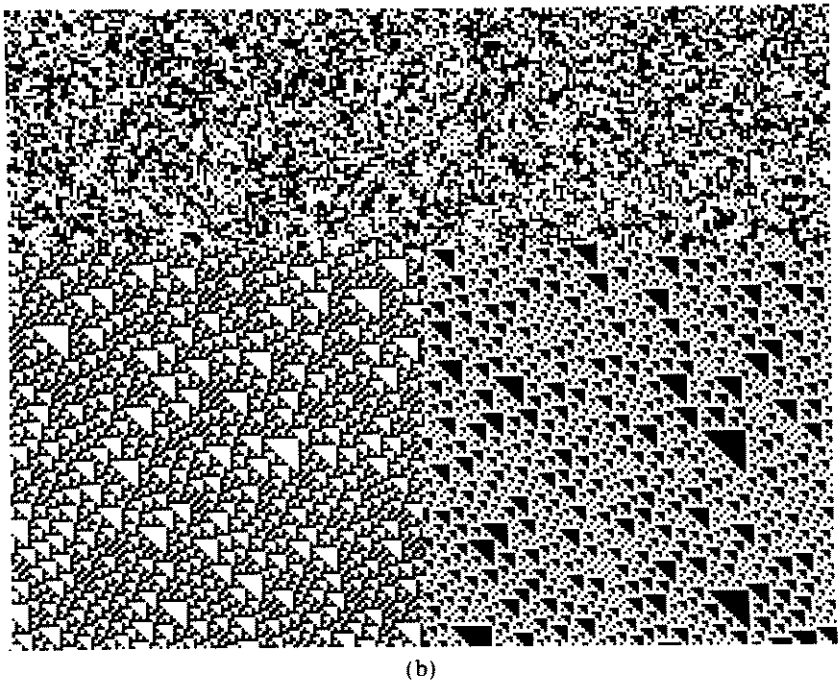
Fig. 9—Similar to Fig. 8, showing that aggregates of elements composed of the same textons cannot be preattentively discriminated. (*Philosophical Transactions*, 1980[25])

One of the simplest *global* computations routinely performed on images by engineers, and recently by psychologists in vision research, is to determine the images' Fourier power spectra. This process involves the decomposition of the images into one-dimensional sinusoidal luminance gratings whose specific amplitudes, spatial frequencies, phases, and angular orientations depend on the spatial characteristics of luminance distributions across the entire image. The amplitude of the spectral components ignoring phase determine the power spectra. When Fourier power spectra of textures are taken, it is a common misconception that differences in these will reveal differences in texture granularity. That the preattentive visual system does not perform Fourier analysis is demonstrated next.

Figure 10a consists of three areas that have identical Fourier power spectra (invented by Julesz, Gilbert and Victor[2]) and yet appear as very distinct textures. [The mathematically sophisticated reader might appreciate that the three areas have identical third-order statistics, and differ only in their fourth-order statistics. Those interested in the definition of *n*th-order statistics should consult Refs. 3 and 4 and the Appendix.] Figure 10b also consists of three areas with identical Fourier power spectra, and again these areas appear conspicuously different. Conversely, in Fig. 11a the lower left quadrant of the bottom



(a)



(b)

Fig. 10—Discriminable texture pairs with identical Fourier power spectra (*a* has even identical third-order statistics[1] based on local granularity (texton) differences. (*Biological Cybernetics*, 1981[2])

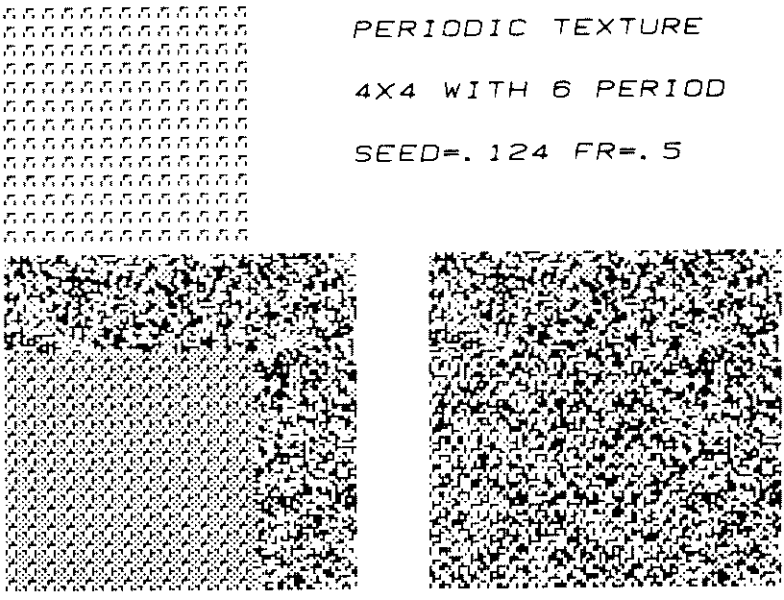
right array has a very different power spectrum from the remainder of the array, yet no preattentive texture discrimination results.[5] The derivation of this texture pair is presented in three steps. The top left array, in Fig. 11a, consists of 4x4 dot elements (8 black and 8 white) with 6-dot periodicity. The bottom left array contains this periodic array in one quadrant, but the 2-dot-wide gaps are filled by a checkerboard screen, while the rest is covered with uniformly random black and white dots. The bottom right array is similar to the bottom left array, except the 2-dot-wide gaps between the periodic patterns are now randomly speckled with dots. Obviously, the periodic patterns in the lower quadrant of the bottom right array in Fig. 11a yield a very different Fourier power spectrum from the rest, which has a flat (white noise) spectrum. The reason that this texture pair is indistinguishable can be easily understood in the light of the texton theory. The periodic patterns are not different from the surrounding random-dot array in the density of elongated blob textons, and therefore are indistinguishable. Indeed, if the 4x4 dot micropattern consists of vertical stripes, which contain textons different from the surrounding random-dot array, as shown in Fig. 11b, the periodic quadrant embedded in randomness is easily perceived.

In all these densely packed dot textures, discrimination is based on local granularity differences that correspond to differences in the density (number) of elongated blobs of certain sizes and orientations. Global statistical descriptors of textures, including the Fourier power spectrum, apparently are ignored in preattentive vision.

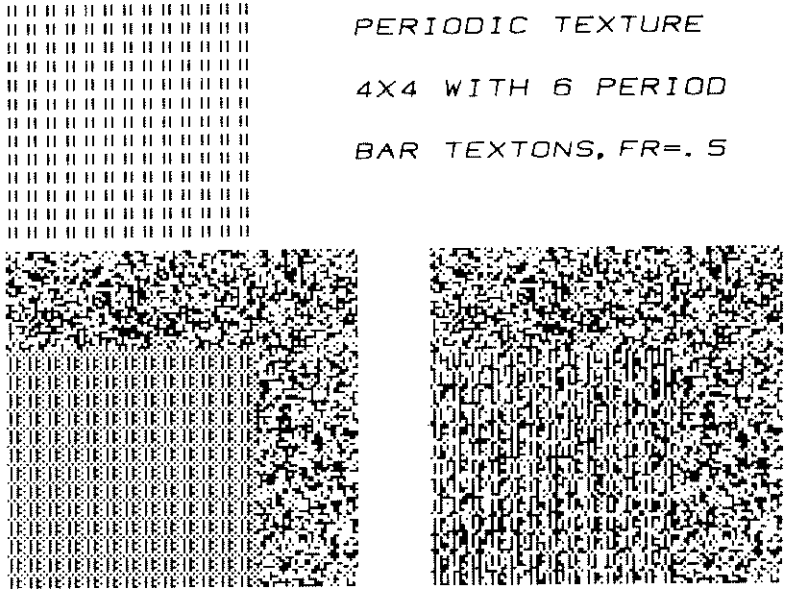
## IV. TEXTURES AND NEUROPHYSIOLOGICAL FEATURE ANALYZERS

We have seen how elongated blob textons are crucial in preattentive texture perception. These human psychological findings have a parallel in primate neurophysiology. Neural units have been found by Hubel and Wiesel[6] in the visual cortex of monkeys that fire optimally for elongated blobs of specific width, length, and orientation. These neural units in the cortex have *retinal receptive fields* consisting of elongated, blob-shaped, excitatory regions, which are surrounded by inhibitory regions. Some of these elongated blob detecting units—which fire optimally for black elongated blobs surrounded by white flanking areas—are called simple "off" detectors. Other neural units are excited optimally by white elongated blobs surrounded by black. These are called simple "on" detectors. The exact shape of the receptive fields of these simple neural units varies a great deal, and is of secondary importance. The important property of these cortical units is that the weighting of the excitatory and inhibitory areas of their receptive fields is about equal, so that for homogeneous stimuli they do not fire.

It should be stressed that the textons reported here were found by




(a)



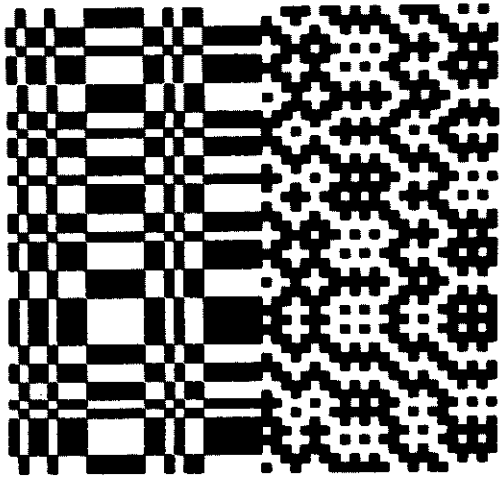

(b)

Fig. 11—Demonstration that the preattentive system cannot perform even such a simple global computation as Fourier power spectra, as described in text. (*Biological Cybernetics*, 1978[5])

psychological methods, and imply that simple neural units found as early as the striate cortex of the monkey might be used in texture perception. However, the relationship between a texton—for example, a perceived line segment—and a Hubel and Wiesel type of neural feature analyzer with a receptive field whose excitatory center matches the shape of the line segment is not a simple isomorphism. As we pointed out years ago (Ref. 7, p. 3), a single simple neural unit might respond equally for a broad line of high contrast or a narrow line of low contrast, while perceptually one can preattentively perceive both the width and contrast of a line segment. Thus, obviously a perceived line segment is encoded by many neural units of similar orientations but tuned to different widths, and having different firing thresholds. It is some combination of these units that would correspond to a perceived line segment. Until more is known about the relationship between perception and neurophysiology, the textons must be defined as perceptual entities, that is conspicuous local features as we actually perceive them. Nevertheless, even though textons and neural units are not simply related, one can easily conceptualize how a "perceptual analyzer" could be built from known neural analyzers that could extract, say, a line segment texton. The question of whether terminators and crossings of line segments—which have been regarded as textons—could be related to the complex and hypercomplex neural analyzers found by the neurophysiologists remains to be seen.[6]

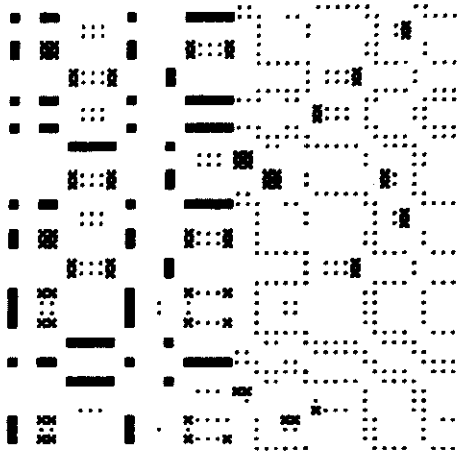
David Marr, in his primal-sketch model of machine vision, also incorporated such elongated blob detectors, by assuming that the neurophysiological findings had direct relevance to vision.[8] The work reported here followed an opposite trend. It took almost two decades to find evidence for the utilization of simple cortical units in texture perception. Caelli and Julesz found the first elongated blob textons that could account for texture discrimination locally, when all global statistical properties of the texture pairs were kept identical.[9] Later demonstrations such as Figs. 10a and b illustrate even more strikingly the importance of local blob textons.

To demonstrate the possible role of the Hubel and Wiesel type of neural units in preattentive texture perception, we developed a computer program called TEXTONS that filters any image with a pool of elongated bar-shaped receptive fields. Each pool of filters consists of "on" and "off" types having the same width, length, orientation, and firing threshold and placed at each point of the array. Figure 12 (bottom) shows the three largest response levels of a pool of 3x3 dot square-shaped receptive fields as this pool processes the texture pair of Fig. 10a, shown also in Fig. 12 (top). These filters have 2x2 dot excitatory centers flanked by one-dot-wide inhibitory margins, as shown in Fig. 12 (right). Of course, there are several pools consisting



(a)

(b)



(c)

Fig. 12—Automatic texture segregation, shown in (c), by applying a texton filter (b) to texture pair (a) (also shown in Fig. 10a).

of filters having receptive fields with some other dimensions and orientations that would be even more effective in segregating the two textures of Fig. 12 (top). Here we stress again that the combination of several filters would be required to yield the best texture segregation, corresponding to human texture discrimination. This combination of filters would correspond to a texton detector.

What our psychological findings show, however, could not have been

guessed by physiologists and theoreticians of artificial intelligence. In preattentive texture perception the various textons are not coupled, that is their relative positions are ignored. T- and L-shaped pairs of line segments cannot be discriminated preattentively in textures. Marr thought that elongated blobs and terminators would form some higher molar unit, which he called place tokens.[8] However, in preattentive vision no such higher interactions are found; the textons appear to be independent of each other.

## V. EXTENSION OF THE TEXTON THEORY TO RAPID PATTERN DISCRIMINATION

The success of the texton theory in predicting phenomena of texture perception is the result of the spatial complexity of the patterns. This complexity over a large area exceeds the capacity of focal attention and thus allows the preattentive system to dominate. This same deemphasis of focal attention can be achieved in simpler patterns by very brief presentation. We will show that under these conditions the same texton theory can be applied.[10]

Because brief temporal presentation is required, the stimuli used in these experiments can be produced only in the laboratory. Consequently, they cannot be demonstrated as the texture discrimination results have been. Thus, in this section we present the main findings as curves describing observers' performance.

The stimuli used in these experiments are shown in Fig. 13. In Fig. 13a there are 35 T's and one L arranged on a hexagonal grid with slight random positional jitter added. In Fig. 13b the T's have been replaced by + elements, and in Fig. 13c only two of the 36 possible positions actually contain an element. In all cases, a disk surrounding the central fixation marker is kept empty. Stimuli of this type are presented for 40 ms, followed by a blank interval of variable duration and a 40-ms erasing field. This erasing field consists of elements, which are the union of the two being discriminated, arranged in the same way as the test field. Use of this erasing technique allows restriction of the inspection interval to times shorter than the duration of the retinal afterimage. The times used are all too short to allow eye movements to be initiated during the presentation. In half of the presentations the test field consists of all identical elements, while in the other half one element is different, as in the examples shown. The task of the observer is to discriminate between these two conditions.

Results obtained using the three stimuli of Fig. 13 are shown in Fig. 14. On the abscissa is the time in milliseconds between the onsets of the test and erasing fields, or the stimulus onset asynchrony (SOA). On the ordinate is the percentage of correct discrimination. The results

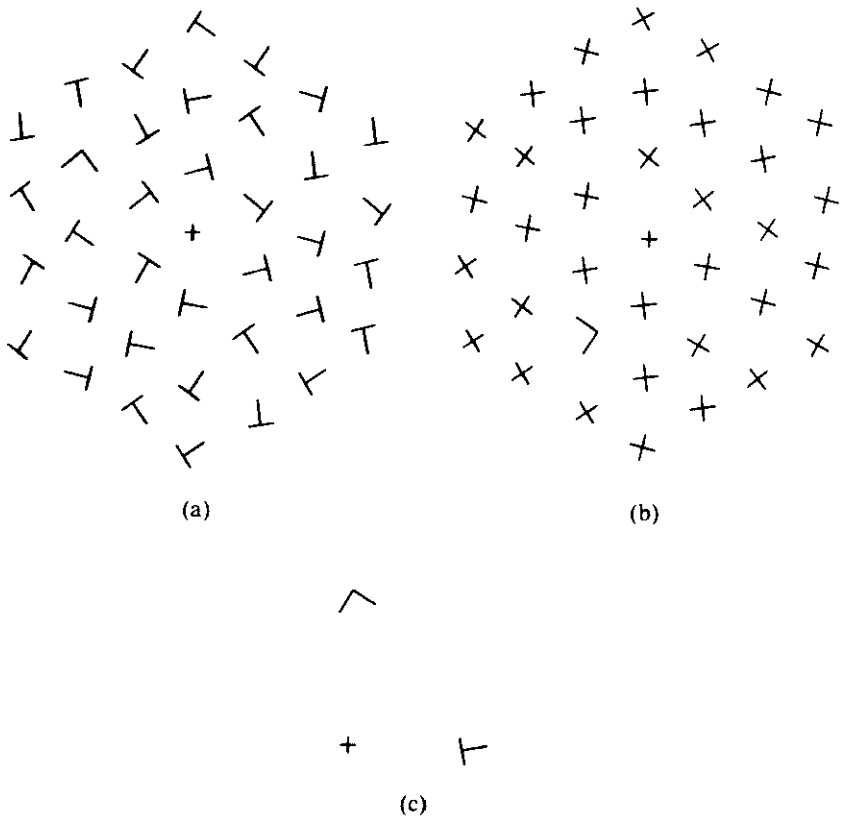



Fig. 13—The three types of stimuli used to obtain the results of Fig. 14. In all cases the task is to discriminate these from a control stimulus in which all elements are identical.

are very different for the case in which the elements share the same textons (T vs L. solid circles) from that in which they contain different textons (+ vs L, open circles). Note that in the T vs L case, not only does performance never exceed 65 percent correct, but it takes over 300 ms to reach this asymptote, while in the + vs L case the asymptote is reached within 200 ms. In fact, by the time the asymptote in the T vs L case is reached, the afterimage resulting from the test flash has largely disappeared. In the case in which only one T and one L are presented (filled squares), the results closely resemble those for 35 +'s and one L. Perceptually, the difference between the same-textons and different-textons cases is simply that the L in the field of T's stands out almost as if presented alone on a blank field, while the same L in a field of T's must be sought out. Attention is rapidly shifted to the L in the former case, while in the latter the search process is apparently still going on after 300 ms have passed. When there are only two elements to choose from, this search time is very brief.

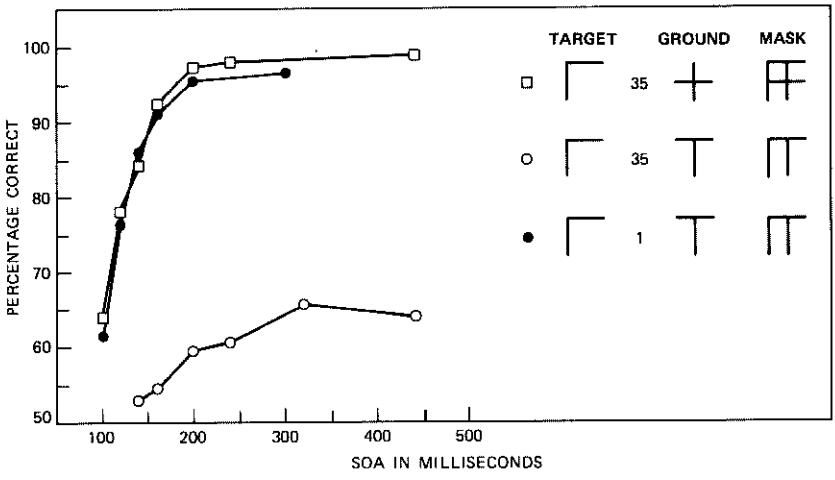



Fig. 14—Results of discrimination experiments using Fig. 13. The open circles, filled circles, and squares correspond to stimuli of Fig. 13a,c,b, respectively. SOA is the abscissa, and percent correct discrimination is the ordinate.

It is interesting to note that the observed asymptotic level of about 65 percent correct is what would be expected if seven or eight of the possible positions could be searched in the time available. Combining this number with the afterimage persistence time of 300–400 ms[11,12] gives a figure of about 50 ms per position inspected.[10]

This process of sequential inspection seems to be essentially independent of the overall angular subtense of the stimulus. Figure 15 shows results from an experiment in which the observer is required to distinguish a stimulus consisting of six T's and one L, or vice versa, from one in which all elements are identical. The stimulus was uniformly contracted so as to fall entirely within the fovea (<3 degrees across), or dilated to extend almost 14 degrees across, with no systematic variation in performance.

Another way of describing this is to say that the measurements are independent of the distance from which the stimulus is viewed, assuming that all of the elements remain resolvable: This independence suggests two important points. First, the fovea is not better than the near periphery in the extraction of this type of visual information. Second, the aperture of attention changes its spatial scale according to the size of the feature being sought. Thus, the same number of sequential fixations of attention are needed when the stimulus is reduced in size uniformly, because the sizes of the features upon which the discrimination is based are proportionally reduced. This extension of the scope of the texton theory from texture perception to rapid pattern discrimination suggests a model of vision in general as described in the following section.

## VI. A MODEL OF THE "TWO VISUAL SYSTEMS"

When a visual scene changes suddenly in time or space, and our attention encompasses the entire visual scene, only those areas in which density differences in textons occur are conspicuous. These textons are elongated blobs with specific colors, widths, lengths, orientations, terminators, and crossings between them. Furthermore, because binocular disparity, movement disparity, and flicker are locally conspicuous features that can be detected in a brief presentation,[7,11,13] they, like color, are also properties of elongated blob textons.

Focal attention is directed to areas of spatial or temporal texton changes. The preattentive process appears to work in parallel and extends over a wide area of the visual field, while scrutiny by local or foveal attention is a serial process, which at any given time is restricted to a small patch. Focal attention can be shifted in 50-ms steps, four times faster than the fastest scanning eye movements. Furthermore, the aperture of focal attention can vary in size and can be a minute portion of the fovea, that is, extending to only a few minutes of arc (as shown in Fig. 15). Therefore, if the visual environment is rich in detail even when slowly changing in time, or is rather lacking in spatial detail but changes rapidly, we perform the major portion of our spatio-temporal processing in the preattentive state.

The focus of visual attention seems to be characterized by a texton class as well as a spatial locus. In particular, just as it apparently is impossible to attend simultaneously to two different places, it also seems impossible simultaneously to attend to very different sizes of features. This fact has been noted previously by other psychologists.[14] Stimuli widely separated in space produce cortical responses which are far apart. Similarly, stimuli of differing sizes often generate re-

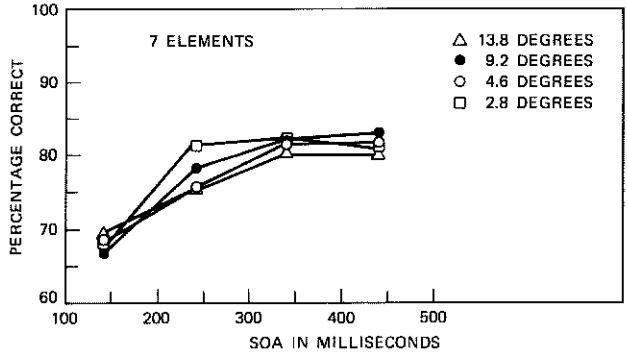



Fig. 15—Size invariance while the angular subtense of seven elements is varied from 2.8 degree of arc diameter to 13.8 degree of arc. Findings imply that the aperture of focal attention can be as small as a few minutes of arc.

sponses in different cortical areas.[6,15] These results seem to imply that the focus of attention is restricted to a very small region of visual cortex, and that stimuli producing responses far apart in the cortex cannot be attended simultaneously.

The essence of our findings is illustrated in Fig. 16. The left array contains a texture composed of "L"-shaped elements (formed by two perpendicular line segments with a gap), except for one "+" shaped and one "T" shaped element (formed by two perpendicular line segments which cross, or have a gap, respectively). The "+" shaped element (target) differs from the many surrounding L's in one texton, namely the "crossing", and perceptually stands out immediately. On the other hand, the T-shaped target can be detected only after some search, by directing the aperture of attention to the target itself.

The right array of Fig. 16 is identical to the left but illustrates our model of vision. The parallel preattentive system instantly detects the location of texton differences and directs the aperture of focal attention to this location, as indicated by the dotted disk around the "+". Since the T contains the same textons as its surround, its detection requires the aperture of attention (symbolized as a "cone" of a searchlight) to scrutinize the texture elements in sequence. Therefore, this

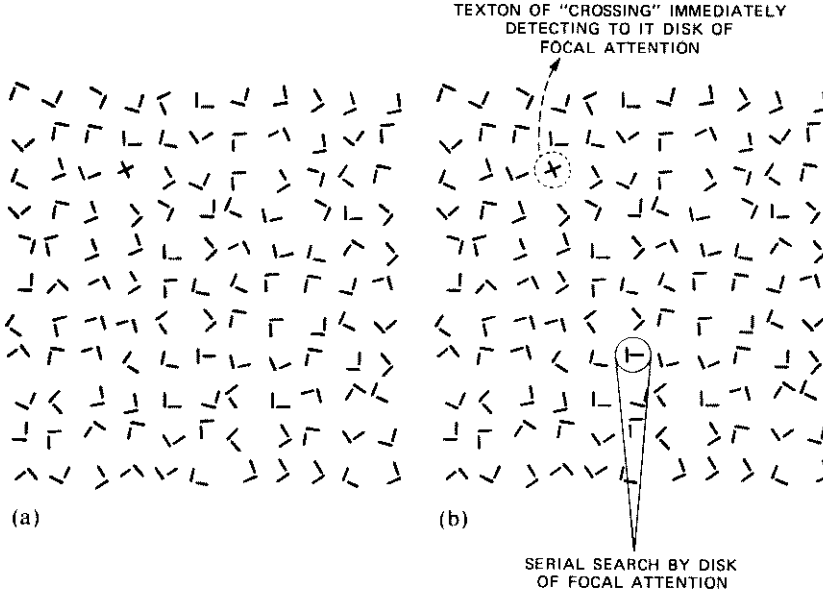



Fig. 16—Model of the two visual systems (b), showing how the preattentive system directs the aperture of focal attention to the loci of texton differences [the + in the L's in (a)], while without such texton differences [the T in the L's in (a)] focal attention requires time-consuming search.

serial search for the T-shaped target depends on the number of texture elements and may take considerable effort and time. However, after the T has been found, and the aperture of focal attention surrounds it, both the + and the T targets are seen with the same clarity. Obviously, form recognition, restricted to the aperture of focal attention, does not depend on the way attention has been directed to the targets. Whether a local difference in textons quickly directed focal attention to the target, or in the absence of texton differences it required time-consuming search to find the target, is immaterial for processing of the target by the attentive visual system.

This mode of behavior of the preattentive and attentive visual systems can also be observed in texture perception, when the reader inspects Fig. 1. The preattentive system immediately detects the texton differences at the boundary of the + and L aggregates, and a quick inspection by focal attention of a few elements on the two sides of the boundary lets the observer conclude that the two areas must contain +'s and L's. Only detailed scrutiny will reveal that the area believed to contain L's only has a region of T's as well.

In summary, the reason that texture discrimination is such a revealing process for showing the workings of the two visual systems is that textures usually cover wide areas of the visual field, while the texture elements are a small portion of the textural area. When the observer is inspecting an extended field, there is an "uncertainty region" in which the relative spatial position of local features is ignored. This is very different from a resolution limit due to visual acuity. In all of the indistinguishable texture pairs, the line segments which make up the texture elements are clearly resolved; nevertheless, if these textons fall within this uncertainty region, it is impossible to tell a T from an L. Many physiologists and psychologists have proposed two visual systems, one ambient and the other focal.[11-20] Yet, without the notion of textons, whose spatial and temporal changes are detected by the preattentive system, which in turn directs focal attention to these loci, the model of the two visual systems is not complete. We hope that the model outlined here gives some useful insights into human vision.

## VII. IMPLICATIONS AND CONCLUSIONS

Some conspicuous local features called textons have been identified by psychological means. These textons, particularly the elongated blobs, are quite similar to features found to stimulate the simple neural units in the striate cortex of the monkey, which are selectively tuned to elongated blobs of certain colors, orientations, width, and length.

Our findings, that in preattentive vision objects are distinguished only through their texton decompositions, might be of considerable

importance. Since in preattentive vision these textons are not coupled, and furthermore the resolution of texton properties—i.e., the perceptual threshold for color, width, length, and orientation differences—is rather limited, the number of distinguishable textons is within practically useful bounds. (For example, the width of periodic bars can be judged with an error of 4 to 6 percent,[21] while accuracy of bar orientation is measured to be only 6 degrees of arc.[22]) This limitation makes practical the devices that simulate preattentive vision. This contrasts with attentive vision for which virtually an infinite number of recognizable patterns exist whose biological, social, or intellectual interest to the observer is unknown. Whether additional textons will be discovered remains to be seen. But as long as they remain independent of the previously isolated textons, the model outlined here will not be importantly affected.

The main implication of our findings is as follows: A considerable amount of vision is carried out by the preattentive system whose workings appear to be much simpler than that of the attentive system. This is important in judging the information requirements of the human visual system realistically. Furthermore, it is important to realize that even in the attentive mental state, with all its prodigious processing powers, complex feats of form recognition are restricted to a small spatial aperture, often as small as a few minutes of arc. Also, changing the position or extent of the aperture of focal attention requires considerable time. The shortest time is about 50 ms when eye movements are prevented, and as long as about 200 ms if saccadic eye movements are necessary.

This dichotomy between preattentive and attentive mental states, the first limited in its power of information processing, the latter limited in its spatial extent, gives a model of human vision that could be exploited in visual communication. Here we do not want to invent specific methods, but only indicate some obvious possibilities. With the advent of fast, perhaps parallel computers, the textons that direct the human observer's attention could be simultaneously extracted by hardware. Detailed images need only be presented in such areas.

Also, one could program computers to extract local features other than textons. For instance, a parallel computer might rapidly detect the difference between an L and a T, rather than between a + and an L. If an observer's attention were directed by such a machine, whose capabilities are very different from human preattentive vision, perhaps a new way of inspecting the visual environment could be made available and possibly learned.

The textons reported here help to discriminate textures, mainly surfaces of objects, without the need of complex familiarity cues. Such an early separation of the visual environment into figure and ground,

or objects and their backgrounds, is a fundamental operation of visual perception. Lack of understanding of this process is, as of now, the greatest bottleneck in machine vision, which in turn is necessary in extending the capabilities of robots.

Regardless of the feasibility of such ambitious schemes, the finding that texton differences can be almost instantaneously perceived over large areas of the visual field can be practically exploited in traffic signs and in directing attention to select areas of visual displays. Traditionally, flickering or static colored lights have been used as traffic signs, or in instrument panels. Now we can add other texton classes—for instance, gaps to increase the terminator number—to enhance visibility. For example, in Fig. 17 we show how a single gap introduced in the conventional alphabet draws attention to the word STOP, which otherwise would require a long time to be segmented and detected. Such slight modification of the alphanumeric characters (amounting to a new "font") might be beneficial in improving legibility. For instance, dyslexic children—children who cannot distinguish well between similar characters with different symmetric transformations such as b, d, or p—might greatly benefit if a gap or stroke were added to one of the characters, so that all characters would differ in at least one texton.

It should be stressed that the textons of preattentive vision only draw attention to certain areas, and we do not claim that these same textons are also the building blocks of form vision. If they were, our findings would prove preattentive vision to be the basis of attentive vision. Even if textons are restricted to vision in the preattentive state, we feel that to know those conspicuous features that grab our attention, wherever they appear, is of interest to everyone who wants to communicate through visual means.

## VIII. ACKNOWLEDGMENTS

We thank our many colleagues who contributed to this research effort throughout the years, and who are mentioned in Section II and in the references. We also thank Max V. Mathews for his helpful comments while reading the manuscript. We are indebted to Walter

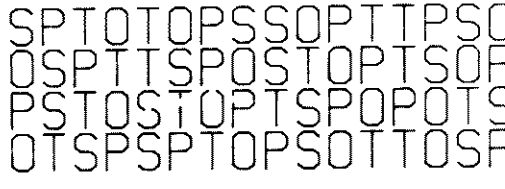



Fig. 17—Demonstration that the introduction of textons into the alphabet (here through increasing the terminator numbers by adding a gap) can help to segment and detect certain areas in a dense letter array.

Kropfl who developed the display hardware, and to Peter Burt who wrote the GENTEX program permitting the rapid generation, display, and manipulation of texture arrays. We thank our summer student, Franklin Schmidt, for developing the TEXTONS program.

## REFERENCES

1. B. Julesz, "Textons, the Elements of Texture Perception, and Their Interactions," Nature, *290* (March 12, 1981), pp. 91–7.
2. B. Julesz, E. N. Gilbert, and J. D. Victor, "Visual Discrimination of Textures with Identical Third-Order Statistics," Biol. Cybernetics, *31* (1978), pp. 137–40.
3. B. Julesz et al., "Inability of Humans to Discriminate Between Visual Textures that Agree in Second-Order Statistics—Revisited," Perception, *2* (1973), pp. 391–405.
4. B. Julesz, "Experiments in the Visual Perception of Texture," Sci. Am., *232* (April 1975), pp. 34–43.
5. B. Julesz, "A Theory of Preattentive Texture Discrimination Based on First-Order Statistics of Textons," Biol. Cybernetics, *41* (1981), pp. 131–8.
6. D. H. Hubel and T. N. Wiesel, "Receptive Fields and Functional Architecture of Monkey Striate Cortex," J. Physiol., *195* (1968), pp. 215–43.
7. B. Julesz, *Foundations of Cyclopean Perception*, Chicago: University of Chicago Press, 1971.
8. D. Marr, "Early Processing of Visual Information," Philos. Trans. R. Soc. London Ser. B, *275* (1976) pp. 483, 524.
9. T. Caelli and B. Julesz, "On Perceptual Analyzers Underlying Visual Texture Discrimination: Part I," Biol. Cybernetics, *28* (1978), pp. 167–75.
10. J. R. Bergen and B. Julesz, "Discrimination with Brief Inspection Times," J. Opt. Soc. Am., *71*, No. 12 (December 1981), p. 1570. Also see "Parallel Versus Serial Processing in Rapid Pattern Discrimination," Nature, *303* (June 23–29, 1983), pp. 696–8.
11. B. Julesz, "Binocular Depth Perception of Computer-Generated Patterns," B.S.T.J., *39* No. 5 (September 1960), pp. 1125–62.
12. E. Averbach and G. Sperling, "Short-term Storage of Information in Vision," in C. Cherry (ed.) *Information Theory, Fourth London Symposium*, London: *Butterworth, 1961*, pp. 196–211.
13. B. Julesz and J. J. Chang, "Interaction Between Pools of Binocular Disparity Detectors Tuned to Different Disparities," Biol. Cybernetics, *22* (1976), pp. 107–19.
14. G. Sperling and M. J. Melchner, "Visual Search, Visual Attention and the Attention Operating Characteristics," in J. Requin (ed.) *Attention and Performance VII*, Hillsdale, NJ: Erlbaum, 1978, pp. 675–86.
15. S. M. Zeki, "The Functional Organization of Projections From Striate to Prestriate Cortex in the Rhesus Monkey," Cold Spring Harbor Symposia on Quantitative Biology, *15* (1976), pp. 591–600.
16. R. Held et al., "Locating and Identifying: Two Modes of Visual Processing, *31*, Psychol. Forsch., pp. 44–62; 299–348 (1967–1968).
17. C. B. Trevarthen, "Two Mechanisms of Vision in Primates," Psychol. Forsch. *31* (1968), pp. 299–337.
18. J. E. Hoffmann, "Hierarchical Stages in the Processing of Visual Information," Perception and Psychophysics, *18* (1975), pp. 348–54.
19. A. Treisman and G. Gelade, "A Feature-Integration Theory of Attention," Cognitive Psychol., *12* (1980), pp. 97–136.
20. B. Julesz, "Visual Pattern Discrimination," IRE Trans. Inform. Theory, *IT-8* (February 1962), pp. 84–92.
21. F. W. Campbell, J. Nachmias, and J. Jukes, "Spatial-Frequency Discrimination in Human Vision," J. Opt. Soc. Am., *60* No. 4 (April 1970), pp. 555–9.
22. J. P. Thomas and J. Gille, "Bandwidths of Orientation Channels in Human Vision," J. Opt. Soc. Am., *69*, No. 5 (May 1973), pp. 652–60.
23. M. Rosenblatt and D. Slepian, "Nth Order Markov Chains With Any Set of N Variable Independent," J. Soc. Indust. Appl. Math., *10* (1962), pp. 537–49.
24. T. Caelli, B. Julesz, and E. N. Gilbert, "On Perceptual Analyzers Underlying Visual Texture Discrimination: Part II," Biol. Cybernetics, *29* (1978), pp. 201–14.
25. B. Julesz, "Spatial Nonlinearities in the Instantaneous Perception of Textures with Identical Power Spectra," Phil. Trans. R. Soc. Lond. B, *290* (1980), pp. 83–94.
26. B. Julesz, "Perceptual Limits of Texture Discrimination and Their Implications to

Figure-Ground Separation," in E.L.J. Leeuwenberg and H.F.J.M. Buffart (eds.), *Formal Theories of Visual Perception*, New York: Wiley, 1978, pp. 205–16.
27. H. L. Frisch and F. H. Stillinger, "Contribution to the Statistical Geometric Basis of Radiation Scattering," J. Chem. Phys., *38* (1963), pp. 2200–7.
28. N. Wiener, "Extrapolation, Interpolation and Smoothing of Stationary Time Series, With Engineering Applications," New York: Cambridge University Press.

## APPENDIX

It required two decades of research efforts to discover that preattentive texture perception depends on local features alone and that global higher-order statistical parameters can be ignored. In 1962, Julesz asked mathematicians to generate stochastic texture pairs that would be identical in their first ($n$-1)th order statistics, but different in the $n$th- and higher than $n$th-order statistics.[20] The $n$th-order statistics are similar to the well-known $n$th-order joint probability distribution of $n$ samples. The $n$ samples are $n$ points of a texture selected at random. However, in random geometry the shape of the $n$ samples is of importance.

These $n$ points can be regarded as the vertices of an $n$-gon. The $n$-gon (or $n$th-order) statistics are obtained when these $n$ points (having the same $n$-gon shape) are selected at random, and statistics indicate that these $n$ points have certain color values. For instance, the second-order statistics can be obtained if a 2-gon (dipole, or needle) is randomly thrown at the texture and the probability is determined that the two end-points of the dipole—of given lengths and orientations—fall on certain color combinations: e.g., black and black; or black and white; or black and gray, etc.

In the intervening years many such stochastic textures were discovered, particularly with identical first- and second-order statistics, but different third- and higher-order statistics.[3,23–25] As a matter of fact, the texture pairs in Figs. 3–6, and 8–10 have this property. The finding that many of these *iso-second-order* texture pairs differing only in third- and higher-orders are indistinguishable suggests that the preattentive visual system cannot compute statistical difference beyond the second order. The recent finding by Julesz, demonstrated in Fig. 11a, suggests that the preattentive visual system cannot even process second-order statistical parameters.[4] From the second-order statistics the autocorrelation function can be uniquely determined—as a matter of fact, for two-tone textures composed of black and white dots, the second-order (dipole) statistic *is* the autocorrelation function[26,27]—and the Fourier transform of the autocorrelation is the Fourier power spectrum.[28] Therefore, all the texture pairs with identical second-order statistics also have identical power spectra. The finding that texture segregation can be obtained in iso-second-order textures, after it was established that the preattentive system cannot process third-order statistics (and, as Fig. 11a demonstrates, not even second-order statis-

tics), implies that this segregation must be based on *local* density differences. Finally, it was proposed that the density changes of certain local conspicuous features, the textons, explain preattentive texture discrimination.[1,25]

## AUTHORS

**James R. Bergen,** A.B. (Mathematics and Psychology), 1975, University of California, Berkeley; Ph.D. (biophysics and theoretical biology), 1981, University of Chicago; Postdoctoral Fellow, Bell Laboratories, 1981–82; RCA, 1983—. Mr. Bergen's work concerns the quantitative analysis of information processing in the human visual system. At the University of Chicago he was involved in the development of a model of the spatial and temporal processing which occurs in the early stages of the system. At Bell Laboratories, his work has concentrated on the effect of visual system structure on the extraction of information from a visual image. Mr. Bergen recently joined RCA Laboratories in Princeton, NJ.

**Bela Julesz,** Diploma, 1950 (Electrical Engineering), Technical University, Budapest; Ph.D., 1956, Hungarian Academy of Sciences; Bell Laboratories, 1956—. Mr. Julesz taught and did research in communications systems for several years prior to 1956. Since joining Bell Laboratories, he has devoted himself to visual research, particularly depth perception and pattern recognition. He is the originator of the random-dot stereoimage technique and of the method of studying texture discrimination by constraining second-order statistics. He has written extensively in the area of visual and auditory perception and is the author of *Foundations of Cyclopean Perception.* Mr. Julesz was Head of the Sensory and Perceptual Processes Department from 1964 to 1982, and in 1983 was made Head of the Visual Perception Research Department. He has been visiting professor of experimental psychology at M.I.T. and other universities. In February 1983 he received the MacArthur Prize Fellow Award in Experimental Psychology and Artificial Intelligence. He was a Fairchild Distinguished Scholar at the California Institute of Technology from 1977 to 1979. Fellow, AAAS, OSA, and the American Academy of Arts and Sciences; Corresponding Member of the Goettingen Academy of Sciences.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# Central Control of Movement Timing

By D. A. ROSENBAUM*

(Manuscript received October 14, 1981)

How do people control the delay between successive finger movements? A reaction-time experiment suggests that the delay is controlled by setting an internal "alarm clock."

## I. INTRODUCTION

A fundamental concern of behavioral science is to understand how we control the movements of our bodies. This paper examines the timing of body movements: As we walk, talk, or type, how are the delays between the phases of these various activities controlled?

A popular means of approaching this question has been to consider two extreme views of how movements might be timed. One view holds that delays between the phases of a movement sequence are controlled by using feedback from each phase to trigger the next. Since the feedback can arise from receptors in the peripheral nervous system—within the muscles, joints, and skin—this view is often referred to as the "peripheralist" theory.[1] The other view holds that movement timing is controlled by plans or "programs" that allow sequences to be performed without the aid of feedback. This view is often called the "centralist" theory.[2]

* Work done at Bell Laboratories. Now at School of Language and Communication, Hampshire College, Amherst, Massachusetts.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

Two main arguments have been leveled against the peripheralist position. One argument is that times between successive movements may often be shorter than the time needed for feedback from one movement to be used to trigger the next.[3] The other argument is that animals and people in whom peripheral feedback is physiologically disrupted can often perform movements effectively.[4,5] Both of these observations indicate than when feedback cannot be relied on, central programs may be used. What the observations do not resolve, however, is whether central programs are also used when feedback is available. The experiment reported here was meant to answer this question. The experiment was also meant to provide detailed information about the nature of the programming of movement sequences when it appears that programming is used.

## II. THE EXPERIMENT

The experiment, which has been reported in detail elsewhere,[6,7] used a simple procedure (see Fig. 1). The subject's task was to perform a sequence of two responses, made with the left and right index fingers, in such a way that the produced interresponse interval approximated a target interval. Feedback given at the end of each trial indicated whether the produced interval was longer or shorter than the target interval and by how much. In addition to producing the specified

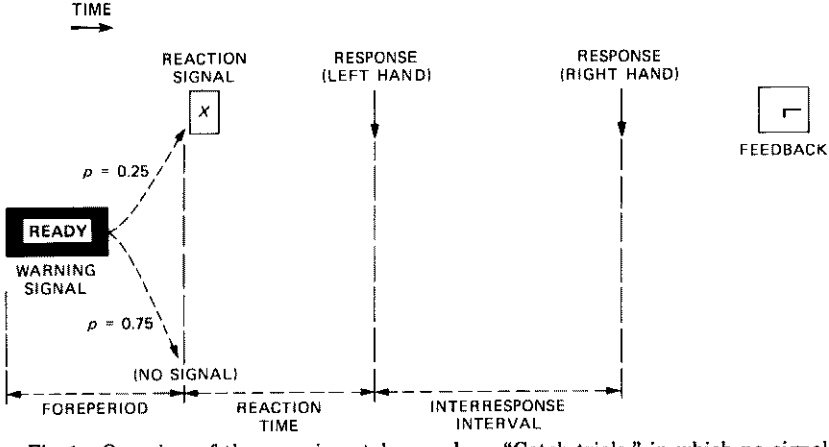



Fig. 1—Overview of the experimental procedure. "Catch trials," in which no signal was presented and no response was allowed, were used to discourage anticipation responses. The warning, reaction, and feedback signals were displayed on a CRT screen. The feedback signal consisted of a vertical line that pointed up if the produced interval was too long and down if the produced interval was too short, where the length of the line indicated the size of the proportional error; the length of a horizontal line indicated how much the Reaction Time (RT) exceeded a target of 130 ms. The foreperiod was fixed at 0.5 second, the reaction signal remained on the screen until the last required response was made, and the feedback signal came on 0.5 second after the offset of the reaction signal and remained on the screen for 2.5 seconds.

interresponse interval, the subject was also required to make the first response as quickly as possible after the onset of a visual reaction signal.

The rationale for the experiment can be understood by noting that several investigators have found that RTs are longer for the first of two responses, performed approximately simultaneously, than for single finger responses.[8–14] If it is assumed that the lengthening of RTs for response doublets results from the demands associated with coordinating two responses, the question that arises is how long the delay between two responses must be before the RT for the first response is as short as the RT for a single, isolated response. According to the peripheralist theory, one would expect the RT for the first of two responses to equal the RT for a single, isolated response whenever the delay between the two responses equals or exceeds the time required to use peripheral feedback from the first response to trigger the second response. Thus, if it takes 200 ms to use peripheral feedback from one response to trigger the next, the peripheralist theory would predict that, whenever the delay between the two responses is greater than or equal to 200 ms, the RT to initiate the two responses in series should be no longer than the RT to initiate the first response alone. By contrast, according to the centralist theory, the RT to initiate the two responses in series should be longer than the RT to initiate the first response alone, even if the delay between the two responses exceeds the feedback loop time; for according to the centralist theory the availability of peripheral feedback does not suffice to control the onset of the second response relative to the first.

Figure 2 shows the results from the main experiment designed to test these predictions. In this experiment, the target interresponse intervals ranged (in a blocked design) from 0 to 1050 ms. In the control condition no second (right-hand) response was required, so the target interval was effectively infinite ($\infty$). The data are averaged over three adult female subjects. In the "stringent" condition, the length of the vertical feedback line was directly proportional to the deviation of the produced interval from the target (nonzero, finite) interval such that the line reached the edge of the CRT screen for deviations exceeding 100 percent of the target interval. In the "relaxed" condition, the scale factor relating the size of the deviation to the length of the vertical feedback line was reduced by a factor of 25 so that generally subjects could only see whether their produced interresponse intervals were too long or too short. About 1300 observations contribute to each point in Fig. 2.

Inspection of Fig. 2a reveals that none of the two-response sequences had mean RTs less than or equal to the mean RT in the control condition, as was confirmed statistically. Since the larger interre-

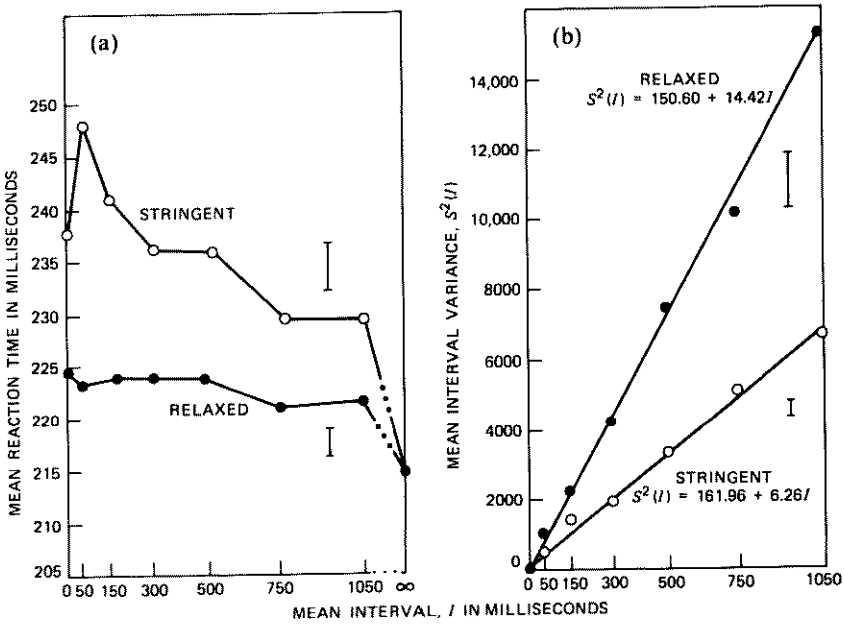



Fig. 2—Results of the experiment. (a) Mean reaction times and estimates of standard error (±SE) for the two accuracy conditions. (b) Mean variances of produced interresponse intervals, fitted linear functions, and estimates of ±SE. The zero intercepts of the fitted linear functions for the two accuracy conditions were not significantly different from one another. (Estimates of ±SE are based on mean squares from fits of mean functions to individual subject data.)

sponse intervals were certainly large enough to allow for the use of peripheral feedback,[15] this result is consistent with the centralist prediction and inconsistent with the peripheralist prediction.

Given that control of the interresponse interval was apparently not based solely on peripheral feedback, what was the nature of the central activity that allowed for control of the intervals? A plausible hypothesis is that the interresponse intervals were controlled by a kind of hydraulic system in which the amount of energy made available to the second response increased as the time until its occurrence decreased. If the total energy available to the two responses was fixed, then the amount of energy available to the first response would decrease as the desired interresponse interval decreased, leading to an inverse relationship between RTs and interresponse intervals. This is a response competition model. (For a more general discussion of response competition, see Ref. 16.)

An alternative hypothesis is that the interresponse interval was controlled by a central clock that was set just before the production of the first response (i.e., during the RT). If the time needed to set the clock depended on the precision of setting, which in turn was reflected

in the variability of the interresponse interval, this alarm clock model predicts that RTs could depend on interval variability rather than interval size.

To test these predictions, we manipulated the analog feedback that subjects received concerning the accuracy of their interresponse intervals. Recall that the analog feedback took the form of a vertical line that pointed up if the produced interval was larger than the target interval and down if the produced interval was smaller than the target interval. By changing the scale factor relating the length of the line to the proportional difference between the produced and target interresponse interval, we could impose stringent or relaxed accuracy requirements on the subject's timing performance. When the scale factor was small, a large error resulted in a small error line ("relaxed" condition), but when the scale factor was large, even a small error resulted in a large error line ("stringent" condition). The relaxed-stringent manipulation was a within-subject variable, which we introduced to get subjects to alter the variability of their interresponse intervals while keeping the means of the intervals constant.

As is seen in Fig. 2a, mean RTs in the relaxed condition were essentially constant over the range of finite intervals. However, mean RTs in the stringent condition were longer than in the relaxed condition and decreased as intervals increased for targets 50 ms or greater. [The discontinuity in the stringent mean RT function at the 0-ms target confirmed subjects' reports that control of the intervals in this condition was qualitatively different from the other conditions (e.g., because the fingers and wrists were kept stiff when simultaneous responses were attempted). Sequences begun with the right hand, which were only permitted in the 0-ms condition, did not have shorter RTs than sequences begun with the left hand.] Thus, Fig. 2a shows that mean RTs were not simply related to mean intervals, contrary to the prediction of the response competition hypothesis. Instead, mean RTs were related to the required precision of the intervals, consistent with the alarm clock model.

The question that now arises is whether the actual precision of the intervals was affected by the differential precision requirements imposed by the stringent-relaxed manipulation. If the actual precision was affected, such that greater precision was achieved in the stringent condition, the evidence would be stronger for the hypothesis that during the RT an internal alarm clock was set to varying degrees of precision. In Fig. 2b variances of produced intervals are plotted against their corresponding means. As is seen in the figure, variances were larger in the relaxed condition than in the stringent condition. That the interval means were approximately equal in the two conditions is demonstrated by the approximately vertical alignment of the filled

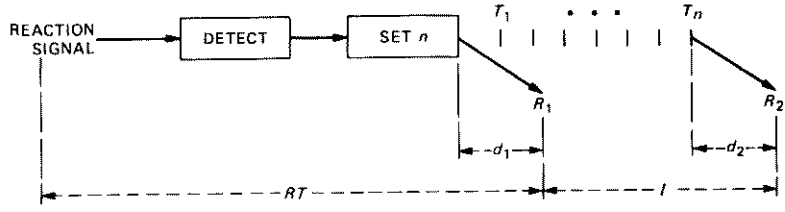



Fig. 3—Overview of an alarm-clock model for performance in the experiment.

and empty points corresponding to each target interval. As is also seen in Fig. 2b, the mean interval variances were well fit by linear functions. Linear regression accounted for 99.7 percent and 99.9 percent of the variance among the mean variances (averaged over subjects) in the stringent and relaxed conditions, respectively. The linearity of the variance functions allows for a simple elaboration of the alarm clock model, which is discussed below.

## III. AN ALARM CLOCK MODEL*

The elaborated alarm clock model is shown in Fig. 3. It says that the size of the delay between the first and second response ($R_1$ and $R_2$, respectively) depends on the number, $n$, of pulses of an internal clock, where the value of $n$ is set during the RT interval. The model assumes that the interval, $I$, between $R_1$ and $R_2$ equals the sum of the times, $T_1, T_2, \cdots, T_n$ for the $n$ clock pulses, plus the difference between the motor delays, $d_1$ and $d_2$, of $R_1$ and $R_2$, respectively. That is,

$$I = T_1 + \cdots + T_n + d_2 - d_1. \tag{1}$$

It is also assumed that the times for the successive clock pulses fluctuate randomly about a mean and are stochastically independent. Hence, as $n$ increases, the variance of the cumulative pulse time increases linearly.[17,18]

The model affords two plausible ways of explaining the difference in slopes for the fitted variance functions in the stringent and relaxed conditions. One possibility is that the variability of interpulse times was greater in the relaxed condition than in the stringent condition. The other possibility, which will be pursued here because of its interesting ramifications for the proposed clock-setting process, is that the variance of $n$ was greater in the relaxed condition than in the stringent condition. It is possible to express precisely how the variance of $n$ in the relaxed condition would be related to the variance of $n$ in the stringent condition, given that the slopes (but not the zero intercepts) of the variance functions differed in the two conditions. Let the

* This is a greatly simplified version of a model presented in Ref. 6.

random variable $N$ be the number of clock pulses between the triggering of response 1 and 2. Let the time between pulses $i-1$ and $i$, for $1 \leq i \leq n$, be a random variable $T$, where the random variables $T_i$ are independent of $N$ and identically distributed as a random variable $T$ with finite mean and variance. Allowing the random variable $D$ to represent the difference $d_2 - d_1$, and assuming $T$, $N$, and $D$ are mutually independent, it is possible to use the expression for the variance of a random sum[19] to obtain

$$\mathrm{Var}(I) = E(N)\mathrm{Var}(T) + \mathrm{Var}(N)E^2(T) + \mathrm{Var}(D). \tag{2}$$

Assuming that the only terms in (2) that differ in the relaxed and stringent conditions are $\mathrm{Var}(I)$ and $\mathrm{Var}(N)$, and using subscripts R and S for the relaxed and stringent conditions, respectively, we can subtract $\mathrm{Var}_S(I)$ from $\mathrm{Var}_R(I)$ and rearrange terms to obtain

$$\mathrm{Var}_R(N) - \mathrm{Var}_S(N) = [E^2(T)]^{-1} \cdot [\mathrm{Var}_R(I) - \mathrm{Var}_S(I)]. \tag{3}$$

Since $\mathrm{Var}_R(I)$ and $\mathrm{Var}_S(I)$ were found to be linear functions of $I$ with what are assumed to be identical zero intercepts, and since $E^2(T)$ is assumed to be a constant, eq. (3) becomes

$$\mathrm{Var}_R(N) - \mathrm{Var}_S(N) = kI, \tag{4}$$

where $k$ is a constant greater than 0. Equation (4) shows that $\mathrm{Var}_R(N)$ and $\mathrm{Var}_S(N)$ are related in a simple way. As is seen below, however, the relationship can be shown to be even simpler.

Consider how the variance of $N$ changes with $I$ within the stringent or relaxed condition. Relying on the fact that we found $\mathrm{Var}(I)$ and $I$ to be linearly related, and noting that

$$E(N) = [E(I) - E(D)]/E(T), \tag{5}$$

eq. (2) can be rewritten to solve for $\mathrm{Var}(N)$ as

$$\mathrm{Var}(N) = \frac{\alpha + \beta \cdot E(I) - \mathrm{Var}(D) - \left[\dfrac{E(I) - E(D)}{E(T)}\right] \mathrm{Var}(T)}{E^2(T)}. \tag{6}$$

All the terms in eq. (6) are constant except for $\mathrm{Var}(N)$ and $E(I)$. Therefore, eq. (6) can be rewritten

$$\mathrm{Var}(N) = u + vE(I), \tag{7}$$

where $u$ and $v$ are constants. Since the clock presumably would not be used (i.e., $N$ would not be set) when $E(I) = 0$, $u$ can be set to 0, leaving

$$\mathrm{Var}(N) = vE(I). \tag{8}$$

On the basis of the simple relation shown in eq. (8), we can indicate how $\mathrm{Var}(N)$ in the stringent condition, $\mathrm{Var}_S(N)$, is simply related to

$\mathrm{Var}(N)$ in the relaxed condition, $\mathrm{Var}_R(N)$. Let $\mathrm{Var}_R(N) = rE(I)$. As we saw in eq. (4), $\mathrm{Var}_R(N) - \mathrm{Var}_S(N) = kE(I)$. It follows, then, that $\mathrm{Var}_S(N) = (r - k)E(I)$. Thus, $\mathrm{Var}_R(N)$ and $\mathrm{Var}_S(N)$ differ with respect to the size of the proportionality constant relating each of these quantities to $E(I)$.

To summarize, the above discussion reveals that the variance of $N$ is proportional to $E(I)$, and that the slope of this function differs in the relaxed and stringent conditions.

The simplicity of these theoretical results allows for the proposal and test of a simple model of the process that may have given rise to the mean RT effects shown in Fig. 2a. The model is suggested by the observation that shorter RTs were associated with larger (predicted) values of $\mathrm{Var}(N)$. Essentially, the model says that the time to set $N$ increases as the range of possible values of $N$ decreases. (A possible reason will be given in Section IV of this paper.) Let $\theta$ denote the time to set $N$ and recall that the range of a distribution is proportional to the standard deviation of that distribution. The model says

$$\theta = \frac{\gamma}{\sqrt{\mathrm{Var}(N)}} = \frac{\gamma}{\sqrt{\delta E(I)}} = c[E(I)]^{-1/2}, \tag{9}$$

where $c = \gamma/\sqrt{\delta}$ is an empirical constant and $0 < E(I) < \infty$. From the previous observation that the proportionality constant relating $\mathrm{Var}(N)$ to $E(I)$ distinguishes the relaxed and stringent conditions, it follows that only the size of $c$ in eq. (9) needs to differ in the relaxed and stringent conditions. Moreover, as is implied in Fig. 3, the RT is assumed to equal $\theta$ plus the time to detect the reaction signal and execute the first response. Hence, the model predicts that the RT in the relaxed and stringent conditions can be expressed as

$$\mathrm{RT}_R = k + \theta_R$$

and

$$\mathrm{RT}_S = k + \theta_S, \tag{10}$$

respectively, or

$$\mathrm{RT}_R = k + r[E(I)]^{-1/2}$$

and

$$\mathrm{RT}_S = k + s[E(I)]^{-1/2}, \tag{11}$$

respectively, where $k$, $r$, and $s$ are empirical constants. Equation (11) has the appealing property that RTs increase with decreases in $E(I)$ at a rate related to the size of the linear coefficient. Thus, changing the linear coefficient in eq. (11) produces a set of curves two of whose members look like the pair of curves seen in Fig. 2a.

For purposes of finding out how well eq. (11) actually fit the RT data, an iterative procedure was used to find the values of $k$, $r$, and $s$ for which it was possible to minimize the sum of squared deviations of the obtained RTs from the predicted RTs. The obtained RT values used were the mean RTs in the conditions where nonzero finite interresponse intervals were required. The best-fitting values of $r$ and $s$ turned out to be 25 and 236, respectively, and the best-fitting value of $k$ turned out to be 221 ms. With these estimates the fitted model accounted for 94.6 percent of the variance of mean obtained RTs. (When $k$ was held at 216 ms, which was the mean RT in the control condition, only 67.5 percent of the variance was accounted for. Because the goodness of fit was so much poorer here, it appears that the time required for detection of the reaction signal and execution of the first response differed in the finite and infinite-interval conditions, or that some extra process, not identified in the model, in fact occurred in the finite-interval conditions.)

## IV. CONCLUDING REMARKS

This paper has examined the mechanisms of movement timing. Of necessity, the experiment that has been reported involved a restricted type of movement, namely simple finger sequences. Yet the experiment appears to have fairly broad theoretical importance for our understanding of how movements are timed. In particular, the experiment suggests that *even when peripheral feedback is available, central programs play a controlling role in movement timing.* The basis for this conclusion is that the mean RT for the first of two responses was longer than the mean RT for a single, isolated response, even when the time between the first and second response exceeded 1 second. That the lengthening of RTs at such long delays was in fact attributable to some aspect of the control of movement timing is suggested by the fact that the lengthening of RTs was greater when the interresponse interval was controlled more precisely.

Besides supporting a general, qualitative principle about movement control, the data from the present experiment also allow for a relatively detailed model of the control of movement timing. The model likens the selection and control of an interresponse interval to the setting and running of a conventional alarm clock: First, the desired number, $n$, of clock pulses is selected (during the RT), and then the clock is allowed to produce the $n$ pulses so that the first response is triggered at the time of the first pulse and the second response is triggered at the time of the $n$th pulse. The proposal that there is a clock-pulsing mechanism is motivated by the finding that the variance of the interresponse interval increased linearly with the interval mean, which is suggestive of a *stochastic wait* process (see Refs. 17 and 18) where

times between successive pulses fluctuate randomly about a mean and are stochastically independent. Through some algebra, it was seen that another factor contributing to the interval variance is the variance of $N$, which was seen to be proportional to the interval mean. The size of this proportionality factor was seen to depend on the slope of the linear function relating the interval variance and mean, and this slope was observed to differ in the relaxed and stringent conditions of the experiment. Having established the above relationships, it was observed that the mean RTs in the nonzero, finite-interval conditions were inversely related to the (predicted) variance of $N$. Then, through a curve-fitting procedure, it was seen that this relationship could account for almost 95 percent of the variance of the mean RTs in the nonzero finite-interval conditions. The inverse relationship between the mean RT and predicted variance of $N$ can be explained by saying that it takes longer to select a particular value of $N$ as the range of allowable values decreases. Such a relationship can be understood in statistical terms as reflecting a decreasing likelihood of selecting an allowable value of $N$ as the range of allowable values of $N$ decreases. If reselection of a value of $N$ is required when an unallowable value has been picked, the average time to select an allowable value would vary in a way consistent with the mean RT results obtained here.

It is interesting to note that the physiological basis for an alarm-clock system has been demonstrated. In the cortex of the cerebellum, rows of regularly spaced neurons known as Purkinje cells are connected to other neurons in such a way that the delay between the entry of a signal into the row and the exit of a signal from the row depends on which Purkinje cell is permitted to conduct the output signal.[20,21] The process of selectively permitting a given Purkinje cell to conduct an output signal on the basis of the delay that would occur prior to that output signal is physiologically analogous to the clock-setting process proposed here.

## V. ACKNOWLEDGMENTS

The author wishes to thank J. L. Knight and J. F. Kroll for their helpful comments during the preparation of this paper.

## REFERENCES

1. J. A. Adams, "Feedback theory of how joint receptors regulate the timing and positioning of a limb," Psychological Review, *84* (November 1977), pp. 504–23.
2. S. W. Keele, "Movement control in skilled motor performance," Psychological Bulletin, *70*, No. 6, Part 1 (1968), pp. 387–403.
3. K. S. Lashley, "The problem of serial order in behavior," in L. A. Jeffress (Ed.), *Cerebral Mechanism in Behavior*, New York: Wiley & Sons, 1951.
4. E. V. Evarts, E. Bizzi, R. Burke, M. DeLong, and W. T. Thach, Jr., *Central Control of Movement*, Brookline, MA: Neurosciences Research Program, 1971.
5. E. Taub and A. J. Berman, "Movement and learning in the absence of sensory

feedback," in S. J. Freedman (Ed.), *The Neuropsychology of Spatially Oriented Behavior*, Homewood, IL: Dorsey Press, 1968.
6. D. A. Rosenbaum and O. Patashnik, "A mental clock setting process revealed by reaction times," in G. E. Stelmach and J. Requin (Eds.), *Tutorials in Motor Behavior*, Amsterdam: North-Holland Publishing Co., 1980.
7. D. A. Rosenbaum and O. Patashnik, "Time to time in the human motor system," in R. S. Nickerson (Ed.), *Attention and Performance VIII*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.
8. M. DiStefano, M. Morelli, C. A. Marzi, and G. Berlucchi, "Hemispheric control of unilateral and bilateral movements of proximal and distal parts of the arm as inferred from simple reaction time to lateralized light stimuli in man," *Experimental Brain Research, 38* (1980), pp. 197–204.
9. N. Hammond and P. Barber, "Evidence for abstract response codes: Ear-hand correspondence effects in a three-choice reaction-time task," Quart. J. Experimental Psych., *30* (February 1978), pp. 71–82.
10. M. A. Jeeves, "A comparison of interhemispheric transmission times in acollosals and normals," Psychonomic Science, *16*, No. 5 (1969), pp. 245–6.
11. J. A. S. Kelso, D. L. Southard, and D. Goodman, "On the coordination of two-handed movements," Science, *203* (1979), pp. 1029–31.
12. J. R. Peterson, "Response-response compatibility effects in a two-hand pointing task," Human Factors, *7* (1965), pp. 231–6.
13. P. M. A. Rabbitt, S. M. Vyas, and S. Fearnley, "Programming sequences of complex responses," in P. M. A. Rabbitt and S. Dornic (Eds.), *Attention and Performance V*, London: Academic Press, 1975.
14. H. C. Ratz and D. Ritchie, "Operator performance on a chord keyboard," J. Appl. Psych., *45* (1961), pp. 303–8.
15. J. A. Adams, "Issues for a closed-loop theory of motor learning," in G. E. Stelmach (Ed.), *Motor control: Issues and trends*, New York: Academic Press, 1976.
16. L. M. Herman and B. H. Kantowitz, "The psychological refractory period effect: Only half the double-stimulation story?," Psychological Bulletin, *73*, No. 1 (1970), pp. 74–88.
17. A. M. Wing and A. B. Kristofferson, "The timing of interresponse intervals," Perception & Psychophysics, *13* (June 1973), pp. 455–60.
18. A. M. Wing and A. B. Kristofferson, "Response delays and the timing of discrete motor responses," Perception & Psychophysics, *14* (August 1973), pp. 5–12.
19. E. Parzen, *Stochastic Processes*, San Francisco: Holden-Day, 1962, p. 56.
20. V. Braitenberg, "Functional interpretation of cerebellar histology," Nature, *190* (1961), pp. 539–40.
21. H. H. Kornhuber, "Cerebral cortex, cerebellum, and basal ganglia: An introduction to their motor functions," in E. V. Evarts (Ed.), *Central Processing of Sensory Input Leading to Motor Output*, Cambridge, MA: The MIT Press, 1975.

## AUTHOR

**David A. Rosenbaum,** B.A., 1973, Swarthmore College; Ph.D., 1977, Stanford; Bell Laboratories, 1977–1981; Hampshire College, 1981—. Mr. Rosenbaum worked in the Human Information-Processing Research Department at Bell Laboratories, Murray Hill, from 1977 to 1981. Currently, he is Assistant Professor of Cognitive Science at Hampshire College, Amherst, Massachusetts, where he teaches and continues research on the cognitive control of human movement.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Experiments on Quantitative Judgments of Graphs and Maps

By W. S. CLEVELAND,* C. S. HARRIS,* and R. McGILL*

(Manuscript received August 18, 1982)

Behavioral studies are essential for devising guidelines for effective communication of quantitative information from graphs. Three experiments in which subjects made quantitative judgments from three different kinds of graphs lead to several recommendations: use pastel rather than highly saturated colors on statistical maps; standardize the point cloud size relative to the frame on a scatterplot; scale circles by making the circle area proportional to the variable represented, but expect widely varying judgments of the areas.

## I. INTRODUCTION

With the proliferation of computer graphics, there is an increasing reliance on visual displays to convey quantitative information. Maps, graphs, and diagrams have been in use for a long time, but in recent years the variety, complexity, and ease of preparing such visual displays have increased greatly. It is often assumed that visual displays allow people to quickly and accurately appreciate quantitative information and relationships that might be much harder to grasp from other representations, such as tables of numbers, equations, or verbal

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

descriptions. Graphs, for example, are regarded as powerful tools for analyzing data[1] and for presenting data to others.[2]

In preparing a graph, the usual assumption is that if the mathematical relationships are represented accurately in visual form, they will convey the correct quantitative impressions. But this may not be right. Numerous experiments with very simple displays have shown that people's perceptual judgments often differ from physical measurements of such attributes as length, area, orientation, separation, brightness, and color (for reference lists, see Refs. 3 through 5). A much smaller number of studies have used displays that are similar to graphs or maps (see Refs. 6 through 9). Considering the enormous usage and variety of graphs, the number of directly applicable experiments is quite small. More information about human factors in graphical judgment is essential for the design of better visual displays.[10,11]

We summarize here three sets of experiments on quantitative judgments of three kinds of graphs. (More detailed accounts are given in Refs. 6, 7, and 12). The experiments differ in the information being conveyed, the coding of the information, the experimental procedures, the subject populations, and the methods of statistical analysis of the results. In passing, we will mention some useful statistical techniques that may be new to many readers.

## II. A COLOR-CAUSED ILLUSION OF AREA

A clear example of erroneous perception of a relatively simple display is shown in a study of the perception of areas of colored regions within a map.[12] Figure 1 is a map of the counties in Nevada. Maps like this, with subsets of the counties colored red or green, were shown to 24 subjects (12 scientists and 12 secretaries, clerks, and craftspeople, all from Bell Laboratories). In each map the total red area differed from the total green area by no more than 0.01 percent. Each subject saw ten variations of the map, with different subsets of counties colored red or green. The maps for one group of 12 subjects had the same partitioning of counties as for the other group, but the counties that were red for one group were green for the other, and vice versa. The maps were paper prints produced by a Bell Laboratories Printing System-Multicolor (PRISM) printer (a computer-driven modified Xerox 6500 copier), using standard options, which include highly saturated colors. Matching the colors with Munsell color chips,[13] under illumination similar to that in which subjects viewed the maps, gave Munsell values of hue = 7.5 red, value (brightness) = 4, chroma (saturation) = 14 and hue = 2.5 green, value = 5, chroma = 12. Thus, the red and green were quite similar in brightness and saturation.

An instruction sheet told the subjects that the colors indicated various geological features in each county. The subjects' task was to

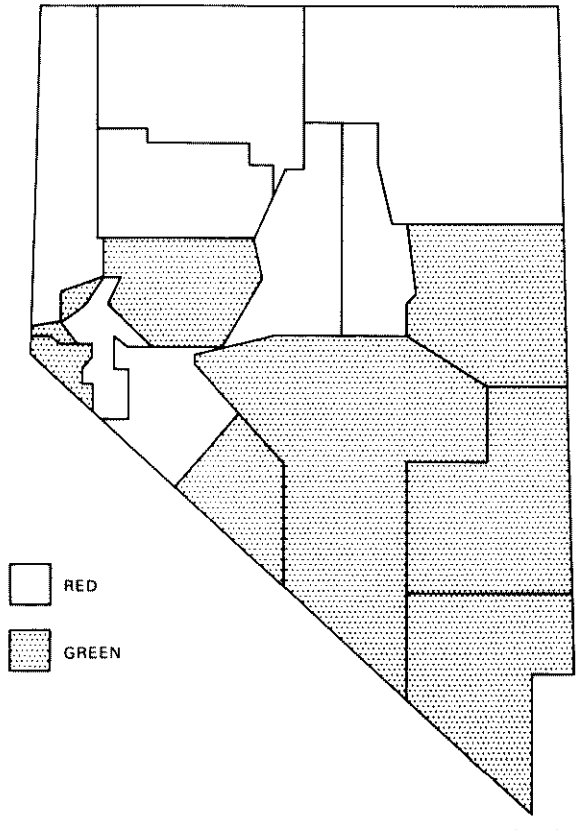



Fig. 1—Map of Nevada, where light and dark areas represent red and green, respectively.

decide which colored area within each map was larger and to mark "more red," "more green," or "same" on a checklist. The outcome was that most of the 24 subjects marked "more red" more frequently than "more green." That is, although the red and green areas were actually equal, they didn't look equal.

Figure 2 shows the data on a trilinear plot,[2] which accommodates within a single graph three variables that sum to a constant (100 percent in this case). Each open circle represents a single subject's data. The vertical axis indicates the percentage of maps for which a subject said the areas of the two colors looked the same. The two diagonal axes measure the percentage of maps in which the red or green area was called larger. Just as in the more familiar Cartesian xy plot, the three percentages for each data point are given by its perpendicular projection onto each of the three axes.

Because each subject made ten judgments, all data points represent multiples of 10 percent. Therefore, to avoid overlapping the values

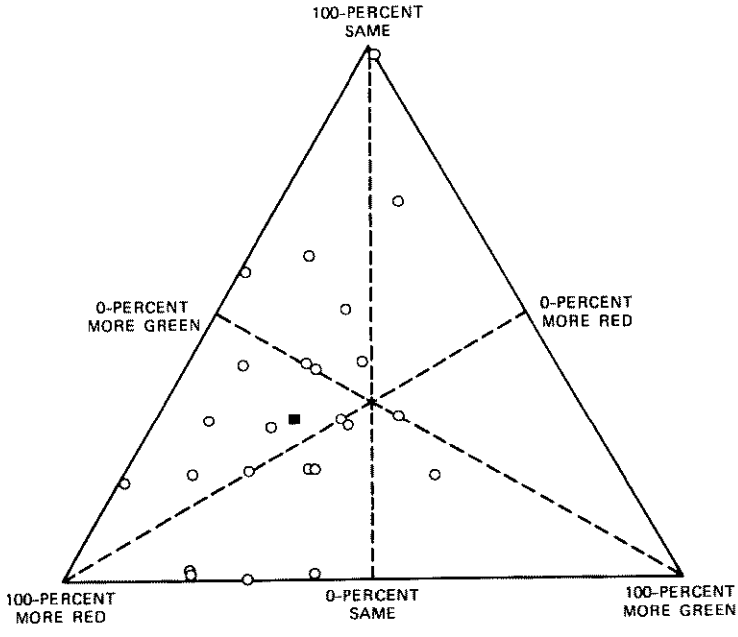



Fig. 2—Area judgments of highly saturated red and green regions.

were jittered: a randomly generated number from the interval −1 percent to +1 percent was added to each value.

First note that most of the 24 subjects' points fall to the left of the vertical axis. This indicates that most subjects judged the red area to be larger more often than they judged the green area to be larger. The filled square is the "robust center of gravity" for the data points. This statistic is calculated by an iterative procedure called "bisquare", using the distances of the data points from the current estimate as input for the next estimate. Using the bisquare[14] produces a statistic that is robust;[15] that is, unlike an arithmetic mean, it is insensitive to outliers, and is highly efficient for a broad class of distributions. The coordinates of this filled square give us another way to summarize the red/green illusion: it falls at about 50 percent "more red" judgments, 20 percent "more green," and 30 percent "same."

To estimate standard errors for the robust center of gravity, the bootstrap method[16] was used. Bisquare estimates were computed on 1000 24-point samples, drawn with replacement. The difference between the green and red coordinates is 49 percent − 22 percent = 26 percent, with a bootstrap standard error of 5.3 percent. Since the bootstrap distribution of the difference was well fit by a normal distribution, the percentage of red-larger judgments is very significantly higher than the percentage of green-larger.

Thus, the likely result of accepting the saturated colors that the PRISM printers normally produce is to make the red areas look too big or the green areas too small. Is there a way around this perceptual distortion? The printer was modified to fill the areas with two unsaturated pastel colors: a half-tone screen composed of red and pale yellow dots and a half-tone screen composed of green and pale yellow dots. The filled square for the robust center of gravity in Fig. 3 shows that for the group as a whole, the percentages of judgments of "more red" and "more green" are nearly equal. Using pastel instead of saturated colors eliminated the overall tendency to call the red area larger than the green.

Additional research could determine whether these findings hold for other pairs of hues, other combinations of brightness and saturation, and other display media (e.g., video displays). For the PRISM printer, it is clear that for accurate judgments of relative area, pastel colors are preferable to the standard saturated red and green.

### III. JUDGMENTS OF CIRCLE SIZES

In most of the perceptions that we label "illusions," some extraneous aspect of the stimulus seems to be distorting perception of another aspect. For example, in the familiar Müller-Lyer illusion the arrowheads added to the ends of two horizontal lines distort perception of

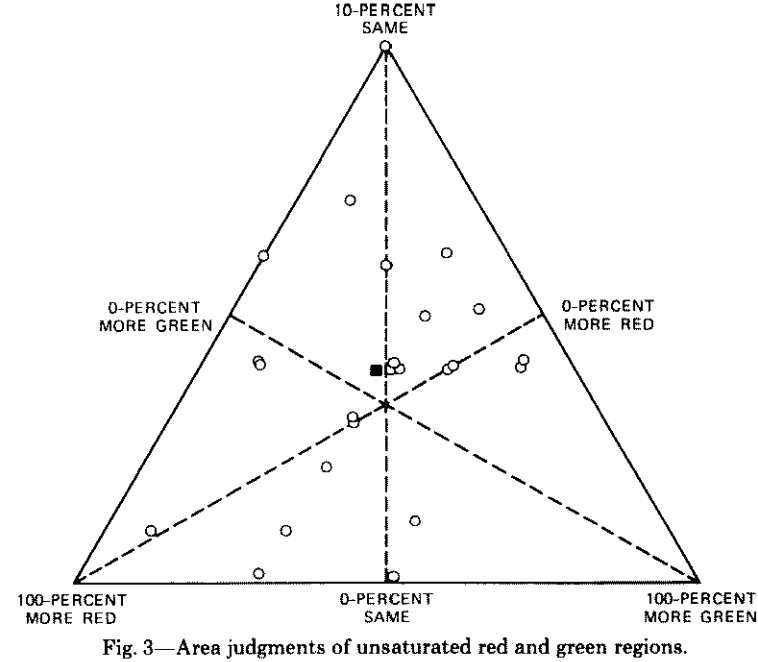



Fig. 3—Area judgments of unsaturated red and green regions.

the lengths of those horizontal lines. Similarly, in the experiment with maps of Nevada, the colors distorted perception of areas. But, in fact, many of our perceptions—perhaps even most of them—are illusions, in the sense that what we see does not match the usual measurements obtained with instruments other than the human observer. For a wide variety of attributes (such as loudness, brightness, tactual roughness, and weight), laboratory studies have found that people's judgments of quantities or intensities seldom vary in direct proportion to these measurements of the physical stimulus. In most cases the relation between subjective judgments and physical measurements can be described by a power function of the form $s = kp^e + c$, where $s$ is the subjective magnitude, $p$ is the physical magnitude, $k$ and $c$ are scaling constants, and $e$ is an exponent that depends on what is being judged, ranging from 0.3 for the brightness of an isolated disk of light to 3.5 for the intensity of an electric shock.[5]

What does this imply for the communication of quantitative information by means of graphs or maps? Consider a common kind of statistical map in which circles of different sizes are used to represent, for example, the number of toll calls from various cities. For laboratory studies of the judgment of area, Stevens (see Ref. 5) gives 0.7 as a typical value for the exponent in the psychophysical function. Such a low exponent would mean that a circle that is double the area of another would be called only 1.6 times as large, and one with five times the area would be called only three times as large, whereas one with 25 percent of the area would be called almost 40 percent as large.

Since the purpose of a map or graph is to convey quantitative relations quickly and accurately, shouldn't areas be scaled to give the correct subjective impressions rather than to be physically correct? Some writers[17,18] have suggested just such a procedure. They recommended scaling the plotted areas to compensate for the low-exponent function found in psychophysical experiments.

However, the exponent found in studies of area judgments might not apply to an actual statistical map. We therefore prepared 24 maplike pages that depicted the average daily telephone toll charges for businesses at different locations in a city.[7] An example is shown in Fig. 4. Fourteen scientists from Bell Laboratories were told that the circle marked with an X represented toll charges of $100. They were to write down, for each of the three circles marked with a dot, their estimate of the dollar amount represented. The word "area" was not used.

The results were quite different from the usual laboratory findings on judgments of area: Most of the exponents in the fitted power functions for individual subjects were closer to 1.0 than to 0.7. There was considerable variation from person to person, with exponents

Figure 467

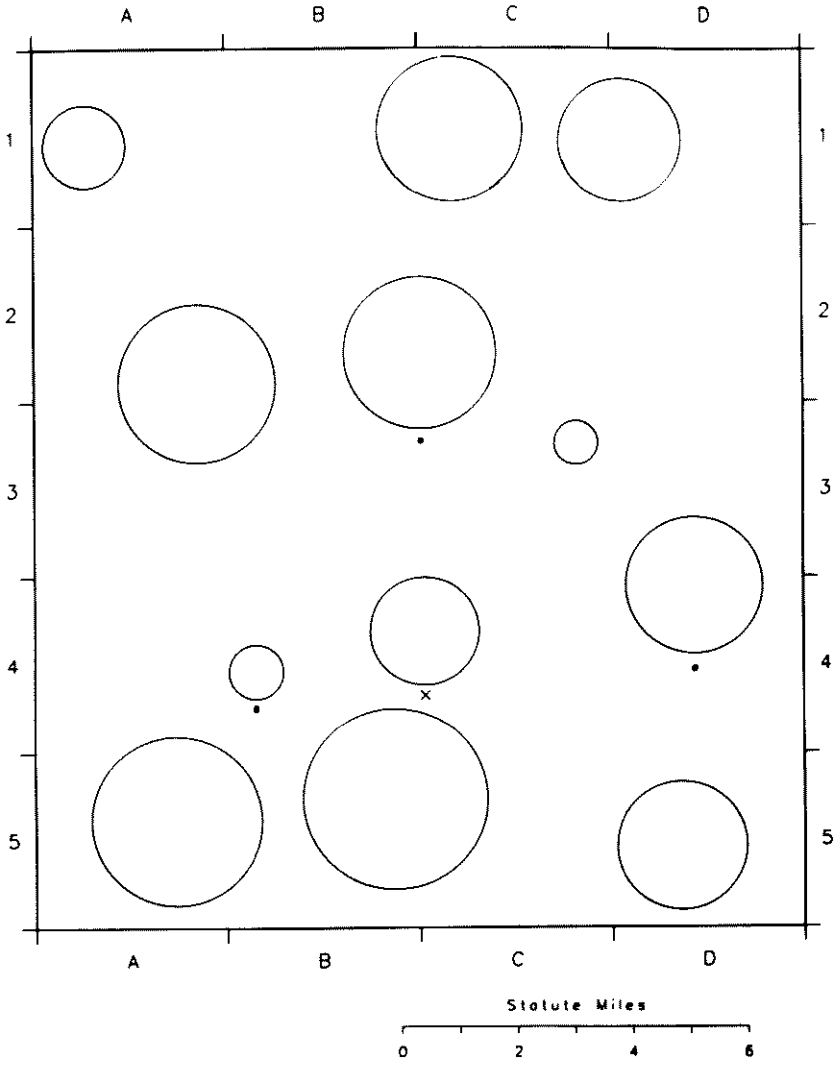



Fig. 4—Maplike display of circles.

ranging from 0.6 to 1.3. This implies that for an actual chart-reading situation, no single form of compensatory scaling of area would give everyone correct impressions. In fact, since the median exponent (0.94) is close to 1.0, these data suggest that the best scale is the simple one with circle areas directly proportional to the represented quantities. (Note that judging diameter or circumference instead of area would

yield an exponent of 0.5 for judgments as a function of area. None of our subjects had an exponent that low.)

Why do our exponents differ from those typically found in psychophysical studies? Is it because we asked for a different kind of judgment—dollars represented, rather than area—and displayed a maplike frame and tick marks? We can answer this question because we had the same subjects make area judgments of the same 24 sets of circles on plain pages, without maplike markings (Fig. 5). All of them did this perceptual task after judging the maplike stimuli. The instruction was to judge the areas of the circles, calling the one marked X 100 units of area. We found no difference between the two types of displays and tasks.

Figure 179

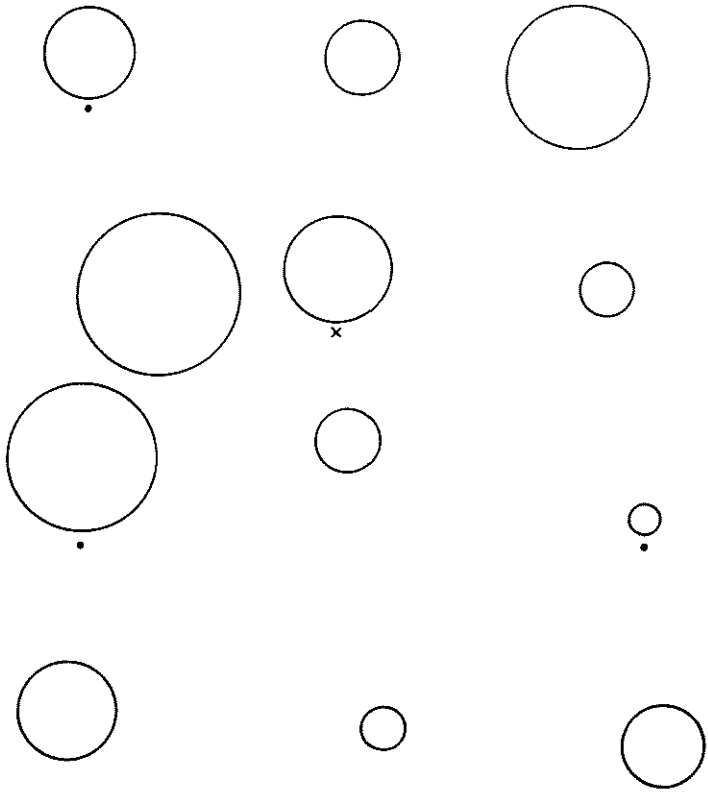



Fig. 5—Display without maplike markings.

Our next thought was that our subjects' scientific training might enable them to judge areas more accurately, on the average, than subjects in previous laboratory experiments. We therefore asked 10 high school students to carry out the same two tasks. Figure 6 presents a comparison of the exponents estimated for the scientists (on the left) and 10 high school students (on the right). Functions of the form $s = kp^e + c$, relating subjective judgment ($s$) to physical area ($p$), were fitted to each subject's data. The digits to the left of the colons are the first digits of the estimated exponents ($e$). Each digit to the right of the colon is the next digit in the exponent for a single subject. The distributions are similar for the scientists and the high school students, with medians at 0.94 and 0.97, respectively. These stem-and-leaf diagrams[19] can be interpreted as enriched histograms. The digits to the left of the colons are the first one or two digits of the estimated exponent, while each single digit to the right of the colon is the next digit in the exponent for a single subject. For example, the stem-and-leaf diagram tells us not only that four scientists had exponents in the range from 1.00 to 1.09, but also that the specific values were 1.00, 1.04, 1.05, and 1.07. There is not much difference between the distributions for the scientists (median = 0.94) and for the students (median = 0.97). Scientific training doesn't have much influence.

The students' data also rule out another possible explanation of the difference from previous psychophysical studies. The scientists judged all of the maplike displays before seeing the ones that contained only circles. When estimating the toll charges, they were free to adopt any strategy they cared to, unlike subjects in psychophysical studies who are told to judge area. Even though the scientists were then asked to judge areas in the second half of the experiment, they might have simply persisted in using the same judgmental strategy as in the first half. This choice of strategy could explain why judgments were the same for maplike displays and plain circles, and also why the results differ from earlier experiments on area estimation. However, half of the students made their area judgments *before* making dollar estimates, and their estimates were quite similar to those made by the students who completed the tasks in the opposite order. The agreement between

| EXPONENTS FOR 14 SCIENTISTS | EXPONENTS FOR 10 HIGH SCHOOL STUDENTS |
|---|---|
| 0.5 : | 0.5 : 8 |
| 0.6 : 3 | 0.6 : 9 |
| 0.7 : 29 | 0.7 : 4 |
| 0.8 : 8 | 0.8 : 8 |
| 0.9 : 01349 | 0.9 : 7799 |
| 1.0 : 0457 | 1.0 : 3 |
| 1.1 : | 1.1 : 8 |
| 1.2 : 7 | 1.2 : |

Fig. 6—Power-function exponents for judgment of circle sizes.

area and dollar estimates shows that neither the type of judgments, nor the order in which they are made, nor the presence of maplike frames is responsible for the high exponents.

One obvious difference between our experiments and many previous ones is that our subjects always judged circles within a simultaneously visible set, whereas in most psychophysical studies the stimuli are presented one at a time. It may be that higher exponents are obtained when the standard circle is visible along with the circles that are compared with it, instead of being displayed and then removed before the judgments are made. Examination of previous studies offers some support for this hypothesis; for several comparable studies (mostly by psychologists) in which the standard was not visible during the area judgments, the median exponent was 0.7, whereas in several studies (largely by cartographers) in which the standard was always present, the median exponent was 0.9.[7]

These findings suggest that when one is preparing statistical maps, it is probably better simply to make circle areas directly proportional to the quantities represented (exponent = 1) rather than to scale with a very different exponent, as a direct application of psychophysical studies of area judgments might suggest. Future research may enable us to specify what variables contribute to higher or lower exponents, but certainly our multiple-circle displays resemble statistical maps more closely than do circles viewed one at a time.

## IV. JUDGED ASSOCIATION IN SCATTERPLOTS

The judgments discussed so far—areas of circles or colored regions on maps—do not require technical training; indeed high school students' judgments proved to be quite similar to those made by scientists. The experiments to be described now, on the other hand, called for subjects with some statistical training. In these experiments[6] all subjects had at least a basic knowledge of statistics (university statistics students and faculty, and practicing statisticians). They were asked to assess the degree of linear association between two variables portrayed by a scatterplot. The most frequently encountered measure of linear association is the correlation coefficient, $r$. The value of $r$ ranges from 0, when there is no linear association, to +1 or −1, when the linear association is perfect and the plotted points fall on a straight line. In many basic statistics courses, $r$ is the only measure of correlation that is taught.

In the study with maps of Nevada, discussed earlier, perceptions of one aspect of the displays—area—was found to be strongly influenced by another aspect—color—which should have been irrelevant. A similar influence was found with judgments of association in scatterplots. The two scatterplots in Fig. 7, projected alternately on a screen, were

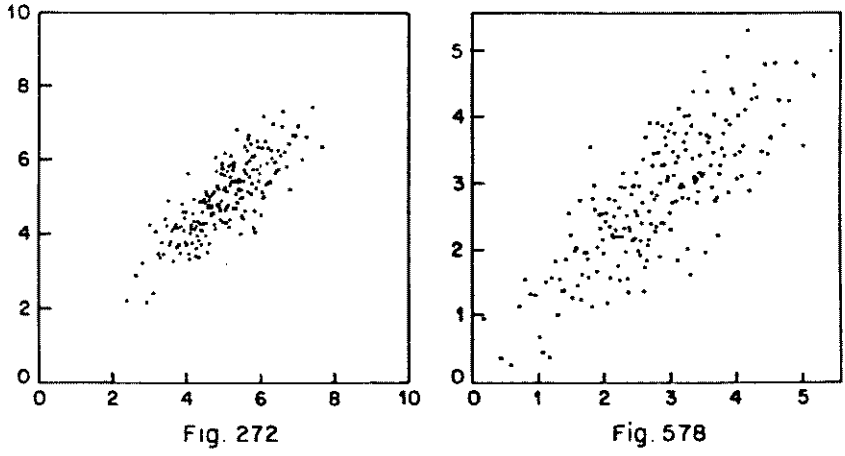



Fig. 7—(Left panel) Scatterplot with correlation $r = 0.8$. (Right panel) Scatterplot with same correlation as in left panel, but with x and y scales expanded.

shown to 109 subjects. They indicated the degree of association of the two variables by assigning a number from 0 to 100 to each plot. In fact, the correlation is identical for the two plots ($r = 0.8$); only the scale of the axes, and hence the size of the point cloud, is different. Judgments of the amount of association on the two plots should therefore be identical also. But they are not; judgments of the association portrayed by the left panel of Fig. 7 were generally higher than for the right panel.

To quantify this difference, each subject's estimate for the right panel of Fig. 7 was subtracted from the estimate for the left panel, and the 10-percent trimmed mean was calculated by dropping the largest 10 percent of the differences and the smallest 10 percent and taking the arithmetic mean of the remaining values. (Unlike means, 10-percent trimmed means are robust measures that are not influenced by extreme outliers. Ten-percent trimmed means can be thought of as a compromise between medians, which are nearly 50-percent trimmed means, and means, which are 0-percent trimmed.) The result (after dividing by 100 to bring the judgments into the range 0 to 1) was a difference of 0.068 between the panels of Fig. 7, with a standard error of 0.011. Thus, the estimated association was significantly higher for the smaller point cloud than for the larger one.

A second experiment corroborated this finding. The scatterplots in Fig. 7 were shown to 32 subjects who were told that the correlation coefficients were identical. The subjects were asked whether one plot *looked* more correlated than the other and if so, which one. Sixty-six percent of the subjects reported that the left panel looked more correlated than the right, 22 percent that they looked the same, and

only 13 percent that the right panel appeared more correlated. So even explicit information that the two correlations are identical does not dispel the illusory impression that the smaller point cloud depicts a higher correlation.

To obtain more detailed information on judgments of association another study was carried out. In this study 74 subjects each judged 19 scatterplots, assigning a number from 0 to 100 to denote the linear association. The 19 stimuli included 10 with the same axis scales, representing 10 different levels of association, plus an additional 9 with 3 different axis scales (that is, different point-cloud sizes) at 3 of those levels of association. A number of other attributes (such as number of points, standard deviations of the variables, sign of the correlation, and size of the square frame) were identical for all of the scatterplots.

The 10-percent trimmed means across subjects are plotted against actual correlation in Fig. 8. The circle radii portray the standard errors of the trimmed means. Thus the circle areas are proportional to the estimated variances of the trimmed means. The numbers to the left of the circles indicate the point-cloud sizes (1 is the smallest). When

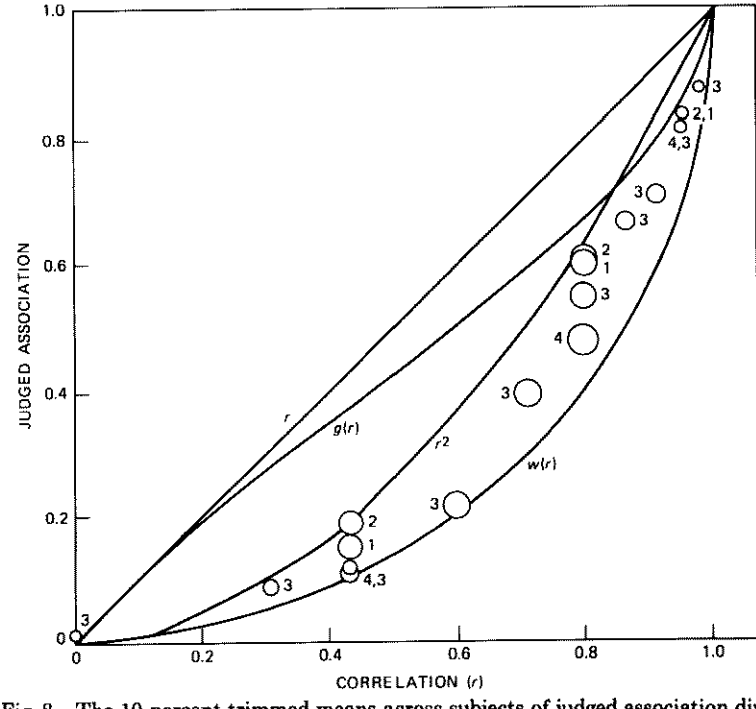



Fig. 8—The 10-percent trimmed means across subjects of judged association divided by 100 for 19 scatterplots are plotted (by the circle centers) against the values of $r$, the correlation coefficient, of the scatterplots.

two numbers are shown, separated by a comma, two circles are nearly coincident and the first number refers to the circle with the smaller trimmed mean. The information on the plot leads to two conclusions: judged association tends to be higher for smaller point-cloud sizes; judged association is not proportional to any of four standard numerical measures of association [the points do not fall on any of the curves, which plot $r$, $g(r)$, $r^2$, and $w(r)$ as a function of $r$]. Thus, the circles for $r$ near 0.4 and 0.8 corroborate the finding that smaller point clouds tend to elicit higher judgments of association.

The new information is that subjects' judgments are far from proportional to the usual measure of linear association, $r$. Since the actual correlation coefficient $r$ is given by the horizontal axis, if the subjects had been judging $r$ accurately, the data would have fallen on the 45-degree line. Instead, the points fall well below it. On the average a correlation of 0.4 was judged to be less than 0.2, a difference of more than a factor of 2.

Two other measures of linear association come closer to fitting the data: $g(r) = 1-\sqrt{(1-r)/(1+r)}$ and $w(r) = 1-\sqrt{1-r^2}$ (see Ref. 20). Unlike $r$, both $g(r)$ and $w(r)$ offer intuitively plausible geometric bases for visual judgments. If we draw the ellipse of the bivariate normal distribution that generated a scatterplot, as in Fig. 9, the ratio of the minor axis to the major axis is $(1-r)/(1+r)$. The smaller this ratio is, the higher the association; so $g(r)$ tells how narrow the ellipse is relative to a zero-correlation circle. The ratio of the area of the ellipse to the area of the rectangle circumscribed around it is $\pi/4\sqrt{1-r^2}$; so $w(r)$ tells how far the ellipse is from filling the rectangle. Unfortunately, neither $g(r)$ nor $w(r)$ fits the data very well, as can be seen from the departure of the data points from both curves in Fig. 8. Another measure, $r^2$, does not fit the data particularly well either. However, there is a two-parameter family of curves, $1-(1-r)^{\alpha}(1+r)^{\beta}$, that includes all four measures of association that we have mentioned (and many others as well). If such a curve is fitted to the data in Fig. 8, the estimates of $\alpha$ and $\beta$ and their standard errors are $0.71 \pm 0.04$ and $0.66 \pm 0.11$, respectively. These estimated parameter values fall between (and are significantly different from) those for $w(r)$ ($\alpha = 0.5$, $\beta = 0.5$) and for $r^2$ ($\alpha = 1$, $\beta = 1$). [For $r$, $\alpha = 1$ and $\beta = 0$; for $g(r)$, $\alpha = 0.5$ and $\beta = -0.5$.] This middle position of the parameter estimates is what we would expect since the circles in Fig. 8 lie between $r^2$ and $w(r)$.

Thus the parameters of the best-fitting curve have values that are significantly different from those of any of the four measures of association that we have considered.

On the face of it, the poor fit of the subjects' judgments to $r$, $r^2$, $g(r)$, and $w(r)$ seems to imply that these highly trained subjects do not base

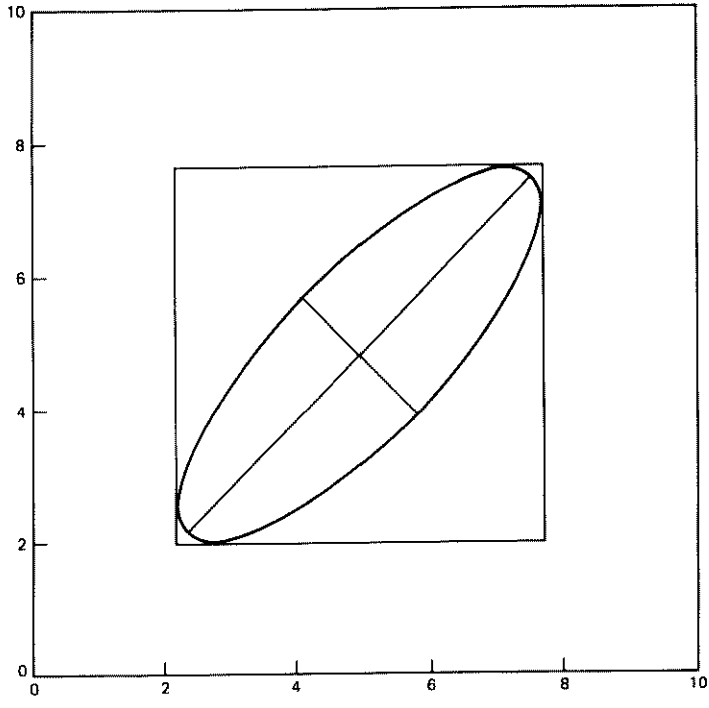



Fig. 9—Geometrical bases for $g(r)$ and $w(r)$.

their estimates of linear association on the visual counterparts of any of these standard measures. However, there is an alternative that remains to be explored. Perhaps subjects do make judgments (for instance, of the ratio of minor to major axis) that are appropriate for one of these measures of association, but misperceive the basic visual attributes on which the judgments are based. For example, if judgments of length were not directly proportional to actual length, then judgments of the ratio of minor to major axis of an ellipse would not be proportional to the actual ratio.

The finding that the size of the point-cloud can have a big effect on judged association means that the axis scaling that many statistical plotting programs apply automatically is not optimal for all purposes. To facilitate comparisons of the degree of association in different plots, it would be wise to make the point-cloud sizes similar; Cleveland and McGill (see Ref. 21) propose a way to do so. The size effect also suggests that when estimates of degree of association are required, the numerical value of a measure of association should also be computed. Even experienced statisticians can have judgments of association affected by extraneous factors.

## V. CONCLUSIONS

Behavioral studies of the kind summarized here are essential for devising guidelines for effective communication of quantitative information. These studies confirm that people can make consistent judgments of a wide variety of visual displays. However, those judgments may not match the usual physical measurements of stimulus attributes: People overestimated bright red areas on maps, relative to bright green areas, and judgments of the degrees of linear association in a scatterplot did not agree closely with any of four standard statistical measures of association [$r$, $w(r)$, $g(r)$, and $r^2$]. Other findings lead to some recommendations about how to convey quantitative relations more accurately. For example, for output from PRISM plotters, substituting pastel colors for saturated red and green reduced the biasing of area judgments. The finding that smaller point-clouds in a scatterplot are judged to portray higher linear association implies that to permit accurate comparisons of association, axes should be scaled to produce similar point-cloud sizes. And finally, our subjects' judgments of arrays of circles suggest that in statistical maps one should not scale to compensate for the distortions in area judgments that are found when stimuli are viewed one at a time; instead, symbol sizes should be directly proportional to the quantities represented.

With reliance on both old and new forms of visual display becoming increasingly widespread, we will need more behavioral studies like the ones summarized here to guide effective communication of quantitative information.

## VI. ACKNOWLEDGMENTS

We acnowledge helpful comments on earlier drafts by Francine Frome, Ram Gnanadesikan, Tom Landauer, Saul Sternberg, Paul Tukey, and several anonymous reviewers.

## REFERENCES

1. J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey, *Graphical Methods for Data Analysis*, Belmont, CA: Wadsworth, 1983 (hardcover); and Boston: Duxbury Press, 1983 (softcover).
2. C. F. Schmid and S. E. Schmid, *Handbook of Graphical Presentation*, New York: Wiley & Sons, 1979.
3. S. Coren and J. S. Girgus, *Seeing is Deceiving: The Psychology of Visual Illusions*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1978.
4. C. S. Harris (Ed.), *Visual Coding and Adaptability*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1980.
5. S. S. Stevens, *Psychophysics—Introduction to its Perceptual, Neural and Social Prospects*, New York: Wiley, 1975.
6. W. S. Cleveland, P. Diaconis, and R. McGill, "Variables on Scatterplots Look More Highly Correlated When the Scales are Increased," Science, *216* (June 1982), pp. 1138–41.
7. W. S. Cleveland, C. S. Harris, and R. McGill, "Judgments of Circle Sizes on Statistical Maps," J. Amer. Stat. Assn., *77* (September 1982), pp. 541–7.

8. H. Wainer and D. Thissen, "Graphical Data Analysis," Ann. Rev. Psychol., *32* (1981), pp. 191–241.
9. L. Wilkinson, "An Experimental Evaluation of Multivariate Graphical Point Representations," Proc. Conf. on Human Factors in Computer Systems, Gaithersburg, Maryland, March 15–17, 1982, pp. 202–9.
10. W. Kruskal, "Visions of Maps and Graphs," Proc. Int. Symp. on Computer-Assisted Cartography, September 21–25, 1975, Amer. Congress on Survey and Mapping and the U.S. Bureau of the Census (1977), pp. 27–36.
11. W. Kruskal, "Criteria for Judging Statistical Graphics," Utilitas Mathematica, *21B* (May 1982), pp. 283–310.
12. W. S. Cleveland and R. McGill, "A Color-Caused Optical Illusion on a Statistical Graph," *The Amer. Statistician*, (1983a), in press.
13. A. H. Munsell, *The Munsell Book of Color—Glossy Finish Collection*, Baltimore: Munsell Color Co., 1966.
14. F. Mosteller and J. W. Tukey, *Data Analysis and Regression*, Reading, Massachusetts: Addison-Wesley, 1977.
15. P. J. Huber, "Robust Estimation of a Location Parameter," Ann. Math. Stat., *35* (February 1964), pp. 73–101.
16. B. Efron, "Bootstrap Methods: Another Look at the Jackknife," Ann. of Stat., *7* (January 1979), pp. 1–26.
17. J. J. Flannery, "The Relative Effectiveness of Some Common Graduated Point Symbols in the Presentation of Quantitative Data," Canadian Cartographer, *8* (December 1971), pp. 96–109.
18. G. F. McCleary, Jr., "In Pursuit of the Map User," Proc. Int. Symp. on Computer-Assisted Cartography, Auto-Carto II, (1975), pp. 238–50.
19. J. W. Tukey, *Exploratory Data Analysis*, Reading, Massachusetts: Addison-Wesley, 1977.
20. M. B. Wilk, unpublished work.
21. W. S. Cleveland and R. McGill, unpublished work.

## AUTHORS

**William S. Cleveland,** A.B., 1965 (Mathematics), Princeton; M.S. and Ph.D., 1969 (Statistics), Yale; Bell Laboratories, 1972—. Mr. Cleveland works on the development of statistical methodology. His current research areas include time series analysis, seasonal adjustment, economic forecasting, atmospheric chemistry, graphical methods for data analysis, software systems for presentation graphs, and graphical perception. Mr. Cleveland is co-author of a book *Graphical Methods for Data Analysis* that was published in 1983. Member, AAAS, National Computer Graphics Association, American Statistical Association (Fellow, Chairman of Statistical Graphics Committee); elected member, International Statistical Institute.

**Robert McGill,** Bell Laboratories, 1969—. Mr. McGill is a Member of Technical Staff in the Statistics and Data Analysis Research Department. Currently, he is working on graphical techniques and perception of statistical graphics. Member, American Statistical Association, Association for Computing Machinery, National Computer Graphics Association, International Association for Statistical Computing, American Association for Advancement of Science.

**Charles S. Harris,** B.A., 1959, Swarthmore College; M.S., 1961, Yale University; Ph.D. (Experimental Psychology), 1963, Harvard University; Assistant Professor, University of Pennsylvania, 1964–1966; Bell Laboratories, 1966—. Mr. Harris has carried out research on various phenomena of visual and kinesthetic perception, including motion illusions, adaptation to visual distortions, selective attention, color and spatial-frequency aftereffects, and perceptual facilitation by object-like context. He has edited a book entitled *Visual Coding and Adaptability.* He is currently a member of the Human Information-Processing Research Department. Member, Sigma Xi, Phi Beta Kappa, Psychonomic Society, Eastern Psychological Association.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Retrospective Reports Reveal Differences in People's Reasoning

By D. E. EGAN*

(Manuscript received January 14, 1982)

Reasoning for a class of transitive inference problems was studied and the following questions were experimentally investigated: (1) Can people give reliable retrospective reports about their reasoning processes? (2) Do people who report different reasoning processes actually reason in different ways? (3) Can people be trained to use different reasoning processes? In the situations studied, subjects' retrospective reports about reasoning contained sufficient information to classify the subjects reliably. Subjects classified as using different reasoning strategies made different amounts and different kinds of reasoning errors. As a result of training, subjects could use reasoning processes that they would not have used spontaneously. These results have implications for developing theories of reasoning and for assessing and modifying reasoning-like processes in practical situations.

## I. INTRODUCTION

Reasoning, the ability to draw conclusions or inferences from given information, is a prized intellectual talent. Reasoning is known to be associated with successful learning of mathematics.[1] A "reasoning

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

factor" plays a prominent role in reading comprehension.[2] Practical problem-solving tasks such as balancing a checkbook or troubleshooting equipment undoubtedly involve reasoning. Because of its pervasive importance, tests of reasoning have been included in virtually every battery of aptitude tests, and experimental psychologists have studied reasoning extensively in various forms.

Reasoning may play an especially prominent role in the future as more occupations and everyday tasks involve computers. For example, reasoning test scores to some extent predict people's success at computer programming,[3] an occupation that is growing in importance. Reasoning probably is required to use the host of new computer-like devices ranging from automated banking machines and appliance timers to systems for editing and data retrieval. The small bit of "programming" required to activate many new telephone services appears to require mental processes akin to reasoning.

### *1.1 Three basic questions about reasoning*

This paper presents results of basic psychological research concerning three questions about reasoning. The first of these questions is, "Can people give reliable retrospective reports about their reasoning processes?" This question is important, because it asks whether a potentially useful kind of data about the reasoning process meets the scientific standard of repeatability. Using subjective reports to analyze reasoning processes poses much more severe problems of reliability than, say, using meter readings to analyze physical processes.

To be useful data, retrospective reports about reasoning processes must, as a minimum, admit to consistent classification. Reports should contain enough information so that two raters would classify a given report in the same category. Different methods of reporting (e.g., verbal reports and nonverbal reports such as drawings or checklists) should agree with each other so that a reasoner would be classified in the same category no matter which particular method of reporting is used. The classification of reports given by the same person on different occasions should be consistent, as long as other indicators show that the person is using the same reasoning process on those occasions. Retrospective reports must meet each of these requirements to be considered reliable indicators of people's reasoning processes.

A second fundamental question considered here is, "Do people who report different reasoning processes actually reason in different ways?" If different people actually reason in different ways consistent with their reports, two conclusions follow. One conclusion would be that retrospective reports about reasoning are valid evidence about the reasoning process. Reports could be valid simply if people who give a particular kind of report tend to use a particular reasoning process,

even if the content of the report does not accurately describe the process. Formally, such reports would have validity because their content would be correlated with performance. A more powerful and interesting kind of validity would require that the content of people's reports accurately reflects how they reason.

Another conclusion would be that data (e.g., reasoning errors) from people giving different kinds of reports do not result from a common underlying reasoning process. To understand reasoning, people using a given reasoning process first may have to be identified and their reasoning data grouped together. Models for the reasoning process appropriate to each group then could be developed.

A very practical aspect of this question is that retrospective reports are often the most easily obtainable data and sometimes they are the only data available in practical situations involving mental processes like reasoning. For example, people's reports about how they do a task or how they use a complex device often are used to evaluate new services or products requiring human interaction. It would be advantageous to have at least one assessment of the validity of these kinds of reports, and to know whether different people are likely to use different reasoning processes spontaneously. The extent to which laboratory studies of reasoning generalize to specific practical situations involving reasoning is not known. However, one benchmark for the validity and variety of retrospective reports about reasoning can be obtained in laboratory studies where it is possible to validate reports.

The third question about reasoning addressed in these studies is, "Can people be trained to use different reasoning processes?" This question introduces a distinction between the reasoning processes people might use spontaneously, and those they could use if trained to do so. In particular, for a specific kind of problem people may be able to follow directions to use a certain reasoning process, even if they would not use that process if left to themselves.

This question has theoretical and practical interest. To the extent that training can induce people to use different reasoning processes, reasoning cannot be conceptualized simply as a stable ability that some people have more of or are better at doing. Instead, a theory of reasoning must explain what factors cause people to discover and use a particular reasoning process when they are competent to use others. Moreover, if people can be trained to reason in different ways, then they probably can be trained to think in various ways about practical tasks involving reasoning. Directions on "how to think about" a task may be an effective aid for learning new procedures (perhaps activating telephone services or interacting with computers) that require reasoning.

### *1.2 Reasoning problems used in these studies*

In all of the studies presented here, people solved reasoning problems involving a simple transitive inference. The problems, known as "three-term series problems," have a common basic form. Examples of the problems are given in Table I.

Three-term series problems were selected for these studies over many other possible kinds of reasoning problems for the following reasons. First, these problems require no specialized content knowledge and therefore seem to be good tools for examining "pure reasoning" uncontaminated by whatever specific information people might know about a problem. Second, the simple verbal structure of problems like those in Table I makes it easy to manipulate certain problem characteristics while controlling others (see below). Finally, three-term series problems have been studied extensively. They have been used on standard tests as markers for deductive reasoning ability[4,5] and much is already known about factors that influence the difficulty of these problems.[6]

Each three-term series problem consists of two premises followed by a question. The premises state a relationship between two of the three terms in the problems. The question concerns the inferred relationship between the remaining pair of terms.

In describing three-term series problems, an important distinction must be made between *relations* and *inverses*. This distinction is based on well-substantiated findings demonstrating that one member of a pair of opposite relational words typically produces better performance than the other member. For example, using the word "above" results in more rapid and accurate performance than the word "below" in a large range of tasks. Similarly, the word "fatter" leads to better performance than "thinner," etc. The psycholinguistic theory of lexical marking[7] may account for some of these differences. For present purposes, an inverse is defined as that member of a pair of opposing relational words that is known to cause the greater difficulty. Conse-

Table I—Sample of the reasoning problems used

| Positional Relations | Visual Comparative Relations |
|---|---|
| Triangle is above circle.<br>Square is below circle.<br>Is triangle above square? | Square is smoother than triangle.<br>Circle is smoother than square.<br>Is triangle smoother than circle? |
| Circle is left of triangle.<br>Square is left of circle.<br>Is triangle left of square? | Circle is darker than square.<br>Circle is lighter than triangle.<br>Is triangle darker than square? |
| Square is in back of triangle.<br>Square is in front of circle.<br>Is triangle in front of circle? | Triangle is fatter than circle.<br>Triangle is thinner than square.<br>Is circle thinner than square? |

quently, "above" will be referred to as a relation, but "below" will be referred to as an inverse. Similarly, "fatter" is called a relation, but "thinner" is called an inverse.

Having thus defined relations and inverses, the terms of the problems will be distinguished as follows: the "A term" is the initial term in the linear order established by the premises (e.g., when using the relation "rougher" or its inverse "smoother," the A term would be the roughest); the "B term" is the pivot, or middle term; the "C term" is the end term of the linear order (e.g., the smoothest when using rougher/smoother).

Several other details of the problem set are noteworthy. The terms and relationships of each problem were selected to encourage the use of spatial mental representations. The terms were the geometric figures circle, square, and triangle. The relationships involved either positional comparisons (relation/inverse pairs above/below, right of/left of, or in front of/in back of), or nonpositional visual comparisons (rougher/smoother, darker/lighter, or fatter/thinner). For each relation/inverse pair, 16 different problem types were generated. These different problem types resulted from a $2^4$ factorial combination of the following factors: (1) the use of a relation or inverse in the premise relating the A term and B term; (2) the use of a relation or inverse in the premise relating the B term and C term; (3) the order of the premises; and (4) the use of a relation or inverse in the problem question. The pattern of reasoning errors made on these different types of problems will be analyzed to test whether various reasoning processes are being used.

## II. CAN PEOPLE GIVE RELIABLE RETROSPECTIVE REPORTS ABOUT THEIR REASONING PROCESSES?

### *2.1 Approach*

Regarding the question of the reliability of reports about reasoning, we will consider data from two experiments.[8] In both studies, high school students solved a large number of three-term series problems, and then described their reasoning processes. In the first study, 12 subjects each solved 384 problems. In the second study, 100 subjects each solved 192 problems.

The problems were presented on an audio-tape-playback machine. Each problem began with a speaker saying "Next," then reading the problem, and then pausing five seconds to allow the subject to answer before beginning the next problem. Subjects answered problems by crossing out "Yes" or "No" or "?" next to the number of the problem on an answer sheet. They were not allowed to write anything else. Answers were scored as errors if either the wrong response or a "?" was used.

Several methods of retrospective reporting were used. In the first experiment, subjects were interviewed by the experimenter, and these interviews were tape-recorded and analyzed. In the second experiment, subjects gave written reports—essentially the equivalent of the oral reports in the first study. Following the written reports, subjects then were asked to draw pictures representing how they thought about the problems. As a final method of reporting, subjects were shown two written descriptions of reasoning processes that attempted to capture the two most common kinds of reports found in the first study. Subjects had to choose which of the two descriptions more closely matched their own way of reasoning.

### *2.2 What people reported*

The most striking aspect of subjects' reports was that different subjects claimed to use quite different sets of processes or *strategies* to deal with simple three-term series problems. The difference was most apparent for reports about the visual comparative relations rougher/smoother, darker/lighter, and fatter/thinner. Some subjects claimed to establish an order for the three geometric figures no matter what relation was used. For these subjects, "rougher," for example, would be identified with one end of a vertical or horizontal scale, as would "darker" and "fatter". Subjects described making the transitive inference by mentally arranging the geometric figures roughest to smoothest, darkest to lightest, etc. Other subjects claimed to attribute physical properties to the geometric figures in the case of the visual comparative relations. For example, these subjects described their representation of a rougher triangle as an image of a triangle having a roughly textured surface, or a lighter circle as a picture of a very bright round object. These subjects described making the transitive inference by scanning the images. Distinctions among reports about positional relations were more subtle (see below). Actual examples of written reports are given in Table II.

Reasoning data from subjects who reported similar reasoning strategies were grouped together. The rule for assigning subjects to groups was that if a subject described a representation that clearly involved physical properties for any relation, then the subject was placed in a group labeled "Concrete Properties Thinkers." A subject who claimed to use an ordered mental array for every relation was placed in another group labeled "Abstract Directional Thinkers."

Using this rule, all subjects in the first experiment could be classified into either the Concrete Properties Group ($N = 5$) or the Abstract Directional Group ($N = 7$). In the second experiment, the consensus rating of written reports by two judges identified 18 subjects as

Table II—Examples of written retrospective reports

| | | |
|---|---|---|
| **Abstract Directional Thinkers** | | |
| Rougher–Smoother | S#005: | "Rather than imagining a rough/smooth figure, I put the figures in a horizontal line, in my mind, in the order of left/right rather than rough/smooth." |
| | S#049: | "I pictured the objects in my mind in a line of sequence." |
| Darker–Lighter | S#003: | "I set up a scale with the lightest on the far right and darkest on the far left and placed the figures on their appropriate spots." |
| | S#051: | "Placed them in a line up and down, darkest being on top." |
| Fatter–Thinner | S#086: | "I also used a mental horizontal grid for this relation with the left side of the grid being the 'thin end' and the right side the 'fat end.'" |
| | S#099: | "Put shapes in order from thinner to fatter." |
| **Concrete Properties Thinkers** | | |
| Rougher–Smoother | S#008: | "I also drew a picture, and if something was rough—I would put craters in it in my mind—smooth was just plain white." |
| | S#098: | "The picture came to mind of corners and smooth edges, then the question was solved." |
| Darker–Lighter | S#062: | "In my mind, I 'colored in' the object that was darkest." |
| | S#080: | "I listened to the problem and tried to solve it mentally, at times picturing the objects colored in or not." |
| Fatter–Thinner | S#022: | "This (fatter/thinner problem) was hard. I had to think of the shapes as squeezed or pulled." |
| | S#100: | "Made them (the figures) fatter and thinner in my head." |

Concrete Properties Thinkers, and 42 as Abstract Directional Thinkers.

The classification rule identifying a subject as a Concrete Properties Thinker on the basis of a single concrete report was motivated by simplicity. Subsequent analyses suggest that the rule, while admittedly crude, did manage to separate people into two groups that used a particular set of reasoning processes fairly consistently. First, subjects who reported a concrete representation for one relation were very likely to report using such a representation for other relations. For example, 17 of 18 subjects identified as Concrete Properties Thinkers in the second study gave reports having a concrete representation for two or more relations. Second, in the statistical analysis of reasoning errors, reasoning groups did not interact with the different relation/inverse pairs, suggesting that each group consistently used one reasoning process. Third, a key-word analysis of the written retrospective reports suggests that the two groups also handled positional relations differently. People classified as Concrete Properties Thinkers used more words in their reports that suggest the use of a visual image (variants of the words "picture" and "draw") for positional problems

(e.g, "I pictured the objects in a row"). People classified as Abstract Directional Thinkers used more words suggesting the use of an order-preserving scale (variants of the words "put", "order", "line," and "horizontal/vertical") for positional problems (e.g., "I put the objects in order on a horizontal line"). This interaction of key-word types and reasoning groups was statistically reliable.

To summarize, the majority of people reported one of two sets of reasoning processes or strategies for the three-term series problems. Some people claimed to preserve the information in the premises of at least some problems by means of an image capturing the visual features or stated position of the geometric terms. Others claimed to preserve the information in all premises by means of an abstract ordering of the geometric terms.

### *2.3 Reliability of retrospective reports*

Several methods were employed to assess various aspects of the reliability of the retrospective reports. The first, alluded to earlier, was to assess the agreement between two different judges who categorized subjects on the basis of their written reports in the second experiment. Each judge classified the 100 subjects as Abstract Directional, Concrete Properties, or Other/Not Clear. Table III shows that the two judges agreed on the classification of 82 percent of the subjects. Almost all cases of disagreement occurred when one judge classified a subject as using one of the two identified strategies, but the other judge classified the subject as using an Other/Not Clear strategy. Compared to several additional studies[9] the consensus shown in Table III is the "worst case." Other estimates of interjudge agreement have ranged up to 95 percent.

A second analysis assessed the agreement among different methods of reporting in the second experiment. The pictures drawn and forced-choice strategy selections made by subjects were compared to the classification of their written reports. Pictures were classified as indicating the Concrete Properties strategy if they depicted geometric objects with altered physical properties (e.g., a pockmarked surface

Table III—Classification of written reports by two judges (Experiment II)

| | 2nd Judge's Categories | | |
|---|---|---|---|
| 1st Judge's Categories | Concrete Properties | Abstract Directional | Other/Not Clear |
| Concrete Properties | 18 | 0 | 6 |
| Abstract Directional | 1 | 42 | 5 |
| Other/Not Clear | 3 | 3 | 22 |

depicting "rougher," shading depicting "darker," etc.). Pictures showing horizontal or vertical orderings of rather standard geometric figures were classified as indicating the Abstract Directional strategy. The classification of drawings agreed with the classification of written reports for 56 of the 60 subjects (93.3 percent) whose written reports had been classified by consensus as Abstract Directional or Concrete Properties. The analysis of forced-choice strategy selections showed that 51 of the 60 subjects (85 percent) chose the strategy description consistent with the classification of their written report.

To assess the long-term stability of reported reasoning strategies, 38 subjects who participated in the second experiment and who had been classified by consensus as Abstract Directional or Concrete Properties Thinkers were recalled six months later for another study. After solving some warm-up problems, subjects gave written reports describing their reasoning strategies. These reports were classified in the previously described manner and this classification was compared to the classification of the subjects performed six months earlier. The results in Table IV indicate that subjects' reports about reasoning have some, but not perfect, stability over time and across different presentation conditions (the former reports were given after listening to problems, the latter after reading problems). The stability of verbal reports estimated by the four-fold point correlation based on Table IV is $r = 0.59$ ($p < 0.01$). Specifically, 95-percent of the subjects earlier classified as Abstract Directional again reported that strategy, but only 59 percent of the original Concrete Properties Thinkers reported that strategy again six months later. The instability of the latter group may have been due to unreliable reports or classification procedures on one hand, or actual changes in reasoning strategies[9] on the other.

### *2.4 Summary*

The reliability of retrospective reports about reasoning has been established in that: (1) different judges show considerable agreement on how to classify reports, (2) the classification of written reports agrees to a large extent with classifications based on other nonverbal methods of reporting, and (3) the classification of reports given by

Table IV—Number of subjects using strategies initially and six months later

| | Strategy Used Six Months Later | |
|---|---|---|
| Strategy Used Initially (Experiment II) | Abstract Directional | Concrete Properties |
| Abstract Directional | 20 | 1 |
| Concrete Properties | 7 | 10 |

people at different times has some stability. On the other hand, retrospective reports about reasoning or perhaps the present procedures for classifying reports are not perfectly reliable. Compared to paper-and-pencil tests having finely graded scores, the reliability of retrospective reports is somewhat low. In particular, the classification of reports given at different times and under different conditions of presenting problems is not always the same. Despite these difficulties, the great majority of retrospective reports about reasoning in these studies contain sufficient information to be classified consistently. This is a necessary condition for reports to be useful in exploring the process of reasoning.

## III. DO PEOPLE WHO REPORT DIFFERENT REASONING PROCESSES ACTUALLY REASON IN DIFFERENT WAYS?

### *3.1 Approach*

Reasoning errors from the two studies previously described will be used to analyze the validity of retrospective reports about reasoning and to gain further understanding of reasoning processes. Error data from the two groups of subjects reporting different reasoning strategies will be compared at successively finer levels of detail. The overall error rates from the two groups will be analyzed first. Then, general patterns of interaction in the error data will be discussed. Next, the effects of specific problem factors hypothesized to affect a particular reasoning process will be tested. Finally, two models of the process of making a transitive inference will be described and tested. The goal of this section is to demonstrate that different models of the reasoning process are required to account for reasoning errors made by the two groups of subjects who reported different reasoning strategies.

### *3.2 Differences in reasoning errors between groups reporting different strategies*

The first attempt at assessing the validity of retrospective reports asked whether the overall reasoning error rate was different for people giving different reports. In the two studies described above, subjects giving Abstract Directional reports made significantly fewer errors than those giving Concrete Properties reports. In the first study, Abstract Directional Thinkers had an error rate of 10.3 percent compared to the Concrete Properties Thinkers' error rate of 38.0 percent. In the second study, the corresponding error rates were 21.0 percent and 27.9 percent. This difference in error rate favoring the Abstract Directional subject now has been found repeatedly.[10] One interpretation of this result is that the Abstract Directional strategy is more efficient for the transitive inference problems used in these studies.

A further study[9] required subjects to describe their reasoning processes at a number of points in a lengthy sequence of three-term series problems. In that study, a change in the strategy reported by a subject was accompanied by a corresponding change in the reasoning error rate. Thus, if a subject reported shifting from the Concrete Properties strategy to the Abstract Directional strategy, the subject's performance improved. For subjects reporting no shift in reasoning strategy, reasoning performance was fairly stable at a low or high level, depending on the strategy reported. This study provides evidence of the validity of different reports given by the same subject. It also suggests why reports by some subjects changed after six months in the previous study: the subjects' reasoning processes may have changed over time.

People reporting different reasoning strategies not only exhibited different overall levels of reasoning errors, but they also exhibited different patterns of reasoning errors. This fact is demonstrated in a general way by a statistically reliable interaction between Report Groups and Problem Types found in the two original experiments.[8] This interaction means that factors causing reasoning problems to be more or less difficult in one group of reasoners were not the same as the factors causing difficulty in the other group. People who gave different reports made different amounts and different kinds of reasoning errors.

Thus far, the validity of reports has been assessed in a formal but rather indirect way. At the next level of detail, we might ask whether the patterns of reasoning errors made by subjects are consistent with the reasoning process subjects claim to be using. Consider reports of Abstract Directional Thinkers who claim to construct an ordered mental array of the geometric terms used in these problems. Previous theories[11,12] have asserted that it should be easier to construct a direct spatial array from the ends toward the middle, rather than from the middle outward. This so-called "end-anchoring principle" leads to a prediction regarding the difficulty of solving various types of three-term series problems. For Abstract Directional Thinkers, reasoning errors in a problem should be directly related to the number of premises that have the middle or pivot term stated first. For people using the Concrete Properties strategy, the end-anchoring principle should be irrelevant.

This prediction was confirmed by patterns of reasoning errors. In both studies, reasoning error rate for Abstract Directional Thinkers was a monotonic function of the number of premises in a problem that began with the middle or pivot term. This factor accounted for highly significant amounts of the variance in the difficulty of different types of problems for Abstract Directional Thinkers (82.2 percent of the variance in Experiment I, 71.8 percent in Experiment II). Data for

Concrete Properties Thinkers were quite different. Reasoning error rate was not a monotonic function of the number of pivot-first premises in either experiment, and this factor accounted for much smaller amounts of the variance in problem difficulty for this group of subjects (17.6 percent in Experiment I, and 20.3 percent in Experiment II). This analysis suggests that Abstract Directional Thinkers are constructing a mental array as they report, and that Concrete Properties Thinkers are reasoning in a different way.

Other analyses of specific problem factors[8] have found that reasoning errors by Concrete Properties Thinkers depend on the number of inverses used in a problem (e.g., using words like "smoother" and "thinner"), as well as the number of times a relation and inverse are alternated in the statement of a problem. Inverses cause extra difficulty for Concrete Properties Thinkers because concrete representations of a property like smoothness may not be as easy to generate as concrete representations of a property like roughness. Abstract Directional Thinkers were less sensitive to such factors. The most difficult kind of problem for Concrete Properties Thinkers was one with premises like, "Circle is rougher than square. Circle is smoother than triangle." In such problems, the Concrete Properties Thinker presumably imagines first a rough circle next to a square, and then imagines a smooth circle next to a triangle. Such problems are difficult to answer when this inconclusive image is scanned.

### *3.3 Two models of reasoning*

Models attempting to capture the reasoning process used by Abstract Directional and Concrete Properties Thinkers are presented in Tables V and VI, respectively. These models try to give a coherent account of the patterns of reasoning errors and the kinds of reports given by subjects. Each model contains parameters representing hy-

Table V—Model for abstract directional thinkers

| Process | Problem Factors | Model Parameters |
|---|---|---|
| 1. Encode Premise 1 | | |
| 2. Establish abstract scale | | |
| 3. Arrange first two terms placing grammatical subject first on scale | | |
| 4. Encode Premise 2 | | |
| 5. Find third term | Is third term grammatical subject or object? | SEARCH |
| 6. Position third term | Does third term fall in "Natural" next position? | POSITION |
| 7. Encode question | | |
| 8. Scan the scale | | |
| 9. Respond | | |

Table VI—Model for concrete properties thinkers

| Process | Problem Factors | Model Parameters |
|---|---|---|
| 1. Encode Premise 1 | | |
| 2. Generate Image Pair 1 by assigning property to grammatical subject | Is difficult (inverse) relation used? | GENERATE |
| 3. Encode Premise 2 | Is relation the same as that in #1? | ENCODE |
| 4. Generate Image Pair 2 by assigning property to grammatical subject | Is difficult (inverse) relation used? | GENERATE |
| 5. Encode question | Is relation the same as that in #3? | ENCODE |
| 6. Scan images | Are the images conclusive? | SCAN |
| 7. Respond | | |

pothetical mental processes that are executed various numbers of times depending on the structure of a specific type of problem.

### *3.3.1 The abstract directional model*

Abstract Directional Thinkers (see Table V) are assumed to encode the first premise and establish a mental scale for a problem. Then, the two terms stated in the first premise are arranged on the scale, the grammatical subject being placed first. The second premise is then encoded, and the subject searches for the third, or missing, term. This search is easier if the third term is the grammatical subject rather than the object of the second premise. The value of the SEARCH parameter (0 or 1, respectively) reflects this difficulty, and accounts for the effect of starting the second premise with the pivot term. Next, the third term is positioned on the mental scale, and it is assumed that there are three distinct cases for this operation. The easiest case (POSITION = 0) occurs when the third term is placed next in the sequence established by the first two terms. For example, if the first two terms are arranged smooth → rough, positioning the third term is easiest if it is roughest. If the first two terms are placed rough → smooth, then positioning the third term is easiest when it is the smoothest. Two more difficult cases exist and correspond to problems beginning with a pivot-first premise. In the easier of these cases (POSITION = 1) the third term does not fall next in sequence, but instead it must be placed at the end of the scale associated with the relation. In the remaining, most difficult case (POSITION = 2), the third term again does not fall in sequence, but must be positioned at the end of the scale associated with the inverse.

### *3.3.2 The concrete properties model*

The model for Concrete Properties Thinkers suggests that these subjects generate and compare images of objects having the stated properties. For each premise, Concrete Properties Thinkers (Table

VI) are assumed to encode the premise and then generate an image pair in which the grammatical subject takes on the property stated in the premise, while the grammatical object remains neutral. After two such pairs have been generated, the question is encoded, and then the two image pairs are scanned for the answer. Differences in difficulty among problems are assumed to arise from three sources, each corresponding to a parameter in the processing model for Concrete Properties subjects. One kind of difficulty has to do with whether the relation or inverse is used in each premise. Using an inverse presumably makes the appropriate image pair more difficult to generate. For a given problem, the parameter GENERATE takes on a value equal to the number of difficult images required (0, 1, or 2). The parameter ENCODE reflects the difficulty of alternately accessing a relation and its inverse. This parameter equals the number of alternations between a relation and inverse as a problem is read (0, 1, or 2). Finally, the parameter SCAN reflects the difficulty of dealing with images that are inconclusive. As noted previously, problems in which the B term takes on a property in one image pair and then takes on the inverse property in the other pair are especially difficult for Concrete Properties Thinkers. Such problems produce inconclusive image pairs in which the A and C terms are both neutral. Confronted with this type of problem, Concrete Properties Thinkers may guess or reformulate one of the premises to arrive at an answer. The SCAN parameter has the value 1 for such problems and 0 otherwise.

### *3.4 Comparing models to data*

The two models were compared to the data of Abstract Directional and Concrete Properties Thinkers from each experiment. The proportion of variance in problem difficulty uniquely associated with each parameter in each model was determined by stepwise multiple regression. The data are shown in Table VII. For both experiments, the Abstract Directional model was the better predictor of performance for Abstract Directional Thinkers, while the Concrete Properties model was the better predictor for Concrete Properties Thinkers. If errors on the various problem types are combined across experiments, the Abstract Directional model accounts for 90.3 percent of the variance in problem difficulty for Abstract Directional Thinkers (the Concrete Properties model accounts for 80.4 percent of the variance for this group). Both the SEARCH and POSITION parameters account for significant and unique portions of variance in problem difficulty for Abstract Directional Thinkers. The Concrete Properties model accounts for 80.1 percent of the variance in problem difficulty for the combined Concrete Properties Thinkers (the Abstract Direc-

Table VII—Proportion of variance* in problem difficulty attributable to parameters of two models

| Parameters | Experiment I | | Experiment II | | Experiment I + II | |
|---|---|---|---|---|---|---|
| | Abstract Group | Concrete Group | Abstract Group | Concrete Group | Abstract Group | Concrete Group |
| Abstract Directional Model | | | | | | |
| 1. SEARCH | 0.576† | 0.021 | 0.668† | 0.105 | 0.668† | 0.076 |
| 2. POSITION | 0.288† | 0.305‡ | 0.208† | 0.201 | 0.235† | 0.311‡ |
| $\sum R^2$ | 0.864† | 0.326 | 0.876† | 0.306 | 0.903† | 0.387‡ |
| Concrete Properties Model | | | | | | |
| 1. SCAN | 0.794† | 0.372‡ | 0.724† | 0.374‡ | 0.768† | 0.474† |
| 2. GENERATE | 0.001 | 0.185‡ | 0.016 | 0.166‡ | 0.011 | 0.221† |
| 3. ENCODE | 0.024 | 0.079 | 0.024 | 0.087 | 0.025 | 0.106‡ |
| $\sum R^2$ | 0.819† | 0.636† | 0.764† | 0.627† | 0.804† | 0.801† |

* These proportions are increments in $R^2$ values due to each parameter. The order in which the parameters are given corresponds to the step at which they entered the regression equation for the group of subjects appropriate to a particular model.

† $p < 0.01$

‡ $p < 0.05$

tional model accounts for 38.7 percent of the variance for this group), and each of the parameters SCAN, GENERATE, and ENCODE accounts for significant and unique variance.

Two aspects of the results of the model-fitting procedure should be clarified. First, the reliability of the error rates on the 16 problem types imposes a theoretical upper limit on the amount of variance for which any model can account. Therefore, it is important to estimate the data's reliability and compare that estimate to the $R^2$ of the best-fitting model.

Reliability estimates suggest that it would be difficult to improve the fits of the models appropriate to each group of subjects in Table VII. The estimated reliability of the Abstract Directional data combined across experiments was 0.951, so the Abstract Directional model ($R^2 = 0.903$) accounted for 0.903/0.951 or 95.0 percent of the reliable variation in the Problem Type data for those subjects. The fit of the Concrete Properties model ($R^2 = 0.801$) actually slightly exceeded the theoretical upper limit of the reliability of the combined Concrete Properties data (estimated reliability was 0.727).

Second, it is important to note that certain parameters of the two models are correlated in the 16 problem types used. The most important example of this confounding occurs for the parameter SCAN in the Concrete Properties model, which is correlated with both the SEARCH and POSITION parameters in the Abstract Directional model. These correlations account for the contribution of SCAN to variance in problem difficulty for Abstract Directional subjects. This interpretation of the contribution of SCAN is consistent with the fact that it is the only parameter in the Concrete Properties model that correlates with performance for Abstract Directional subjects, and that the two-parameter Abstract Directional model accounts for more variance in that group than the three-parameter Concrete Properties model.

### *3.5 Summary*

Retrospective reporting under the conditions studied here apparently is one case in which reports about reasoning processes contain valid information. The fact that the two groups of people who reported different reasoning strategies actually reasoned in different ways is supported by (1) the different overall levels of reasoning errors made by the two groups, (2) the different general patterns of reasoning errors made by the groups, (3) the differential effects of specific problem factors hypothesized to influence difficulty in one group but not the other, and (4) the fits of different process models of reasoning to the reasoning error data of the two groups.

## IV. CAN PEOPLE BE TRAINED TO USE DIFFERENT REASONING PROCESSES?

### *4.1 Approach*

A further study[9] dealt with the question of whether people can be trained to use different reasoning processes. In that study, 65 adult women solved a small number of three-term series problems and reported their reasoning processes. As in the experiments described previously, these reports identified the reasoning strategies that subjects spontaneously used. The subjects were then randomly assigned to two groups. One group received training in applying the Abstract Directional strategy to a new set of three-term series problems. The other group was trained to apply the Concrete Properties strategy to the same problems. Reasoning errors made by the two groups of subjects after receiving training were compared.

### *4.2 Training in reasoning strategies*

The training consisted of short descriptions of the models in Tables V and VI and examples showing how to apply them. The training was tailored specifically to a new set of problems involving the relation/inverse pair happy/sad. The terms for these problems were the names of three imaginary people, "Rich", "Dot", and "Harry". A typical problem was therefore, "Rich is happier than Dot. Harry is sadder than Dot. Is Harry happier than Rich?"

Subjects in the Concrete Properties training group were told to represent premises by vividly imagining faces having different features. Illustrations of the faces were drawn such that the people's names suggested the image of the correct face. Thus, "Rich" was depicted as a man wearing an expensive top hat, "Dot" was drawn with freckles, and "Harry" was pictured with a beard and mustache. Subjects were told to represent each premise by visualizing a pair of faces in which the face of the grammatical subject was smiling or frowning, depending on the wording of the problem. The two pairs of images then were to be scanned to answer the question for each problem.

Subjects given the Abstract Directional training were told to imagine a scale with "Sad" on the left and "Happy" on the right, and to place the names of the people on the scale appropriately as a problem was read. The order of the names on the scale then was to be used to make the transitive inference required to answer the question.

Following training, subjects solved 32 happy/sad problems. Subjects next rated the difficulty of applying the strategy they were trained to use. The rating scale ranged from 1 (extremely easy to use the strategy) to 6 (extremely difficult to use the strategy). After giving these ratings,

subjects described the strategy they would have used if they had not received training.

### *4.3 Results of strategy training*

Reasoning errors made by the two training groups were compared at successively finer levels of detail. The analyses parallel those applied previously to errors from subjects giving retrospective reports of spontaneously adopted strategies.

First, the group trained to use the Concrete Properties strategy had a significantly higher overall error rate (34 percent) than the group trained in the Abstract Directional strategy (17 percent). Second, the two training groups exhibited different general patterns of reasoning errors, as indicated by a statistically reliable interaction of training groups and problem types. Third, the specific problem factors found to distinguish the spontaneous report groups had analogous effects in the training groups. Errors made by the Abstract Directional training group were related strongly to the number of premises in a problem beginning with the pivot term (the "end anchoring effect" described previously), but were not strongly related to the number of inverses in a problem (uses of the word "sadder"). Subjects trained in the Concrete Properties strategy tended to show the complementary pattern. Fourth, when the process models in Tables V and VI were fitted to the error data of each group of subjects, the appropriate process model gave the better fit in each case (see Table VIII).

Two further analyses related the strategy reported by subjects prior to receiving training to their performance after training. In the first of these analyses, subjects were grouped by the strategy they reported prior to training, and the reasoning errors of the different groups were

Table VIII—Proportion of variance* in problem difficulty attributable to two strategy models

| Parameters | Abstract Directional Group | Concrete Properties Group |
|---|---|---|
| Abstract Directional Model | | |
| 1. SEARCH | 0.691† | 0.330‡ |
| 2. POSITION | 0.003 | 0.198 |
| $\sum R^2$ (Percent of Reliable Variance) | 0.694† (93.9%) | 0.528† (57.4%) |
| Concrete Properties Model | | |
| 1. SCAN | 0.372‡ | 0.743† |
| 2. GENERATE | 0.067 | 0.113‡ |
| 3. ENCODE | 0.014 | 0.012 |
| $\sum R^2$ (Percent of Reliable Variance) | 0.453 (61.3%) | 0.868† (94.4%) |

* These proportions are increments in $R^2$ values attributable to each parameter. The order in which parameters are given corresponds to the step at which they entered the regression equation.
† $p < 0.01$
‡ $p < 0.05$

compared (see Table IX). The general result was that subjects trained to use the Abstract Directional strategy made fewer errors than those trained to use the Concrete Properties strategy no matter what strategy was initially reported. This result suggests that people can be trained to use a particular reasoning strategy even if they have not adopted that strategy spontaneously.

Subjects given Concrete Properties training rated their strategy significantly more difficult to use ($\bar{x} = 3.81$) than those given Abstract Directional training ($\bar{x} = 2.36$). These ratings then were related to the strategy reported by subjects prior to training. All subjects given Abstract Directional training found that strategy relatively easy to use no matter what their initial strategy had been. For people trained to use the Concrete Properties strategy, the results were different. The Concrete Properties strategy was rated more difficult to use by Abstract Directional Thinkers ($\bar{x} = 4.41$) than by Concrete Properties Thinkers ($\bar{x} = 3.20$). Virtually all the subjects who initially reported using the Abstract Directional strategy indicated that they would have applied that strategy to the happy/sad problems if there had been no training. This pattern of results suggests that people can appreciate a good reasoning strategy: subjects rated a less efficient reasoning strategy as "difficult to use," especially if they knew a more efficient strategy.

### *4.4 Summary*

People can be trained to use different mental processes for making transitive inferences in three-term series problems. This fact has been demonstrated by analyses of the reasoning errors made by people after receiving training in different strategies. Results closely parallel the results of previous analyses of errors made by people retrospectively reporting the different reasoning strategies. Further results suggest that training in a particular strategy can be effective even for those people who did not report the strategy in spontaneous reasoning. People also appear to be sensitive to the difficulty of using various reasoning strategies, and rate a less efficient strategy "difficult to use," especially if they know a better one.

Table IX—Reasoning error rates made by subjects after strategy training

| | Strategy Reported Prior to Training | | |
|---|---|---|---|
| | Abstract Directional | Concrete Properties | Other/Not Clear |
| Strategy trained to use | $\bar{X}$ (N) | $\bar{X}$ (N) | $\bar{X}$ (N) |
| Abstract Directional | 0.10 (12) | 0.09 (6) | 0.25 (15) |
| Concrete Properties | 0.33 (11) | 0.33 (5) | 0.36 (16) |

## V. GENERAL DISCUSSION

### *5.1 Retrospective reports about reasoning*

These studies provide a direct test of the reliability and validity of retrospective reports about reasoning. Generally, reports by different subjects contained sufficient information to classify the subjects consistently. The content of subjects' reports also was related systematically to the patterns of reasoning errors subjects made.

While retrospective reports proved very useful in these studies, they also had definite limitations. The classification of different people on the basis of their reports was not perfectly reliable. Another limitation was that a sizable minority of people gave reports that were either idiosyncratic or incoherent (Other/Not Clear reports). Subjects also reported metaphors or general descriptions of reasoning strategies (see Table II) rather than complete and detailed models like those in Tables V and VI. Using a crude classification rule, the majority of subjects' reports could be grouped reliably into two categories, and people giving different kinds of reports tended to exhibit different amounts and patterns of reasoning errors.

The fact that different people reported and used different reasoning processes has two practical implications. One is that it may be possible to obtain useful reports of reasoning-like processes in practical situations where a procedure or device is to be evaluated. Reports may lead to redesigning tasks to make them more compatible with people's reasoning processes. Another implication is a methodological suggestion. If reports on reasoning-like processes are used, it may be wise to obtain them from a large number of different people to gauge the range of mental processes people are likely to adopt.

The general conditions under which retrospective reports about thought processes are reliable and valid remain to be established. The method used in these studies was to have subjects carefully describe how they thought through specific kinds of problems immediately after attempting to solve a large number of the problems. Subjects were not asked why they chose a particular strategy, a kind of judgment that people are notoriously poor at making.[13] Subjects also were not asked to "think out loud" while solving problems, a technique that might have yielded more precise descriptions of the reasoning process. The cost of that technique in the present experiments on reasoning is that it would have prevented the collection of unbiased reasoning error data, and therefore would have clouded the test of the validity of reports. For other purposes, the technique of "thinking out loud" may be quite acceptable. What kinds of mental processes are reportable and which techniques are best for reporting them are two questions that must be answered before reports about thought processes generally can be used with confidence.[14]

### *5.2 Differences in people's reasoning*

An important result of these studies is that different people spontaneously adopted different reasoning strategies for solving very simple, highly stereotyped, three-term series problems. Different models of reasoning processes accounted for large amounts of the variance in problem difficulty for people classified as using different reasoning strategies. These models also received support from results of the training study in which people were directed to use one reasoning strategy or the other (Table VIII). While the process models provide a good first approximation to the reasoning processes employed by different people, the more basic and important result is that people spontaneously reason in different ways.

Training has been shown to cause people to adopt different reasoning processes in a rather simple way: people can follow different sets of directions on how to reason. Subsequent studies[9] identified two other factors that influence in a subtle way whether people use one reasoning strategy or another. One factor is aptitude for visualizing spatial transformations of figures. People good at spatial visualization are more likely to adopt the Abstract Directional strategy spontaneously than those who have difficulty with spatial visualization. The use of different reasoning strategies for three-term series problems is not, however, influenced by verbal aptitude. A second factor influencing the adoption of a reasoning strategy is the context in which reasoning problems are posed. The strategy used for a particular problem is influenced by surrounding problems. Some context problems apparently suggest or allow the development of good strategies while others inhibit strategy development.

A new theory of reasoning is required to account for differences in people's reasoning. The emerging picture includes a dynamic and modifiable process in which reasoning strategies are developed. This phase of strategy development may be influenced by people's basic capacities, the context in which the reasoning occurs, and training. The fact that people can rate the difficulty of using different reasoning processes suggests that feedback of this kind also may be involved in strategy development. Certainly this picture is a far cry from the idea that reasoning is a fixed ability in which people differ by the amount they have or how well they can perform. The emerging picture holds the hope that reasoning processes used by different people can be understood, and that the understanding might lead to redesigning some tasks that many people find very difficult.

### *5.3 Training in reasoning*

The training study reviewed here demonstrates that people can learn to use different reasoning processes with consequent effects on

their reasoning performance. A good characterization of the Abstract Directional training would be that it suggested an efficient general approach that most subjects could use, but that some subjects would not have discovered spontaneously. This result tends to confirm informal observation that a suggested strategy for dealing with a complex problem (e.g., thinking of a queueing problem in terms of a "pushdown stack", adopting spatial metaphors for programming problems, etc.) can be very helpful and will lead to certain patterns of performance. Perhaps there are many situations in which directions on "how to think about" a task (e.g., activating a telephone service or interacting with a computer) might reduce errors and lead to more predictable patterns of performance compared to directions in which learners must develop a strategy on their own.

The training study did not address the very important question of whether people can be trained to reason better in general. It is unlikely that the effects of strategy training for a specific simple transitive inference problem would generalize to cause subjects to reason more efficiently in many other situations. People who in general are good reasoners probably have discovered and used a large set of strategies that they can retrieve in given situations. Because of their basic capabilities (e.g., good spatial visualization) and previous experience, good reasoners probably are also very good at generating new alternative strategies for novel problems. The results reviewed here suggest that people can be trained to deal more effectively with some specific situations involving reasoning. This basic fact certainly does not reduce the likelihood of finding ways to train people to reason better in a large range of situations.

## VI. ACKNOWLEDGMENTS

Dorothea Grimes-Farrow collaborated on the first two studies reviewed here. Peter Meany and Anthony Rooney assisted in collecting and analyzing data for several of the studies. E. Z. Rothkopf gave many helpful suggestions for revising previous drafts of this paper.

## REFERENCES

1. V. A. Krutetskii, *The Psychology of Mathematical Abilities in School Children*, J. Teller, Trans; J. Kilpatrick and I. Wirszup, Eds., Chicago: The University of Chicago Press, 1976.
2. F. Davis, "Psychometric research on comprehension in reading," Reading Res. Quart., 7 (Summer 1972), pp. 628–78.
3. D. M. Irons, "Cognitive correlates of programming tasks in novice programmers," Proc. of Human Factors in Comp. Syst., Gaithersburg, Md., March 15–17, 1982.
4. C. Burt, "The development of reasoning in school children," J. Exp. Pedagogy, 5 (1919), pp. 68–77.
5. L. L. Thurstone, "Primary mental abilities," Psychometric Monogr., *1*, 1938.
6. R. J. Sternberg, "Representation and process in transitive inference," J. Exp. Pscyh.: Gen. *109* (June 1980), pp. 119–59.

7. H. H. Clark, "Influence of language on solving three-term series problems," J. Exp. Psych., *82* (November 1969), pp. 205–15.
8. D. E. Egan and D. D. Grimes-Farrow, "Different mental representations spontaneously adopted for reasoning," Mem. & Cog., *10* (July 1982), pp. 297–307.
9. D. E. Egan, unpublished work.
10. D. E. Egan, unpublished work.
11. C. B. DeSoto, M. London, and S. Handel, "Social reasoning and spatial paralogic," J. Pers. and Soc. Psych., *2* (1965), pp. 513–21.
12. J. Huttenlocher, "Constructing spatial images: a strategy in reasoning," Psych. Rev., *75* (November 1968), pp. 550–60.
13. R. E. Nisbett and T. D. Wilson, "Telling more than we can know: verbal reports on mental processes," Psych. Rev., *84* (May 1977), pp. 231–59.
14. K. A. Ericsson and H. A. Simon, "Verbal reports as data," Psych. Rev., *87* (May 1980), pp. 215–51.

## AUTHOR

**Dennis E. Egan,** A.B., 1969 (Psychology), College of the Holy Cross; M.A., 1973 (Mathematics), Ph.D., 1973 (Experimental Psychology), University of Michigan; Naval Aerospace Medical Research Laboratory, 1973–1976; Bell Laboratories, 1976—. Mr. Egan has worked on individual differences in cognitive abilities, especially spatial ability and reasoning. At Bell Laboratories, he has explored ways of adapting instruction to individual differences in abilities, experience, and other learner characteristics. Mr. Egan is currently a member of the Learning and Instruction Research Department.

# New Technological Demands

The articles in this section sample a range of task requirements, from turning a knob to the complex motor activities involved in typing. The common objective in these studies is to find ways to make tasks fit the abilities and prior learning that humans bring to them.

Donegan and Koppes introduce three articles on perceptual and motor factors affecting the design of equipment used by craftspeople who install and maintain telephone equipment. These articles deal with knob characteristics (Kohl), cable splicing techniques (Paul), and the perceptual problem of locating corresponding terminals in two high-density connector fields that are mirror images of one another (Flamm). Finally, Cohen describes a study comparing the perceptual motor effects of typing with membrane keys, which move little when pushed, with conventional travel keys.

No fundamentally new human factors issue is addressed in the articles appearing in this section. They collectively illustrate that old questions will need new answers whenever the human/system interface changes.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# Human Factors Engineering for the Loop Plant

By J. DONEGAN* and D. N. KOPPES*

(Manuscript received

Bell System Loop Plant operations employ 150,000 craftspeople. New technology is changing the demanding jobs they perform, and the range of physical characteristics of craftspeople is increasing as women join the traditionally male work force. Both of these factors indicate the need for human factors contributions to designing the tools and apparatus that craftspeople use. This article introduces three other articles describing different aspects of human factors work for the Loop Plant.

## I. INTRODUCTION

The installation, testing, operation, and maintenance of the Loop Plant (the customer cable network) requires the full-time employment of some 150,000 outside plant and central office craftspeople—a significant portion of the total Bell System craft work force. An important element of the human factors work in the Loop Transmission Division is knowing the characteristics of the craftspeople and their tasks so that we may supply engineers with human factors criteria for the design of loop apparatus and systems.

The published literature on anthropometrics, biomechanics, human factors, and applied psychology is, of course, a valuable resource.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

However, the literature is often incomplete with respect to Bell System tasks and the data often describe what can be done by people under ideal conditions, not what must be done by craftspeople under stressful conditions. Traditionally, physically demanding tasks have been performed by these craftspeople in suboptimal conditions: up on poles, down in manholes, and during foul weather conditions. While such work is still done, new technologies such as loop electronics, fiber-optic cable, and computer-controlled equipment have overlaid old jobs with more sophisticated tasks, requiring different physical and mental characteristics. Consider loop plant splicers responding to a major alarm concerning a fiber-optic cable repair. The first set of tasks could include lifting a 300-pound manhole cover, pumping "water" out of the manhole, and setting up a work station—perhaps 10 feet down, but still 10 feet from the bottom of the manhole. The next set of tasks could then involve the precision and fine manipulative skills described in the article entitled "Human Factors Comparison of Two Fiber-Optic Continuous-Groove Field-Repair Splicing Techniques" by Larry Paul, in which he compares two methods of field splicing fiber-optic cable that are currently under development at Bell Laboratories in Atlanta.

In addition to the introduction of new tasks, an increasing number of people of slight stature, particularly females, has altered the anthropometric characteristics of the loop plant work force. Where possible, the physically demanding tasks are redesigned to accommodate this new work force. A case in point is the ubiquitous use of torque knobs in loop plant equipment. In particular they are used to secure splicing equipment, and female craftspeople complain of difficulty in exerting enough torque to hold this heavy equipment in place. The experiment described in the article by George Kohl, entitled "Effects of Shape and Size of Knobs on Maximal Turning Forces Applied by Females," provided design guidelines for matching torque knob shapes and sizes with female capabilities.

The loop plant also includes two main areas for planned manual access to the network: the main distributing frame, and the feeder/distribution interface. At these points a constant "churning" of the loop network takes place in daily installation, removal, and rearrangement of cross-connecting wires. These areas have been very troublesome in the past and any new equipment, hardware, or tasks designed for these crucial access points are closely scrutinized to preclude wiring congestion or poor work practices. Well-motivated, trained craftspeople can always make equipment work but superior designs are required to accommodate the less skilled craftsperson who at one time or another will work on all of the in-place plant. Such attention to design detail and human factors issues is illustrated in the article by Lois

Flamm entitled "Performance in Locating Terminals on a High-Density Connector." This article describes an experiment that identified and helped resolve visual complexities associated with a new and compact protector block designed for use on the main distributing frame.

Many of the more significant loop plant hardware and systems developments are subjected to a full-scale evaluation at the Chester Field Laboratory.[1] Such evaluations take place prior to final production and utilize New Jersey Bell craftspeople to install, operate, and maintain preproduction models in realistic scenarios. However, laboratory experiments play a significant role at a much earlier stage of the design and development process. They may address questions related to the design of specific apparatus or resolve more general questions related to the capabilities of Bell System craftspeople. The articles by Lois Flamm, George Kohl, and Larry Paul are examples of such human factors experiments carried out in Department 54525 as an integral part of the development process of the new Loop Plant.

## REFERENCES

1. Brent E. Coy, "Putting the human factor into the outside plant," Bell Lab. Rec., 59, No. 1 (January 1981), pp. 17–23.

## AUTHORS

**John Donegan,** B.Sc., 1956, Strathclyde University, Scotland; M.S., 1963, New York University; Rolls-Royce in Aero Engine Design, 1956–1957; Union Carbide Company—Mining Equipment Design, 1957–1961; Bell Laboratories, 1961–1976; American Bell, 1976–1979; Bell Laboratories, 1979—. During his initial years with Bell Laboratories, Mr. Donegan worked on microwave radio relay systems and the underground and buried cable systems. With American Bell he was stationed in Tehran as Manager of Outside Plant Standards. Mr. Donegan has been working in the area of human factors engineering of the Loop Plant since 1979 and is currently Supervisor of the Human Factors Engineering Group II.

**David N. Koppes,** B.C.E., 1959, M.C.E., 1960, Cornell University; Structural Engineer, Aluminum Company of America, 1960–1964; Bell Laboratories, 1964–1971; AT&T, 1971–1973; Bell Laboratories, 1973—. During his initial years at Bell Labs, Mr. Koppes investigated cable plowing techniques, had mechanical design and operations responsibilities for Sea Plow II and Sea Plow III, and worked on maintenance systems for the WT-4 Millimeter Waveguide System. At AT&T, he was an Assistant Manager in Outside Plant Engineering. Mr. Koppes has been doing human factors work in the Loop Plant area since 1975 and currently is Supervisor of the Human Factors Engineering Group I.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Effects of Shape and Size of Knobs on Maximal Hand-Turning Forces Applied by Females

By G. A. KOHL*

(Manuscript received December 23, 1981)

Outside plant craftspeople use knobs to apply turning forces on clamp mechanisms that hold field equipment temporarily in place. A study was performed to develop a set of data that provide guidance for determining knob size and shape characteristics most appropriate for various outside plant working conditions. Forty female participants applied maximal isometric turning force to each member of a set of twenty experimental knobs that systematically varied in shape and size. In half the trials the participants applied force with greased hands and in the other half used nonslip compound. In addition, two arm-wrist positions were observed. In general, triangular knobs allow more hand torque to be generated and require significantly less material than square, pentagonal, hexagonal, or circular knobs of comparable size. However, this effect depends upon the arm-wrist position and grip conditions. A 3.5-inch turning diameter is desirable when both cost and performance are considered.

## I. INTRODUCTION

Outside plant craftspeople in the Bell System use knobs to apply turning forces on screw-operated clamp mechanisms used to hold heavy field equipment temporarily in place. Female craft who use these

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

knobs report that the knobs they currently use do not enable them to easily generate sufficient torque to clamp down equipment. The question raised by these reports and addressed here is: What characteristics of a knob allow turning forces to be easily generated? Considering the variety of torque knobs in the world (e.g., door knobs, small valve wheels, faucets), and the considerable body of knowledge the human factors community has acquired about control knobs, it is perhaps surprising to find that the literature sheds little light on this rather straightforward and practical question (see Refs. 1 and 2 for the best existing treatments of torque knobs).

The following experiment was designed to provide a set of data that can assist designers in developing torque knobs, and we hope will be of value to the hand tool industry in general. The approach taken in this study was strictly empirical and consisted of measuring the amount of turning force that participants can generate using knobs of various sizes and shapes under several different conditions.

## II. METHOD

### *2.1 Participants*

Forty right-handed females from the Whippany, New Jersey, area responded to newspaper and intracompany bulletin board ads, and served as paid participants. Mean and standard deviation of the group's age, height, and weight were respectively: 37 years, SD = 13.5; 64.7 inches, SD = 3.4, and 132 pounds, SD = 27.2.

### *2.2 Independent variables*

Variables were identified that could affect knob-turning performance, including features of the knobs and the conditions under which the knobs might be used. To render the study manageable in size, only five of the most intuitively and/or practically relevant variables to the Bell System application were included in the study.

#### *2.21 Knob shape*

There are a limitless number of possible knob shapes. A shape attribute, "sidedness," was chosen to succinctly capture the spectrum of shapes that could affect knob-turning performance. Five shapes were chosen, as shown in Fig. 1, varying from few sides to infinite sides (i.e., triangle, square, pentagon, hexagon and circle).

#### *2.2.2 Knob size*

An operational definition of size was generated so that knobs of different shapes could be compared. The "diameter" of a knob is the diameter of the circle bounding the outermost points of the knob. Thus the triangular knob always has about half the area of a circular

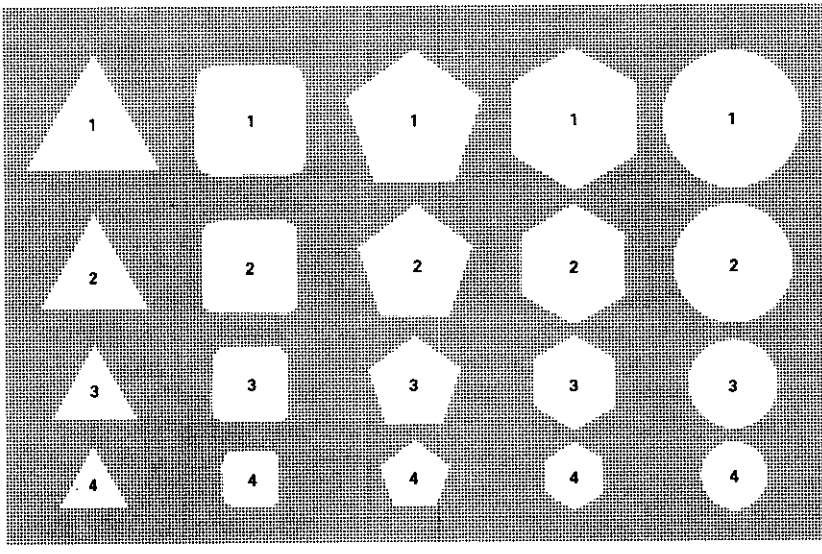



Fig. 1—The set of experimental knobs.

knob with an equivalent diameter. The definition has real-world validity in that the envelope in which a knob turns is likely to be a design constraint. Four diameter sizes were included in the study: 2.5, 3.5, 4.5, and 5.5 inches. Preliminary evidence showed these sizes to represent plausible end- and middle-point values for diameters of knobs to be used to generate turning force.

#### *2.2.3 Grip condition*

The types of knob surface conditions that could be tested are numerous, for example, knurled surfaces, serrated edges, slippery surfaces. The surfaces of the knobs in this study were all the same—smooth, anodized aluminum. However, an attempt was made to study the extremes of surface conditions by including two grip friction conditions under which the knob was turned. In the *greased* condition participants greased their hands before turning; in the *nonslip* condition they applied a nonslip compound.

#### *2.2.4 Position*

The hand-arm position was varied to emulate various real-world conditions, where knobs must be turned from many different vantage points. Two standing positions were included; in both conditions the height of the knob was adjusted to the level of the elbow joint for each participant. In the *front position* the participant stood straight in front of the knob, forearm parallel to the floor and perpendicular to the upper arm and frontal plane of the body, the axis of knob rotation

coincident with the longitudinal axis of the forearm. The wrist was bent so that the hand pointed up, the palm of the hand against the knob face, fingers spread around the knob edge. In the *side position* the participant stood to the side of the knob with the forearm parallel and perpendicular to the axis of rotation of the knob. The palm of the hand pressed against the face of the knob with the fingers spread around the knob edge.

#### *2.2.5 Participant size*

Size was included as a subject variable. Although hand size was originally targeted as the variable that would most affect performance, weight was found to be a better predictor.[3] Four weight groups of ten participants each were formed on a post hoc basis. The mean and standard deviation of the four group's weights in pounds were: 106 pounds, SD = 5.72; 120 pounds, SD = 3.65; 132 pounds, SD = 5.87; 171 pounds, SD = 22.96.

### *2.3 Experimental design*

A complete factorial design was employed (5 knob shapes × 4 knob sizes × 2 grip conditions × 2 positions × 4 participant weight groups). All participants were included in all possible within-participant treatment conditions.

### *2.4 Procedure*

Each participant attended for one day on two consecutive weeks for approximately four hours each day. On each day each participant performed four blocks of twenty trials, each block consisting of one appearance of each of the twenty knobs. For each trial participants were instructed to apply maximal isometric turning force to a knob specified by the experimenter, using the right hand for a period of three one-second beats of a metronome. The order of the knob appearance was randomized for each block of trials for each participant. On each day, the front position was employed on two blocks of trials, the side position on the other two; one of the side position and one of the front position trial blocks were performed with a greased hand. The remaining two blocks were performed using a nonslip compound.

Participants were observed in groups of three or four, every participant performing a single trial before any given participant performed her next trial. Roughly two minutes intervened between any given participant's trials, and breaks of twenty minutes were taken between blocks of trials.

### *2.5 The knobs*

The relative size and shape of the knobs is shown in Fig. 1. They were made of machined aluminum and had an anodized smooth

surface. All corners and edges of the knobs were rounded and the smallest radius on any edge was 0.25 inch. The knobs were 1.25 inches thick.

## III. RESULTS

The data were submitted to an analysis of variance,[4] with knob shape, knob size, grip conditions, position, and day of participation treated as within-subjects variables, and participant weight as a between-subjects variable. The most important results are summarized here.

The most informative effect obtained was a four-way interaction of the shape, size, position, and grip variables [$F(12,432) = 5.2$, $p < 0.00001$]. Fig. 2 shows this interaction. The top two panels show performance as a function of knob shape and size for the side position, the left panel for the nonslip blocks and the right panel for the greased blocks of trials. The bottom two panels show performance using the front position, the left panel for the nonslip condition and the right panel for the greased condition.

Several effects are apparent from a visual inspection of Fig. 2. The greased condition performance (right two panels) is lower than the nonslip condition [$F(1,36) = 289.6$, $p < 0.00001$]. Further, over all other variables, performance decreases as the number of knob sides increases [$F(4,144) = 183.2$, $p < 0.00001$]. This main effect is qualified by an interaction with the grip variable [$F(4,144) = 71.0$, $p < 0.00001$], indicating that the main effect of shape is much more pronounced in the greased condition than in the nonslip condition. Figure 3 shows this two-way interaction.

The effect of size is apparent in Fig. 2; across all other variables, the bigger the knob diameter, the more torque developed [$F(3,108) = 246.8$, $p < 0.00001$]. The left-to-right convergence of the curves in Fig. 2 reflects the two-way interaction of the knob size and shape variables: over all other variables, the greater the number of sides, the less advantage size has [$F(12,432) = 53.6$, $p < 0.00001$]. The fact that the convergence of the curves is more pronounced in the greased (right two panels) than the nonslip panels is reflected in a two-way interaction of knob shape, knob size, and grip condition [$F(12,432) = 8.4$, $p < 0.00001$]. The reversal of the sidedness advantage for smaller knobs in the nonslip, top left panel is counter to the patterns in the other three panels and is reflected as the four-way interaction illustrated in Fig. 2.

Participant weight was only a marginally significant main effect [$F(3,36) = 2.6$, $p < 0.07$] but did interact with the knob size variable [$F(9,108) = 4.8$, $p < 0.0001$], indicating that the advantage larger people have over smaller people dwindles as knob size decreases.

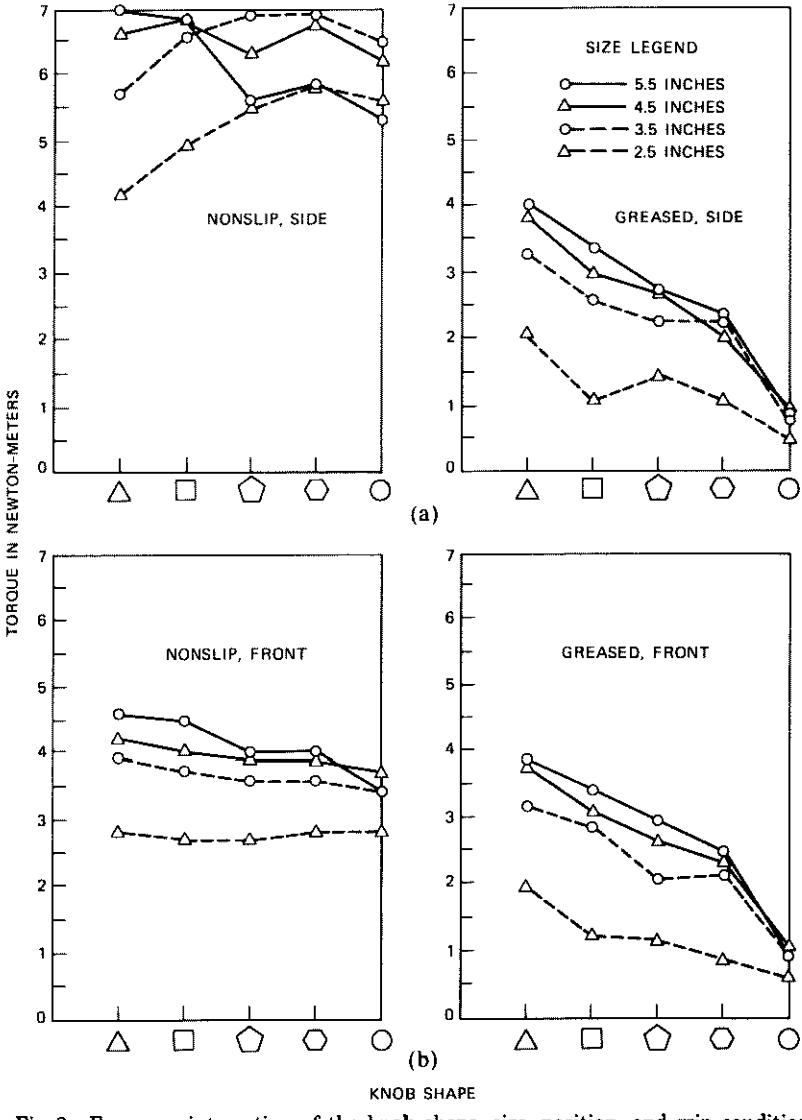



Fig. 2—Four-way interaction of the knob shape, size, position, and grip condition. (a) Performance as a function of the knob shape and size for side position. (b) Performance as a function of knob shape and size for the front position.

Several other main effects and interactions were obtained: position, day, grip × position, grip × knob size, position × knob size, position × shape, day × shape, grip × position × knob size, grip × position × shape, and position × knob size × shape.

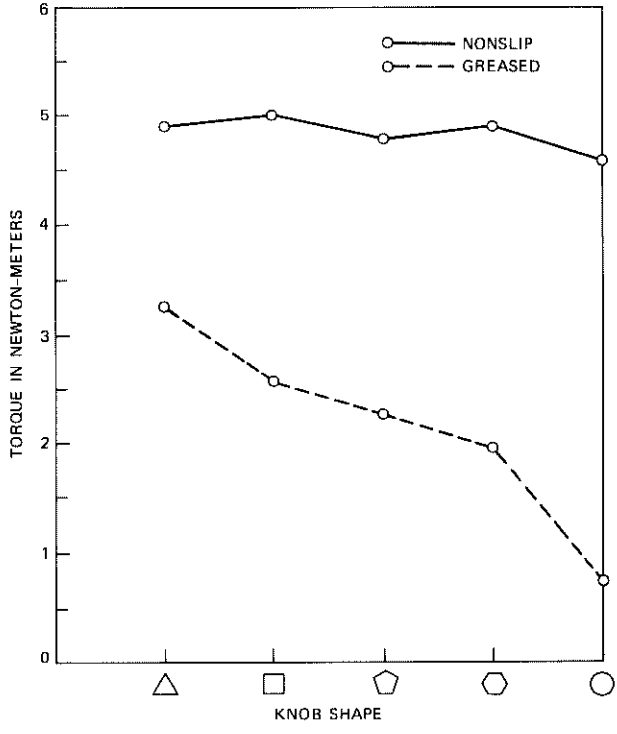



Fig. 3—Knob shape by grip condition.

## IV. SUMMARY AND DISCUSSION

The practical difference between means in Fig. 2 will have to be decided by individual designers. Note that if the application involves a nonslip, front approach, the shape of the knob has little effect, and the knob size is very important (Fig. 2, bottom left panel). However, in both the front and side greased conditions, shape is very important; note the smallest triangle results in better performance than the largest circle (Fig. 2, bottom right panel). The savings in materials in this case is roughly 10 to 1.

In general, triangular knobs allow for the generation of as much or more torque than any of the other shapes. However, subjective comfort ratings indicate that the smallest triangular knob causes discomfort. Also, the smallest triangular knob does not fair well in the nonslip, side condition. Therefore, for 2.5-inch diameter applications the square is probably a better choice.

The biggest jump in performance with respect to knob diameters comes between 2.5 and 3.5 inches; thereafter performance increases as diameter increases, but at a smaller rate.

Considering materials, torque and comfort, the triangular and square 3.5-inch knobs are recommended for general application. If more torque is required than can be obtained with these knobs, larger diameter triangular or square-shaped knobs should be used. For these or any other knobs, all corners and points should be rounded for best performance and comfort.

## V. ACKNOWLEDGMENTS

The author expresses appreciation to Brian G. Bancroft for assisting in the development of experimental apparatus and conducting the investigation, and Elise Darrow and Ha T. Nguyen for assisting in the data analysis and preparation of this report.

## REFERENCES

1. E. O. Sharp, "Maximum Torque Exertable on Knobs of Various Sizes and Rim Surfaces," MRL-TDR-62-17, Behavioral Sciences Laboratory, Aerospace Medical Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 1962.
2. A. D. Swain, G. C. Shelton, and L. U. Rigby, "Maximal Torque for Small Knobs Operated With and Without Gloves," Ergonomics, *13*, No. 2 (1970), pp. 201–8.
3. G. A. Kohl, unpublished work.
4. B. J. Winer, *Statistical Principles in Experimental Design* (2nd Ed.), New York: McGraw-Hill, 1971.

## AUTHOR

**George A. Kohl, Jr.,** B.S., 1968, United States Merchant Marine Academy; M.S., 1974, Experimental Psychology, Purdue University; Ph.D., 1978, Experimental Psychology, Purdue University; U.S. Merchant Marine Engineering Officer, 1968–1974; Assistant Professor of Psychology, Allentown College, 1977–1978; Bell Laboratories, 1978—. Mr. Kohl has been a Human Factors Specialist at Bell Laboratories, performing research, design, and development work on equipment and systems used by central office and outside plant craftspeople. Most recently, he has been a member of the team responsible for the design of a computerized transmission enhancement system to support special services operations. Member, American Psychological Association and the Human Factors Society.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Human Factors Comparison of Two Fiber-Optic Continuous-Groove Field-Repair Splicing Techniques

By L. M. PAUL*

(Manuscript received December 23, 1981)

Two Bell System fiber-optic splicing techniques that are under development were experimentally compared in a human factors study. In addition to splicing technique, the variables explored were the uniformity (evenness) of fiber spacing in the 12-fiber "ribbon," and the hand (preferred or nonpreferred) that was used to splice. Performance was measured by the time to insert fibers, the number of fibers broken, and the time taken to complete a series of insertions for each technique. The preference of the participants between the two techniques was also determined. All of the measures except participant preference showed the vacuum technique to be statistically superior to the hold-down bar technique, with participant preference suggesting the same conclusion, although it was not statistically significant.

## I. INTRODUCTION

A human factors study compared two techniques under development for the repair splicing of damaged Bell System fiber-optic cable in the field. The goal of the study was to determine the preferred technique from a human factors viewpoint. This information combined with economic considerations would then allow the technique's designers

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

to choose one of the techniques for further development and release to operating telephone companies.

## II. DESCRIPTION

Bell System fiber-optic cable consists of one to twelve multiple-fiber "ribbon" arrays. Each ribbon consists of twelve optical fibers embedded in a plastic tape. Currently, each ribbon of the cable is shipped with a silicon connector on each end to facilitate the splicing (joining) of the ribbons of one cable section to the ribbons of the next section.[1] In joining one ribbon to another, the splicer first places silicon wafers on either side of one ribbon array connector, forming a sandwich, and secures this arrangement with a spring clip. The second ribbon connector is then slipped into this sandwich and a second spring clip is added. The final operation consists of the application and curing of a refraction index-matching gel.

These procedures are reasonably straightforward and have been successfully used by operating telephone companies. On the other hand, in certain situations the splicer must refabricate a ribbon's silicon connector in the field.[2] Refabrication is required when the fragile silicon ribbon connector is received damaged, is damaged in craft handling, when it is necessary to reposition ("swap") fibers within the ribbon, or when an additional splice point becomes necessary. In part, refabrication involves grinding and polishing of the individual fiber ends in the reconstructed silicon connector to minimize losses when the connector is spliced (joined) to another connector. Although field refabrication is successfully being done by operating telephone companies as the field-repair method, the two techniques described in this report are being developed to simplify field-repair splicing basically by eliminating the grinding and polishing operations.

Figure 1 schematically depicts the continuous-groove approach shared by the two splicing techniques. In both techniques, the silicon array connectors are first removed from the ribbons to be spliced. After fiber preparation, which includes removing the plastic tape, removing the individual fiber coatings, and developing square fiber ends, fibers of one ribbon are "combed" into the set of 12 continuous grooves. The fibers of the second ribbon are then similarly inserted into the continuous grooves. Once the fibers of each ribbon are inserted, they are brought into proximity and index-matching gel is applied and cured as in the standard joining procedure. Thus connector refabrication, with its necessary grinding and polishing steps, and the subsequent connector joining have been eliminated and replaced by a single operation.

The two continuous-groove techniques that were evaluated differ in the method for inserting fibers into the continuous grooves. The

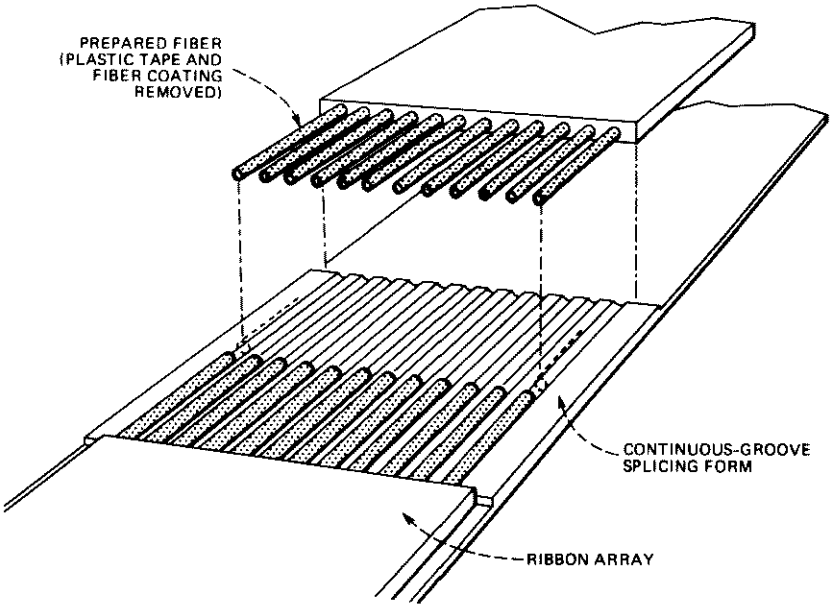



Fig. 1—Basics of a fiber-optic ribbon, continuous-groove splice.

vacuum technique uses a vacuum assist to help secure both the prepared fibers (tape and coating removed) and a portion of the ribbon beyond the prepared fibers. The splicer then uses a lateral motion to "coax" the 12 fibers into their respective grooves. In the hold-down bar technique, the splicer combs the prepared fibers into their respective grooves with the use of an alignment comb without any additional aid. The splicer first registers the twelve prepared fibers in the twelve slots of the alignment comb and then carefully pulls the ribbon back until the ends of the fibers go below a bar in the comb and fall to the continuous grooves below. The splicer then gently pushes the ribbon forward under the bar that holds the ribbon in the continuous grooves (hence the "hold-down bar" name), and pushes the ribbon along the continuous grooves to approximately the middle of the splicing form. The proprietary nature of these techniques, which are still under development, prohibit the use of photographs to illustrate the two techniques.

## III. HUMAN FACTORS EXPERIMENTAL COMPARISON OF SPLICING TECHNIQUES

A human factors study was designed to compare the two field-repair splicing techniques on the basis of craft performance. An analysis of the sequence of operations in the techniques suggested that the only difference between fully developed techniques would be the method of

inserting fibers into the grooves. Both techniques would likely share the same or similar methods of fiber preparation, index-matching gel application, etc. Therefore, it was decided to limit the scope of this study to assessing performance differences between techniques for successfully inserting fibers into the continuous grooves.

### *3.1 Method*

#### *3.1.1 Design*

The experiment was a within-participants design. Each participant was trained and tested under every combination of experimental variables. The three experimental variables were: splicing technique (vacuum or hold-down bar), fiber spacing uniformity (uniform or nonuniform), and splicing hand (preferred or nonpreferred).

Fiber spacing uniformity refers to the uniformity of spacing of the twelve fibers in the ribbon. Previous experience with handling of fibers showed that nonuniformly spaced fibers are more difficult to guide into grooved structures than uniformly spaced fibers. Nonuniformly spaced fibers can result from manufacturing and from the removal of the coatings on each fiber. Since nonuniform ribbons occur infrequently, they were created for the study by making a hole with a small awl in the plastic tape at the point where the prepared fibers emerge from the tape.

The third variable, splicing hand, was chosen to reflect the fact that *all* splicing techniques require the splicer to use both left and right hands on the ribbons coming from the left and right sides, respectively, and thus the potential interaction between splicing hand and field-repair technique is of interest. Although it is possible to rotate the splicing form so that the splicer uses the preferred hand for both ribbons, concern over building in transmission loss caused by bending the fibers precluded further consideration of this approach.

Four dependent measures selected to contrast the two field-repair techniques were:

1. Average insertion time
2. Number of broken fibers
3. Elapsed (clock) time
4. Participant preference.

Average insertion time, measured with the participants' knowledge, was the time between beginning an insertion and its successful completion. If a fiber was broken the clock was stopped, a new ribbon was provided, and the clock was restarted. Similarly, the clock was stopped and restarted if a question was asked or the experimenter wanted to clarify a procedure.

Elapsed (clock) time was simply the total time required to complete all 14 insertions required in each technique. It included training and

questions in addition to the actual insertion time reflected in the average insertion time.

Participant preference was gauged at the end of the experiment by asking the question: "If you were a Bell System splicer, which of the two splicing techniques you used would you prefer to use on a regular basis?"

#### *3.1.2 Experimental apparatus*

The work platform measured 14 × 24 inches and was chosen to closely approximate the current splicing work station. This rather small area is dictated by the limitations of many splicing situations. A stand-mounted 3× magnifier to allow close viewing of the fibers (approximate 0.005″ diameter with coating removed) and splicing form, and two halogen lamps to evenly illuminate the work station were chosen as a result of previous work. A beam splitter plate, television camera, and monitor comprised the observation system that allowed unobtrusive monitoring of participant performance and the demonstration of insertion technique by the experimenter, as discussed below.

#### *3.1.3 Procedure*

Participants were individually tested in a session lasting approximately two hours. They were first familiarized with Bell System fiber-optic cable and current procedures for splicing ribbons together. Half of the participants were then trained and tested on the hold-down bar technique first, with the other half receiving the vacuum technique first.

Training consisted of the experimenter demonstrating the correct insertion procedure. Participants first watched "over the experimenter's shoulder" and then observed the detailed aspects on the closed-circuit television monitoring system. After this initial training, all participants performed the sequence of splicing insertions outlined in Table I.

The first insertion was designed as practice with the participant using a uniform ribbon and his or her preferred hand. In addition, the ribbon was free, i.e., it was not anchored at one end. All subsequent

Table I—Sequence of experimental ribbon insertions for both splicing techniques

| Step | Fiber Spacing | Task |
|---|---|---|
| 1 | Uniform | Practice with free ribbon using preferred hand |
| 2 | Uniform | One practice insertion using preferred hand |
| 3 | Uniform | Six insertions alternating hands |
| 4 | Nonuniform | One practice insertion using preferred hand |
| 5 | Nonuniform | Six insertions alternating hands |

insertions were timed with the 20-inch ribbon anchored approximately 16 inches from the splicing form to simulate a ribbon emerging from a cable.

As shown in Table I, nonuniform ribbons were not introduced until later in the splicing sequence, and thus any comparison of the effect of fiber spacing uniformity on splicing performance within a particular splicing technique is confounded by the additional training for nonuniform ribbons relative to uniform ribbons. This point will be discussed further in Section 3.2.

After completing the insertion sequence using one splicing technique, participants were given a five-minute break and then trained and tested in an identical manner using the other technique. After completing both techniques, participants were asked a short series of questions to obtain more detailed information on their performance. Questions were asked concerning any hobbies requiring fine motor control such as knitting, use of glasses (including bifocals) during the experiment or at other times, general comments, and their preference between the two field-repair techniques.

#### *3.1.4 Participants*

It was considered useful to select two distinct subpopulations as participants, each with some degree of demonstrated motor skill, in order to be able to generalize the results of the experiment to operating telephone company craftspeople. Nine people from the Bell Labs wiring shop and nine from the Bell Labs clerical pool participated.

### *3.2 Results*

Performance of the two subpopulations, wiring shop and clerical, was virtually identical. The subpopulation variable was not significant ($F < 1$) and did not interact with any of the other independent variables. This similarity of results for the two subpopulations presumably reflects the fact that both groups did indeed possess a similar degree of general motor skill with which to approach the learning of this new motor skill.

#### *3.2.1 Average insertion time*

Figure 2 shows average insertion time for the nonpractice insertions (Steps 3 and 5, Table I) as a function of splicing technique and fiber spacing uniformity for all 18 participants. The superiority of the vacuum technique over the hold-down bar technique was significant [$F(1,16) = 19.6, p < 0.001, \mathrm{MS_e} = 0.52$]. This statistical superiority of the vacuum technique was obtained despite the fact that four participants had to be stopped on one or more insertion attempts using the hold-down bar technique and the time to that point used as an

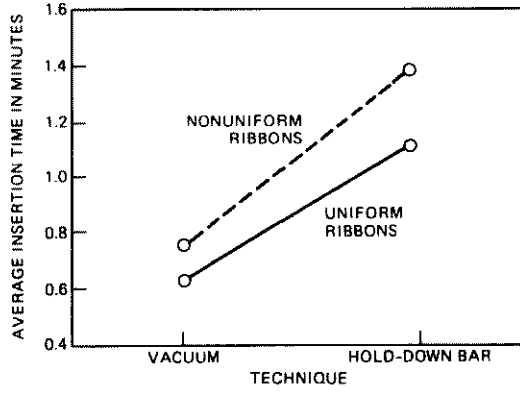



Fig. 2—Average insertion time on uniform and nonuniform ribbons for each splicing technique.

(under) estimate of that insertion attempt. Participants were stopped whenever an attempt exceeded approximately four minutes in order to allow completion of the experiment in a two-hour period.

The smaller average insertion time for uniform ribbons relative to nonuniform ribbons shown in Fig. 2 was statistically significant [$F(1,16) = 5.6$, $p < 0.05$, $MS_e = 0.23$]. As previously noted (and shown in Table I), nonuniform ribbons were presented after uniform ribbons, introducing a confounding between fiber spacing uniformity and training. However, since it is reasonable to assume that performance with the nonuniform ribbons benefitted from this additional training relative to the uniform ribbons, the smaller average insertion time for the uniform ribbons may be interpreted as a valid effect. Neither the interaction between splicing technique and fiber spacing uniformity, nor interactions between any of the other independent variables were statistically significant.

Participants performed marginally better with their preferred hand, although this difference was not significant ($p > 0.05$). The lack of an interaction between splicing hand and technique was surprising since a distinct preferred-hand advantage was expected for the hold-down bar technique given the motor skill required for this technique.

#### *3.2.2 Additional measures*

Table II contrasts the two field-repair splicing techniques using the other three dependent measures. The number of broken fibers was significantly less in the vacuum technique [$F(1,16) = 14.6$, $p < 0.005$, $MS_e = 3.25$]. No other main effects or two-way interactions were found using number of broken fibers as the performance measure. Elapsed (clock) time was also significantly less in the vacuum technique

Table II—Additional ribbon insertion measures

| Measure | Technique | |
|---|---|---|
| | Vacuum | Hold-Down Bar |
| * Number of broken fibers (total) | 4 | 24 |
| * Elapsed (clock) time | 33.6 min | 56.9 min |
| Participant preference (total) | 12 | 5 |

* Statistically significant at $p < 0.005$.

(Wilcoxon test, $p < 0.002$). Participant preference (with one abstention) favored the vacuum technique, although it was not significant as assessed by a Sign test.

## IV. DISCUSSION

### *4.1 Vacuum technique superiority*

The various measures strongly suggest the superiority of the vacuum technique over the hold-down bar technique. This superiority is basically a time advantage realized both directly in using the technique, and indirectly by reducing the number of repeated fiber preparations required when fibers are broken. However, it should be noted that 14 of the 18 participants were successful in completing all 12 of the nonpractice insertions required of them in the hold-down bar technique. Further, even the four participants who had to have one or more of their insertion attempts terminated because of time limitations completed at least 10 of the 12 attempts. This relative facility is important as it was not clear, a priori, whether inexperienced people could be easily trained to use the hold-down technique, especially with the brief (approximately 10-minute) training.

In interpreting the results of this study, it should be remembered that the study focused on only one of the operations required in field repair, i.e., ribbon insertion. Although ribbon insertion is undoubtedly the most difficult operation for the splicer, the time required by the other operations common to any field-repair technique may somewhat overshadow the vacuum technique time advantage. Thus, other factors such as economics and manufacturability are important in selecting the technique to be developed for release to the operating telephone companies.

### *4.2 Unanswered questions*

Two relevant human factors concerns remain unanswered by this study. One unknown is the amount of additional training that would be required to improve the performance of those people experiencing difficulty with the hold-down bar technique. A second question is the stability of the learned field-repair skills over time, i.e., the memory for the skills. The ability to perform either technique will probably

decrease as a function of the time since a person's last splicing experience. This forgetting is particularly relevant to the scenario in which a period of weeks or months elapses between splicing experiences. It is not clear, for example, that the vacuum technique would remain superior to the hold-down bar technique over time. Further studies will help to answer these questions.

## REFERENCES


1. C. M. Miller, "Fiber optic array splicing with etched silicon chips," B.S.T.J., *57*, No. 1 (January 1978), pp. 175–90.
2. H. J. Friedrichsen and P. F. Gagen, "Optical cable field repair using array fabrication," 7th European Conf. on Optical Fiber Commun., September, 1981, Copenhagen, Denmark.


## AUTHOR


**Lawrence M. Paul,** B.S.E.E., 1967, University of Maryland; M.S.E.E., 1969, University of Illinois; Ph.D. (Experimental Psychology), 1975, Purdue University; Design Engineer, Philco-Ford, 1968–69; Digital Project Engineer, Pertec Inc., 1969–71; Assistant Professor of Psychology, Lehigh University, 1975–78; Bell Laboratories, 1978—. Since coming to Bell Laboratories, Mr. Paul has been a human factors specialist consulting on projects relating to operating telephone company outside craftspeople. His principal focus has been on the placing and splicing of lightguide (fiber-optic) cable. Lightguide projects have included the work described in this issue, development of a splicing work station, and the field observation of early lightguide installations to identify human factors concerns. Member, American Psychological Association, Eastern Psychological Association, Human Factors Society, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Performance in Locating Terminals on a High-Density Connector

By L. E. FLAMM*

(Manuscript received October 6, 1981)

In two experiments thirty participants located terminals on variations of a new main distributing frame connector. Abutting individual connectors in mirror image led to a higher termination density. Location times and errors were evaluated using deadline procedures. Experiment 1 results showed a high incidence of parallax, counting, and left-right reversal errors. Design modifications aimed at reducing these errors led to improved performance in Experiment 2. Parallax, counting errors, and location times were significantly reduced. The continued occurrence of left-right reversal errors is discussed.

## I. INTRODUCTION

Main distributing frame (MDF) connectors provide termination points for outside plant cable and protect central office personnel and equipment from harm due to foreign electrical potentials. Some connectors also have provision for making cross-connections. Over the years connector design has responded to increased space demands on main frames. A new 309-type protected connector achieves a higher density of terminations by putting two independent connectors to-

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

gether in mirror image. Although the physical arrangement of the two abutted connectors differs, functionally they are equivalent.

Figure 1 shows prototypes of the original 309-type connector and a modified version with designation labels to identify terminal numbers. The modified connector incorporates features based on recommendations from the human factors studies described below. The direct spatial correspondence between cross-connect and protector fields and the familiar grouping by fives or tens found on previous connectors is missing in the new composite. Because each connector row consists of two cross-connect pairs and five protector units, it was important that the designation labels reflect users' expectations of direction of count and facilitate their making the appropriate associations.

The aim of a preliminary pencil and paper experiment (unpublished) was to choose the most natural numbering scheme. For both connectors, results showed consistent but independent horizontal left-to-right counting patterns associated with cross-connect and protector locations. Concern was generated from the preliminary study about potentially high error rates within connectors and possible confusions between connectors. The loss of spatial correspondence with the new design and the above results led to the hypothesis that shading connector backgrounds would help users to localize specific cross-connect and protector terminals. In the first experiment, two shading schemes were compared with the unshaded connector version in both slow and fast deadline conditions. It was anticipated that any differences in performance between the shaded and unshaded connector versions would be more evident with the rigorous work pace of the fast deadline.

## II. EXPERIMENT 1

### *2.1 Method*

#### *2.1.1 Subjects*

The participants in Experiment 1 were 18 male Bell System employees, several of whom had relevant experience as craftspeople.

#### *2.1.2 Design and Materials*

The experiment can be characterized as a within-subjects design in which each participant located cross-connect pairs and protector units on each of the three connector versions in both deadline conditions. As schematically illustrated in Fig. 2, the version 1 connector represents the original prototype 309 connector with the preferred labeling; in addition versions 2 and 3 include cross-connect field background patterns grouping cross-connect pairs by fives and tens, respectively. The layout of each connector version is a mirror image, but the

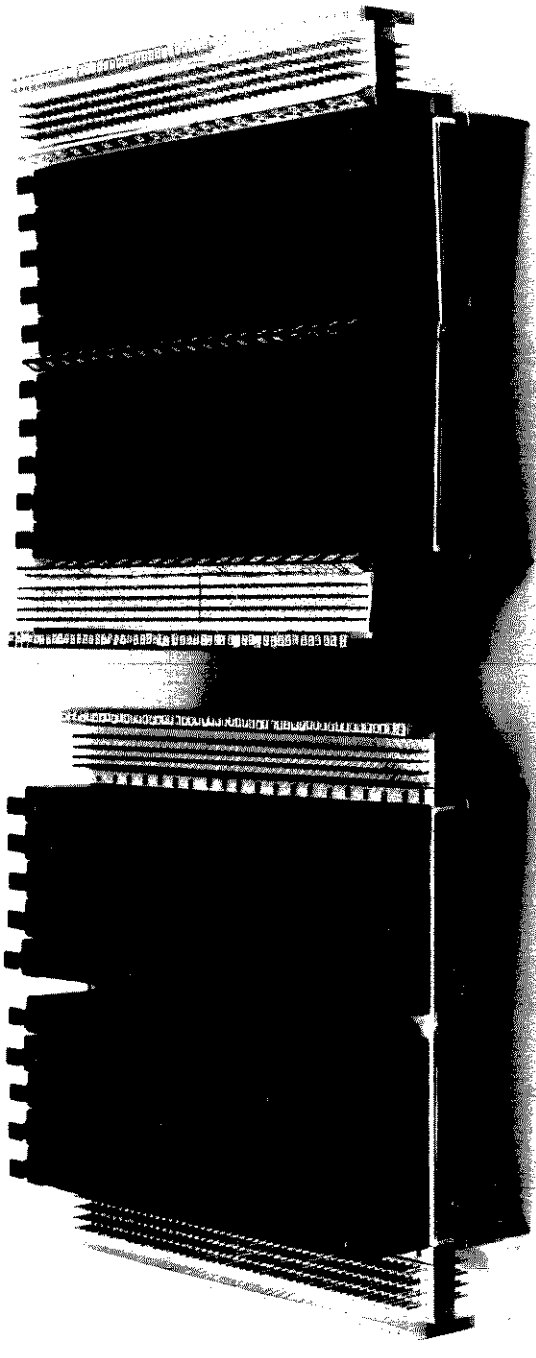


Fig. 1—Originally proposed (left) and modified (right) 309-type protected connector.

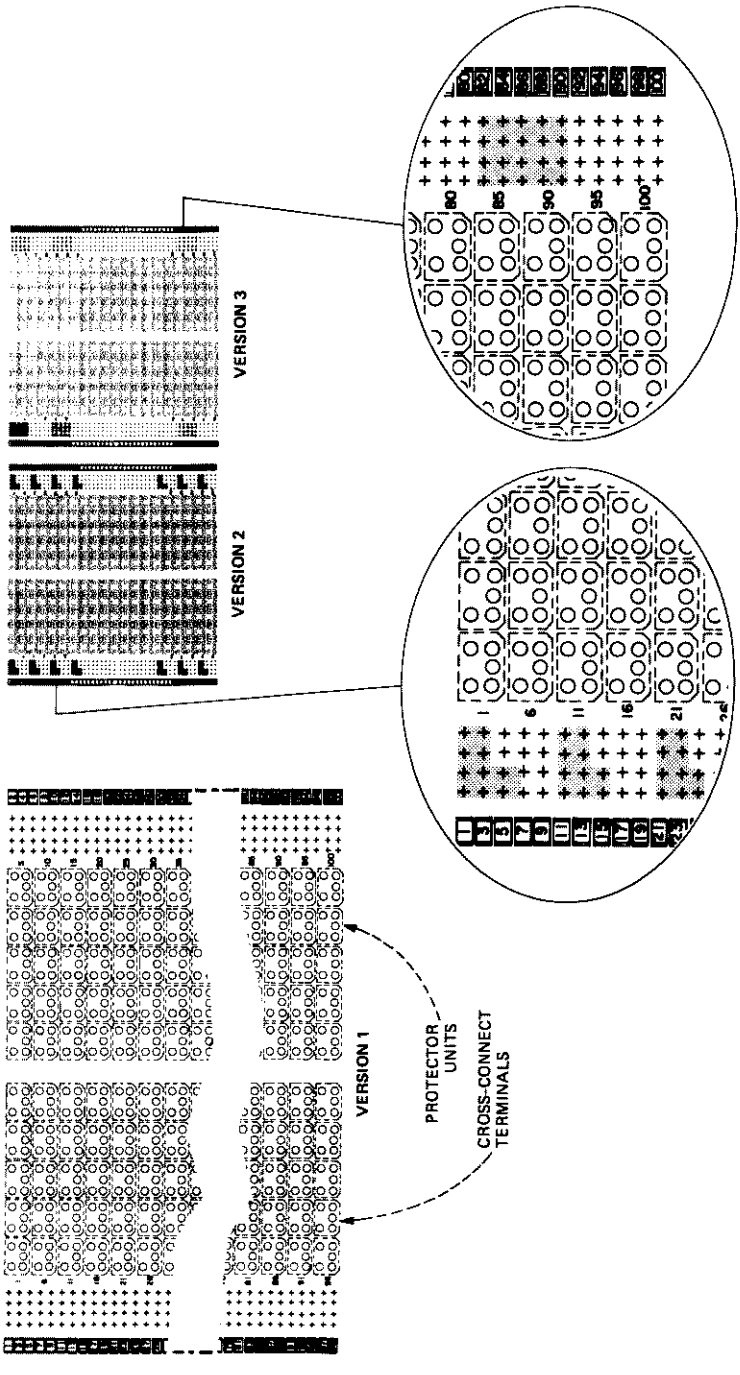



Fig. 2—Schematic illustrations of connectors (Experiment 1).

numbering scheme is not. The option of carrying the cross-connect field background patterns up to the designation strips was rejected as too costly. For purposes of these studies, the independent mirror image connectors of each version are labeled A and B.

A work assignment consisted of twenty trials of locating and marking cross-connect pairs and corresponding protectors. Cardboard trays with 20 plastic cups provided work assignment information. Each cup contained the material for one trial, a cross-connect cap and protector unit each marked with an A or B to designate which connector, and a specific number from 1 to 100 for position within the connector. Time to locate and install each set of cross-connect caps and protector units was recorded on a *Datamyte** Data Collector. If the participant did not complete a trial and press the response button by the imposed deadline, a brief beep occurred. The 34-second slow and 16-second fast deadline times were chosen from the results of a pilot study. In that study, response times measured in an unpaced task showed that 34 seconds was ample time to complete a trial and that a 16-second deadline would be inadequate about 50 percent of the time. Search time was measured as the interval between response button presses. In addition to time data, the position of each cross-connect cap and protector unit was recorded for identification of type as well as number of errors.

#### *2.1.3 Procedure*

Each participant completed a practice work assignment of ten trials followed by one work assignment on each of the three connector versions at each of the two deadlines. Order of presentation of connectors within as well as between deadlines was counterbalanced across participants, and the six work assignments were completely rotated through the six connectors every six participants.

Each participant was read instructions describing the physical arrangement of the A and B connectors and the procedure to be followed. For each trial, the procedure was to find the terminal marked on the cross-connect cap and protector unit, place the cap over the cross-connect pair, replace the corresponding protector with the labeled one, and then press the response button to denote completion of the trial. The experimenter emphasized that a clock which was reset by depressing the response button would be running; the goal was to complete each trial before the buzzer sounded.

### *2.2 Results*

Mean time to locate cross-connect and protector unit sets was used

* Trademark of Data Acquisition Equipment, Electro/General Corporation.

in an Analysis of Variance (ANOVA) computed with subjects (18) and connectors (3) with both slow and fast deadline conditions (2) as variables.[1] No significant differences in search times were found for the three connector versions, nor were there deadline x connector interactions.

In determining errors each cross-connect cap member and protector unit was considered separately. The overall error rate was 4.5 percent. Results of the data analysis showed error rates were not affected significantly by the connector background or by the pace of the work (slow versus fast deadline). Percent and types of errors are summarized in Table I.

Three main error categories were found: parallax, left-right reversals, and counting errors. Parallax errors were responses incorrectly displaced vertically by some number of rows. If a terminal was marked on the wrong connector but in the correctly numbered position, e.g., A3 instead of B3, it was called an L-R reversal error. Counting errors were mistakes in locating terminals in any row within a correct connector that did not fit any obvious pattern.

### *2.3 Discussion*

The addition of background patterns to cross-connect fields did not aid in locating connector terminals. Errors were useful in pinpointing areas for design modifications. Parallax, a problem associated with high-density connectors, is likely to be accentuated because cross-connect and protector fields, as well as numbers referring to them, are in different planes. With the 309-type composite connector, additional difficulties may be related to the abutment of two independent connectors. The large number of L-R reversal errors suggested the need

Table I—Errors in locating terminals

| Cross-Connect and Protector-Unit Errors | Experiment 1 ($N = 18$) | Experiment 2 ($N = 12$) | |
|---|---|---|---|
| Type | Average for All Connectors | Original Connector | Modified Connector |
| Left-right reversals | 1.4% of all responses (31% of errors) | 1.88% of all responses (20.2% of errors) | 1.25% of all responses (46% of errors) |
| Parallax | 1.5% of all responses (34% of errors) | 5.1% of all responses (55% of errors) | 0.5% of all responses (19.2% of errors) |
| Counting | 1.25% of all responses (27% of errors) | 1.5% of all responses (15.7% of errors) | 0.3% of all responses (11.5% of errors) |
| Other | 0.7% of all responses (8% of errors) | 0.8% of all responses (9% of errors) | 0.6% of all responses (23.3% of errors) |

for a clearer demarcation between the two connectors, and counting errors pointed to difficulties in locating terminal positions.

Based on the results of Experiment 1, the design of the 309-type connector was modified and a second experiment was conducted. A modified 309-type connector included an angled center designation strip with a vertical black line over the ridge for more distinct separation between connectors, and angled designation strips between cross-connect and protector fields for labeling rows on both sides. Angled rather than flat strips were used to minimize any change in connector dimensions and confusion between adjacent labels. To help further, cross-connect and protector fields were placed in more nearly the same plane by raising the cross-connect field and shortening the cross-connect terminal pins slightly so that the ends of the pins were in the same plane as the corresponding designation strip (see Fig. 1).

A similar experimental design was used in a second experiment to compare performance on the modified connector version and the original unpatterned connector of Experiment 1. Neither connector was patterned on the cross-connect field since patterning did not prove to be effective in Experiment 1. Only the fast deadline was used.

## III. EXPERIMENT 2

### *3.1 Method*

Twelve new Bell System participants with comparable experience to those of Experiment 1 took part in the study.

Designation strips on the modified connector were angled back 62 degrees from the vertical plane. The original connector was labeled as before. Four of the six work assignment trays and the practice assignment were used. After the practice assignment, the participant completed a 20-set assignment on each of the two connectors, followed by a break and a repeated measure on the two connectors. The order of connector presentation was counterbalanced and work assignments were rotated as before.

### *3.2 Results*

As in Experiment 1, search time and error rate were the dependent measures. Subjects (12), connectors (2), and the repeated measure, trials (2), were the variables for the ANOVAs. Mean times to find and complete a connection on the original connector were 13.6 and 11.9 seconds for the first and second trials, respectively; these were 12.4 and 11.3 seconds on the modified connector. The modified connector version resulted in a modest but significant savings in search time ($F = 9.11$, $df = 1,11$, $p < 0.05$). The improvement in speed from the first

to second trials on both connectors led to a significant practice effect ($F = 34.5$, $df = 1,11$, $p < 0.001$).

The overall mean error rate was 9.25 percent on the original connector and 2.70 percent on the modified version; an ANOVA showed that errors were significantly lower on the modified connector ($F = 10.36$, $df = 1,11$, $p < 0.01$), and improved with practice ($F = 7.31$, $df = 1,11$, $p < 0.05$). Interactions were not significant. Of greatest interest is the marked reduction in errors on the modified 309-type connector. As shown in Table I, using the same error categories, numbers of parallax and counting errors decreased substantially. Unfortunately, L-R reversals did not decrease.

### *3.3 Discussion*

From a human factors standpoint, the lack of a one-to-one correspondence between cross-connect and protector fields and the abutment of two connectors in mirror image are not ideal. On the other hand, there is a real and immediate need for additional space on the MDF.

It is not clear whether the L-R reversal errors were due to difficulties associated with the left-right nature of the task, the mirror image arrangement, or even the proximity of the composite connectors. Although the two sides of each composite connector were called A and B, in all likelihood they were recognized in terms of their relative positions as left and right. People have difficulty discriminating left from right.[2] Despite the random assignment of work order pairs to the A and B connectors, 69 percent of the reversal errors in Experiment 1 and 73 percent in Experiment 2 were preceded by a correct response on the same side, as if participants were not paying attention to which of the connectors they had just worked on, and so continued to work on the same connector.

The terminal location task was chosen as it was assumed it would be sensitive to the spatial arrangement of the connectors. Time-consuming wiring operations were not included in the task as they should be constant. On the main frame, however, these and many other operations would interrupt the left-to-right scan. The search would also necessarily extend over a much wider area since a main frame consists of large arrays of connectors. The identification of main frame connectors at the top of vertical frame modules should also result in vertical top-to-bottom search. Thus, the predominant left-to-right orientation of the task may have been somewhat artificial. It is likely that L-R reversal errors would be reduced on the main frame where the inclusion of other operations should make the craftperson's job more sensitive to a connector's specific position.

While eliminating the one-to-one correspondence between cross-

connect and protector fields results in higher density terminations on the MDF, craftspeople, like the experimental participants, may locate associated elements on the two fields independently. The potentially small difference in time should not affect productivity. In reality, a work assignment would not necessarily require work on both cross-connect and associated protector units. More important for the craftsperson's job may be the advantage of the front-facing work surface. There is some assurance that the craftsperson will keep his/her gaze nearly perpendicular to the angled designation strips on the modified 309-type connector, since numeral positions on the sides of the wedge-shaped strips preclude reading both sides from the same point. Besides providing clear labels and end points for rows, then, the angled designation strips may lead to more concentration that results in greater accuracy in terminal location.

In summary, the number and consistency in errors on the initially proposed 309-type connector were unacceptably high. Performance on a modified version was improved with significant reductions in parallax and counting errors. The resultant 309-type connector represents the integration of design and human factors efforts.

## REFERENCES


1. B. J. Winer, *Statistical Principles in Experimental Design*, New York: McGraw-Hill, 1971.
2. W. S. Farrell, Jr., "Coding Left and Right," J. Experimental Psychology: Human Perception and Performance, *5*, No. 1 (February 1979), pp. 42–51.


## AUTHOR


**Lois E. Flamm,** B.A., Skidmore College, 1966; M.A., 1968, Ph.D. (Experimental Psychology) 1971, Northeastern University; Bell Laboratories, 1977—. Ms. Flamm taught at Texas A & M University from 1972–1976. In 1977 she joined the Human Factors Engineering Group, Loop Transmission Division, at Bell Laboratories in Whippany, where she worked on a variety of interface problems associated with the development and evaluation of new apparatus and methods for outside plant craft. In 1982, she was an intern in the Distributed Computer Systems Research Department, Murray Hill. She is presently in the *UNIX*™ Systems Engineering Group in the *UNIX* Development Laboratory, Murray Hill.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# Membrane Keyboards and Human Performance

By K. M. COHEN LOEB*

(Manuscript received January 7, 1982)

This paper describes systematic human factors research in which typing performance using a membrane keyboard and using a conventional, full-travel keyboard were compared for subjects representing different levels of typing proficiency. Membrane switch technology has become increasingly popular in many consumer-oriented products because of its low production cost and design flexibility. However, the absence of familiar key travel associated with membrane switches removes an important, direct source of feedback to the user with respect to specific keystrokes. Hence, the conventional wisdom has been that membrane switches without key travel are unacceptable for such keyboard applications as typing tasks. The results of the research discussed here indicate that for nontouch typists there was little difference in performance between keyboards. For touch typists, performance with the conventional keyboard was initially much better than with the membrane keyboard. Rapid learning resulted in improvement in typing performance with the membrane keyboard—both within an experimental session and across sessions—such that the advantage of the conventional keyboard over the membrane one for touch typists was reduced substantially, although not completely. Future work will be aimed at measurement of the additional improvement in performance that may result from extended practice with better-designed membrane keyboards.

*American Bell.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

## I. INTRODUCTION

Membrane switch technology has become increasingly popular in consumer-oriented products such as pocket calculators and kitchen appliances. Application of this technology to the design of full-size, alphanumeric keyboards has resulted in products that differ markedly from conventional keyboards in terms of keystroke feedback to the user. Membrane switches usually consist of mechanical contacts on two layers of material, separated by a nonconductive third layer of material (see Fig. 1). The upper, membrane layer is usually a thin polyester material with flexible conductors applied to its underside. Graphics may be silk-screened on this surface or on a second surface placed on top of the membrane layer. The substrate may be either a printed circuit board with conductors or a flexible film with printed conductors. This is usually mounted on a rigid, smooth surface. The spacer layer is an insulating material that separates the membrane and substrate layers by 5 to 7 mils. It has holes through which the upper, flexible layer may be depressed, causing contact closure. When pressure on the membrane layer is removed, the resilient, flexible membrane breaks contact with the substrate and returns to its original position.

Membrane switch technology has become popular for several reasons. Its production costs are low because there are no key-plunger mechanisms. It also affords considerable design flexibility in terms of panel layout, "key" size and shape, and graphic labels. Good switch enclosure can also be assured for hostile environments and for protection against dust accumulation, spills, and vermin infestation. Easy cleaning of the upper surface is also an advantage associated with membrane switch technology.

Although there are several technical advantages of membrane switches, there are potential user-related problems when this technology is applied to full-size keyboard design. First, familiar key travel is absent as a source of keystroke feedback to the user. Second, high actuation forces are often used with membrane switches to prevent accidental switch closure that may occur from pressure exerted by hands or fingers in a resting position on the contact area of the switch. Finally, the actual contact area ("sweet spot") of the switch often is indiscriminable for the user, compared with traditional mechanical switches that use key caps on top of key-plunger mechanisms.

Despite these disadvantages, membrane switch technology is currently being introduced into the design of full-size keyboards because of its considerable cost advantages. There is little question that membrane switches may be used successfully for simple on-off function keys. There is, however, considerable doubt about the use of this technology for keyboard tasks such as typing because of the assumed

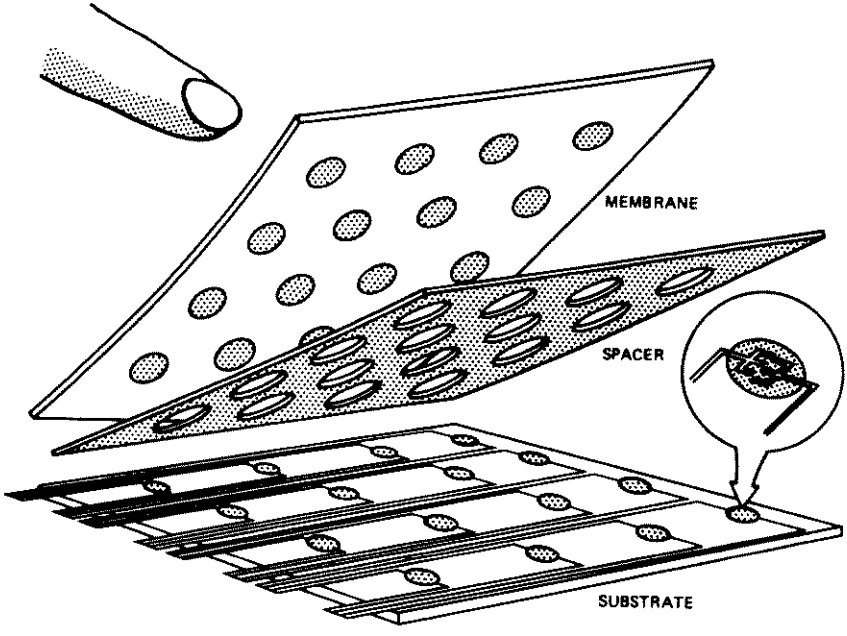



Fig. 1—Membrane switch technology.

importance to the user of key-travel feedback associated with individual keystrokes. Despite an absence of empirical evidence to support this claim, the conventional wisdom has been that membrane keyboards cannot be used as high-speed data- or text-entry devices.

Systematic human factors research was thus undertaken to determine whether performance using membrane keyboards differs from that using a conventional, full-travel keyboard. It was expected that large performance differences between keyboards would be present initially, but that with practice these performance differences might be reduced. Thus, this research was designed to provide subjects with several experimental sessions using each keyboard. It was also hypothesized that typing skill would be directly related to initial performance differences between keyboards and, perhaps, to the speed of learning to use the membrane keyboard. Specifically, it was expected that better touch typists would initially show greater performance differences between the membrane and conventional keyboards than less skilled touch typists because of the greater familiarity of the better touch typists with conventional keyboards. However, it was also possible that skilled touch typists would demonstrate greater transfer of typing skills to a novel keyboard and, hence, would learn to use the membrane keyboard more rapidly than less skilled touch typists.

Unskilled, nontouch typists, who tend to look at the keys as they are struck, were expected to show little difference in performance between membrane and conventional keyboards. A within-subject design with multiple sessions was used so that each subject would serve as his/her own control. This design also facilitated direct comparisons of typing performance on the membrane and conventional keyboards over time.

## II. METHOD

### *2.1 Subjects*

Twenty-one subjects (17 females and 4 males) participated in this study. Subjects were either employees of Bell Laboratories or volunteers from the local community. Volunteers were paid nominally for their participation in the study. Subjects varied in their typing ability and their average typing time per week.

### *2.2 Facilities and equipment*

Two commercially available keyboards were used in this study. The presence or absence of familiar key travel associated with each keystroke was the most distinguishable difference between these keyboards. One keyboard was considered a conventional keyboard in terms of key travel (about 100 mils to contact and an additional 50 mils to bottoming out), average actuation force (68 grams or 2.4 oz.), keytop size (450 mils diameter), and interkey spacing (750 mils, center-to-center). The other keyboard was a flat membrane keyboard with virtually no key travel. The average actuation force was 82 grams or 2.89 oz. A 1-mil-thick grid overlay delineated individual "keys." The "keytop" diameter size was about 600 mils with interkey spacing of 750 mils, center-to-center. Both keyboards had a QWERTY layout of alphanumeric keys and also a DELETE key that permitted subjects to make corrections. Audio feedback associated with each keystroke was achieved upon "key" depression with the key plunger mechanism for the conventional keyboard (an audible "click") and with an internal audio tone generator for the membrane keyboard.

Hardware and software interfaces were developed to permit on-line experimentation and data collection using these keyboards. Both keyboards were interfaced with a MINC-11/03 (DEC) minicomputer via an RS-232 port. The output from the keyboard was displayed simultaneously on a 9-inch video monitor placed directly in front of the subject behind the keyboard and on a VT-105 (DEC) display in front of the experimenter (see Fig. 2).

The study was conducted on-line with the MINC computer displaying appropriate prompts (e.g., "type text") to the subject prior to each of the experimental tasks. Three types of data were collected during each task: the actual characters displayed on the monitor, the actual

Fig. 2—Experimental setup.

characters typed by the subject (including deletions and corrections), and each interkeystroke time interval.

### *2.3 Experimental tasks*

Each subject participated in six sessions, three with the membrane keyboard and three with the conventional keyboard. Three types of tasks were used twice during each session: dialing telephone numbers, typing text, and answering questions. The typing-text and dialing-numbers tasks were viewed as experimental tasks, while the answering-questions tasks were viewed as practice tasks. Each experimental task was considered a "trial" so that there were two trials for each type of experimental task per session. In addition, there were two sets of questions for the practice tasks per session. The stimulus materials for each of these tasks were placed on a typing stand to the right of the keyboard and monitor.

Each "dialing" task consisted of 30 seven-digit phone numbers that were to be dialed from the keyboard. Two different lists of numbers were used for each of three sessions with a given keyboard, for a total of six different lists. The two lists used for each of the first, second, and third sessions with the conventional keyboard were identical to those used for the respective sessions with the membrane keyboard. Hence, direct comparison of performance using the two keyboards for a particular session was possible.

Each "typing text" task consisted of a paragraph from a standard

typing textbook that is used to train and assess typing speed.[1] The paragraphs were of comparable typing difficulty, based upon syllabic intensity ratings. These ratings reflect the average number of syllables per word and are traditionally used to assess the typing difficulty of textual material.[1] As with the phone number list, two different paragraphs were used for each session with the given keyboard, for a total of six different paragraphs. The two paragraphs for each of the first, second, and third sessions for the conventional keyboard were identical to those used for the respective sessions with the membrane keyboard. Hence, direct comparison of performance between keyboards for a particular session was possible. These tasks were considered the most critical ones with respect to the major objectives of this study.

Each answering-questions task consisted of a few questions on a general topic (e.g., favorite sport and how it is played, most recent vacation activities). Subjects answered these questions using the conventional or the membrane keyboard. Two different sets of questions were used during each session, for a total of 12 different sets. This task was primarily designed to provide a structured exercise by which the subject would gain experience using each keyboard.

### *2.4 Procedure*

At the beginning of each session, a standard 3-minute typing test was given to each subject. This test was conducted using an IBM Selectric typewriter. Following this test, each subject completed six tasks on either the conventional or the membrane keyboard. The tasks were administered in the following order: dialing numbers, typing text, answering questions, answering questions, typing text, dialing numbers. This order of tasks allowed the subject about one-half to three-quarters of an hour of practice on each keyboard (e.g., the answering-questions trials) in between the two dialing-numbers and typing-text trials. This order of tasks, then, maximized observation of practice effects over trials within a session but also minimized warm-up and fatigue effects on the critical typing-text trials. There were also four general questions that were answered as a warm-up exercise at the beginning of the first session with each type of keyboard.

Each subject completed three sessions with one keyboard and then three sessions with the other keyboard. Sessions took place on consecutive (or nearly consecutive) days, one session per day. At the conclusion of each session, subjects completed a questionnaire that probed their rating of various features of each keyboard. At the conclusion of the third session with each keyboard, another typing test with an IBM Selectric typewriter was administered to assess changes in typing performance over the course of a session. Though these changes in performance on the IBM Selectric are not reported here, they helped

determine the typing skill classification for each subject. Biographical data regarding touch typing ability, weekly amount of time typing, and prior use of membrane switches were also obtained.

### *2.5 Typing groups*

Subjects were categorized according to their typing ability based upon two criteria. First, assessment was made of their use of touch typing techniques in which all ten fingers and "home row" positioning of the fingers are used. Second, their average gross words per minute (WPM) on the eight typing tests given on an IBM selectric typewriter during the six experimental sessions was computed (i.e., the typing tests preceding each session and the test following the third session with each keyboard). These post-hoc criteria resulted in the following five categories of typing proficiency (see Table I): excellent touch typists (more than 60 WPM), good touch typists (50–60 WPM), fair touch typists (40–49 WPM), poor touch typists (26–39 WPM), and non-touch typists (16–25 WPM). There were 3, 5, 5, 4, and 4 subjects in these respective typing groups, for a total of 21 subjects.

The order in which keyboards were used by subjects was counterbalanced within each touch typing group through random assignment of subjects to one of the two possible keyboard orders. This balancing scheme for keyboard order was limited by the fact that the post-hoc categorization of subjects into typing groups resulted in an unequal number of subjects per group. Hence, to the extent possible, half the subjects in each touch typing group experienced the membrane keyboard first (total = 9 subjects) while the remainder of each touch typing group used the conventional keyboard first (total = 8 subjects). Also, by chance, all of the subjects who were categorized post hoc as nontouch typists used the membrane keyboard first. Hence, for the nontouch typing group, there was no variation across subjects in keyboard order.

### *2.6 Experimental design*

Typing Group (5 levels) was the major between-subject factor. Order of keyboards (2 levels) was considered a minor between-subject factor. Analyses using this factor across the touch typing groups were not

Table I—Number of subjects per typing group

| Touch Typing Group | Gross Words Per Minute | Total Number | Membrane First | Conventional First |
|---|---|---|---|---|
| Excellent | >60 | 3 | 2 | 1 |
| Good | 50–60 | 5 | 3 | 2 |
| Fair | 40–49 | 5 | 2 | 3 |
| Poor | 26–39 | 4 | 2 | 2 |
| Non | 16–25 | 4 | 4 | 0 |

reported since the number of subjects per unit of analysis was extremely small, making interpretation of results complex. In addition, there was no variability across the order variable for the nontouch typists, and the basis for estimating the variance attributable to keyboard order for this group was inadequate. Within-subject factors included keyboards (2 levels), sessions (3 levels), and within-session trials (2 levels). Separate analyses were conducted for the different types of experimental tasks (dialing and typing text). The data from the answering-questions trials were not analyzed.

## III. RESULTS

Several dependent measures were computed for each subject for the dialing and typing tasks. These measures were based upon the speed and/or accuracy of performance using both the conventional and membrane keyboards. Of particular importance to this study were the performance differences between the two keyboards for the various typing groups. Hence, difference scores and percent difference measures were also computed for each subject, and mixed-factor analyses of variance (ANOVAs) were performed. In addition, for the typing tasks the speed and accuracy data were combined to compute each subject's average words per minute, a throughput measure of performance.

### *3.1 Dialing results*

For each dialing task, the number of phone numbers dialed incorrectly and not subsequently corrected was computed. This measure included both misdialed digits as well as insufficient or additional digits per phone number. However, if a subject corrected a dialing error through use of the DELETE key, the resultant phone number was considered correct. There were 30 phone numbers per dialing task. On a few trials, a subject either omitted a phone number or redialed a phone number, a mistake no doubt attributable to the experimental set-up. These omitted or redialed phone numbers were not scored as dialing errors. Hence, the percent error rate for dialing phone numbers for each subject was computed from these data, and an ANOVA was performed.

As shown in Table II, the error rate for the dialing tasks was low for all groups using either the conventional or the membrane keyboard (less than 3 percent). The ANOVA indicated no significant difference in dialing accuracy as a function of keyboard or typing group ($p = 0.259$ and $p = 0.720$, respectively). There were also no significant effects when difference scores between keyboards were computed for each subject and served as the unit of analysis.

Table II—Percent error rate for dialing tasks*

| Keyboard | Touch Typing Group | | | | |
|---|---|---|---|---|---|
| | Excellent | Good | Fair | Poor | Non |
| Membrane | 1.11 | 2.66 | 2.01 | 2.50 | 1.30 |
| Conventional | 2.61 | 2.85 | 1.88 | 2.78 | 1.52 |
| Mean | 1.86 | 2.98 | 1.95 | 2.64 | 1.24 |

*Error rate is for uncorrected errors.
Each dialing task has 30 phone numbers.

### *3.2 Dialing summary*

Accuracy of dialing performance did not vary as a function of keyboard used or typing proficiency level. Given the simplicity of the task, relative to typing text, this result is not surprising. The data do, however, provide an anchor point for comparison of performance between the two keyboards as task complexity increases.

### *3.3 Typing results*

For each typing task, the number of "words in error" that remained after the subjects' corrections was computed according to traditional methods of scoring typing tests.[2] Hence, an incorrect letter(s), a missing letter, a letter reversal, a punctuation error, and/or a spacing error were all scored as a "word in error." The six passages, while equated in difficulty based upon the traditional syllabic intensity level, varied somewhat in the total number of words per passage (number of words per passage, where 5 character spaces equals 1 word, were: 64, 57, 79, 77, 73, 84).

#### *3.3.1 Typing accuracy*

Typing accuracy was measured in terms of both uncorrected errors and corrections. As shown in Table III, the total number of words in error was quite small (range: 1.8 to 4.5 words). The ANOVA did not indicate a reliable difference for this measure among typing groups or between keyboards. There was, however, a significant trial effect [$F(1,16) = 20.01$, $p < 0.0005$], with errors declining as a function of practice (4.2 vs. 2.7 errors).

Unlike standard typing tests, subjects were permitted to correct errors during the typing tests through use of the DELETE key. An ANOVA was performed on the number of corrections made by subjects during each typing task. The results indicated significant effects of keyboard [$F(1,16) = 5.27$, $p = 0.035$] and of session [$F(2,32) = 3.63$, $p = 0.038$]. In general, more corrections were made using the membrane keyboard than using the conventional keyboard (4.61 vs. 3.14), as shown in Table IV. Also, more corrections were made in the second

Table III—Average words typed in error

| Keyboard | Touch Typing Group | | | | |
|---|---|---|---|---|---|
| | Excellent | Good | Fair | Poor | Non |
| Membrane | 1.99 | 1.83 | 3.77 | 3.79 | 4.54 |
| Conventional | 2.56 | 4.13 | 3.90 | 3.13 | 3.79 |
| Mean | 2.28 | 2.98 | 3.83 | 3.63 | 4.17 |

Table IV—Average number of corrections for typing tasks

| Keyboard | Touch Typing Group | | | | |
|---|---|---|---|---|---|
| | Excellent | Good | Fair | Poor | Non |
| Membrane | 3.22 | 4.57 | 6.10 | 6.13 | 2.38 |
| Conventional | 4.00 | 2.47 | 3.83 | 2.96 | 2.67 |
| Mean | 3.61 | 3.52 | 4.97 | 4.54 | 2.52 |

and third sessions than in the first session, even though there was no reliable session difference in typing accuracy (i.e., uncorrected words in error). This session effect for number of corrections may be related to passage length; the average passage length was shorter for the first session (61 words) than for later sessions (78 words), and the probability of making a typing error increases as passage length increases. There was also a significant interaction between keyboard and trial factors [$F(1,16) = 5.48$, $p = 0.033$]. This interaction reflected a more dramatic practice effect (i.e., decline in errors corrected) for the membrane keyboard than for the conventional one.

#### *3.3.2 Typing speed*

The total typing time for each trial was also computed and an ANOVA was performed. The results indicated, as expected, a significant effect of typing group [$F(4,16) = 6.94$, $p = 0.002$]. As expected, typing speed varied directly with typing proficiency level, as shown in Table V. There was also a significant keyboard effect [$F(1,16) = 64.7$, $p < 0.005$]. Typing performance was somewhat faster with the conventional keyboard than with the membrane keyboard (130 seconds vs. 171 seconds).

Though there were other reliable effects related to the trial and session factors on speed of typing, it should be noted that, as with the typing accuracy data, variation in passage length (57 to 84 words) for the six test passages could readily have contributed to differences in typing time per trial and/or session.

#### *3.3.3 Differences between keyboards in typing speed*

Because of the variation in passage length, ANOVAs were performed on the difference in typing time between the two keyboards for

Table V—Average time for typing tasks (in seconds)

| | Touch Typing Group | | | | |
|---|---|---|---|---|---|
| Keyboard | Excellent | Good | Fair | Poor | Non |
| Membrane | 134.1 | 143.5 | 166.8 | 193.6 | 213.1 |
| Conventional | 97.0 | 101.4 | 106.7 | 149.0 | 169.9 |
| Mean | 115.5 | 122.5 | 136.7 | 171.3 | 206.2 |

corresponding trials. Each difference score was thus based upon the same passage typed on each keyboard.

The ANOVA results indicated significant effects of both the session [$F(2,32) = 3.68$, $p = 0.036$] and the trial [$F(1,16) = 27.0$, $p < 0.0005$] factors. These findings reflect practice effects both across and within sessions. There was also a significant interaction between trial and session factors [$F(2,32) = 3.5$, $p = 0.042$], as shown in Table VI. This interaction was attributable to larger practice effects between trials during the first two sessions, compared with those during the last session. This practice or learning effect within a session is based primarily upon greater improvement in typing speed using the membrane keyboard, compared with that using the conventional keyboard.

The main effect of typing group for this measure approached significance ($p = 0.084$). The largest speed difference between the keyboards occurred for the fair touch typists and the least difference occurred, as expected, for the nontouch typists. Post-hoc analyses (Duncan's Multiple Range Test) indicated that the difference in typing speed between the two keyboards for the fair touch typists was reliably greater than that for all groups other than the poor touch typists ($p < 0.05$), and the difference score for the nontouch typists was reliably less than that of all the touch typing groups ($p < 0.05$).

#### *3.3.4 Words per minute*

Perhaps the most relevant measure for the assessment of typing performance is the calculation of each subject's typing ability in terms of words per minute (WPM). This measure reflects a throughput measure of performance that takes into account both speed and accuracy:

$$\frac{\text{number of words typed} - \text{number of words in error}}{\text{total time in minutes}}$$

It was obviously expected that WPM for the typing tasks would vary directly with the typing group categories since this measure for a similar typing task was the basis for classifying individuals into typing groups. It was also likely that the experimental set-up per se might result in lower WPM scores than those calculated from the standard

Table VI—Average typing time difference between membrane and conventional keyboards (in seconds)

| | Touch Typing Group | | | | |
|---|---|---|---|---|---|
| | Excellent | Good | Fair | Poor | Non |
| Session 1 | | | | | |
| Trial 1 | 48.5 | 61.5 | 89.2 | 74.6 | 25.1 |
| Trial 2 | 24.2 | 27.0 | 52.8 | 40.1 | 11.6 |
| Session 2 | | | | | |
| Trial 1 | 59.0 | 67.2 | 62.9 | 52.5 | 32.3 |
| Trial 2 | 30.4 | 45.9 | 50.8 | 34.0 | 7.9 |
| Session 3 | | | | | |
| Trial 1 | 39.0 | 20.6 | 56.2 | 42.4 | 6.1 |
| Trial 2 | 21.7 | 32.8 | 48.7 | 23.6 | 16.5 |
| Mean | 37.1 | 42.5 | 60.1 | 44.5 | 14.0 |

typing tests given to each subject since many touch typists were unfamiliar with an electronic keyboard attached to a visual display.

The ANOVA on WPM scores confirmed these predictions. The main effects of all four experimental factors were significant: typing group [$F(4,16) = 9.45$, $p < 0.0005$]; keyboard [$F(1,6) = 58.7$, $p < 0.0005$]; session [$F(2,32) = 24.6$, $p < 0.0005$]; and trial [$F(1,16) = 23.0$, $p < 0.0005$]. Practice improved performance both across sessions and between trials within a session. The poor and nontouch typing groups differed reliably from the other groups, as shown in Table VII. Performance was also significantly better for the conventional than the membrane keyboard (35.4 vs. 25.9 WPM). However, a reliable interaction between keyboard and typing group [$F(4,16) = 3.74$, $p = 0.025$] indicated that the keyboard effect was attributable to performance differences for the excellent, good, and fair touch typing groups. Post-hoc analyses showed that performance differences between keyboards were not reliably different ($p > 0.05$) for either the poor touch typists or the nontouch typists.

The trial factor interacted significantly with the keyboard factor [$F(1,16) = 1.7$, $p = 0.003$] and with the session factor [$F(2,32) = 4.31$, $p = 0.022$]. The improvement with practice across trials within a session was somewhat greater with the membrane keyboard than with the conventional keyboard. Also, the degree of improvement between trials within a session diminished as practice (i.e., number of sessions) increased, with the largest difference between trials occurring, as expected, during the first session.

#### *3.3.5 Differences between keyboards in words per minute*

The relative, rather than the absolute, performance of each subject using the two keyboards was of primary concern in this study; hence,

Table VII—Average words per minute for typing tasks

| | Touch Typing Group | | | | |
|---|---|---|---|---|---|
| Keyboard | Excellent | Good | Fair | Poor | Non |
| Membrane | 32.1 | 30.4 | 25.4 | 22.1 | 20.2 |
| Conventional | 43.4 | 43.0 | 39.0 | 28.9 | 21.6 |
| Mean | 37.7 | 36.7 | 32.2 | 25.5 | 20.9 |

the percent difference in WPM using the two keyboards was computed to evaluate the relative advantage of the conventional keyboard over the membrane keyboard:

$$\frac{\text{WPM on conventional} - \text{WPM on membrane}}{\text{WPM on conventional}} \times 100.$$

This measure was computed for each subject for each typing task such that each difference score was based upon the same paragraph of text using each keyboard.

The ANOVA based upon this measure indicated significant main effects of the session [$F(2,32) = 3.57$, $p = 0.04$] and the trial [$F(1,16) = 17.6$, $p = 0.001$] factors. Both the session and trial effects reflected a decrease with practice in the percent difference in WPM between keyboards. The third session differed reliably from the first and second sessions (19.0% vs. 26.6% and 24.3%, respectively). The performance difference between keyboards was also reduced from the first to the second trial within a session (26.5% to 20.1%). The main effect of typing group approached significance [$F(4,16) = 2.93$, $p = 0.054$]. The conventional keyboard had relatively little advantage over the membrane keyboard for the nontouch typing group (6.5%), while this advantage was much more pronounced for the other touch typing groups (excellent = 26.0%, good = 25.5%, fair = 34.2%, and poor = 21.7%). The pattern of results for this measure for each group is depicted in Fig. 3. The difference between the touch typing groups and the nontouch typing group as well as the effect of practice are readily apparent. For the nontouch typists there is virtually no advantage of the conventional keyboard over the membrane keyboard; for one trial, subjects even performed slightly better with the membrane keyboard than with the conventional keyboard. For the touch typing groups the performance with the conventional keyboard was initially much better than that with the membrane keyboard. However, the percent advantage of the conventional keyboard over the membrane one was reduced substantially, although not completely, as a function of limited practice.

It is noteworthy that, despite this rapid learning function, the excellent touch typists appear to "forget" during the interval between sessions what they learned during a session. This phenomenon is

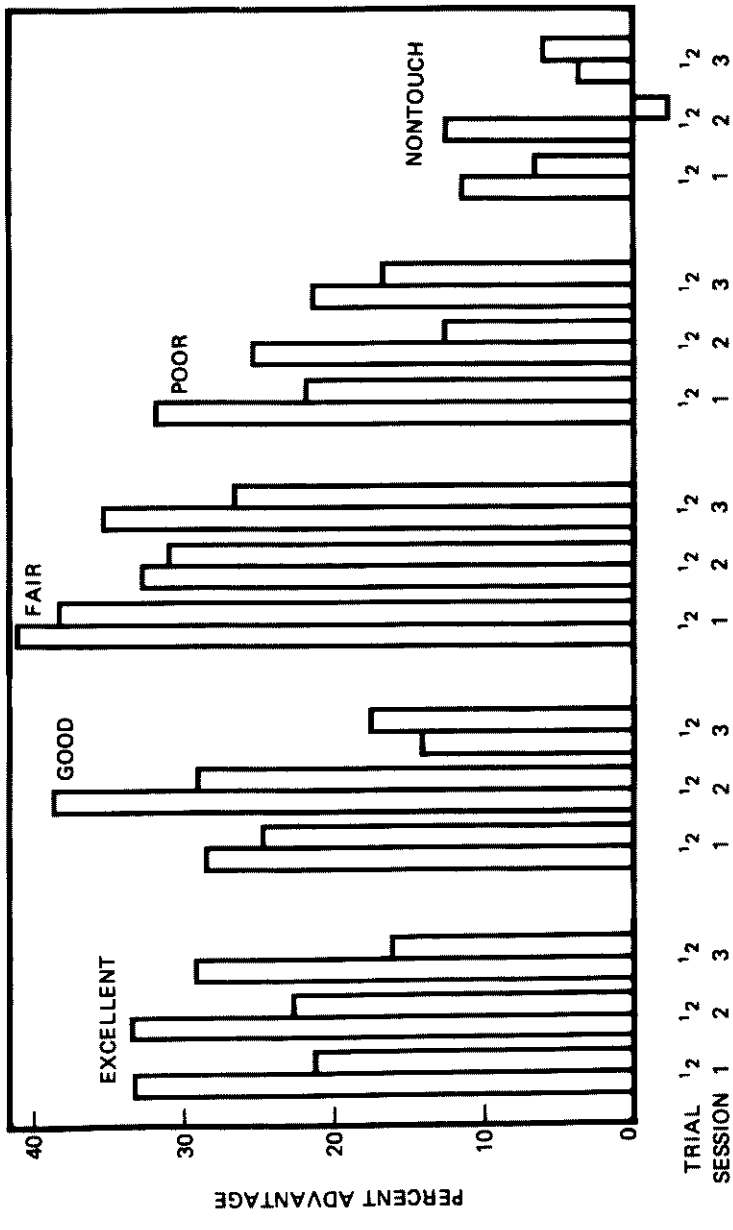



Fig. 3—Average words per minute advantage of conventional keyboard over membrane keyboard as a function of experience and touch typing group.

depicted by the larger advantage of the conventional keyboard over the membrane keyboard for the first trial of a new session, compared with that for the last trial of the prior session. This result, however, is less surprising when biographical data from these subjects are examined; these data indicate that the excellent touch typists spend about 28 hours per week using a conventional keyboard. These subjects were returning between sessions to a keyboard similar to the conventional one used here, and this activity between sessions may have produced retroactive interference, resulting in their forgetting what they had previously learned with respect to the membrane keyboard. Hence, the advantage of the conventional keyboard over the membrane keyboard was most pronounced at the beginning of each new session.

### *3.4 Typing summary*

For the typing tasks, there was evidence of rapid learning that resulted in improved performance on the membrane keyboard, relative to the conventional one, both within and across sessions. Practice effects were found on measures of both speed and accuracy. Keyboard effects were also evident. Performance was faster on the conventional keyboard than on the membrane one. Differences in speed between the two keyboards were most pronounced for the fair touch typists and virtually nonexistent for the nontouch typists. Also, more corrections were made with the membrane keyboard than the conventional one, especially at the beginning of a session.

When speed and accuracy data were combined to produce the WPM measure, practice effects were still evident. Keyboard effects were also evident, but primarily for the excellent, good, and fair touch typists. When the relative difference between keyboards in throughput performance was examined (i.e., percent advantage in terms of WPM), keyboard effects were apparent for the touch typing groups. Performance with the conventional keyboard was initially much better than it was with the membrane keyboard for touch typists. There was, however, rapid learning to use the membrane keyboard such that the advantage of the conventional keyboard, relative to the membrane one, was reduced substantially, although not completely. For nontouch typists there was virtually no advantage of the conventional keyboard over the membrane one. Finally, "forgetting" between sessions for excellent touch typists appears attributable to their customary use of another conventional keyboard during the interval between sessions.

## IV. SUMMARY

The purpose of this study was to compare performance using a membrane keyboard with that using a conventional keyboard for

different levels of typing proficiency. In general, the results indicate little difference in performance between keyboards for nontouch typists. For touch typists the results demonstrate better performance on conventional keyboards than on membrane ones. However, the results also demonstrate rapid learning to use the membrane keyboard. With just three hours of experience, performance differences between keyboards for touch typists reflected, at worst, a 27-percent advantage of the conventional keyboard over the membrane one and, at best, a 16-percent advantage.

These results provide human factors evidence that the difference in performance using membrane and conventional keyboards is much smaller than might be expected, given the absence of key travel feedback associated with each keystroke. Moreover, touch typists who are familiar with conventional, "full-travel" keyboards quickly learn to use membrane keyboards. For most touch typists, there remains an advantage for the conventional keyboard over the membrane keyboard after a limited exposure (only about three hours) to the novel keyboard.

These results are encouraging in terms of the application of membrane switch technology for some keyboard tasks. First, the particular membrane keyboard tested in this study does not necessarily represent an optimal membrane keyboard. Features of this particular keyboard that were identified as bothersome to subjects will be used in guiding the design of membrane keyboards for future evaluation. Second, extended practice with membrane keyboards may diminish residual differences in performance between conventional and membrane keyboards. Given the cost advantages and design flexibility afforded by membrane switch technology, such avenues may be well worth pursuing in the design of new products.

## V. ACKNOWLEDGMENTS

The contribution of the following people to this research project is greatly appreciated: R. Rosebrock for providing the experimental keyboards; P. Vittorio for measuring actuation forces; D. Kelly for developing the hardware and software interfaces; D. Lewis for testing subjects and reducing data; A. Glance, J. Papay, and P. Keany for reducing data; and M. Katz for making helpful comments and suggestions.

## REFERENCES

1. D. D. Lessenberry, S. J. Wanous, and C. H. Duncan, *College Typewriting*, Chicago: South-Western Publishing, 1965.
2. J. L. Rowe, A. C. Lloyd, F. E. Winger, *Typing 300, Volume One: General Course*, New York: McGraw-Hill, 1972.

## AUTHOR

**Karen M. Cohen Loeb,** B.A., 1970 (Psychology and Sociology), Washington University, St. Louis; M.A., 1972, and Ph.D., 1978 (Experimental Psychology and Education), Harvard University; Bell Laboratories, 1979—. In the period 1972 to 1979, Ms. Cohen was a research associate and faculty member in psychology at the University of Denver and a research psychologist at the Denver Research Institute. Ms. Cohen's field of specialization is human visual information processing and cognition. At Bell Laboratories, she has done human factors research in support of product development, market research, systems engineering, and software development. Projects have included videoteleconferencing, terminal design, videotex trials, voice messaging systems, and business communication systems. Member, Human Factors Society.

# Interface Design

Recent behavioral science work at Bell Laboratories and American Bell has focused on a portion of the user-machine interface, best characterized as dialogue design. This work has included the design of dialogues used for human-computer interactions, such as command languages, menu systems, and text editors. All of these are used in systems provided to customers and in the complex operations support systems used by telephone company employees. Dialogue design has also included new methods for interacting with advanced telephone services, such as call forwarding and teleconferencing.

The papers in this section focus on interface design. At best, from a human factors viewpoint, right from the start behavioral scientists were able to influence system design so that the most desirable interface was provided. In other cases, the functioning of machines was already determined, so that the behavioral scientist's job was to provide the best training or instructions to help users with their side of the dialogue.

The papers by Furnas, Landauer, Gomez, and Dumais, and by Streeter, Ackroff, and Taylor, are examples of research that guide the design of user-computer dialogue. The paper by Karhan, Riley, and Schoeffler describes the design of printed instructions to assist users of public telephones. The dialogue between people and telecommunications equipment often includes such interactions with printed material. The paper by Coke and Koether examines the readability of frequently used Bell System documents and the reading abilities of those who use them. Finally, the paper by Holmgren discusses the design of dialogues for users of automatic speech recognition devices.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# Statistical Semantics: Analysis of the Potential Performance of Key-Word Information Systems

By G. W. FURNAS,* T. K. LANDAUER,* L. M. GOMEZ,* and S. T. DUMAIS*

(Manuscript received February 10, 1982)

This paper examines how imprecision in the way humans name things might limit how well a computer can guess to what they are referring. People were asked to name things in a variety of domains: instructions for text-editing operations, index words for cooking recipes, categories for "want ads," and descriptions of common objects. We found that random pairs of people used the same word for an object only 10 to 20 percent of the time. But we also found that hit rates could be increased threefold by using norms on naming to pick optimal names, by recognizing as many of the users' various words as possible, and by allowing the user and the system several guesses in trying to hit upon the desired target.

## I. INTRODUCTION

Computer-based information management systems can store, manipulate, and transmit enormous quantities of information. They can allow almost unlimited organization, multiple indexing and cross referencing, and are capable of performing rapid and complex search operations. Thus, they can provide far more powerful tools for knowl-

* Bell Laboratories.

©Copyright 1983, American Telephone, & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

edge management than have been available previously. Such tools will be important to almost everybody: cooks wanting to find recipes, doctors needing patients' histories, managers tracking inventories, clerks filling orders and keeping records, and buyers looking for products. It would be fine if such systems could be used directly by inexperienced and occasional users, people whose main job or talents lie elsewhere. The hope is that new information-seeking tools could be operated with no more training than is needed to use a card-file or a book index, but with much greater success.

We believe that some of the greatest difficulties blocking this dream are psychological. Certainly, more available speed and capacity, and new algorithmic and data structure developments involving deep and difficult problems will be needed before the arrival of fully satisfying information systems. But, progress along these lines has already been enormous and continues at a brisk pace. Meanwhile, although even existing computational capabilities are very great, our ability to make them easily usable by nonspecialists has been quite limited. An important psychological problem is in understanding the relationship between what people say and what they want. This understanding is the key to designing systems that can infer what services or information users need from the input they provide. This basic capability is needed by information systems of many sorts, bibliographic reference search systems, business management databases, airline reservation systems, plant inventory and customer record systems, and even text processors. We believe that current systems generally fall very far short of the ideal of always knowing just what to give the user.

At this time although it is clear that there is a problem, very little relevant work has been done (see Carroll[1] and Landauer, Galotti, and Hartwell[2]). We do not even know the locus of the problem in the chain of actors and events. For example, the main problems may lie at the source; perhaps the human intellect is basically incapable of forming information specifications that are very precise. If so, perhaps no system could do much better than the current ones. But, it is also possible to believe that if systems could understand people as well as, say, close friends understand each other, there would be much less of a problem.

We have tried to get some idea of the extent of this problem by attacking a small but important part of it. We have asked people to give descriptions of various information objects, and analyzed their responses to determine how well the objects to which they refer can be inferred from what they say. We have begun by studying the referential properties of isolated words and short phrases.

Our goals in this research have been twofold; first, to advance understanding of the psychological processes by which human seman-

tic reference is generated, and second, to model and estimate the strengths and weaknesses of information systems that take human-generated descriptions of sought items as their input. Empirical observations of naming behavior provide the necessary data for both enterprises.

In this article we first describe four sets of object-description data, the way they were collected and reduced, and some of their more interesting features. Then we present a series of analyses in which we treat the collected descriptions both as representative of what users would provide as input to information retrieval systems, and as the source of the information that the system would use in determining user wants. The hypothetical systems we consider are limited to ones in which the user's initial entry or query consists of a single word or short phrase. We do not attempt to analyze the possible performance of systems that make use of sentential syntax, or linguistic or real world context. The reason for this limitation is largely pragmatic; it postpones analysis of many difficult complexities. However, the characteristics and limitations of single-word to object reference that we have investigated have strong implications for many access methods (by which we mean the access method provided for the user, not that used by the program). We are, of course, aware that there are data access methods that do not start with the user entering a key word or phrase; for example, there are strictly menu-driven systems, ones that rely on well-formed queries and restricted query languages, and also ones that attempt some form of natural language understanding. These other methods share some, but not all, of the same conceptual and practical difficulties of key-word methods, and each raises somewhat unique and interesting psychological issues of its own. Where our data bear on these issues in reasonably direct ways, we offer some comments, but we focus primarily on the key-word comprehension process and its ramifications. In the final section of this paper, we discuss some of the reasons why key-word access is as limited as we find it to be and consider several potential methods for overcoming the deficiencies.

## II. DESCRIPTION OF DATA SETS

There are things a computer system has or does to which a user might wish to refer. These "information objects" are just the objects, e.g., operations or data sets, to which commands and queries apply. We have collected descriptions of information objects in four quite disparate domains, chosen in an intentional effort to achieve variety, and also for their relevance to a number of special problems that are outside the concerns of this paper. The four sets are: the verbs used in spontaneous descriptions of the operations needed to perform

manual text-editing operations, descriptions of named common objects designed to induce another person or a computer to return the name in a game like the *PASSWORD*™ game, superordinate category names for items available in a swap-and-sale listing similar to classified ads in newspapers, and index words provided for a set of main-course cooking recipes.

In this section we will describe how the object specifications were obtained from people, how they were reduced to single-word or short-phrase keys, and then summarize a few qualitative and statistical characteristics of the responses.

### *2.1 Text-editing operations*

The first data came from language applied to text editing. The study was one of two conducted to explore "naturalness" in command names.[2] Forty-eight secretarial and high school students were asked to provide instructions to another hypothetical typist describing what operations needed to be performed on text marked by an author for correction. These corrections involved two examples each of 25 sorts of edits: five basic operations (insert, delete, move, change, and transpose) on each of five textual units (blanks, characters, words, lines, and paragraphs).

Preprocessing in this case reduced each response to the main verb or phrase in the instruction describing how to perform the editing operation. While this was in fact accomplished manually here, we believe that a simple parser and English word list could in principle have given nearly identical results. These expressions may be considered candidates for command names for editing systems.

Perhaps the most striking result was that there was extensive disagreement in the verbs people produced. This point is the main focus of the current article, and will be dealt with in considerable detail later. For now let us just make a few preliminary notes, e.g., that the three most popular names for each operation accounted for only 33 percent of the total number of responses. The intersubject agreement, the probability that any two people used the same verb in describing a particular text correction, is only .08. Since each of the 25 sorts of edits occurred twice, we also have a measure of within-subject (with 1200 observations) agreement. The probability that an individual subject used the same main verb in the two cases was .34.

What agreement there was did not favor the terminology used by our locally popular editor (the *UNIX** Operating System text editor ed). For 24 out of 25 of the types of edits, the name in ed was not the most frequent spontaneously given name. Use of the terms "delete" and "substitute" was quite rare, for example. (Landauer et al.[2] went

* Trademark of Bell Laboratories.

on to show, however, that this caused no problems initially in learning the basic editor.) People preferred "add" for the insert operations, "omit" for delete operations, and "change" for the replace operations. There was little consensus in describing the transpose and move operations.

### *2.2 Common object descriptions*

These data were originally collected in a study by Dumais and Landauer[3] that examined how people naturally obtain information from one another. The information objects here were names of 50 common items chosen from 10 "categories." The categories were: cities, proper names, clothing, animals, food, household items, abstract words, a category of words with highly associated opposites (e.g., black, love), and two categories whose members were words selected in such a way that negation might figure strongly in the descriptions, e.g., to eliminate the unwanted set members. A total of 337 New York University students were asked to write down a description that would enable another person (or in half the cases a hypothetical computer) to guess the object. There were no restrictions as to the form or content of the descriptions, except that they could not contain the target word itself. Subjects also indicated whether they had any computer experience. Those with at least one computer course were classified as computer-"experienced" in the data summaries discussed below.

A subsequent study was conducted to evaluate the effectiveness of the descriptions generated in the first study. Twenty-five subjects (6 Murray Hill area homemakers and 19 employees of Bell Laboratories) were each given 150 descriptions randomly selected from those generated by the NYU students. They were asked to: (1) guess (without knowing the alternatives) the item being described, and (2) indicate on a five-point scale their confidence in their guess.

Principal results from the main study include the finding that when communications were intended for computers, people with computer experience were relatively more terse, and nonexperienced people were relatively more verbose than when communications were intended for people. However, there was no simple relationship between verbosity and effectiveness, i.e., guessing accuracy as indicated by the second study. People were somewhat more successful in guessing the target items when the descriptions had been provided by people without computer experience (81.2 percent vs. 78.5 percent), but this difference is not statistically significant.

A point of considerable interest here was the style of specification that subjects used. Subjects' descriptions were not very precise; typically they refer to a whole set of items, not just the intended target (although we have no good measure of this other than informal ratings

of denotative class size). Still, the average successful guess rate of the second group of subjects was over 80 percent. We will return to a discussion of this paradoxically high success rate towards the end of this paper. The most frequent way of specifying target items, used about 60 percent of the time, was to describe them in terms of a superordinate (sometimes followed by characteristics or attributes that distinguish the intended target from other members of the superordinate category). Another fairly common form of description (~20 percent) was the use of exemplars. For several of the target words (e.g., motorcycle, magazine, sports, games, science), subjects listed examples of more specific items falling into the target category (e.g., Harley, Suzuki ..., in the case of motorcycle) instead of attempting a more formal definition. Negations (and opposites) were used less than 50 percent of the time for the words we thought were particularly amenable to this form of description.

For the purposes of the analyses undertaken in the current paper, these descriptions were preprocessed to merge minor variations in as automatic a fashion as possible: uppercase was folded to lowercase, word endings were stripped (plurals, tense markers, etc.), and "non-content" words (including articles, imperatives, conjunctions, prepositions, pronouns, and tenses of the verb "to be") were removed.

An average of 8 words per description were in this way condensed to an average of 5.4 words. The first of the remaining words (i.e., first standardized content word) was tabulated for the statistical analyses.

### *2.3 Superordinate categories for swap-and-sale items*

The major purpose of collecting these data was to develop empirical networks of "ISA" relations, that is, classification hierarchies based on user knowledge and representations, for a set of items to be incorporated in an experimental menu-driven information access system.[4]

The information objects were 64 items taken randomly from roughly 300 entries on a monthly bulletin board listing of items for swap and sale at Bell Laboratories. The subjects were 30 local New Jersey homemakers. Each subject worked with a random 32 of the 64 target items. They were told that they were participating in the study to find out how they classified various items being sold on local bulletin boards. The use of these categories in helping people in future computerized retrieval systems was mentioned. Subjects were instructed to complete successive "All ____ are ____" sentences. Beginning with the specific target they were to give its immediate superordinate (e.g., "all red Delicious apples for sale @10¢ ea. are apples"). Then they copied the first given superordinate ("apples") to the beginning of the next incomplete sentence and finished the new sentence with a still

more general category (e.g., "all apples are fruit"). They were to continue in this way (e.g., "all fruits are food") until they could go no further. They were then to go back and find some category that had another superordinate in addition to the one they had already cited, and list that category with its new superordinate. These data were also standardized by stripping off endings and discarding noncontent words.

On average, an individual subject produced 2.1 different chains of successively more general superordinate categories for each stimulus. The chains averaged 2.5 superordinates each, with superordinate categories named in phrases containing an average of 1.7 standardized (content) words. For the current paper, only the lowest level (most specific) category, from the first generated chain of superordinates, was used. The category name was used in its (standardized) entirety.

The categories given in this study make it apparent that the construction of a network that will faithfully match all users' conceptions of a domain is not an easy matter. People have difficulty in generating superordinates and show considerable disagreement as to how things should be grouped under those superordinates. Categorization and indexing schemes currently in use always depend on a user's either generating the same superordinate as the system knows about, or at least being able to choose the right one from a list. Perhaps the difficulty and lack of agreement among people in categorizing information objects account for much of the perceived deficiency of current menu-driven data access methods.

### *2.4 Recipe index words*

The original motive for this data set was to study the effect of domain expertise (i.e., cooking skill) on indexing and key-word usage.[5] The information objects were 188 main-course cooking recipes (French, Italian, Mexican, and American cuisine) taken from 12 cookbooks of explicitly varied sophistication (ranging from a garden club's cookbook,[6] through *The New York Times Cookbook*,[7] to *The Art of French Cooking*[8]).

There were three groups of eight subjects each: experts, who taught cooking classes; and intermediates and novices, selected from local homemakers who came out at the high and low extremes of several self-rating scales on culinary sophistication.

Subjects were told their task was to describe each of the recipes in key-word form, selecting at least three but no more than seven descriptive words or brief phrases for each recipe. They were told that their job was similar to that of a librarian who is creating an index or card catalog, and that the descriptions should be useful to another person trying to locate that recipe in a large set of recipes. Half of the subjects

in each experience group were instructed to direct their descriptions to expert cooks using the index, the other half to novice cooks. The description task was self-paced by each subject in her home. Subjects required between 5 and 10 hours to complete the task.

Terms here were again preprocessed to remove word endings and noncontent words. All multiple-word productions were scored by two judges (the experimenters) to determine if the phrase could be decomposed into its constituent words and maintain its meaning. Subjects produced an average of 5.4 key words per recipe, the first, "most important" of which was studied here.

Again we found considerable diversity. For the 188 recipes, a total of 303 different word types were used by the 8 experts, 220 by the 8 intermediates, and 252 by the 8 novices. It is interesting to speculate on the reasons why these groups differ. Perhaps the experts have a large and specialized vocabulary and the novices have an unruly, haphazard one. In any case, there seems to be something more conventional about the word use of intermediates, a point to which we will return later.

### *2.5 General comments on the data sets*

These data sets all pertain to information objects that one might want accessible on a computer. They were also all of modest size. Other than that, though, they tapped very different knowledge domains, they asked for specification in a number of different ways, and they were provided by different kinds of people. Moreover, the method of reducing the free-form descriptions given by our participants to single words and short phrases varied somewhat from one case to the other. This variety of data is important for our purposes. In order for results to have any pretense of robustness, it is important that they be obtained on a sufficient variety of cases to assure that it is not the particulars of the objects at hand that are responsible for the observed characteristics. We know of no way to actually sample data domains, descriptive methods, and reduction methods in a representative way. However, we believe that results that hold for all of the disparate sets that we have studied stand a good chance of holding for most others.

For each domain, our data can be represented as a table in which the rows are words provided by the subject, the columns are the objects to which these words were applied as descriptions, and the cell entries indicate the number of times each word was used in the description of a given information object. The questions we ask concern how the information contained in such a table might be used to guess from an input word what object is intended. Two partial tables are shown in Tables Ia and b. Table Ia is derived from the text-editing study (for five objects) and Table Ib from the common object data. The numbers

Table I—Word-object data

(a) Sample data from the text-editing study

| Words | Objects | | | | |
|---|---|---|---|---|---|
| | insert | delete | replace | move | transpose |
| change | 30 | 22 | 60 | 30 | 41 |
| remove | 0 | 21 | 12 | 17 | 5 |
| spell | 4 | 14 | 13 | 12 | 10 |
| reverse | 0 | 0 | 0 | 0 | 27 |
| leave | 10 | 0 | 0 | 1 | 0 |
| make into | 0 | 4 | 0 | 0 | 1 |
| ... | | | | | |

(b) Sample data from the common object study

| Words | Objects | | | | | |
|---|---|---|---|---|---|---|
| | calculator | lime | Lucille Ball | pear | raisin | robin... |
| small | 17 | 0 | 0 | 0 | 7 | 4 |
| machine | 4 | 0 | 0 | 0 | 0 | 0 |
| green | 0 | 18 | 0 | 7 | 0 | 0 |
| bird | 0 | 0 | 0 | 0 | 0 | 21 |
| fruit | 0 | 1 | 0 | 19 | 1 | 0 |
| red | 0 | 0 | 8 | 0 | 0 | 7 |
| female | 0 | 0 | 2 | 0 | 0 | 0 |
| ... | | | | | | |

in the tables represent the frequency with which a word was used to refer to an object.

In fact, there is an implicit third dimension to these tables, representing the person from whom the description was obtained, and sometimes a fourth dimension, representing which of several words of a multiple-word description provided by a given subject is involved. However, for most of the analyses we consider only the first word given, and the matrices are all very sparse, so we have chosen to collapse across subjects. It is worth noting that the tables are not sparse simply because we have failed to collect enough data. Word usage tends to resemble Zipf's distribution[9] (supposedly straight line relation between occurrence frequency and rank frequency when both are plotted logarithmically) in that a few words are used very frequently and many words (over 340 here) used only once (see Fig. 1). As more and more data are collected some cells increase in frequency, but the number of unique words also grows so that the sparseness of the table tends to be preserved. Moreover, most words refer only to a limited number of objects, so that such tables usually have a large number of empty cells.

## III. INTRODUCTION TO ANALYSES TO DETERMINE REFERENT EFFECTIVENESS

In the analyses that we report in Section IV, we have been interested in how much information about referent object identity is contained

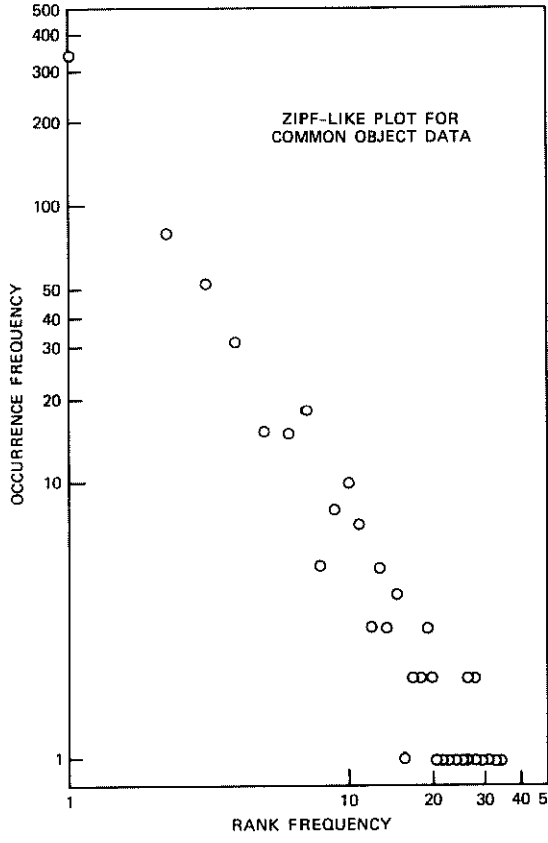



Fig. 1—Plot of common object word usage data.

in the words used to describe them. To consider how good a word or expression is as a reference to some object, it is necessary to suppose some sort of a mechanism by which the expression is comprehended. We can only estimate the value of inputs as determinants of output if we can specify the function that takes one into the other. The inputs of our problem are the words provided by people in specifying information objects. The output is a guess or set of guesses about which object was intended. Our approach has been to specify various ways in which the input can be made to yield the output. We have considered models or functions that are intended to mimic what happens in typical information systems and also ones that try to improve on current methods in various ways. We also develop models that allow us to estimate what ideal input/output mappings could accomplish. It is in this sense of ideal or asymptotic performance that we can appraise the limits of word reference.

The models that we examine have in common that they use the frequency table of users' word reference, and only this data, to guide the system in guessing users' intents.

### *3.1 The general model*

In the sections below we consider a number of special cases of the following rather general baseline model. To retrieve a desired item the user is assumed to enter one (though in some instances, $K$ different) key word(s). The system is assumed to make either one or $M$ guesses as to the user's intent. Success is counted if the intended object is among the system guesses.

As an actual system, our baseline model would operate as follows. The user would enter one (or more) key words or phrases. If it could recognize the input at all, the system would return one or more potential objects according to some rule, perhaps relying on data in a table of observed word-use frequencies. (If it could not recognize the input, it would return nothing.) In cases where the system returns several alternative objects, it can be assumed that the user would be able to select from among them the one intended without error. This corresponds to a simple case of what might be called a "menu-on-the-fly" technique, in which the first entry is a freely chosen key word, and the next step is a choice among a menu of items selected on the basis of the system's knowledge about referents the human user might have intended by that key word.

We will now describe the models that we have analyzed, beginning with the simplest ones, those closest to the way typical systems are currently set up, and proceed to more sophisticated models. Each model is characterized by a set of assumptions or constraints on the input/output function of the system.

Sections 3.2 to 4.7 consider a number of models and analyses in considerable technical detail. Some readers may wish to skip these sections and continue to a summary of the highlights in Section 4.8.

To give a preview, the results convince us of three major points. First, the vocabulary problem is a serious one for untutored users of computers, particularly given the poor naming techniques typical of current practice. Second, the difficulties come from a severe and fundamental lack of consensus in the language community on what to call things. Finally, our research suggests that, by using a very non-standard approach, namely trying to make the best possible guesses on all user words, substantial improvements could be made in meeting the underlying objective of providing access to the things people want.

### *3.2 Formalizing the general model*

Let us make things more formal. We have two sorts of entities, the user-generated inputs, referred to here as "words," "terms," or "de-

scriptors," and the system outputs, called "information objects," "targets," or just "objects." We are concerned with two relationships between these entities. On the system's side are the associations dictating how the system will respond to a user's words—which words will be associated with what system objects. On the user's side are the actual intended associations—what the user in fact meant by a given word. Both the system's and the user's relationships can be shown in the form of a table. The *system table* gives the input/output mapping of which object the system will retrieve (execute, etc.) for what words, with a 1 wherever a word is associated with a system object, and a 0 otherwise. The *user table* shows the frequencies with which people say what words for what objects. Examples of a user table were shown in Table I. This table can be used in the evaluation, and, if one wishes, in the design of the system table.

#### *3.2.1 The system table*

The system table has two aspects. First is what might be called the set of structural constraints. These concern how much and what variety of input the system will respond to, and are imposed by capacity limitations on memory, processing, data collection, or sometimes by the computer algorithms used. Second is the choice of a particular instance of the input/output system table, that is, the assignment of exactly which words the system will respond to, and with what objects. One design problem centers on whether assignments will be made in an effective or possibly optimal way.

Structural constraints in the system's input/output table correspond to general restrictions on which cells may be used in the mapping of input to output. We consider patterns based on limiting the number of cells in rows and columns (e.g., the number of words understood as referring to each object and the number of objects taken as possible referents for each word). In particular, we will be considering models where, in the system table, there are various restrictions on:

1. Total words recognized by the system (number of nonzero rows in the system table)
2. Number of words recognized for each object (column totals)
3. Total objects referenced (number of nonzero columns)
4. Number of objects referenced by each word (row totals)
5. Total word-object associations "understood" by the system (grand total of cells).

Once a general structure has been determined, one still must choose which associations to give the system, i.e., exactly which cells to include in the mapping. We consider three cases that give rise to different versions of each model. The first is a purely random assignment. Within the specified structural constraints, the cells defining

the system's associations between words and objects are chosen randomly. Clearly, this is not a realistic system model, but it is useful in the conceptual analysis of other models and as a baseline of comparison. The second case is a weighted random assignment, assigning a word to an object with the same probability that the word and object occur together in usage. That is, input/output associations are chosen with probabilities proportional to their frequency of use, as indicated by the user table. This is, in fact, a very important case to consider, since it approximates the way many systems are currently designed. We will often refer to this as the "armchair" method: a single human designer sits in his or her proverbial armchair and makes the name-object associations by some sort of intuitive guessing. The user table is in effect a compendium of many humans' "armchair" attempts to assign good words to the system objects. A weighted random sample from the table thus corresponds to a random person's "armchair" nomination. Such weighted sampling therefore allows us to estimate the effectiveness of "armchair" design. The last assignment technique is by the best available optimization procedure. Optimization methods make generous use of the empirical data in the user table to pick the best cells. For some system constraints we do not know a combinatorially feasible way to find the best configuration, but it is possible to improve substantially upon the weighted random method. Note that the success of any optimization attempt is also limited by the quality of the data available in the user table.

#### *3.2.2 The user table*

The user table is a data matrix compiled from the studies mentioned earlier in this paper. It records the frequency with which untrained users employed various terms in referring to the system objects.

We assume that the descriptions given by our subjects for the information objects in the various sets bear a close resemblance to the entries similar users would make in trying to specify the same objects in a database system. This assumption may be wrong in detail; in using actual systems people might give descriptions somewhat different from those induced by our instructions. However, when we varied the intended recipient of the descriptions, as in the recipe (experts, novices) and common object data (people, machines), there were only small changes in the descriptions. So we believe that descriptive language would change little in attempts to communicate with real, as compared to hypothetical, systems. (We cannot, however, estimate the effect that prolonged interaction with a given system might have on a user's vocabulary.)

We also assume that collection of user descriptions is a good basis for predicting actual user intent from key-word input. In approaching

a data access system, the user must have some information object in mind, and must give a description to the system. In actual use, it is difficult to know what the user really has in mind. Sometimes the user has no clear idea at all. It is not apparent whether this "diffuse target" case is easier or harder for a system to handle. A user with an unclear goal may or may not be more easily satisfied, but is probably less likely to give precise specifications.

It is obvious, however, that the diffuse target case is harder to study. Reliably inducing such a state in the user is difficult, and the system's success is not easily appraised. Therefore, we have collected data and investigated models based on the clear target case.

Thus, in these models it seems to us that the best information the system could start with is to know what typical users would give as descriptions, given the intended object was well known and clearly defined. Our assumption is that the descriptions people give of objects when we specify them will closely resemble the description they would give if they had thought of the specific objects themselves. This seemed a practicable and, we hope, realistic starting point.

#### *3.2.3 Going through the models*

Insofar as possible, we will use a consistent expository framework to describe each model. The framework is as follows:

1. Name: A mnemonic title, by which the model will be referenced.
2. Interpretation: What the model is all about.
3. Motivations: Where and why it might come up.
4. Structural constraints: The general form of the system's input/output table ("system table") for the model. This will be given as clearly as possible in English, with a more formal summary of the constraints for each model appearing in Appendix A. This appendix is a very useful reference source in comparing the models.
5. Analyses: For each of three methods of selecting cells for the system table (random, weighted random, and optimized), the following information will be given whenever possible.
   (a) Version: what the method means in this model.
   (b) Evaluation by: what statistic is used to assess its success.
   (c) Result: Statistics are computed for the four data sets:
      (1) Edit command data
         (a) All 25 Operation x Text-Unit combinations (abbreviated: "Ed25")
         (b) For purposes of comparison we include a second analysis of the same data: the five basic operations collapsed across textual units ("Ed5")
      (2) Common object data, all 50 objects ("CmOb")
      (3) Swap-and-sale data, the 64 sale items ("Swap")
      (4) Recipe data, the 188 main course recipes ("Recp")

The following statistics will be given:

Recall probability: how often the method succeeds in returning the desired object.

Mean number returned: when the system is able to make a guess about the user's desired object; this is a record of how many guesses it makes to achieve the reported probability of recall.

A few comments are needed about the "number returned" statistic. First note its importance in considering recall success probabilities. If the system returns a large number of guesses, it can obviously be expected to have a greater chance of including the target. Thus, comparisons of performance demand that this number be kept in mind. Second, note that these indicate the number of things returned when the system in fact recognizes the user's input and so is in fact able to hazard a guess. For some systems it will be common to have insufficient data and to make no guess at all for many user words.

We denote the number of objects, or columns, in the table by $C$, and the number of user words, or rows, by $R$. The former is in part well defined by the designers of the system but the latter, the vocabulary size, is usually an artifact of data collection, as it would tend to expand if more data were collected. Here it refers to the size of the vocabulary in the corpus of data we collected.

## IV. ANALYSES TO DETERMINE REFERENT EFFECTIVENESS

### *4.1 Model 1: "One name per object."*

#### *4.1.1 Interpretation*

Each object in the system is assigned a single term or name. The user enters one term. Success depends on the user's word coinciding with the system's.

#### *4.1.2 Motivations*

This approach is common in computer systems—each entity has one and only one name. The name is usually chosen by the designer or by an expert indexer. The designer hopes to establish a convention about what system objects will be called. Users must either learn or guess the names to make the system work. Such learning may be feasible in small systems or for highly practiced users. The growing community of novice and infrequent users, however, are often reduced to trying to second-guess the system, even to find documentation. This is often frustrating and we shall see that in principle the approach is far from adequate. Moreover, the real difficulties of such a scheme often go unremarked simply because traditional computer users have come to accept as normal the necessity of extended learning, repeated second-guessing, lengthy searches, or expert consultation in finding the correct names for programming commands, file names, or information categories.

#### 4.1.3 Structure

The only constraint is that there be exactly one word for each object. Note especially that this model allows the possibility that any given word can be used to name more than one object. This is a situation that designers often try to avoid. It nevertheless arises in several situations, as when programs, commands, file names, or index categories are assigned by many independent users, or to highly similar objects (like bibliographic subject or author specifications, or the case of several functions being collapsed under a single command name). We evaluate models that disallow such "collisions" later. Note that for the likelihood of recalling a given object at all, the aspect of the problem on which we focus in these first analyses, assigning a word to more than one object as we do here, can only improve the expected system performance.

#### 4.1.4 Analyses

***4.1.4.1 Version: random.*** For each object one name is chosen randomly from the total vocabulary for that object, i.e., one cell from each column. That is, the name of each object is a random choice among all the total set of descriptors given to all the objects.

*Evaluation: by theoretical value.* In this case we can calculate the expected performance of the system exactly. The success rate is given simply by the ratio of $t$, the total number of cells included in the system mapping, to $RC$, the total number of cells in the matrix. A simple proof of this appears in Appendix B. See Results Table 1.

Results Table 1

| | |
|---|---|
| recall probability = | $C/RC = 1/R$, i.e., the reciprocal of the total number of words that users use for all the objects |
| | Ed5 Ed25 CmOb Swap Recp |
| recall probability = | .012 .008 .002 .001 .001 |
| mean number returned = | $1 + (C - 1)/R$ |

The "mean number returned" is a measure of the amount of ambiguity or imprecision in the terms as the system understands them. It is the average number of things the system knows by a given name that the system recognizes; the system must return all of these objects when trying to guess a target. For some of our models the number of objects that the system returns will be fixed ahead of time by design, but here it is the mean of a random variable, easily calculated to be $1 + (C - 1)/R$.*

* Under the pure random method, each object has one of the $R$ words associated with it randomly and independently. Thus, for each object, any given word arising or not becomes a Bernoulli event with probability $1/R$. So having chosen a word for one object,

*4.1.4.2 Version: weighted random.* A system name is assigned to an object with the same probability that users attributed the term to the object. Here the user enters one key word; and the system has had a key word assigned to each object, based on one other person's nomination. This model mimics, more or less, a currently fairly typical situation for key-word systems (or program-name or command-name access systems) in which the system designer has provided an entry name for each object, obtained only from one usage datum (the designer or indexer's "armchair" introspection), and the user is required to enter just that name. We assume that system designers or indexers are like our subjects in their choice of names, so that the relative frequency with which a name was given to an object by our subjects provides an estimate of the likelihood that a designer would choose that name for that object. (One source of support for this assumption is that experts gave no more consistent names than novices; see below.)

*Evaluation: by the "column repeat rate" statistic.* We estimated how well such a system is likely to work by estimating the probability that a given word chosen randomly from our population (e.g., by one user on one occasion) would match another word chosen from the same population (e.g., by one designer). This probability is known as the repeat rate.[10] If we index rows (words) by $i$, and columns (objects) by $j$, in a word-by-object table, then repeat rate for a given object, $rep_j$ is defined as:

$$r_j = \sum_i p_{ij}^2.$$

This formula can be understood by considering that a match between two randomly chosen words occurs when, for any given word first chosen, the second word is the same. Say the first word was $word_i$; the second word will match it with the probability of $word_i$ occurring in the population, $p_{ij}$. The probability of a $word_i$ being the first word is also $p_{ij}$, so the probability of $word_i$ being involved in a match is $p_{ij}^2$. Summing across all possible words, we get the equation given above.

An unbiased estimate of the population probability of such a match comes from calculating the true probability of such a coincidence in drawing from our sample without replacement, given by:

---

the number of the other $C - 1$ objects having randomly been assigned that same word is given by a binomial distribution with parameters $p = 1/R$ and $C - 1$. Thus the mean number of other objects with the same name as our given object is the mean of this distribution, $(C - 1)/R$. The system will return any of these objects plus the original object, or $1 + (C - 1)/R$ objects.

$$\hat{r}_j = \sum_i \frac{n_{ij}}{N} \frac{n_{ij} - 1}{N - 1},$$

where $N$ is a column total and $n_{ij}$ is the frequency of the $i$th cell in the column $j$.

Here we are interested in the average probability of success, given any particular target, throughout the table. So we decompose the overall probability by conditionalizing on columns, calculating the repeat rate for each column, and then average these success probabilities using weights proportional to the column total frequencies. (In our data these column totals are approximately equal by design.) These weighted average column repeat rates are given below for our four sets of data. These numbers may be interpreted as answering the following question: Given that all objects are equally often the desired target, what is the probability that the name given by a user trying to specify a target would match the name assigned to it by a designer? See Results Table 2.

Results Table 2

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .07 | .11 | .12 | .14 | .18 |

While there is some variation among the values, they are all quite small. People do not agree with one another very well as to the first word or phrase with which to label an object. The probability that two typists will use the same main verb in describing an edit operation is less than one in fifteen. The probability that two people will use the same first key word for a recipe is less than one in five. (These numbers also tell us something about the size of the set of alternatives that people use in their disagreement. It is a property of repeat rate that the set of alternatives must be at least $1/r$, and can be quite a bit larger if they are not equally likely, as is the case here.)

Most of the interesting comparisons will be between the different models presented (e.g., this model with the subsequent ones), and not the different data sets. To facilitate model-to-model comparison, all results appear together in a summary table in Appendix C.* The reader is strongly encouraged to refer to this table throughout Section IV.

* Of course, some comparisons between data sets are also of interest. Note for example that the value for Ed25 is higher than for Ed5. This says that the set of words applied to the individual objects is more sharply restricted than for the collapsed classes. This is to be expected, since any diversity between objects in the pattern of terms applied becomes within-class diversity, when the objects are collapsed together, driving the column repeat rate down. Only if all objects in a class had identical naming patterns would the repeat rate not decrease.

The mimicked method is one in which the designer provides the system with only one entry word that it can understand, and the user enters just one key word. This is clearly unsatisfactory for untrained users. The usual solution has been for system designers to rely on users learning the chosen vocabulary, i.e., to try to force the user's table to adapt to a fairly arbitrary system table. When the system is small and the user's interaction frequent, this can work quite well. Indeed, Landauer et al.[2] have shown that using unrelated random names has little or no detrimental effect on initially learning to operate a small editor. But, if the system is large and its use intermittent and nonexpert (as for example in large-scale information retrieval systems like library catalogs, recipe files, or classified product catalogs), it is simply unreasonable to require users to learn a specialized vocabulary.* Despite the designers intentions, the uninitiated will try to make the system work without memorizing extensive naming conventions. Thus the problem remains a real one.

One approach we might consider at this point is to seek expert advice in choosing names. This is a fairly common approach, taken in the hope that experts in a given subject area know what things are, or should be, called and so might generate words of more general currency. Indeed, the indexes to books, libraries, user manuals, and other information sources are customarily created either by subject matter experts or by professional indexers.

We have collected some data relevant to this issue in the recipe study. Our key-word providers represented several levels of expertise. The situation is not unrepresentative; usually the indexes of cookbooks and recipe collections are created by cooking experts, presumably on the assumption that their characterization and labeling will be superior, even for less sophisticated users. We calculated repeat rates separately for the three groups of key-word providers (experts, intermediates, and novices) subdivided by whether they had produced the key words under instructions to make them appropriate to novices or to experts. The results are shown in Table II. The repeat rates shown were calculated in a special way. For each cell, the index words were provided by particular subsets of describers, and the proportion of matches was calculated on a pool of descriptor words provided by other subjects. Thus, the (ee, ee) cell estimates how likely a word provided by one expert for other experts was to match that provided by another expert with the same, expert, audience in mind. The (en, ne) cell estimates the probability that a word provided by an expert

* In intermediate cases, like program and command names for an operating system, the method may be satisfactory for expert users, while leading to dissatisfaction for others (see Refs. 11 and 12).

Table II—Repeat rate measures of agreement between indexers and users for the recipe data set

| Group | ee | en | ie | in | ne | nn |
|---|---|---|---|---|---|---|
| ee | .11 | .17 | .14 | .15 | .10 | .22 |
| en | .17 | .11 | .18 | .17 | .16 | .21 |
| ie | .14 | .18 | .20 | .20 | .21 | .23 |
| in | .15 | .17 | .20 | .19 | .17 | .26 |
| ne | .10 | .16 | .21 | .17 | .16 | .16 |
| nn | .22 | .21 | .23 | .26 | .16 | .32 |

for novice users would match the word provided by a novice for expert users. Clearly, the differences among marginal values (i.e., the averages of repeat rates for different index providers) are not large, and more clearly still, expert cooks do not provide better descriptions for the use of either other experts or novices. (If anything, novices do the best in using each other's words.)

The usual armchair approach, even if undertaken by subject matter experts, has only a small rate of success. The obvious step at this point is to seek explicitly optimal choices of names, treating this as the empirical question that it clearly is.

***4.1.4.3 Version: optimized (best).*** If we want to use the name that has the greatest currency among subjects, we must choose the term that is in fact maximally used by subjects for each object. We pick the maximum cell in each column and use the corresponding term.

*Evaluation: by various estimates of the range of expected performance.* The lower bound is based on split halves analysis; the upper bound is based on a transformation of the column repeat rate and from an analysis that assumes that the sample data exactly reflect the population probabilities.

To know the performance of this model we need to know the true population magnitude of the maximum cell in a column. That cell is the one we would choose according to an optimum name assignment scheme, and its size would be the proportion of future users' terms for the given object that would coincide with our optimal choice. Problems arise in that there are no known distribution-free, unbiased estimators of the population magnitude of the maximum probability cell. We have, however, been able to devise a few techniques that let us put bounds on the performance of this system. The first uses a split-halves technique to give a lower bound estimate. The data in each column are split into two halves, and the maximum cell chosen on the basis of the first half (as though we had to design our "optimal" system on the basis of half the data). This cell is then matched against the second half to see how well it succeeds. Thus, the second half acts as a virtual experimental test of the performance of the "optimal" method. The split-half results shown here are the average performance of ten independent splits of each data set. See Results Table 3.

Results Table 3

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .15 | .19 | .26 | .26 | .31 |

These numbers are an unbiased estimate of how well a system would do using this optimum strategy but constrained to a small amount of data (namely the amount in each half). It clearly underestimates how well one could do with more data. We have used two approaches to obtaining an upper bound. One is based on an interesting inequality relation that holds between the size of the maximum cell and the repeat rate statistic described above: It can be shown that the former is no greater than the square root of the latter.*

Thus, the performance of the optimum model is expected to be no greater than the square root of the performance of the weighted random (armchair) model. This statistic will overestimate performance to the degree that individual words other than the maximal one are also applied frequently to a given object. Since our own data suggest that this is commonly true, this upper bound is likely to be quite generous. It has an important pragmatic advantage, however, in that it is independent of sample size and easy to obtain. Other estimates of optimal performance require collecting detailed data on the precise pattern of naming. This estimate requires only that one observe the probability that two people use the same name (i.e., the repeat rate), without even having to note what particular terms they use. Taking the square root then yields an upper bound estimate for the best possible single name per object. For our data, these quantities are listed in Results Table 4.

Results Table 4

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .27 | .32 | .33 | .35 | .42 |

The final way to estimate the performance is to let the data predict itself. The observed largest proportion falling in a cell is taken as the estimate of the population maximum. A familiar result in rank order

* This relation follows from two inequalities. First note that the repeat rate is the sum of the squared cell probabilities. This sum is clearly greater than or equal to any of its terms, since all are positive. In particular, it is larger than the square of the maximum probability cell. Thus, the expectation of the repeat rate is larger than the expectation of the squared maximum cell. Second, note that the expectation of any squared variable is always greater than or equal to the square of that variable's expectation. Putting these together, the expectation of the repeat rate is greater than or equal to the square of the expected magnitude of the maximum cell. Thus, an upper bound on the maximum cell is estimated by the square root of the repeat rate.

statistics is that in the presence of error the observed maximum of a number of observations is expected to be larger than it should be. Thus, using the maximum to estimate itself is likely to be an overestimate for any limited sample size. This is particularly true for small samples. In the extreme case note that if a column of cells has only one observation, the observed maximum will be in the one cell where the single observation fell, which will have an estimated probability of 1.0, regardless of the true underlying probabilities. This will become more relevant later when we calculate similar statistics on rows of the matrix, where many of them will involve small numbers of total observations in each row. For now, however, the samples are not too small, and the results are presented in Results Table 5.

Results Table 5

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .16 | .22 | .28 | .34 | .36 |

#### *4.1.5 Discussion*

It appears that performance would be about twice as good when using an optimum naming strategy than when using the weighted random model (which we believe to be an approximation to typical current practice) in a one-word-per-object system. Still, the overall levels are not very impressive. It should be noted that these optimal strategies represent the best *any* single-name scheme could do. No expert, human or otherwise, could choose single names that would work better.

Is this the best that we can hope to do for people? The answer is no. There are other, more effective approaches. One is to use multiple names, sometimes called "aliases", for each object, which leads us to our second general structure for a system model.

### *4.2 Model 2: "Several names per object."*

#### *4.2.1 Interpretation*

Each object in the system is assigned $M$ terms or names. The user enters one term. Success depends on the user's word coinciding with any one of the system's $M$ words.

#### *4.2.2 Motivations*

Giving things single names is not adequate for the uninitiated, so a reasonable next step is to try giving the system several names by which to recognize each object.

#### *4.2.3 Structure*

The constraint is that exactly $M$ words are stored for each object.

#### 4.2.4 Analyses

***4.2.4.1 Version: random.*** For each object $M$ names are chosen randomly from the total vocabulary, i.e., $M$ cells are chosen from each column. The recall probability is $M/R$, and in our sample tables would range from .01 to .04 for three names per object ($M = 3$).

***4.2.4.2 Version: weighted random.*** The first name is chosen with a probability proportional to its frequency in the given object's column of the table. Each successive name is then similarly chosen, without replacement, from the remaining cells. This is as though someone "sitting in an armchair" thought up several distinct names, any one of which the user could use with success. We make an independence assumption that the probability of a word being chosen is independent, except for renormalization, of the words already chosen. This is probably a faulty approximation for a single human generating a series of words, where the first word thought of may influence the next. If the words were separately proposed by different designers, this independence assumption might be more nearly correct.

Note that by reversing the roles of the system and the user we obtain a very interesting dual interpretation of this model. That is, let the system store a single word and the user make $M$ different guesses. Success would be counted if any of the user's $M$ words matches the system word. Either of these interpretations merely involves the probability that a single sample from the column will match one of $M$ samples drawn without replacement from the cells of the same column (that is, without replacing the whole cell, not just the observation from the cell).

*Evaluation: by "M-order repeat rate" statistic, within columns.* These success probabilities can be estimated using an extension of the repeat rate statistic to this case of multiple samplings without replacement of types. The probability for column $j$ can be shown to be estimated by:

$$\hat{r}_j^{[m]} = \sum_{i_1} \frac{n_{i_1 j}}{N} \frac{n_{i_1 j} - 1}{N - 1} \left[ 1 + \sum_{i_2 \neq i_1} \frac{n_{i_2 j}}{N - n_{i_2 j} - 1} \left[ 1 + \sum_{i_3 \neq i_1, i_2} \frac{n_{i_3 j}}{N - n_{i_2 j} - n_{i_3 j} - 1} \left[ 1 + \cdots \right. \right. \right.$$

$$\left. \left. \left. \cdots \sum_{i_m \neq i_1, i_2, \ldots, i_{m-1}} \frac{n_{i_m j}}{N - \left( \sum_{k=2}^{m-1} n_{i_k j} \right) - 1} \left[ 1 \right] \cdots \right] \right] \right].$$

This formula requires a calculation time that grows exponentially in $M$. We therefore limit results to $M \leq 3$. See Results Table 6.

Results Table 6

| recall probability | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| ($M = 1$) | .07 | .11 | .12 | .14 | .18 |
| ($M = 2$) | .14 | .21 | .21 | .25 | .33 |
| ($M = 3$) | .21 | .30 | .28 | .34 | .45 |

Note that allowing the system (or, equivalently, the user) several words for each object in trying to match its partner achieves considerable gain. Performance almost doubles and triples as we go to two and and three guesses.

It should be remembered that the estimated probabilities of success for 1, 2, and 3 guesses were calculated on data from subjects' first responses only. Under the interpretation of this model in which subjects make repeated guesses at a system's single word, this is equivalent to assuming that successive guesses, given that previous guesses had failed, would resemble other first-provided words. To get a true estimate of the expected performance of a system that actually used this technique, we would need data on people actually making successive guesses. One would have to know how many such guesses can actually be made and with what quality before one could know what the maximum performance would be. However, if one supposes that the system, or perhaps the users' desires and abilities, limited input to three guesses, performance of such a system would not be likely to exceed about one chance in three of correct return.

***4.2.4.3 Version: optimized.*** Choose as the $M$ names for each object those terms that are maximally used by subjects for the object; that is, pick the highest $M$ cells in each column and use the corresponding terms. This is a simple generalization of the single-name case.

*Evaluation: by split halves and self-estimation.* We again run into the problems involved in estimating maximum, and now also the nearly maximum, cells. The split-half and self-estimation procedures were used here to estimate upper and lower bounds. The split-half results represent the average performance of ten independent splits of each data set. The results are presented in Results Table 7.

Results Table 7

| recall probability | Ed5 | Ed25 | CmOb | Swap | Recp | |
|---|---|---|---|---|---|---|
| ($M = 1$) | .15 | .19 | .26 | .26 | .31 | (split half) |
| | .16 | .22 | .28 | .34 | .36 | (self-estimation) |
| ($M = 2$) | .26 | .32 | .36 | .36 | .49 | (split half) |
| | .28 | .37 | .41 | .50 | .56 | (self-estimation) |
| ($M = 3$) | .37 | .42 | .42 | .45 | .58 | (split half) |
| | .38 | .49 | .48 | .59 | .67 | (self-estimation) |

#### 4.2.5 Discussion

Considerable benefit is obtained both from using data to optimize choices and from giving the system several names for each object (or, equivalently, allowing the user several guesses). Though it was not undertaken here, it would be interesting to explore the possibility of letting both the user and the system use several names to try to match each other.

The improvement gained by storing multiple words has associated with it a potential cost in ambiguity. Nowhere in either this model or the previous, one-name model has there been any concern that the names be distinct. The names for each object were picked from the corresponding column, with no regard for what names were being chosen for other objects. This means that two objects could be assigned the same name, with consequent ambiguity should the user give that name. The system would be unable to tell which object the user intended, and would have to present the user with a menu of choices to differentiate among them. (Recall that at the outset we stated that we would be assuming such a system, and, moreover, that choices among the items on the subsequent "menu-on-the-fly" would be assumed error free.)

Naming choices will collide when two different objects have the same words applied to them. One might expect this to happen, for example, if two objects are very similar, so that the same words apply to them. We have collected some data that confirm the existence of this similarity effect. For both the recipe and the common object data we compared the probability of a naming collision for random and similar objects in the set. Subjects were given 5 "focal" objects and 25 "match" objects, all drawn at random from the set. For each of the focal objects, they were told to select the one match object that was most similar to it. Using the weighted random naming method, the naming collision rates were compared for these five pairs of similar objects and the same five focal objects paired with random members of the match set. That is, we calculated the probability that a word applied by one user to the first object would be the same as the word applied by another user to a second object. Using subject differences as a random error estimate, there was a highly significant increase in naming collisions for the pairs of objects judged to be similar [$t(14) = 3.86$ and $t(24) = 5.37$ for the recipe and common object data, respectively].

To examine the effects of trying to avoid collisions, we studied the next model.

### 4.3 Model 3: "A distinct name for each object."

#### 4.3.1 Interpretation

Each object in the system is assigned a single term, with the proviso

that no term may be used more than once. Success depends on the user's spontaneously produced word coinciding with the system's distinct name for the intended object.

#### 4.3.2 Motivations

Note that in typical naming circumstances the intent is often to establish conventions about terminology so as to avoid the ambiguity that would otherwise arise. Naming conventions are used not only to set by fiat the name by which an object will be known to a system, but also to proclaim that *only* that object will be so known. Often interaction will reach a stage where complete precision is needed, e.g., for actual command execution.

It may be the designer's motivation in selecting names that users learn terminology that allows them to be precise. As had already been pointed out, however, the designer's intent may not correspond with the user's reality; the untutored may try to deal with the system anyway. Thus, we explore the user-guess success for systems designed with the unique name constraint.

The models discussed here are just like those of the "one-name" case, except that words cannot be used more than once.

#### 4.3.3 Structure

One distinct word is stored for each object, i.e., this is the same as Model 1, except that the words must all be distinct, so that at most one object is referenced by each word. This imposes strong constraints on the system table, on both row and column totals, and the number of rows and columns used overall. This high degree of constraint makes mathematical treatment difficult, as will be discussed shortly.

#### 4.3.4 Analyses

***4.3.4.1 Version: random.*** For each object, one name is chosen randomly from the total vocabulary. Once a vocabulary item is used, however, it is eliminated from any future consideration.

Recall probability would be $1/R$ and range from .002 to .014 for these data. The "number returned" is easy to give for this case, as it is set by the structural constraint, that each name have a unique referent. It is therefore 1 for all models having this structure. That is, when the system finds a target it returns exactly one candidate. However, there are many occasions when the user word matches no system word, and so no target is returned.

***4.3.4.2 Version: weighted random.*** Here the name is chosen with a probability proportional to its frequency in the given object's column of the table. Then the chosen word is eliminated, and a name is chosen in an analogous way for another object from the remaining words, etc.

The results are clearly influenced by the sequence in which objects are dealt with. The approach used here was to choose in the following way: a word/object pair is chosen by a weighted random sampling from the whole table. This gives the right distribution within each column and allows appropriate representation of each column. Once such a cell is chosen, the corresponding object has been named and the word used, so both are eliminated, and the procedure is iterated on the table, now reduced by one row and column, until all objects have been named.

*Evaluation: by Monte Carlo simulations with split halves.* We could devise no way to evaluate this model analytically, so we used a Monte Carlo simulation on split halves. We divided the data in the user table in half and randomly picked names, according to the model outlined above, using the data from one half. The second half of the data was then used to evaluate the effectiveness of the names thus chosen. The results presented here are the average for ten split halves with ten independent Monte Carlo simulations of name selection in each. See Results Table 8.

Results Table 8

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .07 | .08 | .11 | .12 | .09 |
| number returned = 1.0 (by design) | | | | | |

Note that, as might be expected, success is less than for the comparable model in which names did not have to be distinct. In some instances this decrease is small, as with the edit studies; in others it is large, as with the recipes. This should depend on whether there were a few high-frequency words that were used for many different objects.

***4.3.4.3 Analysis of precision versus popularity of terms.*** In all of our data sets there was a slight trend for high-frequency words to be less discriminating than low-frequency words. In studying this relation quantitatively, the measure of discriminating power of a word was given by the repeat rate statistic, this time applied within each row. This row-repeat rate is a measure of the probability that two uses of a word refer to the same object. When this probability is high, the word is very discriminating. Below we present the correlation (Spearman $r$) of row-repeat rates with the marginal frequency of the words. See Correlations Table 9.

Correlations Table 9

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| corr: | −.28 | −.30 | −.21 | −.24 | −.16 |
| ($N$ words): | (74) | (84) | (301) | (145) | (167) |

This correlation points to the cause of the difficulties run into by the constraint of distinct names. There is something of a conflict. If one chooses high-frequency names, they will be likely to collide. If one chooses less frequent names, the chances of collision will be somewhat less, but fewer users' queries will be handled. Unfortunately, there is no correspondence between the size of these correlations and the size of the decrease in performance for each data set. The reasons for this are not clear.

***4.3.4.4 Version: optimized (though not best possible).*** The idea is to choose the distinct names such that the total probability in the cells chosen is maximal. The method used here is a greedy algorithm that is not truly optimal, but it is a reasonable improvement over the weighted random method. It begins by picking the highest cell in the matrix; then, after eliminating the corresponding row and column from the matrix, it iterates, picking the next highest cell, etc., until all objects have been assigned a name. Algorithms like this are called "greedy" because at each step they take the biggest possible chunk of what is left, without regard for what later problems that may cause. In this case, it is possible that an early choice will eliminate a word that would be very good for another object, when there was an alternate word that would have done almost as well at no such cost. Algorithms that would take such future complications into consideration, or equivalently consider so many possibilities at once, are typically combinatorially explosive. Thus we present the results of the straightforward greedy algorithm approach.

*Evaluation: by split halves.* There is no simple method to estimate the performance of such an approach. Again we turned to the split-halves technique of dividing the data in half, applying the algorithm to one half and then testing the chosen names on the other half. The results, averaged over ten independent splits, are given in Results Table 10.

Results Table 10

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .14 | .11 | .23 | .19 | .11 |
| number returned = 1.0 (by design) | | | | | |

### *4.3.5 Discussion*

As we expected again, the optimization attempt, even though imperfect, has a dramatic effect, in many cases doubling the performance of the system. Even in the best case, however, the system succeeds only about one quarter of the time, and typically little more than a tenth of the time. We note that the numbers here are substantially lower than in the case where there was no requirement that names be distinct. In fact this improved method really does no better than an

armchair (weighted random) version of the unconstrained model, and often worse. The lesson here is that adding the requirement of uniqueness, common in establishing conventions, hurts naive users quite a bit. The need for such conventions is not denied, it is simply asserted that auxiliary aids will be needed for systems that make heavy use of such conventions. It is worth noting that, for our data sets, the number of objects that are not disambiguated and would have to be presented for a second stage choice is always small, and the gain in recall always large. Thus, the "menu-on-the-fly" method implicit in several of the models presented here appears to be very promising as an aid to unsophisticated users.

### *4.4 Model 4: "Distinct names, augmented with M extra referents."*

#### *4.4.1 Interpretation*

Each object is given a distinct name. $M - 1$ other referents for those words are also stored.

#### *4.4.2 Motivations*

Part of the inferiority of the distinct-name models, when compared to the unconstrained models, came from the fact that people often want to use the same names to refer to several objects. The motivation for the next model is to recapture access to some of those other interpretations of the term. A simple extension of the distinct-name structure is to begin with the situation where each object is associated with a distinct name, one that can be memorized and used unambiguously by experts, but also to store $M - 1$ other objects that the term applies to, explicitly for use in a "menu-on-the-fly" for the untutored. A reasonable system implementation might be to give the "distinct" name a special status (the "real" meaning of the term), and to admit the other interpretations as secondary, perhaps to be verified in the context of use.

This is the first model in which we explicitly design in more than a single system guess as to the intended object. To be sure, several guesses for the meaning of a term could (and would) have arisen in the uncontrolled name cases of Models 1 and 2, but here we predetermine exactly how many guesses the system makes and evaluate performance as a function of this parameter.

As more objects are returned, the likelihood of including the intended target is increased, but a certain cost is incurred—the cost associated with discerning the true target among all the returned objects. Data searches can fail not only by not giving access to a desired object, but also by returning too many unwanted objects. In information science the problem is familiar as the trade-off between *recall*—the number of desirable items returned—and *precision*—the

proportion of items returned that are desirable. This is also similar to the hit versus false alarm trade-offs of signal detection, familiar to psychologists and communication theorists. In the latter context the trade-off is sometimes examined by tracing out operating characteristics and interpreting them in terms of parameters of an underlying statistical theory. There does not currently exist a corresponding statistical theory for our precision and recall rates, but it is useful to be able to examine, or in this case, control, precision explicitly and see what gains in recall result. This yields an operating characteristic. Any version of the general baseline model in which the system is set up to return an explicit number of guesses provides a direct way to do this.

#### *4.4.3 Structure*

At least one name is given to each object, but each name that is used by the system also refers to $M$ different system objects.

#### *4.4.4 Analyses*

***4.4.4.1 Version: random.*** Here the primary cells are again chosen completely blindly, eliminating rows and columns already used in an iterative manner exactly as in the distinct-name case. After the $C$ primary cells are chosen, the $M - 1$ additional or secondary cells are chosen randomly from each new row chosen.

The resulting probablity is $M/R$ and would range from .001 to .035. The number returned is, of course, $M$ by design for all versions of Model 4.

***4.4.4.2 Version: weighted random.*** Cells are chosen at each step with a probability reflecting their magnitudes. Thus, first $C$ distinct names are picked, as in the distinct-name model (Model 3). Then, in the row of each cell just picked, $M - 1$ more objects are chosen with a probability equal to their relative frequencies in the rows.

Again, the weighted random case is an approximation to what an individual designer might do without collecting data from other people. The process of coming up with additional interpretations requires a bit of further explanation. The asymmetries found in free association data suggest that people starting from terms and generating objects would yield probabilities quite different from those obtained from people starting with objects and generating terms.[13] This asymmetry means, in this model, that the additional interpretations cannot be thought of as the result of the designer sitting in an armchair and thinking up other interpretations for the terms. The efficacy of such an approach is not estimable from our data.

A scenario that might better satisfy the assumptions of this model would have the system memorize the first $M - 1$ nonstandard uses of its known terms that it comes across.

*Evaluation: by Monte Carlo simulation on split halves.* The data are given in Results Table 11.

Results Table 11

| recall probability = | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| ($M = 1$) | .07 | .08 | .11 | .13 | .08 |
| ($M = 2$) | .13 | .15 | .15 | .17 | .15 |
| ($M = 3$) | .18 | .21 | .18 | .20 | .20 |

The number of guesses that the system is requiring to achieve this performance is of course, $M$, the number returned by design.

***4.4.4.3 Version: optimized (though not best possible).*** Again it was not feasible to find the completely optimal choice of cells. Instead, the primary cells were found in the same "greedy" way used for the simple distinct-name "optimal" case. The subsequent choice of secondary referents for the words so selected can be optimized by choosing the $M - 1$ largest cells remaining in the row. Note that it is possible for some of these remaining cells to be larger than the primary cell, as when they refer to an object that was eliminated before the row was chosen. (It is the vagaries of the need for convention in names that brings about this ironic use of the term "primary.")

*Evaluated: by split halves.* In the now familiar procedure, the data are split and the first half used to choose the cells following the algorithm just outlined. Then the second half of the data is used to evaluate the choice of cells. See Results Table 12.

Results Table 12

| recall probability = | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| ($M = 1$) | .14 | .11 | .22 | .21 | .11 |
| ($M = 2$) | .21 | .20 | .28 | .27 | .17 |
| ($M = 3$) | .27 | .29 | .31 | .30 | .22 |
| number returned = $M$ (by design) | | | | | |

These results, for one to three total guesses, are plotted in Fig. 2. The model curves can be viewed as operating characteristics, giving total recall as a function of decreasing precision. The dashed diagonal lines represent expected chance performance if the system returns guesses at random. Note that in the case of the Ed5 data, the random performance is *better* than that of this model. This is possible because this model, like most current computer systems, makes no response at all to any but a few words. By failing on so many of the words it encounters, it can be surpassed by a system that only guesses, but at least guesses for all words. Admittedly, in the context of command execution any level of guessing without confirmation may be danger-

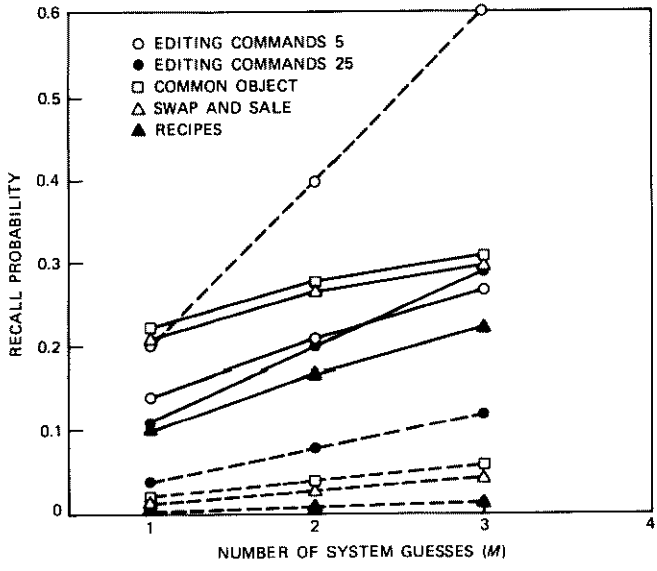



Fig. 2—Recall as a function of the number of system guesses for Model 4 ('distinct names, augmented with $M$ extra referents') is shown by the solid lines. The dashed lines represent expected chance performance if the system returns guesses at random. (Different chance performance for the different data sets simply reflects the different number of objects in each set.)

ously out of place, but there are safer arenas, like help facilities, that could benefit. In any case the cost of ignoring so many user words is made clear. Notice that the total probability does not approach one, even in the case of command verbs for editing operations where there were only five objects. The reason is again that many words were used by our subjects to describe these objects, and the algorithm picked only a maximum of $C$ of them to recognize. The modeled assumption was that if a subject supplied any word not recognized by the system, no choices would be returned and a failure would result.

#### *4.4.5 Discussion*

These models use the same number of cells as the multiple-names models, yet do not do as well. The implication, presumably, is that there is considerable cost in limiting oneself to a small number of words. The users distribute their descriptors too broadly for any such limitation to be very satisfactory.

Thus we are motivated to try a completely different approach. All the models up to this point have inherently focused on what the system brings to the interaction. The modus operandi has been to set all the system's objects before us and then try various ways to guess what users will call them, albeit with some improvements as we give

deference to the empirical characteristics of the user's language as seen in the user table.

Suppose we turn the tables, so to speak, and focus on what the users bring to the interaction: the terms they use when trying to specify things. Start with the terms, all of them, and concentrate on trying to guess to what object they refer. This brings us to our next model.

### *4.5 Model 5: "Recognize one referent for every word."*

#### *4.5.1 Interpretation*

The system stores every word that it can, and for each has one guess as to the intended object.

#### *4.5.2 Motivations*

One of the principal problems with the approaches studied so far has been that a large proportion of the recall failures can be attributed simply to not recognizing many of the user's words. The systems modeled have paid too little attention to the wide variety of the terms people use spontaneously. Here we focus on this essential character of what the users bring to the interaction. For every word produced during the collection of the data for the user table, a guess is made. The different versions vary as to how these guesses are made.

It is useful to note that these models, and their generalizations in the next section, are really duals of the single- and multiple-names models. In those models the system had words for each object, and it succeeded when its name was the same as the word the user would use. Thus, to evaluate those models we calculated the conditional probability that, given a target object, the system and user names would coincide. Then we summed probabilities across objects, weighting each by its column's marginal probability (all essentially equal in our data, by design). Here we conditionalize the other way. Given the word used, we find the probabilities that the system and user associate the same object with the word. Then we sum across words (rows), weighting by the very uneven observed marginal frequencies for each word, to get the overall performance.

#### *4.5.3 Structure*

The system makes use of every word in the table, associating each with one, and only one, system object.

#### *4.5.4 Analyses*

***4.5.4.1 Version: random.*** Here the system will associate random objects with the input words. That is, the system will simply make a

random guess for anything the user says. The recall probability is just $1/C$, and ranges from .005 for the recipes to .200 for the Ed5 data. The number of guesses being made to achieve these performances is predetermined by design to be 1 for all these models.

***4.5.4.2 Version: weighted random.*** Here an object is associated with each word by a random method that takes each object with probability proportional to its relative frequency in the row. Arguments similar to those made for the augmented distinct-name model here imply that this is not the analog of having the designer sit down with the list of words and hazard guesses as to what they mean; the data used here go in the other direction. The more appropriate scenario is as if a system records the first encounter with a new word, and its intended referent, and makes a single pointer based on this single observation, forever freezing the meaning of the word. This is perhaps an unreasonable scenario, but the model is worth investigating because the statistics that result have other valuable interpretations.

*Evaluation: by row-repeat rate statistic.* The probability of success in this case is the probability that two users will mean the same thing by a given word. The results below are the average row-repeat rates, weighted by total row frequency. (Rows with a frequency of one were excluded, since the repeat rate is not calculable in that case.) This number is an unbiased estimate of the overall probability that any two occurrences of any word will be in reference to the same object. See Results Table 13.

Results Table 13

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | .41 | .15 | .52 | .62 | .13 |
| number returned = 1.0 (by design) | | | | | |

We note how variable the probabilities are. They are low where the same words mean different things, and high where any one word refers to only one object. Apparently these domains differ in this aspect. There are at least two possible explanations. First is the similarity effect discussed in connection with Model 3. There it was shown that pairs of objects judged to be similar had higher overlap in the patterns of names applied to them than did random pairs. The resulting naming collisions meant that it was more difficult to find distinct names for similar pairs. Here we are explicitly interested in a different, though related, effect. We can use the previous experimental data (see Section 4.3.4.3) again, this time to demonstrate the similarity effect on a row-repeat rate measure. In this version we consider pairs of cells in each row: one cell from the target column, and one from either a column

rated similar or from a random control column. The repeat rate is calculated on the two cells and summed (weighted by row sum) down all the rows. Since a low repeat rate for a given word indicates that it does not discriminate well between the objects, we predict a lower repeat rate for the similar pairs of objects, and that is what we obtain. In the recipe data the mean repeat rate of the similar pairs was .73 and for the random pairs was .91 [$t(14) = -5.83$]. For the common object data, the similar pair repeat rate was .89 and the random repeat rate was .95 [$t(24) = -5.71$]. Thus similarity has a strong effect on repeat rate. Unfortunately, what we need to make sense of the varied results is some measure of the relative internal similarities of the various domains. Such data are neither available nor easily obtained. Still, it might be agreed that the set of common objects is more diverse than is a set of cooking recipes, or text editor operations, corresponding to observed differences in average repeat rates.

There is another factor that might be helping the swap-and-sale descriptors. Analysis of the other data sets was either strictly limited to single words, or else only to short phrases with few content words. Swap-and-sale descriptors contained an average of 1.7 content words, where there were fewer than 1.2 content words in the others. If these words tend to be used in a conjunctive sense and if they are not redundant, they should contribute to the high selectivity of the descriptors seen for the swap-and-sale data.

A final note should be made of the higher repeat rate for Ed5 compared to Ed25. Part of this is due simply to the fact that even with a purely random, undiscriminating distribution of name usage across objects, there would be an increase in repeat rate as the number of objects is decreased. If there are only two objects, people will have to mean the same object by any given term at least half the time. A more intriguing possibility is that the classes are more distinct entities than are the individual objects, and that this makes word usage more discriminating, above and beyond the chance effect just mentioned.

***4.5.4.3 Version: optimized.*** In this case the user matrix is used to find the best possible choice for a word's referent. This is done by picking the maximum cell from each row, the object to which the word has most commonly referred.

*Evaluation: by split halves, square root of the row-repeat rate, and self-prediction.* Evaluation is again problematic. While there is no question that picking the maximum cell is the best possible strategy, there is no unbiased estimate of its true magnitude. Thus we resort to the same three methods used when similarly confronted in the optimal version of the one-name model (Model 1): the split-half technique, an unbiased estimate of how well one could expect to do with half the data; and two estimates of upper bounds. See Results Table 14.

Results Table 14

| | Ed5 | Ed25 | CmOb | Swap | Recp | |
|---|---|---|---|---|---|---|
| recall probability = | | | | | | |
| | .49 | .18 | .43 | .35 | .13 | (split half) |
| | .62 | .35 | .65 | .72 | .28 | (square root of row-repeat rate) |
| | .54 | .26 | .69 | .81 | .25 | (self-prediction) |
| number returned =1.0 (by design) | | | | | | |

#### *4.5.5 Discussion*

There is considerable variability, both in the performance and in the range of these estimates. The largest range is for the swap-and-sale estimates. This is due to the very large number of descriptors that occurred only a few times. The data for each such descriptor are statistically unreliable, so the maximum cell cannot be accurately identified; this problem is severely aggravated by splitting the data and only using half to make the identifications. Note that this is not an experimental artifact. A very long tail on a descriptor distribution is a legitimate real-world problem, because it means that new terms will keep arising that the system will not have seen before, and about which it will not be able to make any educated guesses. A wide range in the estimate reflects the fact that very large amounts of data would be needed to approach asymptotic performance.

As for the overall diversity of scores from domain to domain, the arguments about similarity and number of words in the descriptors, given above for the repeat-rate case, are equally applicable here.

The most important point to be made about this optimized model is that it represents the best one could possibly do. We can do no better than to recognize every word users use and make an optimal guess as to its meaning. Clearly, if we have insufficient data to do this well, either preventing us from recognizing a word, or from being able to estimate the modal referent, performance will suffer. But the upper bound estimates represent the real, strict limits of performance. No other pattern of structural constraints could do better, except at the cost of precision. This trade-off with precision is explored in our final model, in which an explicitly prespecified number of multiple guesses is returned to increase the chance of including the user's intended object among them.

### *4.6 Model 6: "M referents for every word."*

#### *4.6.1 Interpretation*

The system stores every word that it can, together with a set of $M$ possible referents. Whenever the user enters a word, the system returns the $M$ guesses, possibly ranked in some way.

#### 4.6.2 Motivations

The "one referent for every word" model (Model 5), in its optimized version, gave the best possible performance for a system that hazards only one guess for a user's word. The only way to increase the chance of returning the user's intended object is to return more than a single guess. These guesses could be returned in the form of a menu of $M$ items, among which users would choose.

As we mentioned, this is the dual of the multiple-name set of models, and several of the evaluation procedures differ only in that rows have traded roles with columns.

#### 4.6.3 Structure

Like the previous model, the system makes use of every word in the table, but here it associates $M$ system objects with each.

#### 4.6.4 Analyses

***4.6.4.1 Version: random.*** The system makes just $M$ pure guesses, without replacement, from the set of system objects. The recall probability for these cases is thus exactly $M/C$, and the number returned is $M$, by design, for all versions of Model 6.

***4.6.4.2 Version: weighted random.*** In a manner following the single referent weighted random model, the $M$ cells are chosen with a probability that is proportional to their relative sizes. The $M$ choices are required to be distinct, and so a cell is excluded from further consideration once it has been selected.

The scenario that this might correspond to would be one in which the system learns the first $M$ distinct referents of the word that it comes across in use. Its encounters would be governed by exactly these probabilities.

*Evaluation: by M-order repeat rate statistic, within rows.* We want the probability that the user's single intended referent coincides with any of the system's candidates, when both sets are drawn from the same probability matrix. These success probabilities are estimated using the same extension of the repeat rate statistic used in evaluating success of the many-names-for-one-object model (Model 2, Section 4.2.4.2). Here though, it is applied to the cells within a row, rather than within a column.

It should be noted that the formula requires there to be $M$ nonempty cells in the row, to prevent some of the denominators from going to zero. For example, it is not possible to give an unbiased estimate of how well three guesses would do if the data only show two objects. Note that while rows with highly discriminating words should in general have high $M$-order repeat rates, it is exactly such rows that will be less likely to satisfy this requirement, especially at lower

marginal frequencies. Thus, ignoring rows where this statistic cannot be calculated biases the average value of this statistic downward. Though it is not clear how to get an upper estimate, there is another way to get a lower bound, by recognizing that performance for $M$ guesses is at least as good as performance for $M$-1 guesses. Thus, a lower bound on the average performance comes from using for each row the calculable repeat rate of highest order not greater than $M$. See Results Table 15.

Results Table 15

| | Ed5 | Ed25 | CmOb | Swap | Recp |
|---|---|---|---|---|---|
| recall probability = | | | | | |
| ($M = 1$) | .41 | .15 | .52 | .62 | .13 |
| ($M = 2$) | .66 | .26 | .65 | .74 | .21 |
| ($M = 3$) | .81 | .36 | .71 | .80 | .27 |

***4.6.4.3 Version: optimized.*** This version takes the optimal strategy for multiple guesses. The user gives one word, and the system makes guesses that the user means one of the $M$ most likely things the word has meant in the past, as deduced from the data in the user table. That is, it returns the objects associated with the $M$ highest cells in the row corresponding to the word given. (A sequential version of the model would give the user the guesses in decreasing observed frequency, beginning with the maximum cell.)

*Evaluation: by self-estimation and split halves.* As in other cases where evaluation of performance depends on the estimation of true population ordered frequencies, we must resort to indirect methods to give a range. We use split halves to give a lower bound and self-estimate to give an upper bound. See Results Table 16.

Results Table 16

| | Ed5 | Ed25 | CmOb | Swap | Recp | |
|---|---|---|---|---|---|---|
| recall probability = | | | | | | |
| ($M = 1$) | .49 | .18 | .43 | .35 | .13 | (split half) |
| | .54 | .26 | .69 | .81 | .25 | (self-estimation) |
| ($M = 2$) | .70 | .30 | .53 | .43 | .20 | (split half) |
| | .76 | .42 | .82 | .92 | .37 | (self-estimation) |
| ($M = 3$) | .83 | .40 | .58 | .47 | .26 | (split half) |
| | .88 | .53 | .88 | .96 | .46 | (self-estimation) |
| number returned = $M$ (by design) | | | | | | |

The recall rates, as estimated by the conservative split-half procedure, are given in Fig. 3. The curves can be interpreted as operating characteristics; the distance between the curves and the corresponding dashed diagonal line reflects how much the implied system could be expected to outperform a simple menu system based on the pure random model given above, i.e., on a random choice of $M$ objects. All

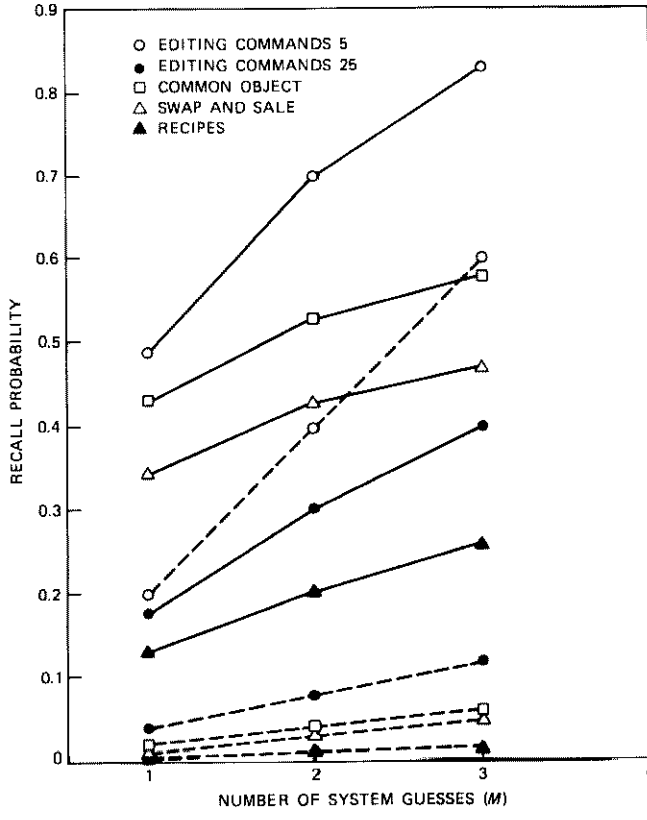



Fig. 3—Recall as a function of the number of system guesses for Model 6 ('*M* referents for every word') are shown by the solid lines (split-half estimation). The dashed lines represent expected chance performance if the system returns guesses at random. (Different chance performance for the different data sets simply reflects the different number of objects in each set.)

of our models do substantially better than chance. More importantly, they do much better than the augmented distinct-name model (Model 4) presented in Fig. 2.

#### *4.6.5 Discussion*

There are really two important points to be made about these results. The first is that these performances are variable and but moderately high. Note that the very high value of the three-system-guess case in Ed5 is rather vacuous, since there were only five objects to guess from. In the common object case, however, the high values are quite meaningful, since there were 50 objects.

The legitimate high values lead us to the second observation, that substantial improvement has been gained over the traditional, one-

distinct-name approach. The presumption is that this gain is due in part to the increase in capturing all the user's inputs. There are, after all, two ways for the system to fail in retrieving the user's target. One is by making wrong guesses, the other is by being unable to make any guesses at all. To give an idea of the size of this problem: for three of our data sets, no system using only $C$ words (e.g., one for each object) could recognize even half the user's words. Thus, recall rates can only increase as the system is taught more words, and often this can be a sizable improvement. The only conceivable decrements (except those associated with time and space in index management) would be if the additional words were less precise, thus decreasing precision for a fixed recall rate, or decreasing recall for a fixed precision. But our finding of a consistent, albeit small, negative correlation between a word's frequency and its selectivity suggests that the opposite is true. As a greater number of lower-frequency words are included, precision will go up.

Recall also increases dramatically if the choice of algorithms is optimized, though it never exceeds the square root of the performance obtained by "armchairing", i.e., making pointers based on a single observation. The cost is in the trouble it may take to collect data. In some cases the variety of terms is so great that data collection must continue for quite a time before asymptotic performance is approached. Performance also increases as the system is allowed to make more guesses; this involves a direct trade of recall probability for reduced precision. In many circumstances, particularly in help facilities, this presents little problem. The user need only be given the various options and whatever additional information it takes to decide what is really sought.

### *4.7 Other possible models and some limits*

The models presented so far have covered a large share of the space of simply constrained models where the user makes one guess, but they do not exhaust the ways in which a system could use empirical data on user descriptions to guess the user's desired objects.

One might consider, for instance, models that take advantage of redundancy in the tables. A good theory or description of what such tables are like could be used to augment the data to make better estimates of true population usage probabilities. For one example, if we knew that some single shape of distribution characterized all columns, we could use the data to estimate parameters of that distribution instead of individual cell probabilities. This would reduce the number of parameters that need to be estimated from the data, and consequently improve stability and reliability of predictions.

A related, and perhaps more interesting, idea would be to look for

latent structure in the similarity between objects, and between words, and use this to improve predictions. For example, if we knew or could infer from the data that two words were used almost identically, we might pool the cell frequencies for the two words; or if we knew that the objects and words related to each other by some more elaborate structure (e.g., a hierarchical tree), we could base predictions on calculated indirect reference paths.

While these ideas might be useful, the research presented in this paper puts strict limits on the success of any such approaches. As noted above in Models 5 and 6, the self-prediction evaluation procedure (determining the best guesses from the data and then using the same data to estimate success) and the "square root of repeat rate" evaluation procedure yield upper bounds on expected performance. Indeed, no analysis method of the kind we have just been discussing, no matter how clever, could improve on this result. The reason is as follows. Even using inherent structure to improve predictions could do no more than increase the accuracy of estimation of the population values for the input/output tables. Even with true population probabilities, however, the guessing decision rule would be the same; choose first the most probable referent of each word, and so forth. The input/output tables are inherently probabilistic, as a result of disagreement in word usage, not just of the estimation uncertainties. For a given user population, with a given degree of training, a certain amount of disagreement in word usage will occur. Even with perfect knowledge of the probabilities describing this usage, we could not predict individual referents perfectly. Now, the objective of having true population probabilities as cell entries is mimicked by our upper bound estimating procedure. It pretends that the values observed for each cell are exactly the probabilities that would exist in further sampling from the population. Thus, no method of "cooking" the data to reduce sampling error could achieve a better result than is illustrated by our upper bound values.

To improve performance beyond these limits, it would be necessary to construct systems that use different input, e.g., multiple words or interactive dialogues, or make the user learn to use more easily interpretable language. The last of these is the current default approach—make the user learn everything—and it has its limits for large systems or occasional users. Thus, we believe a more fruitful direction to explore further is multiple-word inputs.

The simplest models of multiple-word query might assume only content words (no syntax) and a statistical independence between words. Independence would allow the performance of such a system to be estimated from single-word data of the sort we have collected. Multiple words could be used disjunctively. In this case the system

would return any objects that matched any of the input words. This would result in an increase in probability of recall. Alternately, the words could be used conjunctively, so that the system would return only objects matching all the key words. The result would be higher precision. Mixtures of these two approaches (e.g., conjuncts of disjuncts) then clearly could be used to improve both precision and recall.

Of course, from our data we do not know how people can or do use multiple words, even without considering syntax. The popular balances of conjunction and disjunction are unknown, and spontaneous multiple guesses are no doubt not independent. But what sort of nonindependence is common? For example, do multiple terms tend to be more unrelated, or more redundant, than independence predicts? These questions must be explored before multiple-word systems can be theoretically evaluated, or optimized for the naive user.

The consideration of syntax leads to a whole new set of problems, those often encountered in Artificial Intelligence work on natural language understanding. We will not address them here.

### *4.8 Highlights*

In the previous sections we used the tables of observed word usage to explore various models of how systems could name, and thereby give access to, their contents. We devised six general schemes for assigning names to objects. To test the schemes, we approximated user behavior by sampling appropriately from the tables.

Thus, for example, we simulated what word a designer might assign to an object (pick from the table), what word a user might choose (pick again from the table), and looked at how frequently the two words were the same. The results were taken to indicate how well a set of spontaneously generated single names for objects can be expected to work for untrained users. This particular example was called the "armchair model." The table is, after all, only a compendium of many people's spontaneous armchair attempts to give the best possible names for these objects. Thus, sampling from the table mimics designers picking names from their armchairs.

We used this general technique to explore variants on our basic model of interaction, in which we assume the designer builds in certain connections between names and the system's objects or services. These names were in some cases purely random, and in other cases were a simulation of a designer's best armchair guess; in still other cases the names were chosen more systematically, with a goal of optimality. In all cases, we assume that the user must "guess" the right name because we are concerned with untutored users who do not know the system words. The user's choice of words was always mimicked by sampling from the tables.

Of the models and variations, summarized in Appendices A and C, perhaps the most important are:

(1b) "One name per object: weighted random"
(1c) "One name per object: optimized"
(6c) "M referents for every word: optimized."

The first of these (1b) was the so-called "armchair" naming method just described. Expected results from the armchair method are shown in Fig. 4 model (1b). Note that despite the fact that the four domains differ dramatically in content, data collection style, and subject population, the results are quite consistent. They are all low. Two people—the designer and the user—will come up with the same name only about 15 percent of the time.

These results indicate that the current practice is deficient; it guarantees that it will be difficult for people to tell machines what

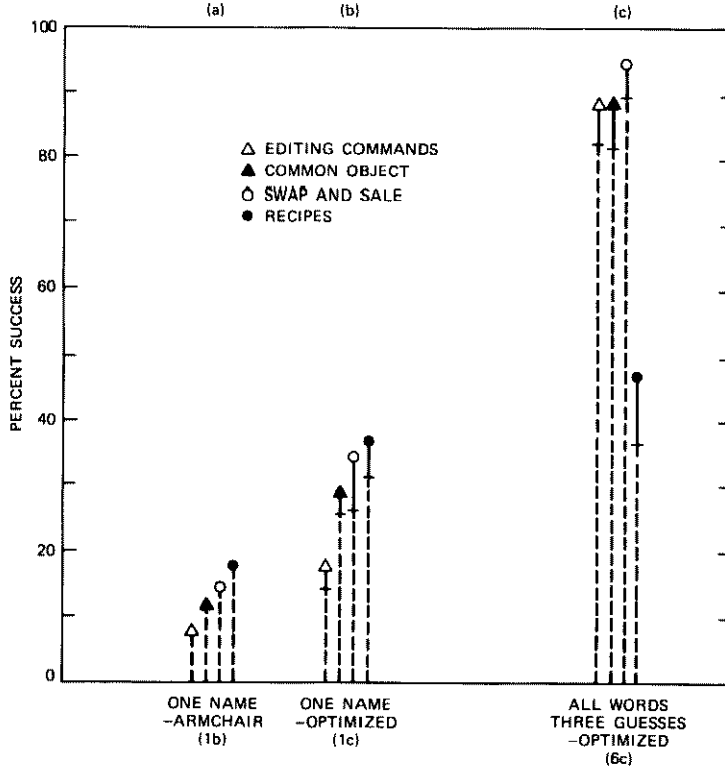



Fig. 4—Summary of expected success for three of the most important models presented. (a) "Armchair" model (1b); (b) Best possible single-name model (1c); (c) Model where the system recognizes all words and returns three guesses (6c).

they want, unless they already know what to say. Note that designers often have other, legitimate motives in assigning names—e.g., lack of ambiguity, memorability, or cuteness. But such naming practices only make the current problem worse; they lead to even less likely names.

The source of the problem is this. There are many names possible for any object, many ways to say the same thing about it, and many different things to say. Any one person thinks of only one or a few of the possibilities. Thus, designers are likely to think of names that few other people think of, not because they are perverse or stupid, but because everyone thinks of names that few others think of. Moreover, since any one person tends to think of only one or a few alternatives, it is not surprising that people greatly overrate the obviousness and adequacy of their own choices.

To improve performance of the "armchair" method we investigated whether experts in one of the content areas could pick better words. We had expert chefs choose key words for cooking recipes and found essentially no improvement: experts do not seem to do noticeably better picking terms from their armchairs than does anyone else (Section 4.1.4.2).

The next model of major interest (1c) substituted objective data for armchair suggestions. It used our data tables to identify the name that was most commonly used for each object. Thus, for example, for the operation we referred to as DELETE, the most commonly used term was "omit", with a frequency of 110, and that was the name used in this model. This empirical approach, occasionally advocated by human factors people, achieves the levels of expected success illustrated in Fig. 4 model (1c).

As discussed in Section 4.1.4.3, for statistical reasons we can only estimate certain upper and lower bounds on performance here. The lower one indicates how well one could do with a limited amount of empirical data from which to try to pick the best names. The upper one liberally estimates how well one could do with an infinite amount of data. The improvement over the armchair method is substantial, typically almost doubling the hit rate. This makes the point that the best name is a good bit better than the typical name. But it is also clear that even the best one is not very good.

It is critical tc note that these numbers represent the best one could possibly do by assigning a single name to each object. The problem is not solved by finding the right name. Different people, contexts, and motives give rise to so varied a list of names that no single name, no matter how well chosen, can do very well. Figure 5 is a plot of how many names are needed to account for a given percentage of the users' attempts, here for the common object data. (Here again we have the statistical estimation problem, so the boundary is double.) Note that

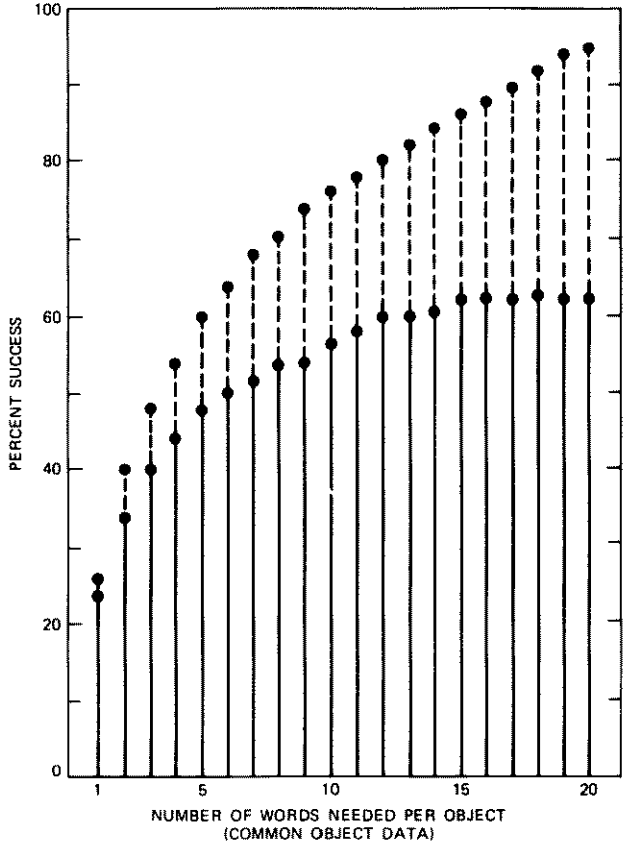



Fig. 5—A plot of how many different names (or "aliases") are needed for each object to account for a given percentage of success. This plot is based on data from the common object domain. The dashed lines indicate ranges.

even 15 words (word types or "aliases") per object account for only 60 to 80 percent of the words people apply. That is, even with 15 names stored per object, the computer will miss 20 to 40 percent of the time. The lesson clearly is that systems must recognize many, many names.

After considering several other models, we turned the problem around a bit. Instead of starting with system objects and looking for names, the new strategy was to begin with user's words and look for interpretations—that is, try to recognize every possible word that the users generate, and use empirical data to determine what they mean by that word.

We inject a cautionary note here: when word meanings are to be surmised from word usage data, there is always a risk that the same word may have been used in reference to several objects in the system.

In such a case, the system would not have a unique guess. (This was actually an implicit problem discussed for several of the earlier models.) In our simulations, we limited the system to returning just its single best (Model 5, Section 4.5) or best three (Model 6, Section 4.6) guesses as to what was wanted, given the word the user said. For discussion here we will mention only this last model (6c). The results in Fig. 4 show what can be achieved if all words are saved by the system, and it uses behavioral data to make the three best guesses as to what was meant. (Here too we can only present estimates of upper and lower bounds on performance.)

These results are encouraging. The big improvement is due primarily to the fact that the earlier methods, like most in current use, failed to recognize all the rare words. The unfortunate truth is that the majority of words used are "rare" words, and so it is a mistake not to reckon with them. Indeed, it turns out (see Section 4.3.4.3) that less common words tend to be more precise, more discriminating, and so are particularly valuable for the computer to know.

While we studied quite a wide variety of intermediate models, this last one really is the best. At the risk of belaboring the point, here it is summarized again. It requires: (1) the computer to keep all words that people use; (2) that data be collected on what users use each word to refer to; (3) that when someone uses a word, the computer goes to the data table and looks up the top few candidates to determine what its interpretation should be (what the word is most probably referring to); and (4) the computer presents the candidates to the user as a choice, since the interpretations may be in error. This model amounts, really, to a rich, empirically (i.e., behaviorally) defined cross index. Without it, performance is near abysmal, with it, potentially quite good.

Further improvements are shown to require the input of multiple words. This would entail means for representing and evaluating relations as well as reference, and matters of logic, syntax expression, and language comprehension. Detailed analyses of the complicated usability issues that would be involved are beyond the scope of this paper, but a number of related issues were addressed in Section 4.7.

## V. DISCUSSION

We have seen that the object referent of a word cannot be predicted with great accuracy from knowledge of its past referential use. The use of statistical data on reference behavior can improve the choice of a single name by roughly a factor of two over the common procedure, in which only a single designer-chosen word is stored as an entry point for each object. Even the best system input/output function of the general kind we have described, while adding another factor of two or

so to performance, can be expected to perform at well below perfect reliability. In this section, we touch on some of the reasons for this limitation and suggest some avenues for further exploration of better means for understanding object reference.

Obviously, one of the main difficulties in predicting an intended object from a provided word is synonymy. There are many different words that can refer to the same object. Even though the receiver may know several of them, the communicant (or user) may choose another. It is the long-tailed distribution of the word usage for objects, as illustrated in Fig. 1, that is the villain. Unfortunately, however, this is only a part of the problem. Indeed, our upper bound estimates assume that this problem does not exist, that we know every word that will be used, and with what probability each will be applied to every object. Still our best upper bound predictions are quite error prone. Another part of the problem is polysemy; each word means many different things and can refer to many different objects. In our observed data, words that were frequently used tended to be applied to several objects. Clearly, if a word is used for two or more objects, there is no way that we can guess from the use of that word which unique object is its referent. The more similar the objects in the domain, the more likely it is that a single word will include more than one in its scope (Section 4.5.4.2). In general, then, one might expect that the more similar the objects in a set seem to people, the more difficulty they will have in describing them uniquely and the more difficulty a receiver of their descriptions would have knowing to what they refer.

How might one interpret descriptions of objects more accurately? Observe that in the common object experiment, human subjects were able to make a single guess as to the intended object with over 80 percent success. This is well above any of our performance estimates for systems based on the statistical information in single-word input/output tables. How do people do this? What other sources of information, either in the input description or in the receiver's mind, are brought to bear? What is needed to build a system that would approach human capacities?

One possibility is that the limits we observed may be due to utilization of only a single word or phrase from the input. Perhaps if multiple words, or the combined meaning implied by conjunctions of several words and their syntactic order, were utilized, much better reference could be achieved. The conjunctive use of words primarily serves to more narrowly specify the object of discourse. This presumably increases the precision of reference and would reduce the chance that we would guess an incorrect object from a provided description. But, under models that assume the comprehender can return several guesses as to the likely object, the recall provided by a full description

would not necessarily be much greater than that provided by the unrestricted meaning of its most important word. However, if users can provide many independent words, each an essentially new attempt, the recall rate could be raised substantially, as Model 2 suggests. Thus, allowing longer, more complex specifications and learning how to understand them is one promising direction to be investigated.

Recall also that using experts to generate armchair key words did not work well, at least in the recipe data where we tested it. We also note that in informal demonstrations, programmer subjects do not fare well in providing a name for a program to be matched by other programmers. Similarly, there have been a number of studies of indexer reliability in bibliographic indexing.[14-18] These come from quite favorable circumstances, in which the index terms are chosen from restricted vocabularies and the indexers are highly trained. The chances of one indexer choosing the same categories as another, even under these circumstances, are usually disappointingly low. Thus, the chance that professional indexers will agree with the first word entered by an untrained user does not seem to offer a promising route.

A possibility for improvement that seems worthy of further investigation is the study of the structure of the conceptions or mental representations of the objects in a domain to be referenced. The polysemy aspect of the problem arises because two or more objects are not linguistically separable. If we could learn how to group such objects into "super-objects," then we could potentially improve at least our ability to predict which of these "super-objects" is being sought. Similarly, informal impressions from the swap-and-sale superordinate data suggest that certain levels of superordination give rise to more consistent naming than others. If means can be found to reveal and represent strong hierarchical structure in the concepts being named, then possibly one can choose the levels or nodes in such structure that are best represented as the objects to be found in a data set. These "super-objects" also might be more amenable to automatic reference by the means we have modeled.

We have generated statistics relevant to this hypothesis, in the double treatment given to the editing terms. The five objects in the Ed5 data are in fact "super-objects", made by condensing the 25 editing operation–text unit pairs into categories involving just the editing operation. A look back at the numbers involved shows a uniform superiority in the retrieval and discrimination of "super-objects" over that of the individual constituent objects. In part this is just because there are fewer objects to be dealt with, but our hypothesis is that performance is further enhanced by the lower overall similarity at the super-object level. Suppose a system were built to capitalize on this kind of situation, that is by first empirically discovering the lowest

level of subdivision of a given domain at which natural descriptive language is adequate. Freely chosen key-word entries might well be effective for specifying objects at this level. But the whole problem of retrieval would not yet be solved. Users who wanted to access particular subordinates of these super-objects would have to be provided with some further mechanism. A hybrid approach in which early stages of specification are done by freely chosen keys and later stages by menu selection seems a promising approach.

An especially important matter, which we have so far neglected, is the prior probabilities of a user's intending various objects. Our data have been aimed at estimation of the input/output pointer functions of words to objects, and were collected with equal numbers of occasions for nomination of key words for each object. In real life, and in a real system, people seek different objects with different frequencies, perhaps with steep distribution functions resembling Zipf's law.[9] Our ability to predict what object a person has in mind by a word could be greatly improved by taking into account its prior, unconditional likelihood of being sought. Again, our informal impressions from the kinds of descriptions provided in the common object data is that human describers and recipients take great advantage of such frequency information. For example, in specifying the Empire State Building as a tall building in midtown Manhattan, the describer probably assumes that the receiver would choose, from the large set of possible objects, the one that is most likely to be the object of specification.

It will also require further work to see how such information could be incorporated in an automated access system. A start would be to consider an adaptive system that keeps track of the frequency with which objects are sought (as well as the frequency with which the particular input is satisfied by a particular output). Then object prior-probability data could be combined with input/output conditional probability data, like that we have investigated, by the use of Bayes' rule.[19] This would almost certainly yield much better predictions than our models estimate, or any that are currently achievable in available systems. Such a scheme might also take advantage of individual differences in cases where the same people will use the system repeatedly.

There are still other plausible means for circumventing the limitations our models suggest. There are other kinds of data access devices, such as menu-driven systems, query languages, or natural language understanding systems. These approaches all have something to offer, and probably can overcome some of the deficiencies of the pure key-word entry method. But each also has problems of its own. For example, menus cannot list a very large number of alternatives at once, so the desirable feature of turning the recall or production

problem into comprehension and choice is limited. (Note also that in menus the user must understand the system's terse descriptions of objects, much as the system has to understand those of the user of key words.) When there are many objects, a menu system must use a successive search method that relies on some kind of hierarchical tree or other presentation of the relations among the objects. How to do this in a way that leads to correct user choices at each level, to good overall performance, and to acceptable convenience are unsolved issues. Query languages generally require users to input well-formed relational algebra or Boolean expressions that are unlike anything seen in the data specifications provided in our password study. Such expressions require the kind of logical thought that is known to be extremely difficult for ordinary people.[20] Natural language understanding systems, as so far implemented at least, have yet to deal adequately with the lexical reference problem. They have usually glossed over the issue by restricting themselves to very limited domains and limited lexical input, for which they can store a reasonably adequate "hand-tailored" synonym list. We suspect that when such systems are developed for use in real data applications they will have to solve the synonymy and polysemy problems in some of the same ways (e.g., by the use of statistical input/output tables and prior object probabilities that we have been suggesting here. It is probably a mistake to take a too naively optimistic view of the value of "natural language" input to a computer device. For example, although our human subjects were much better than our model automatic systems at predicting the referent of a description, it is not obvious that their success was based on being able to "understand" the natural language of the input, at least if this is taken to mean successful syntactic parsing, etc. We believe that it is also possible that human success is based largely on statistical knowledge of the likelihood of objects and the likely referents of words, and that this is a matter that could be incorporated into a system without it doing highly intelligent natural language understanding.

## VI. SUMMARY

The data we have collected on people describing objects have allowed us to estimate the likely performance of several mechanisms for understanding the references of words to information objects. We have found that input/output functions based on normative naming behavior of users will work much better than systems based on a single name provided by designers. We have also shown that a system that made several best guesses, and/or allowed the user to make several tries, and then returned a menu-like set of guesses to be chosen among, would substantially improve performance beyond current popular

methods. The best approach was to focus on what the user brings to the interaction, namely a great variety of words, and for each word have the system make one or more best guesses as to what the user meant.

But such a system for understanding references will still not perform nearly as well as a human receiver would. Thus, this model of the process of reference certainly does not fully capture what goes on in people's minds. Thus, we are clearly not yet ready to hazard a theory of how humans succeed as well as they do in this task, or to propose an automated method that would do as well or better. However, we believe that the evidence and analyses reviewed here lead to some promising suggestions for further exploration in both regards.

## REFERENCES


1. J. M. Carroll, "Learning, using, and designing command paradigms," Human Learning: Journal of Practical Research and Application, *1* (January 1982), pp. 31–62.
2. T. K. Landauer, K. M. Galotti, and S. Hartwell, "Natural command names and initial learning: A study of text editing terms," Commun. ACM, *26*, No. 7 (July 1983).
3. S. T. Dumais and T. K. Landauer, unpublished work.
4. G. W. Furnas, unpublished work.
5. L. M. Gomez and R. Kraut, unpublished work.
6. *Our favorite recipes: Inverness Garden Club*, private printing, 1977–78.
7. C. Claiborne, *The New York Times Cookbook*, New York: Harper and Row, 1961.
8. J. Child, L. Bertholle, and S. Beck, *Mastering the Art of French cooking*, Vol. I, New York: Knopf, 1979.
9. G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Reading, MA: Addison-Wesley, 1949.
10. G. Herdan, *Type Token Mathematics: A Textbook of Mathematical Linguistics*, Mouton: S-Gravenhage, 1960.
11. D. A. Norman, "The trouble with UNIX," Datamation, *27*, No. 12 (November 1981), pp. 139–50.
12. M. E. Lesk, "Another view," Datamation, *27*, No. 12 (November 1981), p. 146.
13. G. Keppel and B. Z. Strand, "Free association responses to the primary responses and other responses selected from the Palermo-Jenkins norms," in L. Postman and G. Keppel (Eds.), *Norms of Word Association*, New York: Academic Press, 1970, pp. 177–87.
14. R. S. Hooper, "Indexer consistency tests—origin, measurements, results, and utilization," IBM Washington Systems Center, Bethesda, MD, 1965 Congress Int. Federation for Documentation, October 10–15, 1965.
15. J. Jacoby, "Methodology for indexer reliability tests," RADC-IN-62-1, Documentation, Inc., Bethesda, MD (March 1962).
16. D. J. Rodgers, "A study of intra-indexer consistency," General Electric Company, Washington, D. C., (September 1961).
17. J. F. Tinker, "Imprecision in meaning measured by inconsistency of indexing," American Documentation, *17* (April 1966), pp. 96–102.
18. J. F. Tinker, "Imprecision in indexing (Part II)," American Documentation, *19* (July 1968), pp. 322–30.
19. V. E. McGee, *Principles of statistics: Traditional and Bayesian*, Englewood Cliffs, NJ: Prentice Hall, 1971.
20. P. C. Wason and P. D. Johnson-Laird, *Psychology of structure and reasoning: Structure and content*, Cambridge, MA: Harvard University Press, 1972.


## APPENDIX A

### *Formal Model Structures*

In Table III below we present the formal structures for each of the six models studied. The constraints critical to the definition of each

Table III—Summary of constraints defining each model

| Model | Description | Words Used | | Objects Referenced | | Total Word-Object Pairs |
|---|---|---|---|---|---|---|
| | | Per Object | Total | Per Word | Total | |
| 1 | One name per object | **1** | $\leq R$ | $\leq C$ | **C** | $C$ |
| 2 | $M$ names per object | **M** | $\leq R$ | $\leq C$ | **C** | $M \times C$ |
| 3 | Distinct name for each object | **1** | **C** | **1 or 0** | **C** | $C$ |
| 4 | Distinct names, augmented with $M-1$ extra referents | $1 \leq n \leq M$ | **C** | **M or 0** | **C** | $M \times C$ |
| 5 | Recognize one referent for every word | $\leq C$ | **R** | **1** | $C$ | $R$ |
| 6 | $M$ referents for every word | $\leq C$ | **R** | **M** | $C$ | $M \times R$ |

Defining constraints for the models are in boldface. C = number of columns (objects), R = number of rows (words).

model are marked with asterisks. (Note that these constraints are to be met whenever possible, and for simplicity, the numbers below assume that this is always possible.)

## APPENDIX B

### *Evaluating the Pure Random Versions of the Models*

The success of any pure random model considered in this paper is given simply by the ratio of $t$, the number of cells included in the system mapping, to $RC$, the total number of cells in the matrix. To see this simply note that any structural constraints we might impose are only defined up to a permutation of the rows and columns. Thus, consider any table and its group, i.e., all its variants obtained by row and column permutations. The table that is the cell-wise total of all these tables must, for reasons of symmetry, be everywhere the same, and its grand total must be $t$ times the number of tables in the group. The success of any individual table is given by the dot product of the user and system matrices (treating the matrices as vectors, i.e., sum the products of corresponding cells). The total success for the group is the sum of these dot products for the members of the group. But the sum of the dot product of one vector with several others is the dot product of that vector with the sum of the others, so the total success for the group is just the dot product of the user matrix and the total matrix. But since the total matrix is uniform, and we divide by the size of the group, we conclude that the average matrix then is just $t/rc$ times the sum of user matrix. If the user matrix is in relative frequencies, the success is a probability.

## APPENDIX C

### *Summary of Recall Probabilities*

Table IV—Summary of recall probabilities

| Model | Description | Version | $M$ | Recall Probabilities Ed5 ($n$ = 5) | Ed25 ($n$ = 25) | CmOb ($n$ = 50) | Swap ($n$ = 64) | Recp ($n$ = 188) |
|---|---|---|---|---|---|---|---|---|
| 1 | One name per object | WGT | 1 | .074 | .106 | .117 | .142 | .182† |
| | | OPT | 1 | .151 | .194 | .257 | .258 | .312* |
| | | | | .271 | .322 | .329 | .353 | .419† |
| | | | | .163 | .221 | .279 | .337 | .362‡ |
| 2 | Several names per object | WGT | 1 | .074 | .106 | .117 | .142 | .182† |
| | | | 2 | .144 | .205 | .210 | .253 | .332† |
| | | | 3 | .211 | .295 | .284 | .337 | .445† |
| | | OPT | 1 | .151 | .194 | .257 | .258 | .312* |
| | | | | .163 | .221 | .279 | .337 | .362‡ |
| | | | 2 | .263 | .319 | .358 | .358 | .490* |
| | | | | .281 | .369 | .406 | .496 | .564‡ |
| | | | 3 | .365 | .424 | .420 | .452 | .578* |
| | | | | .381 | .492 | .478 | .595 | .670‡ |
| 3 | Distinct name for each object | WGT | 1 | .073 | .076 | .105 | .124 | .086* |
| | | "OPT" | 1 | .140 | .109 | .226 | .194 | .105* |
| 4 | Distinct names, augmented with $M$ extra referents | WGT | 1 | .067 | .076 | .109 | .127 | .084* |
| | | | 2 | .125 | .145 | .151 | .172 | .149* |
| | | | 3 | .175 | .212 | .181 | .196 | .200* |
| | | "OPT" | 1 | .140 | .109 | .221 | .208 | .105* |
| | | | 2 | .214 | .204 | .277 | .266 | .170* |
| | | | 3 | .266 | .287 | .306 | .296 | .220* |
| 5 | Recognize one referent for every word | WGT | 1 | .413 | .153 | .516 | .617 | .128† |
| | | OPT | 1 | .489 | .176 | .434 | .351 | .129* |
| | | | | .620 | .353 | .647 | .715 | .276† |
| | | | | .541 | .258 | .686 | .813 | .247‡ |
| 6 | $M$ referents for every word | WGT | 1 | .413 | .153 | .516 | .617 | .128† |
| | | | 2 | .656 | .264 | .646 | .743 | .207† |
| | | | 3 | .811 | .358 | .713 | .795 | .266† |
| | | OPT | 1 | .489 | .176 | .434 | .351 | .129* |
| | | | | .541 | .258 | .686 | .813 | .247‡ |
| | | | 2 | .699 | .302 | .532 | .427 | .203* |
| | | | | .761 | .416 | .819 | .919 | .373‡ |
| | | | 3 | .830 | .405 | .577 | .465 | .259* |
| | | | | .884 | .531 | .881 | .956 | .455‡ |

* Split halves
† One of the several repeat rate related statistics
‡ Self-prediction

## AUTHORS

**George W. Furnas,** B.A. (Psychology), 1974, Harvard University; Ph.D. (Psychology), 1980, Stanford University; Bell Laboratories, 1980—. Mr. Furnas is a member of the Human Information Processing Research Department. He is doing research on cognitive aspects of person-computer interaction, with emphasis on access to information structures. Particular topics of recent interest include adaptive indexing, multidimensional scaling, and a "fisheye lens" viewer for large structures. Member, Psychometric Society, Classification Society, Society for Mathematical Psychology.

**Thomas K. Landauer,** B.S. (Anthropology), 1954, University of Colorado, Ph.D. (Social Psychology), 1960, Harvard University; Dartmouth College,

1960–1964; Stanford University, 1964–1969; Bell Laboratories, 1969—. During his first 10 years at Bell Laboratories, Mr. Landauer studied a variety of problems involving human memory. He worked on optimal scheduling of practice and study to maximize learning, did research aimed at understanding how humans retrieve information from their own memories, and developed a mathematical model of human memory that assumes it to be a content-addressable parallel access device that imposes no inherent structure on its content. For the last three years he has done research on psychological problems of human-computer communication. Fellow, AAAS; member, ACM, American Psychological Association, Psychonomic Society, Society for Cross-Cultural Research, Classification Society.

**Louis M. Gomez,** B.A. (Psychology), 1974, State University of New York at Stony Brook; Ph.D. (Psychology), 1979, University of California, Berkeley; Bell Laboratories, 1979—. Mr. Gomez is a member of the Human Information Processing Research Department. His current interests are in human factors of information retrieval, and individual differences in the acquisition of computer skills.

**Susan T. Dumais,** B.A. (Mathematics and Psychology), 1975, Bates College; Ph.D., (Psychology), 1979, Indiana University; Bell Laboratories, 1979—. Ms. Dumais is a member of the Human Information Processing Research Department. Her primary research interests are in the area of cognitive aspects of human-computer interface. She is currently involved in work on information retrieval, including menu selection, describing categories, and browsing for information. Member, AAAS, Midwestern Psychological Association, Metropolitan Chapter of the Human Factors Society.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# On Abbreviating Command Names

By L. A. STREETER,* J. M. ACKROFF,† and G. A. TAYLOR†

(Manuscript received May 21, 1982)

Test subjects' abbreviations of command names and randomly selected English words were examined for production regularities. Abbreviation rules based primarily on a word's number of syllables were devised to capture regularities observed in people's productions. This rule set was compared to two simpler abbreviation rules—vowel deletion and truncation. In subsequent learning experiments, separate groups of subjects learned the rule-derived abbreviations for words, while other groups learned the most frequently given abbreviation for each word. Subjects who studied rule-derived abbreviations remembered substantially more of them when prompted with full words than did subjects who studied the most frequently given abbreviations. Moreover, the rule-based abbreviations were superior even for those for which the rule-produced and the most frequently produced abbreviations were identical. When the task was reversed (recall the source term given an abbreviation), performance was best for vowel deletion abbreviations and worst for the rule set abbreviations. We suggest that both memorability of abbreviations and the probability that people will spontaneously produce a "correct" abbreviation are increased by: (1) selecting abbreviations using a vowel deletion rule for one-syllable words and an acronym rule for multiple-word terms, as well as (2) allowing variable length truncations of words.

* Bell Laboratories. † American Bell.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

## I. INTRODUCTION*

Zipf[1] observed that in a number of languages frequently used words tend to be short. Further, this relation appears to be a causal one: as a word or phase increases in usage, it becomes shorter. Examples of word or phrase abbreviation are plentiful in English (e.g., "television" becomes "TV"). Abbreviating is particularly prevalent in government, corporations, and disciplines sharing a specialized vocabulary. Often the abbreviation process is so complete that the antecedent term is forgotten and only the abbreviation is retained (e.g., "cathode ray tube" becomes "CRT" or "light amplification by stimulated emission of radiation" becomes "laser"). Given the frequency and importance of this phenomenon, it seems reasonable to ask whether the process is systematic. That is, given a word or phrase, can one predict how it most likely will be abbreviated? If one could formulate abbreviation rules rooted in natural abbreviating behavior, would these abbreviations be "better" than abbreviations not suggested by behavioral data?

Discovering ways to generate abbreviations that are easy to learn and use has practical significance, since in many circumstances brevity is needed or desired. For instance, while infrequent users of a computer command language may be content to enter complete command names and arguments, frequent users often prefer terse command strings. To avoid typing long command strings, mechanisms should exist for abbreviating commands and their arguments.

There are some common ways to provide abbreviation facilities. One, truncation, allows the user to enter only as much of the command name (from the beginning to some point less than the whole word) as necessary to differentiate it from all other commands in the language. Another provides the user with standard system abbreviations. It is also possible to use both of these methods.

Selecting "good" abbreviations is a complicated problem. There is probably no single measure of an abbreviation's goodness, since there are some competing requirements to consider. For instance, one might want abbreviations in a system to be unique, easy to type, memorable, and natural (i.e., users who knew only the command would be more likely to produce this particular abbreviation than any other). However, some of these requirements are at odds with one another. Consider "easy to type" and "memorable." As Experiment II shows (see Section III), abbreviations produced by a simple vowel deletion rule are easy to remember or generate. Thus, given the command "journal,"

* Some of the material contained in this paper was presented at the 52nd Annual Meeting of the Midwest Psychological Association, St. Louis, May 1980 and the 52nd Meeting of the Eastern Psychological Association, New York City, April 1981.

subjects can generate the requisite vowel deletion abbreviation, "jrnl," with nearly perfect accuracy. However, this abbreviation is probably not easy to input, since the user must spell the whole word and omit every vowel. While we have no production data, it is likely that using a vowel deletion abbreviation rule for long words is no faster than entering the complete word. In fact, it may be slower in many cases.

Consider "easy to type" and "unique." Truncation produces abbreviations that seem easy to type (at least no more difficult to type than the whole word for a skilled typist and probably easier than the whole word for a nonexpert typist), particularly if one truncates to only a few letters. However, the likelihood of collisions among commands with truncation increases as the number of letters used decreases. If one requires uniqueness of abbreviations and has a truncation rule, the user must learn the minimum number of letters required to distinguish a particular command from all other commands. An additional burden may be imposed on the user if the command lexicon grows to include words whose truncations suddenly conflict with old abbreviations. If the lexicon is large enough, conflicts between abbreviations are bound to arise. One way to resolve conflicts in a rule-based system is to allow exceptions to the abbreviation rule. However, to what degree rule-generated systems can tolerate exceptions is an open issue worthy of future investigation. That is, at what point do exceptions to the rule render the rule worthless? It could well be that rules are useful only when there are relatively few or even no exceptions.

While each of these abbreviation requirements is important and critical from a design standpoint, the present study focuses on abbreviation naturalness and memorability. We examined abbreviations to determine how much regularity existed in those produced naturally. Next we incorporated the observed regularity into a set of abbreviation rules and compared the learning of rule-based abbreviations to the learning of the most frequently produced abbreviations for each word. The learning process was also reversed to include recalling of a word from its abbreviation, when the abbreviation was produced either by rules or by consensus.

## II. EXPERIMENT I: REGULARITIES IN NATURALLY PRODUCED ABBREVIATIONS

### *2.1 Introduction*

To determine how people abbreviate, we solicited abbreviations for a number of command terms. We then looked for regularities and the amount of agreement in these productions.

There were already data indicating that long and short words are abbreviated differently. Hodge and Pennington[2] examined how people

abbreviate words that varied in letter length. They observed that subjects used two principal mechanisms. One method, "contraction" (omitting word internal letters), tended to be used to abbreviate short words, whereas the other method, "truncation," was favored in abbreviating long words. However, they did not examine other linguistic properties of words associated with each of these strategies.

To examine abbreviations, we partitioned them into classes based on whether a particular abbreviation was an instance of a vowel deletion rule, a truncation rule, or an acronym formation rule. (Here acronyms are formed from multiple words, in which the first letter of each word is taken for the abbreviation; e.g., "cathode ray tube" becomes "CRT.") We examined whether the proportion of abbreviations accounted for by each rule varied with the number of syllables or words in the abbreviated term.

### *2.2 Method*

#### *2.2.1 Stimuli*

The vocabulary consisted of 81 command names and arguments that were to be used in a large computer system. Many of these terms, such as move, copy, and insert, are common to other computer systems. Others were unique to this particular application, such as usage billing number. The items varied from one to three words. Table I gives statistics for the commands and arguments used in the present study in terms of number of words and syllables and average number of vowels and consonants.

#### *2.2.2 Subjects*

There were two groups of subjects; one group consisted of 19 psychologists doing human factors work at Bell Laboratories, Holmdel, New Jersey, while the other set consisted of 30 adult females living in the Holmdel area. The Bell Laboratories employees volunteered their time; the other subjects were paid for participating.

#### *2.2.3 Procedure*

We gave each group a randomized list of the command terms and asked them to produce "a good abbreviation" for each, that is, one that would be easy to use and remember. Because the command list grew, the human factors psychologists saw a subset of 42 terms, whereas the other subjects saw 81 terms. The paid subjects produced the abbreviations and then performed a 30-minute text-editing task. After the intervening task, these subjects recalled as many of the command terms as they could. There was no free recall task for the psychologists.

### 2.3 Results and discussion

To determine the consistency with which subjects produced abbreviations, we examined the abbreviation that was most frequently produced for each item. Overall, the concurrence across the 81 terms was 37 percent; that is, on the average the most commonly produced abbreviation for each item comprised 37 percent of the abbreviations. There were no discernible differences between the psychologists and local residents in their abbreviations of the terms. The mean percentage concurrence was identical to the nearest percent for the two groups. The groups did not differ on other production measures, such as number of different abbreviations produced or the most commonly produced abbreviation. Consequently, data for the two groups were pooled in subsequent analyses.

Table II shows the average concurrence across items and the standard deviation both for all terms combined and separately by number of syllables. Concurrence decreased markedly as the number of syllables in a word increased and, correspondingly, the number of different abbreviations given for each term increased substantially. Thus, ab-

Table I—Composition of command names and arguments

| | Overall N = 81 | One-Syllable Words N = 15 | Two-Syllable Words N = 42 | Three-Syllable Words N = 12 | Four-Syllable Words N = 2 | Multiple Words N = 10 |
|---|---|---|---|---|---|---|
| Mean number of: | | | | | | |
| Consonants | 4.11 | 2.60 | 3.80 | 4.58 | 4.50 | 7.10 |
| Vowels | 2.77 | 1.67 | 2.55 | 3.17 | 4.50 | 4.50 |
| Syllables | 2.19 | 1.00 | 2.00 | 3.00 | 4.00 | 3.50 |
| Words | — | — | — | — | — | 2.10 |

Table II—Experiment 1: Abbreviation production

| | Overall N = 81 | One-Syllable Words N = 15 | Two-Syllable Words N = 42 | Three-Syllable Words N = 12 | Four-Syllable Words N = 2 | Multiple Words N = 10 |
|---|---|---|---|---|---|---|
| Mean percentage concurrence | 37% | 56% | 36% | 32% | 28% | 24% |
| Standard deviation | (16%) | | | | | |
| Mean number of different abbreviations | 11.95 | 5.73 | 11.13 | 16.00 | 18.00 | 18.90 |
| Standard deviation | (5.68) | | | | | |
| Average number of terms recalled | 3.25 | 4.87 | 3.20 | 2.50 | 2.00 | 2.20 |
| Standard deviation | (3.15) | | | | | |

breviation production was more homogeneous in simpler linguistic environments.

The one or two most frequently elicited abbreviations for each term were studied to determine whether abbreviating could be characterized by some "rule-governed" process. If one ignored the linguistic composition of the words, the abbreviation process appeared unorderly. However, if the terms were partitioned into three classes (monosyllabic, polysyllabic, and multiple words), there was some consistency within each class. By inspection there appeared to be three different rules operating in people's productions—vowel deletion (delete word internal vowels), truncation, and "acronym formation" (for multiple-word terms, select the first letter of each word).

We derived rules to account for the regularity observed in the abbreviating behavior. Table III shows the set of rules that appear to describe the data. The application of a particular rule depended on the number of syllables and/or number of words in the command term.

Table IV compares the proportion of abbreviations accounted for by the three different rules (truncation, vowel deletion, and the rule set from Table III) with the most frequently given abbreviation from each term ("popular"). We combined all truncations from the first letter to one less than the number of letters in a word. We defined vowel deletion as the deletion of all vowels following the first consonant in a word ("a," "e," "i," "o," "u," and "y," when it functioned as a vowel). It is important to remember in interpreting Table IV that popular abbreviations and those produced by the three rules are *not necessarily mutually exclusive.* For example, for one-syllable words there is a great deal of overlap between the rule set and vowel deletion. The rule set for one-syllable words is essentially a vowel deletion rule with only a few other features. However, there was no overlap between the truncated one-syllable abbreviations and the rule set or the vowel

Table III—Abbreviation rules

- Monosyllabic words:
  1. Take initial letter of the word and all subsequent consonants.
  2. Make adjacent double letters single.
  3. If more than four letters remain, retain the fifth letter if it is part of a functional cluster (such as *th, ch, sh, ph, ng*); otherwise, truncate from the right.
  4. Delete the fourth letter if it is silent in the word.
- Polysyllabic words:
  1. Take the entire first syllable.
  2. If second syllable starts with a consonant cluster, add it.
  3. If first syllable is a prefix (such as *de, re, in*) add the second syllable.
  4. Make final double consonants single.
  5. Truncate to four letters (but always retain entire first syllable).
- Multiple words:
  1. Retain the first letter of each word as the abbreviation.

Table IV—Proportion of abbreviations generated for each rule (variable length truncation, vowel deletion, rule set) compared with popular abbreviation for each term

| | Truncation | Vowel Deletion | Rule | Popular |
|---|---|---|---|---|
| | | Overall, N = 81 | | |
| Mean | 0.286 | 0.216 | 0.283 | 0.373 |
| Median | 0.286 | 0.143 | 0.267 | 0.367 |
| Standard error | 0.022 | 0.023 | 0.022 | 0.017 |
| | | One Syllable, N = 15 | | |
| Mean | 0.272 | 0.519 | 0.556 | 0.556 |
| Median | 0.233 | 0.533 | 0.533 | 0.533 |
| Standard error | 0.053 | 0.052 | 0.036 | 0.036 |
| | | Two Syllable, N = 42 | | |
| Mean | 0.313 | 0.192 | 0.208 | 0.358 |
| Median | 0.330 | 0.173 | 0.200 | 0.333 |
| Standard error | 0.025 | 0.024 | 0.023 | 0.018 |
| | | Three and Four Syllable, N = 14 | | |
| Mean | 0.408 | 0.077 | 0.247 | 0.317 |
| Median | 0.388 | 0.051 | 0.224 | 0.265 |
| Standard error | 0.040 | 0.020 | 0.056 | 0.088 |
| | | Multiple Words, N = 10 | | |
| Mean | 0.020 | 0.024 | 0.237 | 0.237 |
| Median | 0 | 0.027 | 0.173 | 0.173 |
| Standard error | 0.012 | 0.022 | 0.038 | 0.038 |

deletion set. For two-syllable words all rule set abbreviations were a subset of truncation abbreviations. In the case of multiple words, where an acronym formation rule applied, there was no overlap among the truncation, vowel deletion, and rule set.

Overall, truncation and the rule set were not significantly different from each other, but each was different from the most popular abbreviation (truncation vs. popular, $t_{80} = 3.507, p < 0.001$; rule vs. popular, $t_{80} = 5.316, p < 0.001$). For one-syllable and multiple words, the rule set was more often the same as people's natural abbreviations than was truncation. However, for two-, three-, and four-syllable words, truncation surpassed the rule set abbreviations.*

It is worth considering the results of allowing variable length truncation and the rule set abbreviations to be used in the same system. If both were allowed for this word set, what proportion of generated abbreviations would be subsumed? In this case the proportion equals 0.415 overall, 0.828 for one-syllable words, 0.313 for two-syllable words,

* In some cases the results of detailed statistical analysis are not presented. We have in all cases given variability estimates for each cell mean. For the most part, the differences among experimental conditions in all experiments were substantial and large relative to the observed variance.

0.408 for three- and four-syllable words, and 0.257 for multiple words. Thus, it is clear that allowing multiple abbreviations accounts for significantly more of the naturally produced abbreviations than any single abbreviation mechanism (truncation vs. combined rules, $t_{80} = 5.512$, $p < 0.001$; rule set vs. combined rules, $t_{80} = 9.846$, $p < 0.001$).

## III. EXPERIMENT II: LEARNING RULE-GENERATED VS. POPULAR ABBREVIATIONS

### *3.1 Introduction*

In the second experiment we examined whether learning a set of abbreviations generated by the rules was better than learning those abbreviations most frequently given by the subjects in Experiment I. That is, does placing abbreviations in an internally consistent set result in more memorable abbreviations than people's natural abbreviations?

### *3.2 Method*

#### *3.2.1 Materials*

The materials consisted of the 81 command terms and arguments from Experiment I and their abbreviations. There were two abbreviation conditions:

1. Rule Condition: The 81 abbreviations produced by the rule set shown in Table III.
2. Popular Condition: The most frequently given abbreviation for each of the 81 terms.

(Note that for 65 percent of the terms, the abbreviations were identical in the two conditions. That is, for the rule set 65 percent of the terms produced the same abbreviation as the one most frequently given by subjects in Experiment I.)

#### *3.2.2 Subjects*

We paid 44 high school students from the Murray Hill, New Jersey, area to participate. There were 23 subjects in the "rule" group and 21 subjects in the "popular" group.

#### *3.2.3 Procedure*

Subjects were randomly assigned to either the rule or the popular group. The task was paired-associate learning of the terms and their abbreviations. The rule subjects saw each term paired with the rule-generated abbreviation, whereas the popular group saw each term paired with the most frequently given abbreviation. Subjects in each group were given a randomized deck of 81 cards, each of which contained one term-abbreviation pair. Each subject received a different random order. Subjects studied each card for five seconds, and at the

sound of a buzzer, flipped to the next study card. During an intervening period of 20 minutes, subjects solved logic problems. Then they were given a list with only the terms and were asked to supply the abbreviations they had learned. If unsure, they were to make their best guess.

### *3.3 Results and discussion*

Table V shows the proportion of correct responses for the two experimental groups for all words combined, separately for the abbreviations that were identical for the two groups, and for those that were different for the two groups. It also shows the standard error of the mean calculated across terms. The difference between the rule and popular conditions was highly reliable ($p < 0.001$) in all three cases. Thus, learning an internally consistent set of abbreviations facilitated later recall or generation. (Note that if subjects implicitly or explicitly knew the rules, they should have been able to generate the correct abbreviation for a given term.)

We performed additional analyses to determine in which linguistic environments the rules most facilitated performance. Table VI shows the mean proportion correct as a function of experimental group for one-, two-, three-, and four-syllable words, and multiple words.

The ability of the abbreviation rules to predict the popular abbreviations was far from uniform across the syllable/word classes. For one-syllable words and multiple-word terms, the rule and popular abbreviations were identical. While there was still a recall advantage for the rule abbreviations, it was small and not statistically reliable. However, this was not the case for the other three word classes. Evidently, the abbreviation rules for one-syllable and multiple words are reasonably straightforward and either tacitly known or easily learned by subjects.

The rules and natural productions were in only moderate agreement

Table V—Proportion of abbreviations correctly recalled and/or generated as a function of experimental group

| | Rule | Popular |
|---|---|---|
| Overall (81 terms) | 0.70 | 0.54 |
| Standard error of the mean | 0.02 | 0.03 |
| Abbreviations same in two groups (53 terms) | 0.76 | 0.63 |
| Standard error | 0.03 | 0.03 |
| Abbreviation different in two groups (28 terms) | 0.54 | 0.37 |
| Standard error | 0.04 | 0.04 |

Table VI—Proportion correct in rule and popular conditions as a function of the linguistic class of term

| | One-Syllable Words N = 15 | Two-Syllable Words N = 42 | Three-Syllable Words N = 12 | Four-Syllable Words N = 2 | Multiple Words N = 10 |
|---|---|---|---|---|---|
| Mean proportion correct | | | | | |
| Rule group | 0.78 | 0.62 | 0.59 | 0.50 | 0.97 |
| Popular group | 0.73 | 0.42 | 0.45 | 0.44 | 0.91 |
| Proportion of items for which rule abbreviation and popular abbreviation were the same | 1.00 | 0.45 | 0.58 | 0.50 | 1.00 |

for two-, three-, and four-syllable words, and for these word classes, using the rules to produce abbreviations had the greatest facilitative effect on performance. One could presume that there is more variation in terms of linguistic composition for these classes. Owing to this complexity, people's internal abbreviation rules may be inconsistent, incomplete, and therefore, more variable. However, if rules are formulated for subjects, the subjects can use them to learn abbreviations.

### *3.4 Conclusions*

The results and conclusions to be drawn from Experiments I and II are relatively straightforward. First, the process by which we produce abbreviations is not entirely idiosyncratic, but is to a large degree regular. We derived abbreviation principles that characterized a majority of the most frequently elicited abbreviations for each term. The abbreviation rule produced abbreviations that were easier to learn than the most frequently produced abbreviations. There was a substantial advantage of learning abbreviations in a mutually consistent set, even when the abbreviation being learned *was the same* as would be frequently given by subjects. However, the generality of these results is limited, since the rules were derived from the same set of words that served as stimuli in the recall.

## IV. EXPERIMENT III: ABBREVIATION PRODUCTION AND LEARNING OF RANDOM WORDS

### *4.1 Introduction*

To test the generalizability of the rule set used in the previous experiment, we applied the rules to a random set of words selected to have the linguistic properties considered relevant in the rule set. A replication was needed, since it is possible that the terms used in the

command set studied in Experiments I and II have properties that differ in some unknown way from ordinary English words. A priori, it is conceivable that the command name words might be different semantically, phonetically, and grammatically from "ordinary" English words. Thus, to know whether rule-based abbreviations are better in general, the test words should be a random sample of words. Also, since the rules were generated from examining subjects' abbreviations of these words, the rule set may include conditions that are the result of idiosyncracies in the particular commands selected.

### *4.2 Method*

#### *4.2.1 Word selection*

A total of 200 English nouns and verbs were randomly selected from Kucera and Francis[3]. Words were either one ($N = 40$), two ($N = 80$), or three ($N = 80$) syllables. For the two- and three-syllable words, 40 had prefixes (e.g., "de," "dis," "pro," "post," and "mis"), while 40 did not.

#### *4.2.2 Production subjects*

Thirty-two paid Rutgers University undergraduates supplied abbreviations for the random words.

#### *4.2.3 Procedure*

The word set was randomly divided in half with half of the subjects supplying abbreviations for 100 words.

### *4.3 Production results*

We analyzed the data in the same way as in Experiment I. Tables VII and VIII show the proportion of abbreviation productions accounted for by variable length truncation, vowel deletion, the rule set, and the most frequently given abbreviation (popular). Table VII shows proportions for all words combined and separately for one-, two-, and three-syllable words. Note that the proportions in Table VII are reasonably close to those found in Experiment I.

Table VIII compares two- and three-syllable words with and without prefixes. The existence of a prefix decreased abbreviation agreement in the popular condition ($t_{79} = 2.790, p < 0.01$). Thus, prefixes affected production. However, it does not appear that prefix conditions in the rule set (as described in Table III) managed to capture any of the differences between words with and without prefixes. Abbreviation behavior was best represented by a truncation rule irrespective of whether the word had a prefix.

Table VII—Abbreviations generated for randomly selected English words

| | Truncation | Vowel Deletion | Rule Set | Popular |
|---|---|---|---|---|
| | | All words, N = 200 | | |
| Mean | 0.396 | 0.234 | 0.304 | 0.406 |
| Median | 0.375 | 0.188 | 0.312 | 0.312 |
| Standard error | 0.013 | 0.014 | 0.015 | 0.015 |
| | | One-Syllable Words, N = 40 | | |
| Mean | 0.228 | 0.483 | 0.473 | 0.512 |
| Median | 0.188 | 0.500 | 0.469 | 0.500 |
| Standard error | 0.027 | 0.033 | 0.034 | 0.027 |
| | | Two-Syllable Words, N = 80 | | |
| Mean | 0.382 | 0.245 | 0.274 | 0.374 |
| Median | 0.375 | 0.250 | 0.281 | 0.375 |
| Standard error | 0.018 | 0.017 | 0.019 | 0.015 |
| | | Three-Syllable Words, N = 80 | | |
| Mean | 0.493 | 0.099 | 0.250 | 0.383 |
| Median | 0.500 | 0.062 | 0.188 | 0.375 |
| Standard error | 0.019 | 0.008 | 0.024 | 0.017 |

## V. LEARNING RULE-BASED AND POPULAR ABBREVIATIONS FOR RANDOM WORDS

### *5.1 Method*

#### *5.1.1 Subjects*

A total of 90 paid adult recruits from the Holmdel area participated in the experiment.

#### *5.1.2 Procedure*

The procedure was identical to that reported in Experiment II with the exception that the interpolated task was underlining "important concepts" in a text-editing manual. There were three conditions. Thus, in separate conditions, each word was paired with (1) its rule set abbreviation, (2) its popular abbreviation, and (3) its vowel deletion abbreviation. There were 30 subjects in each of these three conditions.

### *5.2 Learning results*

Table IX shows the proportions and standard errors across words for each of the three groups. In each row each condition was significantly different from every other condition. (The largest matched-paired $t$ probability of occurrence was less than 0.02. However, this probability value is uncorrected, i.e., it does not take into account the number of post comparisons.) Thus, the ordering of generation or recall performance was best for the simple vowel deletion rule, second best for the rule set condition, and worst for the popular abbreviations.

Thus, the original results that rule-governed abbreviations were

Table VIII—Abbreviations generated for randomly selected two- and three-syllable words with and without prefixes

| | Truncation | Vowel Deletion | Rule Set | Popular |
|---|---|---|---|---|
| | | Two-Syllable Words, No Prefix (N = 40) | | |
| Mean | 0.425 | 0.259 | 0.326 | 0.411 |
| Median | 0.438 | 0.250 | 0.312 | 0.375 |
| Standard error | 0.028 | 0.025 | 0.030 | 0.021 |
| | | Two-Syllable Words, Prefix (N = 40) | | |
| Mean | 0.339 | 0.231 | 0.222 | 0.338 |
| Median | 0.312 | 0.188 | 0.250 | 0.312 |
| Standard error | 0.020 | 0.024 | 0.023 | 0.021 |
| | | Three-Syllable Words, No Prefix (N = 40) | | |
| Mean | 0.483 | 0.109 | 0.308 | 0.406 |
| Median | 0.500 | 0.094 | 0.344 | 0.375 |
| Standard error | 0.029 | 0.012 | 0.037 | 0.024 |
| | | Three-Syllable Words, Prefix (N = 40) | | |
| Mean | 0.503 | 0.089 | 0.192 | 0.359 |
| Median | 0.500 | 0.062 | 0.125 | 0.312 |
| Standard error | 0.024 | 0.012 | 0.029 | 0.025 |

better reproduced than abbreviations produced by consensus were strongly replicated. Note that a simple rule produced the best performance in this task. The vowel deletion rule appears to be a particularly easy rule for subjects to abstract. Thus, performance improved as the rules governing abbreviations became more straightforward.

## VI. EXPERIMENT IV: RECALLING THE SOURCE WORD GIVEN ITS ABBREVIATION

### *6.1 Introduction*

One of the anecdotal observations frequently made about new computer users is that they learn abbreviations for command terms without ever learning what some of the abbreviations mean. In this sense, the abbreviations represent an alternate name space. There is usually some care given to the choice of names for commands, devices, programs, etc, the assumption being that the use of appropriate names tends to make learning easier (but see Landauer, Galotti, and Hartwell[4], where random words assigned to text-editing functions were no easier or harder to learn or use than appropriate names). However, this care is not usually taken in choosing abbreviations. Most people's goal is to choose abbreviations that are easy to remember, given that the user remembers the right name in the first place. In this experiment we were interested in the decoding process: determining how well the abbreviations represented the words they abbreviated.

There is evidence that the decoding and encoding (determining an abbreviation, given a source word) are asymmetric processes. In par-

Table IX—Mean proportion of abbreviations recalled/produced correctly, given words

| | Popular | Rule Set | Vowel Deletion |
|---|---|---|---|
| Overall (200) | 0.466 | 0.650 | 0.811 |
| Standard error | 0.015 | 0.014 | 0.006 |
| 1-syllable (40) | 0.579 | 0.634 | 0.875 |
| Standard error | 0.029 | 0.023 | 0.011 |
| 2-syllable (80) | 0.428 | 0.622 | 0.815 |
| Standard error | 0.031 | 0.035 | 0.013 |
| 3-syllable (80) | 0.446 | 0.685 | 0.774 |
| Standard error | 0.039 | 0.032 | 0.011 |

ticular, rule-based abbreviation schemes are not as effective for decoding purposes as for encoding purposes. With a well-formed rule it is possible to produce the abbreviation accurately when given the word, but the converse is not true. That is, knowing the abbreviation rule and the abbreviation does not necessarily allow one to generate the correct source word.

Rogers and Moeller[5] compared decoding of conventional Navy sonar abbreviations and rule-based abbreviations (produced by truncation). (The subjects in their experiments were sonar operators.) Reaction time to decode the abbreviation was measured. When the scoring criteria were strict, the rule-based abbreviations were worse than the conventional military abbreviations. The difference was due primarily to conventional abbreviations giving a better indication of the endings of words.

Using two 20-word lexicons, Hirsh-Pasek, Niedelman, and Schneider[6] examined decoding in a wide variety of conditions: truncation to four letters, vowel deletion with truncation to four letters, minimum number of letters to distinguish, a phonics system, and user-supplied abbreviations. Vowel deletion and the phonics system produced the fewest number of errors, whereas minimum to distinguish produced the most. Truncation was only slightly worse in terms of errors (perhaps not significantly) than vowel deletion and phonics.

Ehrenreich and Porcu[7] compared decoding of abbreviations formed by vowel deletion and by truncation. There were no reliable differences between these two methods with liberal scoring procedures—65 percent and 57 percent correct, respectively. With stricter scoring criteria, truncation abbreviations were easier to decode (54 percent) than vowel deletion abbreviations (48 percent).

### *6.2 Method*

#### *6.2.1 Subjects*

We paid 90 residents from the Holmdel area to participate.

### 6.2.2 Procedure

We prepared cards using the 200 random English words used in Experiment III. One word and its abbreviation appeared on each card. There were three groups: one group learned popular abbreviations, another the rule set, and a third, the vowel deletion abbreviations. The procedure was the same as in previous experiments, except that after the 20-minute retention interval, subjects were given lists of *abbreviations* and asked to *write down the word* each represented.

### 6.3 Results and discussion

Table X compares the proportion of words recalled correctly in each of the three groups. Note that in Experiments II and III subjects recalled or produced abbreviations formed by the rule set more often than subjects who learned abbreviations arrived at by consensus. We attributed this improvement in performance to the fact that the existence of the rule lent some organization to the set of stimuli being learned. When we looked at ability to recall what the abbreviations represent, however, it appeared that the rule set seemed to interfere with subjects' ability to recall what the abbreviations stood for. Thus, there is an asymmetry between recalling the abbreviation from seeing the command term and recalling the command term from seeing its abbreviation. The superiority of the vowel deletion rule in this task is evident for all but the one-syllable words, and for this case the rule set is essentially a vowel deletion rule.

These results are somewhat at odds with those of Ehrenreich and Porcu.[7] First, in our experiment, the vowel deletion condition produced very high performance. Also, the overall level of performance was much higher in our study (79 percent here vs. 48 percent in the Ehrenreich and Porcu study). Ehrenreich and Porcu claim that when subjects produce words from vowel deletion abbreviations they are apt

Table X—Mean proportion of words recalled/produced correctly, given abbreviations

| | Popular | Rule Set | Vowel Deletion |
|---|---|---|---|
| Overall (200) | 0.626 | 0.467 | 0.788 |
| Standard error | 0.022 | 0.024 | 0.018 |
| 1-syllable (40) | 0.662 | 0.635 | 0.641 |
| Standard error | 0.046 | 0.050 | 0.052 |
| 2-syllable (80) | 0.628 | 0.500 | 0.799 |
| Standard error | 0.037 | 0.038 | 0.027 |
| 3-syllable (80) | 0.605 | 0.350 | 0.850 |
| Standard error | 0.035 | 0.034 | 0.024 |

to make spelling errors. People who misspell a word are more likely to err on a vowel than a consonant. Their terms were Army terms—ours random words; their subjects were Army personnel—ours primarily spouses of Bell Laboratories employees. These differences may have produced the resulting performance differences.

It would seem that vowel deletion should have produced the best results, since consonants carry more information than vowels and the abbreviations in this condition were longer than in the other two conditions (vowel deletion = 4.43 letters, standard deviation = 1.13; rule condition = 3.38 letters, standard deviation = 0.62; popular = 3.94 letters, standard deviation = 1.08). Thus, the results vary directly with the number of letters in the abbreviation alone.

In summary, there was a large asymmetry between the encoding results for the popular and rule condition in Experiment III and the decoding results in Experiment IV. Rule abbreviations were better than popular for remembering the abbreviation given the word, whereas popular abbreviations were better than rules for remembering the word given the abbreviation.

## VII. GENERAL DISCUSSION

While there is much variability in people's natural abbreviations, the process is far from random. People are most likely to abbreviate one-syllable words by deleting word-internal vowels, and multiword terms by deleting everything except the first letter of each word. Abbreviations of polysyllabic words are best characterized by truncation, that is, deleting letters from the end of the word. While these rules accounted for the majority of the most frequently generated abbreviations for each term, the proportion of all abbreviations covered by these rules was only about 0.30 in the two abbreviation production studies. Thus, there is much that people do when abbreviating that is idiosyncratic, variable, and not particularly amenable to rule-based descriptions.

We showed that if abbreviations are formed by applying rules derived from people's behavior, subjects can use these principles. Thus, subjects make use of the internal consistency of the set to learn the abbreviations. Performance with a set of rule-based abbreviations was better than with the corresponding set of most frequently produced abbreviations even when the two abbreviations were identical in the two conditions.

That people extract principles in complex stimuli is not a new finding. For instance Reber,[8,9,10] among others, has demonstrated that people's behavior when learning artificial languages in an experimental setting indicates that they have "learned" the grammatical rules of the language. However, very often they are unable to verbalize these rules

and seem unaware that they have extracted principles. It appears that rule formulation is often an automatic and unintentional process.

Throughout, we have explicitly assumed that rules based on behavior are better than arbitrary rules for abbreviation. One could argue that our own data contradict this assumption. That is, the vowel deletion rule used in Experiments III and IV produced the best performance in two tasks—recalling the abbreviation given the word and conversely recalling the word given the abbreviation. Don't the data then argue that one should abbreviate using vowel deletion? Yes, if the abbreviation is used to recall the term, that is, a decoding task. However, the ability to construct a correct abbreviation, given the word, is only one aspect of a good abbreviation. Another aspect that "abbreviate" itself denotes is that it should be short. Abbreviations produced by vowel deletion increase in length as the length of the source word increases, whereas abbreviations that people spontaneously produce and the ones produced by the rule set better maintain length constancy [e.g., a one-syllable word on the average is abbreviated to 3.4 letters in all conditions, whereas a three-syllable word is abbreviated to 5.2 letters (vowel deletion), 4.3 letters (popular), and 3.3 letters (rule)]. Thus, in terms of total number of keystrokes, the simple vowel deletion rule is least efficient.

A more serious failing for simple vowel deletion is that forming the abbreviation requires first producing the whole word, then systematically deleting the vowels, and finally outputting the remaining consonants. We suspect that this is difficult for long words. When a word is short, the entire word can be dealt with as a "chunk." For a single chunk it is probably not difficult to remove a vowel or two. Beyond simple, short words, the amount of mental bookkeeping increases. If one's goal is to enter command strings *quickly*, vowel deletion is probably not a serious candidate.

Truncation, on the other hand, has properties that make it the best abbreviation mechanism for polysyllabic words. Resulting abbreviations are short and easy to produce—the user does not need to pervert the assumed normal output strategies, just cut them short. It is a simple rule to teach[5–7] and users are most likely to abbreviate multisyllabic words in this way. Thus, it seems to have most of the desirable properties of an ideal abbreviation mechanism—natural, short abbreviations, easy to remember abbreviations, and easy to produce abbreviations. The one thing that may be sacrificed is uniqueness.

There have been suggestions on how to maintain uniqueness of abbreviations, which we now discuss. Many of the recommendations require the user to be flexible and relearn abbreviations from time to time, and some recommendations undermine the consistency of a rule-based set. Ehrenreich and Porcu[7] discuss two ways to resolve conflicts

among abbreviations: (1) use of alternate rules and (2) minimum number of characters to distinguish.

Using an alternate rule when conflicts exist has some serious problems. First, the user must know that there are two or more rules, and that one words most of the time, but when it doesn't, to use the alternate rules. Furthermore, the user has to know which one of the conflicting words requires which rule. The user doesn't know in advance when the primary rule will not work and, consequently, will probably have to learn by trial and error. When the number of exceptions to the primary rule is large, we suspect that the advantages gained by selecting abbreviations by rule will disappear. That is, the mixed rule set performance may approach or may even be worse than performance on abbreviations selected by consensus. However, the degree to which rule-based systems tolerate corruption warrants investigation.

In the case of minimum to distinguish, the user truncates to the number of characters required to differentiate that word for all other admissible words. For example, for "transport" and "transfer," the minimum to distinguish is in each case six characters—"transp" and "transf." There are a few problems worth noting with respect to minimum to distinguish. First, different words require different numbers of letters. Hirsch-Pasek et al.[8] have found in a paired-associates task that minimum to distinguish was more difficult to learn than the other abbreviation schemes they studied; truncation to four letters was the easiest. This may be only a relatively minor problem, which depends on the terms of the command set and the size of the lexicon. It could be dealt with by telling the user the minimum number of letters needed to distinguish all command names. However, a fixed minimum based on all commands will be unnecessarily long for many commands. The user could of course be told that fewer keystrokes will often suffice and learn the minimum to distinguish on a command-by-command basis. The second potential shortcoming occurs when new commands that produce collisions are introduced into the language. Truncations that worked previously are now ambiguous and require relearning. However, introducing software that informs the user of a conflict could solve these problems. If the input is ambiguous, the system could present the user with the collisions and ask the user to select one of the alternatives.

One way to minimize the likelihood of collisions is to make contextual information available to the system. For example, there may exist three words—"debug," "delete," and "define." If collisions were considered on the basis of the entire lexicon, three letters would be needed to differentiate these words. However, if "define" were a command while "debug" and "delete" were command options that occurred in

different environments, it could be the case that a "d" would suffice for either "debug" or "delete." Thus, if the command/option tables are structured hierarchically, such that a command points only to options meaningful in the context of that command, the potential range of conflicts is reduced from the entire command language to a small subset of the language.

In summary, considering all available evidence on producing abbreviations given command terms (encoding), truncation appears to be the best single abbreviation scheme. Truncation also best captures people's natural abbreviations in all environments except two—monosyllablic words and multiple-word terms. In these cases, we recommend using vowel deletion for the former and acronym formation for the latter. If, on the other hand, one's task requires generating full names, given abbreviations (decoding), vowel deletion abbreviations are better than other rule-based schemes.

## REFERENCES


1. G. Zipf, *The Psycho-biology of Language*, Boston: Houghton Mifflin, 1935.
2. M. Hodge and F. M. Pennington, "Some Studies of Word Abbreviation Behavior," J. Exp. Psychol., *98*, No. 2 (1973), pp 350–61.
3. H. Kucera and W. N. Francis, *Computational Analysis of Present-Day American English*, Providence, Rhode Island: Brown University Press, 1967.
4. T. K. Landauer, K. M. Galotti, and S. Hartwell, "Natural command names and initial learning: A study of text editing terms," Commun. ACM, *26*, No. 7 (July 1983).
5. W. H. Rogers and G. Moeller, "Evaluation of Abbreviations in Sonar Displays," 52nd Annual Meeting of the Eastern Psychological Association, New York City, April 24, 1981.
6. K. Hirsh-Pasek, S. Niedelman, and M. L. Schneider, private communication.
7. S. L. Ehrenreich and T. A. Porcu, "Abbreviations for Automated Systems: Teaching Operators the Rules," *Directions in Human Computer Interaction*, A. Badre and B. Shneiderman (Eds), Norwood, NJ: Ablex Publishing Company, 1982, pp. 111–135.
8. A. S. Reber, "Implicit Learning of Artificial Grammars," J Verb. Lear. Verb. Behav., *6* (1967), pp. 855–63.
9. A. S. Reber, "Transfer of Syntactic Structure in Synthetic Languages," J. Exp. Psychol., *81*, No. 1 (1969), pp. 115–9.
10. Reber, A. S., "Implicit Learning of Synthetic Language: The Role of Instructional Set," J. Exp. Psych., Human Learning and Memory, 2, No. 1 (1976), pp. 88–94.


## AUTHORS


**Lynn A. Streeter,** B.A. (Psychology), 1969, University of Michigan; Ph.D. (Psychology), 1974, Columbia University; Bell Laboratories, 1969—. Ms. Streeter first joined Bell Laboratories in 1969 as a member of the then Human Factors Department in Holmdel. She rejoined Bell Laboratories in 1974 as a member of the Linguistics and Speech Analysis Department. The abbreviation data were collected when she was Supervisor of the Human Factors Engineering Group for the Advanced Communications Service project. Currently, she is working on mobile telephony interface design problems in the Communications Methods Research Department.

**John M. Ackroff,** B.A. (Psychology), 1971, Williams College; M.S. (Experimental Psychology) 1975, Ph.D. (Experimental Psychology) 1981, University

of Wisconsin–Milwaukee; Bell Laboratories, 1977–1982; American Bell, 1982—. While at Bell Laboratories, Mr. Ackroff worked on the Advanced Communications Service project, where he was responsible for the design and specification of the command language aspects of the user interface. At American Bell, he is responsible for developing an enhanced version of the command language parser.

**Glen A. Taylor,** B.S. (Psychology), 1969, Louisiana State University; M.A. (Experimental Psychology), 1975, Ph.D. (Experimental Psychology), 1977, University of Kansas; Bell Laboratories, 1979–1982; American Bell, 1982—. Mr. Taylor joined Bell Laboratories as a member of the Human Factors Engineering Group for the Advanced Communications Service project (now AIS/Net 1000 service). He was responsible for designing the user interface for the Net 1000 editor and for investigating various aspects of computerized electronic mail services. In 1981, he became a member of the User/Station Interface Software Group responsible for the development of on-line documentation retrieval software.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Designing and Evaluating Standard Instructions for Public Telephones

By C. J. KARHAN,* C. A. RILEY,† and M. S. SCHOEFFLER†

(Manuscript received July 30, 1982)

Difficulty finding and understanding information on public telephone instructions has led to problems using public telephones. To overcome these problems a new instruction card was designed. In this paper we discuss the information needs of the public telephone user, a conceptual solution to information presentation, a new standard instruction card design, and laboratory and field evaluations of the new design.

## I. INTRODUCTION

Many people have reported that placing a call from a public (pay) telephone is a problem. Individuals and consumer groups have complained to their telephone companies and even their public utility commissions. A major reason for this problem is the difficulty of finding and using information on public telephone instruction cards. In this paper we describe a series of studies conducted to develop standard instructions for public telephones that are easier to use.

### *1.1 The public telephone operating environment*

If all public telephones worked the same way, the task of designing

* Bell Laboratories. † American Bell.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

instructions would be easy. However, there is considerable variation in the operating procedures of public telephones in the Bell System. In addition, new public telephone services are being introduced or will be introduced soon. For example, at some public telephones the user must deposit coins equal to the cost of a local call before dialing any call (coin-first telephones). From the user's perspective, the coins work like a switch to turn on the telephone, and they are returned if no payment is required for the call. At other telephones no coins are needed to turn on the telephone, and a large class of calls, such as collect, credit card, emergency, and operator assistance, can be dialed without depositing any coins (dial-tone-first telephones). On some other telephones coins cannot be used at all (Charge-a-Call telephones). At most telephones, coins are deposited before a local call is dialed, but at some telephones coins are deposited after the person at the other end answers (post-pay telephones). From some telephones you can dial long distance calls; from others the operator must dial for you. From some telephones you can dial in your credit card number; from others you must tell your credit card number to the operator. In some areas you must dial "1" before long distance station-to-station calls; others require no prefix. Furthermore, there are many variations in the numbers for emergency help, directory assistance, and repair service.

Because of these variations, no single instruction card can be used for all public telephones. More importantly, because of these variations, even the most experienced public telephone users must sometimes consult the instructions to place a call correctly.

### *1.2 The need for standard instructions*

Prior to the standard developed from these studies, there was no standard for instruction card design in the Bell System. Various operating telephone companies have used different layouts, word choices, and color schemes to convey the same information. There are also variations within single telephone companies on the format and type of information included. Generally, telephones that operate differently, if only in minor respects, have instructions that vary in many ways that have nothing to do with operational differences. For example, instructions on how to place a long distance call may be different on a telephone where the initial deposit is 10 cents than on one where a 20-cent deposit is required, even though the dialing procedures are identical. There are also tremendous differences in the emphasis (location, color, and type size) placed on particular pieces of information. Finally, information is often provided in "telephonese."

The problem for users is not that all instruction cards are poorly

designed. While instruction cards in some locations may be particularly difficult to read and understand, the biggest problem for users is the inconsistency of design.

The use of a standard instruction card design for all public telephones should significantly reduce the problems of finding and understanding needed information. While identical instructions cannot be placed on all telephones because of differences in operating procedures, consistent language, graphics, and placement of information can be used. The information provided on the standard instruction card should match the information needs of the user. The instructions must be understandable, and the design should be attractive. A good design will help users; and, given a good design, a greater benefit for users will come from the universal application of this design.

### *1.3 Approach to the problem*

The new instruction card design and the studies that supported the development of that design are described in the sections that follow. First, we studied public telephone users to identify their information needs, how they used instructions, and the telephone terminology they used and understood. Based on this information we formed a model of the instruction user. Second, we developed a conceptual solution to the problem. Third, we presented this solution (along with detailed information about public telephone operating procedures) to graphic designers at Henry Dreyfuss Associates, who created designs for the instruction cards. Fourth, we evaluated and refined the designs in several phases. First we used our own and the designer's judgment, and then conducted a laboratory test, and, finally, a field test. During all phases of the project we consulted with the AT&T Public Services Marketing organization.

## II. THE INFORMATION USER

Before any attempts were made to design new instruction cards, we analyzed the information needs of the public telephone user.

### *2.1 Interview study*

A total of 500 users were interviewed in Arizona, Florida, and New Jersey. The individuals interviewed were approached following the completion of a call from a public telephone. They were asked to participate in a study of public telephone service. The individuals who agreed to participate were asked a series of questions about instruction use, such as: "Did you read the instruction card?" "Did you look at the instruction card?" "What information did you need?" "Did you find the information you needed?" They also were asked if they had

any difficulty finding or understanding information, and were asked to define several telephone terms, such as "collect call" and "dial-tone-first telephone."

Few of the people interviewed (about 10 percent) reported that they read the instruction cards. These individuals looked for procedural information necessary to place a specific call. This information included the cost of the initial deposit, when to deposit money, how to dial local calls and various types of long distance calls, and specific numbers, such as directory assistance or repair service.

Some respondents reported that they were unable to find information that they needed on the cards. In almost all of these cases, the individual was looking for more specific information than could ever be placed on the instruction card, such as the area code for Los Angeles, or the cost of a particular long distance call, or the telephone number of the local veterinarian.

Our interviews showed that people generally approach the instructions with particular questions in mind, and that the questions have two general forms. The first form is one in which a specific fact or procedure is needed. Examples are "How much does a local call cost?" or "How do I place a collect call?" The second form of question is asked by the more sophisticated telephone user who is already familiar with alternative operating procedures. This user might ask, "I know there are two kinds of telephones, one where I can dial long distance calls myself, and one where the operator dials all long distance calls; which type of telephone is this one?"

A few people look at the instructions without needing any specific information. These users may also be characterized, however, as having a question in mind before reading (looking at) the instructions, which can be phrased as, "Is there anything (new) here that I need to know about?"

Interview questions about terminology showed that telephone users generally understand terms referring to types of calls, such as collect, credit card, person-to-person, and station-to-station. However, language on instruction cards referring to the type of telephone, such as "dial-tone-first" telephone, was not understood.

## III. THE CONCEPTUAL SOLUTION

Most people do not read instructions on telephones; in fact, there are few circumstances in which anyone reads any instructions at all. (See Ref. 1 for a discussion of this issue.) Our goal was not to induce more people to read the instructions, but rather to make the instructions easier and more effective when they were read.

Our conceptual solution to instruction card design had several

components to simplify searching and understanding. The design should:

1. Provide the information that people need in words they understand.

2. Always place information about a particular aspect of a telephone's operating procedure in the same spatial location on the card. The card is thus like a form, with fields for each type of information. If there is no relevant information to be placed in the field, it should be left blank and not used for some other information.

3. Place frequently needed information on the upper card and detailed procedural information on the lower card.

4. Divide detailed information into related categories. Use the category labels to construct an index.

5. Provide visually obvious cues that are redundant with the text, e.g., colors, pictograms, and other symbols. This information should be visible at a glance and allow knowledgeable telephone users to answer the question "What type of telephone is this?" without reading the text. By systematically varying the visual characteristics of the instruction cards, telephones that have different operating procedures will have instructions that *look* different. Instruction cards on all telephones that operate the same way will look the same. Thus, information obtained at a glance can be informative. If the card "looks" different from cards the user is familiar with, then the instructions should be read.

6. As a final goal provide as much information as possible in a language-independent form to make the cards more usable by people who either do not read English or do so with difficulty.

## IV. THE NEW STANDARD DESIGN

Figures 1 to 3 show examples of the new instruction cards, designed according to the principles described in the section above, and revised based on the results of the study described in the following sections. (Space constraints prohibit illustration of the cards through all the design iterations.) Figure 1 is an instruction card for a telephone that requires a coin deposit before any calls can be dialed, and from which operator assistance is required for all long distance calls. Figure 2 is for a telephone from which credit card, collect, emergency, and other "free" calls can be made without depositing any coins; long distance calls can be dialed by the caller. Figure 3 shows an instruction card from a "Charge-a-Call" telephone, from which no coin-paid calls can be made.

### *4.1 Design features*

The instructions shown in Figs. 1 through 3 are for telephones that

**Local calls** Deposit 10¢ before dialing
**Long Distance** Dial 0
**Operator** will handle all Long Distance calls

**SOS dial 0** for Emergency help

KEENE N H EXP 5

| | | |
|---|---|---|
| **Charge and Person-to-Person** | **Credit Card, Collect & Person-to-Person calls** | Operator |
| **Station-to-Station calls** | Local numbers beginning with: 239, 242, 352, 363, 399, 446, 563, 585, 756, 827, 835, 847, 876 | Number |
| | All other numbers | Operator |
| **Free calls** | **Directory Assistance** | Operator |
| | **Toll Free 800 Numbers** | Operator |

AREA TYPE 2 EXP 5 — Operator assisted rates apply to all toll calls from this telephone.

Fig. 1—Instruction card set for a telephone that requires a coin deposit before any call can be dialed, and from which the operator dials all long distance calls. Colored areas are tan.

differ considerably in operation; however, the common organization of the instructions is easily observed. Differences in visual appearance are also readily apparent. The features of the design are listed below:

1. Color: All instructions are printed on a white background; detailed instructions are in black type and emergency information is in red. Other colored sections—including the band across the top of the upper card, the "0+" and "1+" symbols, index headings and rules on the lower card—are color coded based on the basic operational procedure of the telephone. Telephones that require a coin deposit before any call can be dialed (coin-first) have tan instruction card coloring; telephones that do not require a coin for initial activation (dial-tone-first, including post-pay and Charge-a-Call) have blue instruction card coloring.

2. Upper-left symbols: Coin-first telephones have a white handset in a tan disk (coin); dial-tone-first telephones have a blue handset on a white background. The handset on the Charge-a-Call telephone has

**Local calls** Deposit 20¢ before dialing
**Long Distance** Dial all calls directly
0+ needed for Charge & Person-to-Person calls
1+ needed for Station-to-Station and Free calls

**SOS dial 0** for Emergency help
**SOS marque 0** para Emergencia

TSPS/DTF UPPER-EXP - 11

| | | |
|---|---|---|
| **Charge and Person-to-Person calls** | **Credit Card, Collect & Person-to-Person** | |
| | Within this Area Code | 0+ Number |
| | Outside this Area Code | 0+ Area Code + Number |
| **Station-to-Station calls** | Within this Area Code | 1+ Number |
| | Outside this Area Code | 1+ Area Code + Number |
| **Free calls** | **Directory Assistance** | |
| | Local | 411 |
| | Within this Area Code | 1+ 555-1212 |
| | Outside this Area Code | 1+ Area Code + 555-1212 |
| | **Repair Service** | 611 |
| | **Toll Free 800 Numbers** | 1+ 800 + Number |

TSPS/DTF STERLING LOWER-EXP - 11 Operator assisted rates apply to all toll calls from this telephone.

Fig. 2—Instruction card set for a telephone from which credit card, collect, emergency, and other "free" calls can be made without depositing any coins, and from which long distance calls can be dailed by the caller.

a broken horizontal stripe, representing credit card and speed. The functional meaning of these symbols is explained in the text immediately to the right of the symbol.

3. Dial tone/deposit sequence pictograms: The pictograms in the upper right corner illustrate the sequence for listening for dial tone (depicted as a handset with waveform) and coin deposit. Numbers, as well as spatial position, show the sequence. The post-pay instruction card (not shown) has the additional modifier "Deposit only after answer" between the handset and the coin deposit. The disk representing the coin also contains the initial deposit rate (cost of a local call).

4. Summary instructions: Summary instructions for placing local and long distance calls are located directly below the colored band (for the two-card design). In the long distance case, these instructions are intended to be sufficient for the knowledgeable user to identify the

Fig. 3—Instruction card from a "Charge-a-Call" telephone.

general procedure. The "0+" and "1+" represent the prefixes required before dialing certain types of calls, when needed. The colored word "Operator" indicates that long distance calls cannot be dialed by the caller. Experienced users should be able to recognize the symbols at a glance, and identify the procedure without reading the instructions.

5. Emergency: "SOS" is the internationally recognized emergency symbol, and it is used to identify the number to dial for emergency assistance. Its red color makes it very salient. The use of a unique identifying color makes possible useful public relations messages in non-English-speaking communities. For example, the Chinese press in New York could carry a message stating (in Chinese) "If you need emergency help, dial the red number." Where dual language instructions are required by regulation, the SOS instruction can be repeated in the second language. Where separate numbers for fire, police, and ambulance are required, the SOS is supplemented with specific symbols (flames, a shield, and a cross) paired with each number.

6. Local instructions: The space directly under the dial tone/deposit symbols is reserved for locally varying instructions. Operating companies may use this space for special instructions or to publicize a new service.

7. Lower card: Specific dialing instructions and service numbers are listed on the lower instruction card. The major sections are divided by a colored rule, and each section has an index heading to simplify searching.

### *4.2 Preliminary design differences*

The instructions shown in Figs. 1 to 3 differ in several respects from the instructions used in the laboratory evaluation. Changes were made

based on the findings of that study, as we discuss in Section V. Briefly, the earlier designs differed from the new designs as follows: (1) There was no summary instruction for local calls; (2) the word "Operator" was not in color on the upper card for operator-dialed long distance calls; (3) all operator-dialed calls were combined into a category called "All other calls" on the lower instruction card; and (4) the local directory assistance number was listed in the same line as the heading "Directory Assistance" when that number was different from the "Within this area code" number. In addition, the handset with dial-tone pictogram and Charge-a-Call symbol were modified between the laboratory test phase and the field test.

## V. LABORATORY EVALUATION

### *5.1 Study goals*

The instruction cards were designed to satisfy two major goals. First, users should be able to search successfully and efficiently for the information they need. Second, knowledgeable users should be able to tell at a glance what services and procedures are available on a given telephone. The major purpose of the laboratory evaluation was to determine whether these design goals were met. We used an information search task to assess the new designs and identify needed improvements. The following questions were addressed:

1. Could users find the answers to questions about telephone operation?
2. Did users understand the information provided?
3. Were users able to find information quickly after some experience using standard instructions?
4. Did users understand the symbols and pictograms used on the instructions?
5. Did users infer the meaning of the redundant visual codes (e.g., color) that would allow them to get information at a glance?

In addition, several different formats were evaluated.

### *5.2 Method*

#### *5.2.1 Materials*

Three formats for the upper instruction card and two formats for the lower card were used. The same content and design elements described in the preceding section were used in each format. The only differences between the formats were in the arrangement of the information. The formats of the three upper cards were:

1. Horizontal: the design illustrated in Figs. 1 through 3.
2. Two-panel: the information contained in the stripe on the horizontal design was placed in a square patch on the left third of the

card; the other information was listed on the remaining two thirds of the card.

3. Three-panel: the left third was the same as in the two-panel design; the remaining space was divided into two equal sections by a blue or tan vertical bar. The center section contained long distance dialing and local information, and the right-hand section contained emergency information.

The formats of the two lower cards were:

1. Ruled: illustrated in Figs. 1 through 3.

2. Unruled: the colored lines separating categories were removed, the colored headings were placed directly above the listings for that category, and the listings were centered on the card.

Instructions for eight different telephones were created. The operating procedures and available services varied on the eight telephones. Three of the telephones had coin-first service, two had dial-tone-first, one had dial-tone-first/post-pay, and two had Charge-a-Call. Other operating differences, such as initial deposit rate, long distance dialing procedures, emergency numbers, directory assistance numbers (local and long distance), and repair numbers were systematically varied across the eight telephones. Only combinations of services and procedures that actually exist in the operating telephone environment were used. A set of instruction cards was constructed in each of the formats described above for each of the eight telephones. (For this study, upper and lower housing cards were designed for Charge-a-Call.)

#### *5.2.2 Experimental groups*

Seventy-two people (six groups of 12) participated in the study. Six conditions were formed by combining each set of upper instruction cards with each set of lower cards. Each group of participants saw cards from only one condition.

#### *5.2.3 Participants*

Participants were recruited from the Holmdel, New Jersey, vicinity. The 67 female and five male participants had a median age of 34 years, and a median education of one year of college. They were paid to participate.

#### *5.2.4 Procedure*

An information search task was used. Participants in the study were asked to answer many questions about the operating procedures of telephones, based on information obtained from the instructions. Errors in responding to questions indicated that information was not

found or understood. Response times to questions were measured to assess the efficiency of information search.

After the series of main questions were answered, participants were asked to answer a second set of questions. The instruction cards shown in this series had all text removed. Thus, the second set of questions measured incidental learning of the meaning of colors, symbols, and pictograms that occurred in the first part of the experiment. Finally, participants were asked to complete a questionnaire about overall ease of use and appearance of the instructions, as well as further questions about the colors, symbols, and pictograms.

The participants sat at a table that contained a response panel and a 50-cm-square rear projection screen. The viewing distance was two feet, approximately the viewing distance when reading instructions on a public telephone. The response panel contained four buttons, labeled "Ready," "Yes," "No," and "Don't Know." Participants were instructed to place their index fingers on the "Yes" and "No" buttons, and a thumb on "Ready." They were told to imagine that they were traveling, and that they would need to make numerous telephone calls from public telephones. Since telephones have different operating procedures, they would need to use the instructions on the telephones to answer questions before placing the calls.

Each trial began with the presentation of a slide containing a question, such as "Does a local call cost 20 cents?" The participants were told to study the question until they understood and remembered it, and then push the "Ready" button. Pushing the button started a sequence that turned off the question slide, and turned on a slide containing a picture of a telephone with instruction cards on it. The pictured telephone and instruction cards were shown at actual size. Participants were told to look for the answer to the question they had just seen on the instructions, and answer either "Yes" or "No." They were told that all answers were available in the instructions, but if they could not find it to answer "Don't Know."

The experiment was run under computer control; responses and response times were recorded. Response times were measured in milliseconds from the time the picture appeared on the screen until the response button was pressed. There were 96 trials separated by a 2-second intertrial interval, with a 1-minute break between blocks of 24 trials.

#### *5.2.5 Structure of the question series*

The questions came from eleven categories: cost of local calls, dial tone/deposit sequence, types of calls that require coin deposit, emergency numbers, long distance calls, calls requiring a "1" prefix, dialing of credit card numbers, possibility of coin use, directory assistance

numbers, repair service numbers, and the address of the telephone (provided on the experimental instruction cards). The 96 trial series contained eight questions from each category except emergency numbers and dial tone/deposit sequence, from which there were 12 questions. For half of the questions the correct answer was "Yes." Questions from each category were paired with each of the eight telephones, except where inappropriate (e.g., no questions about the dial tone/deposit sequence were asked about Charge-a-Call telephones). Question order and left-right position of the "Yes" and "No" buttons were counterbalanced.

Following the 96 questions, 16 questions were presented using the cards with text removed. Participants were told to try to figure out the answers based on the information remaining. Questions were from four categories, emergency numbers, types of calls requiring coin deposit, long distance dialing, and possibility of coin use.

### *5.3 Results*

#### *5.3.1 Main question series*

Mean response times were computed for each question category within each trial block (24 questions). An analysis of variance was done on these mean response times. Factors in the analysis were Upper Card Format, Lower Card Format, Trial Block, and Question Category.

No statistically significant ($p < 0.05$) differences in response times were observed between formats. The main effect of Trial Block was statistically significant ($p < 0.001$). Mean response times for trial blocks 1, 2, 3, and 4 were 13.4, 7.9, 6.3, and 5.8 seconds, respectively. Post hoc comparisons of the means showed that block 1 > block 2 = block 3 = block 4. Thus, participants responded more quickly to later questions, which indicates that they could find information more quickly once they had some experience using the cards.

The main effect of Question Category and the Question Category by Trial Block interaction were also statistically significant ($p < 0.001$). Mean response times differed widely across the different question categories. Slow response times, especially later in the question series, are an indication that users may be having difficulty finding or understanding information. In trial blocks 2 through 4, mean response times were relatively short (less than 6 seconds) for emergency numbers, possibility of coin use, repair numbers, cost of local call, and address; response times were slower (8 to 11 seconds) for "1" prefix for long distance, calls requiring coins, long distance dialing, dial tone/deposit sequence, directory assistance, and availability of credit card number dialing. Possible sources of difficulty will be discussed below.

The total number of errors in each trial block and in each question

category was computed for every subject. Both incorrect and "Don't Know" responses were counted as errors. The overall error percentage was 11.4 percent.

An Upper Card Format by Lower Card Format by Trial Block analysis of variance was done on the error data. Only the main effect of Trial Block was statistically significant. There were more errors in the first trial block (15.4 percent) than in the last three trial blocks (9.8 percent). The decrease shows the same facilitative effect of experience that was seen in the response time data.

An Upper Card Format by Lower Card Format by Question Category analysis of variance again showed only the main effect of Question Category statistically significant ($p < 0.001$). The pattern of errors is similar to the pattern of response times. The correlation between mean error rates and mean response times in the 11 categories was high, $r = 0.77$. High error rates are also an indication that users are having difficulty finding or understanding information on the instruction cards.

#### *5.3.2 Incidental learning question series*

The instruction cards used in this question series contained only the information that appears in color on the instruction cards. The purpose of this question series was to determine whether subjects had learned the meaning of this information during the main question series. All of the questions could be answered with the information available.

The total number of errors and mean response times were computed for each of the four question categories for every subject. An Upper Card Format by Lower Card Format by Question Category analysis of variance was done on the error data and on the response time data. In both analyses only the main effect of Question Category reached statistical significance. No differences between card formats were observed.

The questions on long distance calling ("Can you dial your own long distance calls?" or "Does the operator dial all long distance calls?") caused the most difficulty. The percentage of errors (58 percent) does not differ from the performance expected from guessing. The long distance category also elicited the highest percentage of "Don't Know" responses (7 percent). Clearly, the participants did not learn that the presence of the "0+" and sometimes "1+" symbols on the upper card indicated that long distance calls could be dialed, and that the absence of "0+" meant that the operator dials all long distance calls.

Performance on the other question categories was substantially better than what would be obtained by guessing, showing that some incidental learning had occurred. However, questions on the types of

calls that required coin deposit did have a fairly high (15 percent) error rate. The answers to these questions could be inferred from either the color of the card (blue or tan) or from the dial tone/deposit sequence.

The "SOS" emergency symbol was well understood. Responses were fast and few errors were made.

#### *5.3.3 Questionnaire results*

The questionnaire contained four rating-scale items. The instructions were rated on the ease of finding needed information, the ease of understanding the words in the instructions, the overall ease of understanding the instructions, and the attractiveness of the instructions. The responses were scored from 0 to 5, where 5 was the most positive response. An Upper Card Format by Lower Card Format analysis of variance was done on the ratings for each item. No statistically significant differences between formats were found for any item. Overall, participants rated the ease of finding needed information between "somewhat easy" and "easy" (mean rating 3.6), the ease of understanding words between "easy" and "very easy" (4.3), the overall ease of understanding between "somewhat easy" and "easy" (3.7), and the attractiveness between "somewhat attractive" and "attractive" (3.6).

When asked if any information seemed to be missing from the instructions, the majority of participants responded negatively (67 percent). Of the people responding affirmatively, most of the comments were about the lack of information on placing local calls or about the dial tone/deposit sequence. A few general comments were made, such as "Instructions too wordy," or "Instructions not detailed enough." Similarly, the majority of participants did not report any specific problems understanding the instructions in response to an open-ended probe. The problems that were reported were also about the lack of information on local calls, and confusion about when to deposit coins. In addition, some people were confused about the "0+" and "1+" symbols.

The majority of participants gave no suggestions for improving the instructions. Some general suggestions were made, such as "Make the instructions simpler," or "Make the instructions easier to understand." A few participants commented on the small size of the print. Most suggestions, however, concerned the variety of procedures described on the instructions. The most frequent comment was "Make the instructions the same on all telephones." The nature of the comments made by these participants implied that some thought the operating procedures were determined by the instructions, rather than the reverse.

In the last section of the questionnaire the participants were asked to explain the meaning of the colors and symbols. Most of the participants did not learn the meaning of the blue and tan colors but did learn the meaning of the Charge-a-Call symbol. Ninety-six percent of the participants correctly explained the meaning of "SOS," and most of these recalled that the "SOS" symbol was red.

The "0+" and "1+" on the upper instruction card were generally explained as "Dial 0 (1) and then the number." However, a third of the participants said that "0+" meant "Dial the operator" or "Operator dials long distance." A number of the participants who said that "1+" was for direct-dialed long distance also stated that "0+" was for operator-dialed long distance. Certainly these participants had not learned the distinction between 0+ and 1+ dialing.

The symbol sequence for the dial tone/dial/deposit coin sequence was confusing in some cases. The symbol sequence for coin-first operation was understood by 79 percent of the participants. In contrast, there was some confusion about the meaning of the dial-tone-first symbols. Fifty-four percent of the respondents correctly answered either "Dial tone first" or "No coins needed for dial tone." However, a third of the participants confused the typical dial-tone-first case (deposit coins before dialing local calls) with the post-pay procedure. This pattern of responses is consistent with the general confusion over the dial tone/coin deposit/dialing procedures observed in the experiment.

### *5.4 Discussion of problems and recommendations for improvement*

The major findings in this study relate to the content of the instructions, not the formats.

The highest overall error rate in the main question series occurred on questions about directory assistance numbers. There were two major sources of difficulty. First, the listing for local directory assistance appeared in the same line as the heading, and was hard to find. Since no separate listing for "Local" was made on some of the instructions, the participants tended to infer that the number for "Within this Area Code" was always the local number. This problem was subsequently remedied by adding a separate line for "Local" on the lower instruction card. The other confusion has no obvious solution. The participants expressed confusion over the need for separate listings for "Local" and "Within this Area Code." The confusion may stem in part from the fact that all participants were from New Jersey, where only a single number is used for directory assistance within an area code.

The dial tone/deposit sequence questions contained two subsets of

questions. One subset contained questions about the temporal sequence of hearing the dial tone and depositing a coin, the other about the temporal sequence of depositing a coin and dialing a call. The problems occurred in the latter subset. Forty-eight percent of the participants responded that coins were deposited after answer for the typical (pre-pay) dial-tone-first case. Some people appeared to equate the pre-pay and post-pay dial-tone-first telephones. The post-pay telephone is uncommon (except in a few areas outside of New Jersey), so participants may have been confused about the two types of telephones. The cards tested in this experiment contained no written instructions on how to place local calls. Based on the difficulty seen in this area, some statement about when to deposit coins on local calls appeared warranted.

The same confusion appeared to affect the results in the category of calls requiring coin deposit. Most of the errors occurred on the question "Do you need to deposit any coins to make a local call?" Thirty-eight percent said "No." The additional written instruction described above should also help this problem.

The questions about long distance calling can also be divided into two subsets, one about whether customers can dial their own long distance calls, and the other about the specific digits to dial for long distance calls. Most errors occurred in the latter category, on telephones from which long distance calls were dialed by the operator. These instructions had no specific instructions for long distance on the lower card, just a listing for "All other calls" (Dial the Operator). Some improvement should be obtained if these instructions contained all categories, even though the instruction in every case is "Dial the Operator."

On the incidental learning question series, participants were asked whether long distance calls could be dialed. The answers to these questions had to be inferred from the "0+" and "1+" symbols. If these symbols were present, then long distance calls could be dialed; the absence of a symbol implied that the operator dialed the calls. The problem may have occurred because this upper card information was never used in the main question series. However, the problem may have occurred because of the lack of a positive indicator for operator-dialed long distance. The subsequent addition of such a cue should make the information on the upper card clearer.

Errors were made on 19 percent of the questions about the need for a 1 prefix on direct-dialed station-to-station long distance calls. These errors are hard to interpret since the prefix is not used in New Jersey. The other category that caused some confusion was about customer-dialed credit card numbers. As in the case of the 1 prefix, this information was explicitly stated on the lower instruction card when

appropriate; however, the new procedure was not familiar to New Jersey residents.

In summary, the improvements recommended were:

1. Add a separate line for "Local" directory assistance.
2. Add a written instruction on when to deposit coins for local calls on the upper card; the heading should be "Local calls" and be placed above "Long distance calls."
3. Add a symbol for "Operator dialed long distance" to contrast with the "0+" and "1+" symbols in the long distance section of the upper card.
4. Remove the heading "All other calls" from the lower card and replace it with the same headings used on the other cards "Charge and Person-to-person calls" and "Station-to-station calls."

These changes were incorporated into the instructions used in the subsequent field evaluation. The choice of format (horizontal and ruled) used in the final design was made by AT&T Public Services Marketing and the designer.

## VI. FIELD EVALUATION

The major purpose of the field study was to assure that the instruction card design, modified based on laboratory results, was acceptable to telephone users, and that telephone users could use the instructions to find needed information. Because the overall success of the new instructions depends on the use of standard instructions throughout the Bell System, we could not expect to show all the advantages of the new instructions.

### *6.1 Field trial sites*

The field trial of the proposed standard instruction cards was conducted in Illinois Bell Telephone (IBT) Company and New York Telephone (NYT) Company at three sites in each company: a major city airport, a large suburban shopping center, and a small, self-contained city. The sites were selected so that a variety of public telephone users would be exposed to the trial instructions: travelers, local residents, local callers, long distance callers, credit card callers, etc. In addition, a second shopping center was selected in each company. The new cards were not installed at the second shopping center.

A broad range of services and procedures was available at these sites. Among the variations in operating procedures were telephones requiring an initial deposit for all calls, telephones requiring a deposit only for coin-paid calls, and Charge-a-Call telephones. At some locations callers could dial their own long distance calls, at others operator assistance was required. At some locations a "1" prefix was required on calls to other area codes; on others it was not required.

### 6.2 Instruction card content

The information on the trial cards was the same as the information on the instructions they replaced, although some changes were made to conform to the new standards. The changes mainly involved the removal of specific place names. For example, on the existing instructions at Roosevelt Field shopping center in NYT, long distance dialing instructions were given for "Nassau and Suffolk" and "All other places." These specific names were replaced by the general, but equivalent, instructions for "Within this Area Code" and "Outside this Area Code."

### 6.3 Interview study

Interviews were conducted at selected public telephones in December 1979 (three months after card installation in New York, two months after installation in Illinois). People who had just used a public telephone were interviewed face-to-face by professional interviewers. The interviewer asked a series of questions about the call just made and the use of the instructions, then showed instruction cards identical to the ones on the telephone just used and asked a series of questions about how to place various types of calls. The interviews addressed the following areas:

—Did people look at the cards?
—What information was needed?
—Was the information easy to find?
—Was the information easy to understand?
—Could people find and report dialing instructions correctly for various types of calls?
—What was the overall judgment of the appearance of the card?
—Could people understand the pictograms used on the cards?
—Did people understand the meaning of "SOS?"

A total of 650 interviews were conducted at trial locations, 100 at each of the trial sites, plus an additional 50 interviews at Charge-a-Call telephones at O'Hare International airport. Two hundred additional interviews were conducted at the two shopping centers where the original instruction cards had been left in place. The locations provided a comparison of telephone users' acceptance of and problems using the trial cards. Since the advantage of the new cards depends upon their use in all locations, and since the old cards were familiar to the users, this comparison is not intended to be the proof that the new cards are better.

### 6.4 Interview results

For the following discussion, the normal approximation to the

difference between two observed and independent proportions was used to test the significance of the differences between results from the trial and nontrial shopping centers.

Less than 25 percent of the people interviewed reported looking at or reading the instructions cards at any location except at the Charge-a-Call telephones, where 63 percent of those interviewed reported reading the instructions. Charge-a-Call telephones look different from regular coin telephones; 88 percent of the Charge-a-Call users who read the instructions reported doing so to find out how to use the telephone, or because the telephone looked new or different. No significant differences were found between the number of interviewees who looked at or read the instructions at the trial and nontrial shopping centers.

The people interviewed reported needing a variety of types of information. The types of information requested were: how to use the telephone; how much money to deposit; how to make collect, credit card, or person-to-person calls; how to dial long distance; and what numbers to dial for service calls. Ninety percent of those interviewed who reported reading the instructions said that the information they needed was easy to find; 94 percent reported that the information was easy to understand. No significant differences were found between the trial and nontrial shopping centers.

Interviewees were asked to use a set of instructions cards identical to the cards on the telephone they had just used to find and report dialing instructions for the following types of calls: police emergency, local directory assistance, collect, and coin-paid station-to-station long distance calls. In all cases, most of those interviewed were able to correctly report the dialing procedures. Most of the errors occurred when the interviewee reported a dialing procedure that would be correct in other settings, for example, reporting "911" as the police emergency number at a location where "0" was the correct response. This type of error was made in both trial and nontrial locations. No significant problems in finding or understanding dialing instructions were identified with the new instructions. The interviewees had no trouble in finding the emergency information associated with the "SOS" symbol.

Pictograms were used on the upper instruction card to represent the dial tone/deposit sequence. The numerals "1." and "2." were used to indicate the sequence.* A coin-like disk indicated both the action of depositing and the amount of the local call deposit. A handset with an

* A separate study conducted by M. L. Viets reported that sequential steps should be indicated by numbers followed by a dot. Omitting the dot led people to interpret 1 and 2 as magnitudes rather than as ordinal steps in a sequence.

expanding waveform indicated the concept "listen for dial tone." Overall, a high percentage of interviewees (88 percent) were able to provide a correct or partially correct meaning for the dial tone/deposit pictograms. Thirty-nine percent gave correct responses that included the correct meaning of both pictograms and the correct action sequence. The remainder gave responses that included correct meanings for some of the elements.

Interviewees were asked to rate the overall appearance of the instruction cards on a six-point scale ranging from very attractive (a rating of 1) to very unattractive (a rating of 6). Overall, the trial cards were rated attractive; the median response was 2 at all locations except the New York shopping center, where the median response was 3 (somewhat attractive). The median rating of the cards at the Illinois nontrial location was 3; at the New York nontrial location the median response was 2.

Interviewees were also asked an open-ended question about their overall reaction to the instruction cards. Most of the comments were positive; however, about 20 percent of the respondents had some criticism of the instructions. The most common complaints were that the print was too small, the appearance of the card was cluttered, and the color unattractive. Overall, 9 percent of the trial location respondents complained about the print size (10-point type). In Illinois, the nontrial cards were dual-language cards. All instructions were printed in both English and Spanish. This required the use of smaller print (8-point type) than on a single-language card. Thirteen percent of the respondents at the Illinois nontrial location complained about the size of the print, and 13 percent more complained about the use of Spanish, indicating that it made the instructions harder to read. Overall, 6 percent of the trial location respondents complained that the cards were cluttered (too much information); however, less than 1 percent of the airport respondents had this complaint, compared to 11 percent in the small cities and shopping centers. Generally, public telephone users in the cities and shopping centers made local calls, and had little need for most of the information on the instruction cards. Seven percent of those who saw tan cards complained about the color of the cards; less than 1 percent disliked the blue cards.

### *6.5 Field trial conclusions*

The proposed standard public telephone instruction cards were acceptable to users. They were rated as easy to use. Telephone users were able to use the trial instructions at least as well as the nontrial instructions, and no major problems were detected. The new card establishes a standard that satisfies users' needs.

## VII. CONCLUSIONS

Based on the results of the laboratory evaluation and on the reception of the new instructions cards in the field, AT&T Public Services recommended to all Bell Operating Companies (BOCs) that the new instruction cards be used on all public telephones. The new cards have been introduced in most BOCs. In addition, the new design was made available to the Independent Telephone Companies for use in preparing their instruction cards. We anticipate that the widespread use of the new standard instructions will make public telephones easier to use.

## REFERENCES

1. P. Wright, "Feeding the information eaters: suggestions for integrating pure and applied research on language comprehension," Instructional Science, 7 (July 1979), pp. 249–312.

## AUTHORS

**Christine A. Riley,** A.B., (Applied Mathematics and Psychology), 1971, Brown University; Ph.D. (Psychology), 1975, Princeton University; University of Iowa, 1975–1977; Bell Laboratories, 1977–1982; American Bell, 1982—. At the University of Iowa Ms. Riley was an Assistant Professor of Psychology. In 1977 she began working for Bell Laboratories, and in July 1982 moved to American Bell Research and Development. She is currently a Supervisor in the Data Network Architecture Department. She has worked on a variety of human interface design issues, including the design of instructions for telephone services and the design of human-computer interfaces. She is currently supervising a group responsible for the user interface to AIS/Net 1000 Service.

**Max S. Schoeffler,** B.S. (Psychology), 1950, Rutgers University; M.A. (Psychology), 1953, Indiana University; M.A. (Mathematics), 1958, University of Michigan; Ph.D. (Psychology), 1954, Indiana University; University of Michigan, 1956–1960; Bell Laboratories, 1960–1982; American Bell, 1983—. At Bell Laboratories Mr. Schoeffler worked in the areas of human interface design for operator service systems; effects of system delay on user preference; design of dialing plans; and standardization of tones, symbols, instructions, etc. At American Bell he is supervising a group working on user needs for new services in office automation. Mr. Schoeffler has chaired the human factors working party of CCITT. He also holds four patents. Fellow, American Psychology Association; member, Psychonomic Society.

**Cynthia J. Karhan,** B.S. (English Literature), 1969, Indiana University of Pennsylvania; M.A. (Experimental Psychology), 1975, New School for Social Research; Bell Laboratories, 1974—. Ms. Karhan is a Member of Technical Staff in the Stored Program Control Network Services Department. She has worked on automated operator services, including Automated Coin Toll Service and an automated credit card service. She is currently working on the application of automatic speech recognition to network services.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# A Study of the Match Between the Stylistic Difficulty of Technical Documents and the Reading Skills of Technical Personnel

By E. U. COKE* and M. E. KOETHER*

(Manuscript received November 24, 1981)

The reading grade level of a sample of Bell System technical documents was assessed using the Flesch Reading Ease Index, which estimates the reading skill needed to cope with a document's writing style. The reading skills of two groups of Bell System craft and management trainees were also estimated from their performance on a standardized reading test. We found that most of these employees had sufficient reading skill to deal with the writing styles of the sample documents. This observed match between readers and documents resulted because most of our students were proficient readers, not because the documents' writing styles were easy. We compared the reading skills of students as they entered and completed training to see whether less able students were eliminated during training. Results did not support training selection as an explanation of our readers' proficiency. While results indicated that many Bell System employees have the reading skills necessary to cope with the technical documents they read, others may find that the writing styles of technical documents are real barriers to success on the job. We can identify a document that may place an undue burden on its readers by estimating the likelihood of a mismatch between the document's readability and the reading skills of its users.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

## I. INTRODUCTION

How well technical documents communicate with their readers depends in part on the style in which they are written, since the *way* ideas are expressed can influence how easily they are understood. Readers may find a document particularly difficult to read when its writing style places an undue burden on their reading skill. For example, use of words that are unfamiliar to readers can obscure a document's message. Also, a document can be unclear if ideas are embedded in sentences whose syntax is too complex for its readers. Other characteristics of style are described elsewhere in this issue of the *Journal*.[1]

The stylistic difficulty of written communication can be appraised with the help of readability formulas.[2,3] These formulas predict the reading difficulty that may result from a document's writing style but not from its content, organization, or format. Readability formulas estimate the reading skill needed to cope with a document's writing style, as manifested in the writer's choice of surface structures. A readability index is calculated from measures of text features that are thought to be indicators of stylistic difficulty. For example, many formulas use the average length of a text's words and sentences to calculate readability. These two text measures, word length and sentence length, function primarily as indicators of the lexical and syntactic difficulty of the text. In English, shorter words are likely to be more familiar to readers than longer words, since shorter words tend to occur more frequently. Shorter sentences are apt to be easier for readers to process than longer sentences because shorter sentences tend to be less complex syntactically.

While readability indexes are often derived from relatively simple formulas, they can provide useful estimates of sytlistic difficulty.[4] For example, readability indexes tend to agree with human judgments of text difficulty,[5] and it is usually less time-consuming and costly to calculate a readability index than to collect human judgments of reading ease. Readability indexes have also been shown to predict behavioral measures of reading ease, such as reading rate[2,6] or duration of eye fixations,[7] in experimental settings. The readability level of documents has also been related to their effective use in work situations[8,9] and will be discussed later in this section.

Readability formulas have their limitations. Formulas only provide estimates of the stylistic difficulty of texts. These formulas are derived from correlations between text measures related to style and measures of reading difficulty such as comprehension test scores. Almost all formulas have been developed using a multiple regression technique to select the linear combination of stylistic measures that best predicts the scaled difficulty levels of a set of texts. Most of the commonly

used readability formulas, such as the Flesch and Dale-Chall formulas, produce estimates that are accurate to within ±1 reading grade level.[10] The Kincaid formula[11] is slightly less accurate, having a standard error of 1.41 as compared to the Flesch, which has a standard error of 0.81.[12]

Readability formulas are often criticized because editorial revisions that produce acceptable readability indices do not necessarily result in a more readable text.[3,13,14] For example, shortening words or sentences will lower a text's readability index without necessarily making it easier to read. This should not be surprising since readability formulas are derived from *correlations* between measurable text attributes and readers' performance on comprehension tests.[3] As pointed out in the preceding paragraphs, easily quantifiable text features such as sentence length gain their predictive power by indexing underlying contributors to text difficulty that are less easily quantified. Some of the important *determinants* of reading difficulty, such as syntactic complexity and the use of the passive voice, are discussed elsewhere in this issue of the *Journal*.[1]

Just how much difficulty a particular group of readers will have with a document's style depends on their level of reading skill. If a given reader's skill matches or exceeds that required by a document, its writing style alone is not likely to cause trouble for that reader, even though the document's subject matter may be difficult. Conversely, a reader with more limited skill is likely to find that the document's writing style alone presents a challenge.[2–4] By comparing a document's readability index (an estimate of stylistic difficulty) with an estimate of reading skill, we can predict the difficulty a reader might be expected to have with a document's writing style. An estimate of overall reading competence can be obtained from a person's performance on a standardized reading test. These tests sample a reader's behavior on tasks that have been found to discriminate between skilled and unskilled readers.

Estimation of the match between a document's readability and the reading skills of its audience can be helpful in evaluating the effectiveness of technical writing that contains job-related information. Many companies, such as the Bell System, rely heavily on written communication to support job performance. When the writing style of a job-related document is too difficult for its users, job performance may suffer in a number of ways. Studies have investigated the consequences of this mismatch between document style and reading skill. Jobs may take longer than usual to perform because needed information is hard to access.[9] Errors may increase because the descriptions of work procedures are unclear.[8] Perplexed readers may turn to co-workers for help, thus reducing their co-workers' productivity.[9]

In the present study, the match between the readability of docu-

ments and the reading skills of readers was estimated for a large sample of Bell System technical documents. These documents are a major source of information about the operation and maintenance of equipment. The readability of these documents was measured by computer and compared with the reading skills of two groups of Bell System employees who use the documents. Levels of reading skill were estimated from the employees' scores on a reading test.

## II. METHOD

### *2.1 Readability survey*

#### *2.1.1 Documents*

Readability measures were obtained for the text of Bell System technical documents called Bell System Practices (BSPs). These documents are a major source of information about the operation and maintenance of equipment. They are used throughout the system by both craft and management personnel. There are well over 250,000 BSPs so only a sample of these documents were analyzed. Most of these BSPs are typeset under computer control and consequently were available in computer-readable form.

The BSPs selected for analysis were those likely to be used frequently. To meet this goal, documents were chosen from three important areas of company activity—outside plant, customer equipment, and *ESS** switching equipment. The BSPs dealing with outside plant and customer equipment were taken in a random fashion from handbooks provided to craft personnel for use on the job. The BSPs about electronic switching systems were chosen in a random fashion from those describing the operation of two different systems, 1 *ESS* and 2 *ESS* switching equipment. Table I shows the number of BSPs analyzed from each category of company activity. Altogether, readability indices were obtained for 140 BSPs consisting of a total of 378,359 words.

#### *2.1.2 Readability measure*

The Flesch Reading Ease Index[12] was chosen to measure readability. The formula for calculating this index, given in Table II, uses two features of a text to predict stylistic difficulty: (1) the average length of a text's words in syllables, and (2) the average length of its sentences in words. These two measures function primarily as indicators of the lexical and syntactic difficulty of a passage. Available evidence[2–4] suggests that formulas using these simple word and sentence measures are satisfactory predictors of the stylistic difficulty of texts. The

* Trademark of Western Electric.

Table I—Number of BSPs in the readability survey from each category of company activity

| Category | Number of BSPs | Total Number of Words |
|---|---|---|
| Outside plant | 49* | 101,162* |
| Customer equipment | 48* | 83,927* |
| ESS maintenance | 50 | 213,373 |

* Seven BSPs that were in both the outside-plant and customer-equipment handbooks were included in the totals for both categories.

Table II—Formula for the Flesch Reading Ease Index and the algorithm for converting a vowel count to a syllable estimate

1. Flesch Reading Ease Index (FL):

   $FL = 206.84 - 84.6\ WL - 1.015\ SL$

   Where
   WL* is the average length of a word in syllables
   SL is the average length of a sentence in words

2. Algorithm for estimating the syllables (SYL) in a sample of words

   $SYL = 0.998\ VOW - 0.343\ W$

   Where
   VOW is total vowels (aeiouy)
   W is total words

* WL = SYL/W.

Kincaid formula[11] also uses these two measures and is the basis for the Writer's Workbench advisory on readability.[1] The Flesch formula was chosen for the present survey because it has been more widely used than the Kincaid formula and has been shown to be one of the most accurate predictors of readability.[10]

The Flesch Index ranges from 0 to 100 with a higher index representing a more readable text. A Flesch Index can also be expressed as a reading grade level, which is the average school grade of readers who would be expected to score 75 percent on a comprehension test for the reading material. Originally, a point on the Flesch Index scale corresponded to one-tenth of a grade,[12] but Flesch found that his formula underestimated the reading grade level of more difficult passages. He proposed a nonlinear relationship between the Flesch Index and reading grade level,[15] and this relationship is used in the present study to assign a reading grade level to an index value.

The Flesch Index was calculated by computer,[16,17] since the BSPs were in computer-readable form. A program was written to compute average sentence length for (complete) sentences in the text of the BSPs. The program was able to exclude tables, headings, and titles

that were flagged in the text. To calculate the average length of a word in syllables (WL), the program estimated the number of syllables in a word sample from a count of the number of vowels in that sample. This vowel count was then converted to syllables using the formula[18] shown in Table II.

When syllables are estimated from a vowel count, the presence of words without vowels, such as abbreviations, acronyms, and numbers, may be a problem. Their inclusion in the computer's estimate of syllables can lead to an underestimation of average word length and a consequent underestimation of reading difficulty. This is a problem when analyzing technical documents. In some of the BSPs selected for the present survey, over 11 percent of the words did not contain vowels. One solution to this problem is to base the syllable estimate on words that conform to the syllable estimation algorithm's assumption that a word contains at least one vowel. This solution was adopted in the present study and has been implemented in the Writer's Workbench. Only nonnumeric words with at least one vowel were used in the estimation of syllables from a vowel count. This solution seemed to work. Flesch Indexes were computed using hand counts of syllables for 16 BSPs that had a large number of words without vowels. These manually produced indexes were compared with computer-produced indexes. When words without vowels were omitted from the computer estimation of syllables, 11 of the 16 texts were assigned the same reading grade levels by the computer-produced and manually produced indexes. When words without vowels were included in the syllable estimation, only 4 of the 16 BSPs were classified at the same readability level by the computer-produced and manually produced indexes. This solution finds precedence in Flesch's suggestion[12] that counts of syllables be based on words whose pronunciation is unambiguous.

### *2.2 Reading skills survey*

#### *2.2.1 Reading test*

The reading skills of Bell System personnel were estimated from their performance on a standardized, commercially produced reading test, the Nelson-Denny Reading Test, Form D.[19] This test had been standardized by giving it to a sample of over fifteen thousand high school students, representing the student population of the United States in 1972. The test performance of students in this sample was used to estimate the reading skill represented by a test score. Level of reading skill is often expressed as a score's reading grade level. A test score is assigned a reading grade level that corresponds to the *average* educational level of students who made that test score in the standardization sample. Thus, a reader's reading grade level is the school grade for which his or her test score is typical. It should be emphasized

here that the educational level of a person and the reading grade level of his or her test score are not identical.

The Nelson-Denny Reading Test was chosen for three reasons: (1) it can be given in a reasonably short period of time, (2) it covers a range of reading abilities appropriate for Bell System employees, and (3) it was well constructed according to standards for norm-referenced tests.[20]

The Nelson-Denny Test is composed of a vocabulary test and a comprehension test. The vocabulary test contains 100 multiple-choice items consisting of a sentence stem and four words that can complete the sentence. The comprehension test consists of 36 multiple-choice questions that relate to five short prose passages. Readers work on the vocabulary test for 10 minutes and on the comprehension test for 20 minutes.

Performance on the test is expressed as a total score. This score is the sum of two subscores: (1) the number of correct vocabulary items, and (2) the number of correct comprehension items multiplied by two. The reading grade equivalents for total scores are given in the Examiner's Manual of the Nelson-Denny Test.[19] In terms of accuracy, the total score has a standard error of measurement* of about 5.7.[19] Translated into reading grade level, this standard error is approximately a 0.5 reading grade level.

In the present study, total scores were used in all statistical computations. However, the results were also presented as reading grade equivalents of test scores. Reading grade levels were used because: (1) they are often a more meaningful way of expressing reading skill, and (2) the readability of the technical documents was expressed as reading grade levels.

#### *2.2.2 Subjects*

The Nelson-Denny Reading Test was given to 471 technical management employees and 210 craft employees. Both groups were students at Bell System training centers. The management students were taking a variety of technical courses covering such diverse subjects as switching systems, data management, building engineering, and finance. The craft students were taking the final course in an *ESS* switching equipment training program that involves approximately 700 hours of training.

These employees were chosen because they were likely to use doc-

* The consistency of a test can be expressed in terms of the standard error of measurement. Approximately two-thirds of the hypothetical "true" scores on a test would lie within one standard error of measurement from a score obtained on the test.

uments such as the BSPs sampled in the readability survey. In addition, students at the training centers are more readily available for testing than are employees in work situations.

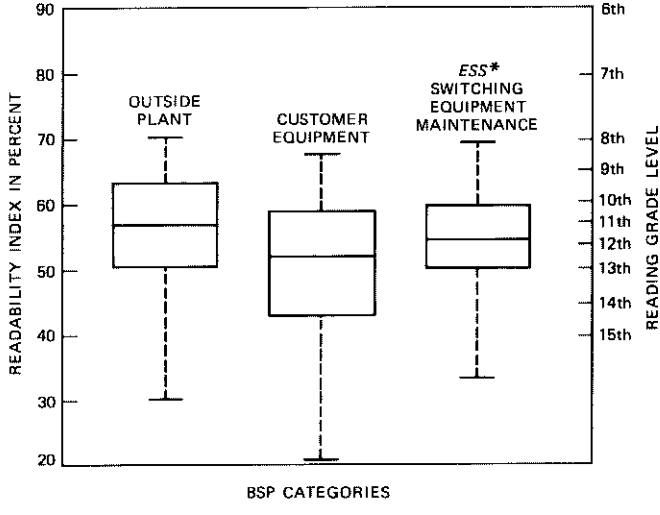
Each student took the Nelson-Denny Test and filled out a questionnaire requesting information about his or her age, sex, years of formal education, job level, years of service in the Bell System, and region of the country in which he or she worked. The study was conducted in the students' classrooms in groups ranging from 10 to 30 students. All students in the selected classes participated in the study although they were not required to do so. To ensure the anonymity of the participants, each student used a number to identify all test material and the questionnaire.

## III. RESULTS

### *3.1 Readability survey*

Figure 1 shows the distributions of Flesch Indexes for BSPs in each of the three categories: outside plant, customer equipment, and *ESS* maintenance. The distribution for each category is displayed as a box plot. The ends of a box are drawn at the upper and lower quartiles of the distribution, while the bar across the box represents the median. The extremes of the distribution are plotted as horizontal lines joined to the box by dashed lines.

Figure 1 suggests that considerable reading skill is required to cope with the writing styles of many BSPs. Thirty-three percent of all the




*TRADEMARK OF WESTERN ELECTRIC.

Fig. 1—The distributions of readability for the three categories of BSPs.

BSPs sampled had Flesch Indexes below 50. An appreciation of the stylistic difficulty of these BSPs can be gained by comparing their readability levels with those obtained by Flesch[12] for some representative magazine articles. Flesch found that academic journals such as the *American Scholar* or the *Yale Review* had Flesch Indexes ranging from 30 to 50, while quality magazines such as the *New Yorker* had Indexes ranging from 50 to 60. Over half of the customer-equipment BSPs were comparable to academic journals in their level of stylistic difficulty. The writing style of three-quarters of the *ESS* switching equipment maintenance BSPs was at least as difficult as that of quality magazines. In terms of required reading skill, the writing style of one-third of the BSPs would be likely to cause difficulty for readers with skills below the 13th reading grade level. The writing style of over half the documents (63 percent) would be likely to cause difficulty for those employees reading below the 11th reading grade level.

### *3.2 Reading skills survey*

#### *3.2.1 Reader characteristics*

The questionnaires provided information about the students who took the reading test. Sixty-five percent of the technical management students were first-level management, 27 percent were second-level management, and the remaining 8 percent were third- (4 percent) and fifth- (4 percent) level management. As Table III shows, the technical management students represented a broader range of age, education, and years of service than did the craft students. The two groups differed primarily in educational level. More technical management

Table III—Age, formal education, and years of service

| | | Percentage of Students | |
|---|---|---|---|
| | | Technical Management | Craft |
| Age Level | 20–29 | 26 | 29 |
| | 30–39 | 39 | 52 |
| | 40–49 | 24 | 14 |
| | 50 or older | 11 | 5 |
| Educational Level | Some high school | 1 | 3 |
| | High school grad | 30 | 55 |
| | Some college | 29 | 36 |
| | College grad | 26 | 6 |
| | Some grad school | 14 | |
| Years of Service | 1–5 | 19 | 9 |
| | 6–10 | 26 | 49 |
| | 11–20 | 30 | 30 |
| | over 20 | 25 | 12 |

students (69 percent) than craft students (42 percent) had continued on to college after graduating from high school. There were also more women among the technical management students (30 percent) than among the craft students (9 percent).

### *3.2.2 Reading scores*

The cumulative probability distributions of total scores on the Nelson-Denny Reading Test are shown in Fig. 2 for each group of students. The reading grade equivalents of test scores appear at the top of the figure. The median, upper, and lower quartiles of the distributions are also shown. The maximum total score is 172, while the highest reading grade level that can be assigned to a score is 15.

Test scores indicate that, on the average, both groups of students were competent readers. Mean scores were 81 and 94 for craft and technical management students, respectively. For comparison, the Nelson-Denny manual[19] gives mean scores for samples of freshmen, sophomores, juniors, and seniors in college. These scores were 75, 85, 93, and 98, respectively. Over half the craft and technical management students had test scores at or above the 14th (college sophomore) reading grade level. The reading grade levels of the mean scores were 14.0 and 14.9 for craft and technical management students, respectively, and this difference between student groups was greater than would be expected by chance ($t = 3.24$, $df = 671$, $p < 0.05$).

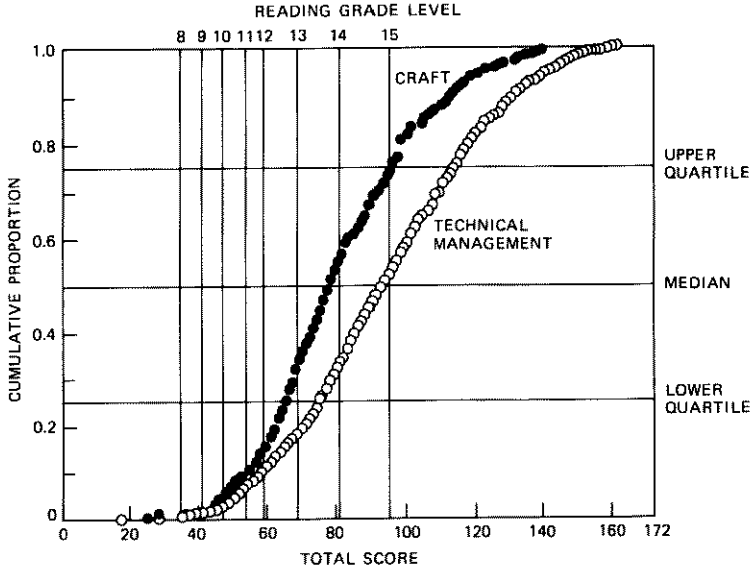



Fig. 2—Cumulative frequency distributions of reading test scores showing the median, upper, and lower quartiles of distributions.

### 3.3 *Estimating the stylistic difficulty of BSPs*

An estimate of the difficulty a group of readers may have with the writing style of a specific document can be made by finding the proportion of readers whose reading grade levels on a reading test are below the reading grade level of the document. If a document's reading grade level is 12, for example, and one third of its users have reading skills below the 12th reading grade level, then the likelihood of a mismatch is 0.33 (e.g., the probability that the users' reading skills will be below the 12th reading grade level).

This likelihood of a mismatch between audience and document can be extended to a set of documents. In this case, two measures are needed for each reading grade level represented by the set of documents:

1. The proportion of documents in the set whose reading grade levels are greater than a specified reading grade level (that is, the proportion of documents likely to be difficult for readers at that grade level);

2. The proportion of readers in the group at that reading grade level.

The likelihood of a mismatch is the sum of the products of these two estimates over all reading grade levels. This likelihood is the probability that a member of the audience will select a document from the set that exceeds his or her reading grade level.

The likelihood of a mismatch between BSPs and the technical management and craft students were calculated from the information in Table IV. This table shows the proportion of BSPs whose reading grade levels exceeded each level in the table. Also shown are the

Table IV—Proportions for calculating the likelihood of a mismatch between BSPs and two employee groups

| Reading Grade Level | Proportion of Difficult BSPs | Proportion at Each Reading Grade Level | |
|---|---|---|---|
| | | Technical Management | Craft |
| below 7.0 | 1.00 | 0.00 | 0.01 |
| 7.0–7.9 | 0.99 | 0.00 | 0.00 |
| 8.0–8.9 | 0.90 | 0.01 | 0.01 |
| 9.0–9.9 | 0.74 | 0.01 | 0.02 |
| 10.0–10.9 | 0.63 | 0.04 | 0.05 |
| 11.0–11.9 | 0.44 | 0.04 | 0.05 |
| 12.0–12.9 | 0.33 | 0.06 | 0.15 |
| 13.0–13.9 | 0.20 | 0.16 | 0.26 |
| 14.0–14.9 | 0.11 | 0.19 | 0.18 |
| 15 or greater | 0.00 | 0.49 | 0.27 |

proportion of technical management and craft students at each reading grade level.

The likelihoods of a mismatch between BSPs and the student audiences was 0.14 for technical management students and 0.20 for craft students. This result suggests that technical management students were likely to have some difficulty with the writing style of about 14 percent of the BSPs in the sample, and craft students with about 20 percent. These estimates of difficulty suggest that the mismatch between students and BSPs is not great, but we do not really know how to interpret such results. For one thing, the Flesch Reading Ease Index assumes 75-percent reading comprehension. For technical documents, such as these BSPs, comprehension closer to 100 percent is probably desirable. In addition, although losses in job efficiency may result when employees have difficulty with a document, simplifying its writing style can be both costly and time-consuming. At present we know that it is useful to match documents to readers but we don't know how to calculate the benefits and costs involved in achieving this match.

## IV. DISCUSSION

The results of the present study suggest that the writing style of many Bell System documents used on the job and in training requires considerable reading skill. Yet, the reading skill levels of the majority of the employees we tested matched or exceeded the estimate of required reading skill provided by the readability analysis. Our two samples of Bell System employees appeared to have fewer people with limited reading skills than might be expected from their educational levels. This was especially so for the craft students who had an average reading grade level (level 14—college sophomore) that was two grade levels *above* their average educational level (level 12—high school graduate).

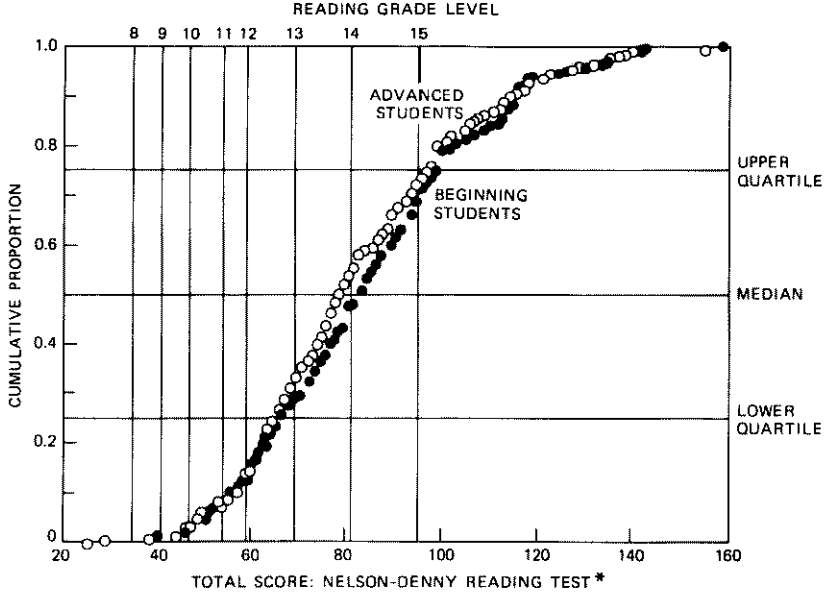
A number of selection factors may have acted to make our sample of Bell System employees such able readers. One of these factors may have been the documents that these students read during training. The craft students we tested had successfully completed a series of courses (approximately 700 hours of training) that required the use of highly technical documents, including BSPs. These technical documents may have acted as selective filters, tending to eliminate students with limited reading skills. This explanation assumes that students who have reading difficulties or who are averse to reading difficult documents will tend to leave training. If our filter explanation is correct, we would expect to find proportionately more skilled readers among students who were about to complete training than among students who were starting a training sequence. We explored this filter

explanation by comparing the reading-test performance of craft students who were just beginning an *ESS* switching equipment training sequence with students who were about to complete this sequence.

The Nelson-Denny Reading Test and a questionnaire were administered to 160 craft employees who were enrolled in introductory *ESS* switching equipment courses given at the training centers of several operating companies. Their performance was compared with that of the 210 craft students discussed earlier (Fig. 2), who were finishing their last course in the ESS training sequence.

The distributions of total scores on the reading test are shown in Fig. 3 for the beginning and advanced students. There was little difference between these two groups in their performance on the reading test. These results do not support the contention that training documents tend to filter out less able readers during *ESS* training, but rather they suggest that our sample of Bell System craft students was not biased by selective factors during *ESS* switching equipment training. Thus, their level of reading skill is likely to be representative of Bell System craft employees with similar training.

Training documents may not have worked as filters because of the procedures for selecting employees who enter *ESS* switching equip-




*MAXIMUM SCORE IS 172

Fig. 3—Cumulative distributions of reading test scores for advanced and beginning students.

ment training. These procedures, themselves, may lead to the choice of able readers. Operating companies undoubtedly choose their more able employees for *ESS* training because it is expensive and time-consuming. In addition, all those who start *ESS* training must complete a mini-course that helps to select employees who are likely to do well in *ESS* switching equipment self-instructional training. Even the perceived mental demands of operating and maintaining a complex switching system may discourage employees with limited intellectual skills. Although *ESS* trainees are not chosen explicitly for their reading skills, all the factors mentioned above could lead to the selection of competent readers as trainees.

## V. CONCLUSION

In the present study, the reading skills of the majority of employees we tested were likely to be matched to the reading demands made by the writing styles of the majority of the technical documents they read. This match resulted because so many of the employees were proficient readers and not because the documents were easy to read. In many technical areas, such as *ESS* switching equipment, readers may have more reading skill than would be expected on the basis of their education. One implication of this finding is that demographic information about education may not be sufficient to predict an audience's level of reading skill. More direct estimates of reading skill may be needed, such as the Nelson-Denny Reading Test.

Although we found a reasonable match in grade levels between documents and readers in the present study, this does not ensure that the documents are easy to read. Documents can be identified as difficult but not necessarily as easy to read, because the readability index does not include such important determinants of difficulty as organization, content, or format. For this reason, our findings should not discourage Bell System writers from writing as clearly and simply as possible. For many employees, the writing styles of technical documents may be a real barrier to success on the job. Screening devices that combine information about the readability of documents and the reading skills of their audiences, such as the likelihood of a mismatch, can help identify documents that may be too difficult for their intended readers.

## VI. ACKNOWLEDGMENTS

We are grateful to E. Z. Rothkopf for his helpful comments and suggestions and Western Electric at Winston-Salem, North Carolina, for providing us with the BSPs. We also wish to thank the management and instructors at the Bell System Center for Technical Education,

Lisle, Illinois, and the Bell System Technical Center, Dublin, Ohio, for their assistance. J. A. Leedy of Network Operations Training at AT&T and the management and instructors at the following Training Centers also lent us their assistance: Plant Training Center, Pennsylvania Bell; Birmingham Training Center, South Central Bell; Decatur Training Center, Southern Bell; Houston Training Center and St. Louis Training Center, Southwestern Bell.

## REFERENCES

1. N. MacDonald, "*UNIX*™ Writer's Workbench Software: Rationale and Design," B.S.T.J., this issue.
2. G. R. Klare, *The Measurement of Readability,* Ames, IW: Iowa State Univ. Press, 1963.
3. G. R. Klare, "Assessing Readability," Reading Res. Quart., *10,* No. 1 (1974–75), pp. 62–102.
4. G. R. Klare, "A Second Look at the Validity of Readability Formulas," J. Reading Behavior, *8,* No. 2 (Summer 1976), pp. 129–52.
5. G. R. Klare, "Judging Readability," Instructional Science, *5,* No. 1 (January 1976), pp. 55–61.
6. E. U. Coke, "Reading Rate, Readability and Variations in Task-Induced Processing," J. Ed. Psych., *68,* No. 2 (April 1976), pp. 167–73.
7. E. Z. Rothkopf, "Copying Span as a Measure of the Information Burden in Written Language," J. Verbal Learning and Verbal Behavior," *19,* No. 5 (October 1980), pp. 562–72.
8. K. A. Johnson, R. P. Relova, and J. P. Stafford, *An Analysis of the Relationship Between Air Force Procedural Manuals and Discrepancies Involving Noncompliance with the Procedures,* Air University: Air Force Institute of Technology, (NTIS No. AD-750 917), 1972.
9. *Reading for Working: A Functional Literacy Anthology,* T. G. Sticht (Ed.), Alexandria, VA: Human Resources Research Organization, 1975.
10. R. D. Powers, W. A. Sumner, and B. E. Kearl, "A Recalculation of four Adult Readability Formulas," J. Ed. Psych., *49,* No. 2 (April 1958), pp. 99–105.
11. J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel.* CNTT Research Branch Report 8-75, Naval Air Station, Memphis, Millington, TN: Chief of Naval Technical Training, 1975.
12. R. F. Flesch, "A New Readability Yardstick," J. Appl. Psych., *32,* No. 3 (June 1948), pp. 221–33.
13. A. Davison, R. N. Kantor, J. Hannah, G. Hermon, R. Lutz, and R. Salzillo, *Limitations of Readability Formulas in Guiding Adaptations of Texts* (Technical Report No. 162), Urbana-Champaign, IL: Center for the Study of Reading, 1980.
14. "Forum—Readability Formulas: Used or Abused?" IEEE Trans. on Professional Commun., *PC-24,* No. 1 (March 1981), pp. 43–54.
15. R. F. Flesch, *The Art of Readable Writing,* Revised Edition, New York: Harper & Row, 1974.
16. M. E. Koether and E. U. Coke, *A Scheme for Text Analysis Using Fortran,* Annual Meeting of the American Educational Research Association, New Orleans, February 28, 1973.
17. E. U. Coke, "Computer Aids for Writing Text," in *The Technology of Text,* D. H. Jonassen (Ed.), Englewood Cliffs, NJ: Educational Technology Publications, 1982.
18. E. U. Coke and E. Z. Rothkopf, "Note on a Simple Algorithm for a Computer-Produced Reading Ease Score," J. Appl. Psych., *54,* No. 3 (June 1970), pp. 208–10.
19. *Nelson-Denny Reading Test, Form D,* Boston, MA: Houghton Mifflin, 1973.
20. O. K. Buros, *The Sixth Mental Measurement Yearbook,* Highland Park, NJ: Gryphon Press, 1965.

## AUTHORS

**Esther U. Coke,** B.A. (Zoology), 1951, Wellesley College; Ph.D. (Psychology), 1968, New York University; Bell Laboratories, 1957—. Ms. Coke is a member of the Learning and Instruction Research Department. Her research interests include human learning, stylistic factors that influence the effectiveness of technical writing, and computer techniques for evaluating writing. Member, APA, AERA, Psychonomic Soc., Sigma Xi, AAAS.

**Mary E. Koether,** B.A. (International Relations), 1951, Mount Holyoke College; Bell Laboratories, 1965—. Ms. Koether is a member of the Learning and Instruction Research Department. Her professional interests include the computer analysis of text and factors that influence text comprehension. She also participated in the design and programming of a computer-based course maintenance system. Member, AERA.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

*Human Factors and Behavioral Science:*

# Toward Bell System Applications of Automatic Speech Recognition

By J. E. HOLMGREN*

(Manuscript received July 30, 1982)

Advances over the past few years in the field of Automatic Speech Recognition (ASR) have brought more attention to potential Bell System applications of this technology. Before reaching the point of ASR implementation, several human factors problems have to be overcome. This paper describes the central human factors issues, then summarizes the initial steps at Bell Laboratories in attempting to deal with those issues. Findings from observations of customers speaking credit card numbers to operators are described, followed by summaries of three studies investigating control of the speech of ASR system users.

## I. INTRODUCTION

The potential of Automatic Speech Recognition (ASR) in telecommunication applications has long been recognized,[1] but until recently the state of ASR technology did not warrant effort beyond the existing laboratory activity. In the past few years, ASR has emerged from the laboratory into commercial use. Several systems are now available providing speaker-dependent word recognition.[2] Such a system must be trained by each user before it can recognize that user's speech.

* American Bell.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

Most of these systems require isolated speech in the sense that each vocabulary item must be spoken with a short silent interval preceding and following each word, although a few systems allowing connected speech are now available.[2] Speaker-dependent systems have been applied primarily in industrial settings that require hands-free data entry, such as inspection and quality control.

Speaker-dependent ASR systems are inappropriate for many attractive telecommunications applications. Instead, speaker-independent systems, which need no prior training by users, are required. Various network operator services, such as credit card calling and directory assistance, are examples of applications in this category. Through the use of ASR, such services could be automated and used from any telephone that has access to the existing network. A few isolated-speech, speaker-independent ASR systems are already in use in the United States and Japan. They provide service to selected groups of users in applications such as private network call-routing and banking. Laboratory work is under way to develop a connected-speech, speaker-independent ASR capability.[3]

The use of ASR in universally accessible services raises many human factors questions. This paper summarizes Bell Laboratories initial human factors work leading toward the first network applications of speaker-independent ASR. Our work began in the context of considering the use of ASR in one particular network service: credit card calling. Today, most credit card calls require giving a credit card number (CCN) to an operator. A new service known as Calling Card Service (CCS) automates the handling of credit card calls from *Touch-Tone** telephones by allowing customers to enter their CCNs on the *Touch-Tone* telephone number pad. However, calls from rotary telephones must still be handled by operators. ASR would allow automation of all credit card calls.

Our first step in investigating the credit card application was to identify the critical human factors issues that require attention. Several of these issues are common to almost all potential network applications of speaker-independent ASR. A description of the common issues provides perspective on our subsequent human factors work.

## II. THE CENTRAL HUMAN FACTORS ISSUES

Issues that surround the user-system dialog were selected as the focus of the work reported here. Although the other issues are only touched on in this paper, they are no less important to any successful network application of ASR.

---

* Trademark of AT&T.

### 2.1 User-system dialog

For the foreseeable future, ASR systems will be limited by several aspects of human speech, such as the vocabulary they can recognize, the maximum rate of speech they can handle, their ability to ignore extraneous words and sounds, and the accuracy of recognition per spoken vocabulary item. Thus, care must be taken in the design of any user-system dialog to overcome these limitations.

#### 2.1.1 Instructions

Appropriate instructions are needed to control the speaking rate and vocabulary that untrained speakers use when they encounter an ASR system for the first time. These instructions also must allow experienced users to proceed without unnecessary delay.

#### 2.1.2 Feedback

While current ASR systems are achieving impressive recognition accuracy for limited vocabularies, their accuracy is less than that of a human listener. Therefore, many applications, particularly those where errors are costly, require feedback to the user to ensure correct recognition by the ASR system. For many attractive telecommunications applications of ASR, user input will consist primarily of strings of digits (e.g., telephone numbers or credit card numbers). Several digit feedback provisions are possible. For instance, feedback could be given after each digit, after each group of digits, or after entry of the entire number. The optimal method depends on the nature of the application and the recognition accuracy of the ASR system.

#### 2.1.3 Error correction

Methods are needed that allow users to correct both their own speaking errors and, when feedback is given, recognition errors on the part of the ASR system. Again, several options are available and the optimal method for each application is unclear.

#### 2.1.4 Problem speakers

No matter how good an ASR system may be, there will always be some speakers whose speech cannot be reliably recognized by the system. Any service incorporating ASR will have to provide for some type of alternate treatment for such individuals; this will often mean transfer to an operator or attendant. Detection of problem speakers early in the dialog and swift alternate treatment will be necessary to ensure both service efficiency and customer satisfaction. The best way to detect these speakers has not yet been established.

### *2.2 Isolated vs. connected speech*

Connected speech is preferable to isolated speech for use as input to an ASR system. However, when speaker-independent ASR systems that accept connected speech become available for use, isolated input will still be more accurately recognized. For this reason, in any application of ASR it will be necessary to decide whether the trade-off between the greater ease of use of connected speech and the greater accuracy of isolated speech favors the former or the latter.

### *2.3 Vocabulary choice and expansion*

User and system considerations may often conflict when a vocabulary is selected for any given application of ASR. The most appropriate vocabulary for the speaker may be particularly difficult for the ASR system. For instance, while spoken, spelled input might be a natural way to specify a name to a directory assistance system, the spoken alphabet is a singularly difficult vocabulary for any ASR system to accept.[4] Words in a vocabulary such as the international word-spelling alphabet (Alpha, Bravo, Charlie, etc.) would be much more accurately recognized by machine, but much less convenient for most users. Closely related to vocabulary selection is the problem of vocabulary expansion. Careful selection of new vocabulary items is necessary because adding new words to a vocabulary may change the system's performance on the original set of words.

### *2.4 Integration of* Touch-Tone *service with ASR*

For any network service using ASR, a large percentage of the customers will be calling from *Touch-Tone* telephones. Thus, it is necessary to consider the possibility of mixed *Touch-Tone* telephone and voice input to an automated service. This raises several questions regarding integration of the two, such as whether to provide both input options at every point in a service, whether to encourage the use of one option over the other, how to make voice input compatible in some sense with *Touch-Tone* telephone input when both are available, etc.

### *2.5 Template construction*

In an ASR system templates represent the words to be recognized in a given application. Template construction is in many respects the most critical hurdle in applying speaker-independent ASR because it is largely the quality of those templates that determines the recognition accuracy of the system across the population of users.[5] Template considerations are included here because of the human factors problems involved in building the speech database needed to construct them.

### *2.6 System evaluation*

The performance of a speaker-independent ASR system in any application depends not only on the characteristics of the system but also on the vocabulary used in the application, the set of templates constructed for that vocabulary, the transmission conditions, and the characteristics of the spoken input. Therefore, evaluating the adequacy of a system in an application involves more than simply obtaining some overall measure of recognition accuracy. Information will be needed about variation in recognition accuracy across segments of the user population, across vocabularies, and across transmission conditions. Other critical aspects of performance will be system response time and the rate of false recognition for words outside the application vocabulary.

## III. TSPS OBSERVATION STUDY

To study user-system dialog issues in the application of ASR to credit card calling we gathered data from customers speaking their credit card numbers to Traffic Service Position System (TSPS) operators, who handle all nonautomated credit card traffic. Such data were needed to identify any customer behavior changes necessary to interact successfully with an ASR system.

The particular aspects of customer speaking behavior that we investigated were:

1. Customer segmenting of CCNs. Segmenting, as used here, refers to the tendency of most customers to break a CCN into spoken segments by pausing briefly after speaking a group of three or four digits. Segmenting is important because the per-item recognition accuracy of speaker-independent ASR systems that can accept connected speech is likely to be highly sensitive to the number of items in a connected sequence.
2. Customer vocabulary (e.g., "zero" versus "oh," "hundred" versus "zero zero," etc.).
3. Occurrence of words or sounds other than those used to give the CCN.
4. Frequency of customer mistakes in speaking the CCN and spontaneous correction of those mistakes.
5. Frequency of operator requests for repetition of a portion or all of the CCN. Our interest in this stems from the fact that reliable system performance is possible only when the customer corrects system-recognition errors using feedback from the ASR system; thus, it is useful to have information on the current frequency of operator-requested repetitions.
6. Speaking rates distribution for current customers.

### 3.1 Summary of results

#### 3.1.1 Number of observations

A total of 3040 credit card calls were observed at three different TSPS offices, 1157 in Milwaukee, 742 in Louisville, and 1141 in Boston. The relatively low number of observations for Louisville reflects a low overall credit card call volume during the observation period. Since the card numbers for the observed calls at each site were not recorded, the number of distinct CCNs among the 3040 calls is not known.

#### 3.1.2 Segmentation of the spoken CCN

The majority of CCNs were spoken in consecutive segments of 3, 3, 4, and 4 digits. Of all observations, 69 percent fell in this category. This is not surprising, since the format of most CCNs is NPA NXX XXXX XXXX and in the three operating companies visited the number is printed on the credit card with that segmenting.

The next most common segmentation was 3 3 4 3 1, used in 17 percent of the observations. The frequency of this segmentation probably reflects carryover from the previous 10-digit-plus-one-letter format used for CCNs up until two months before data collection. In most cases, the new 14-digit numbers were constructed from existing CCNs by adding an initial NPA and changing the terminal letter to a digit. Thus, despite the fact that the new number was printed on the credit card with the 3 3 4 4 segmentation, customers accustomed to speaking their old number with a 3 4 3 1 segmentation carried that habit over to the new number.

Only four percent of the observed numbers were spoken with the digits run together. A number was classified as run together if five or more digits were spoken without an intervening pause. This was the third most common segmentation category. The remaining 10 percent of the calls were distributed among several infrequently occurring segmentation categories.

#### 3.1.3 Variation in vocabulary

Regarding the use of "zero" or "oh" when speaking the digit zero, 29 percent of the 3040 spoken CCNs contained a spoken "zero," 51 percent contained a spoken "oh," 9 percent contained both "zero" and "oh," and 11 percent contained neither. It is interesting that customers frequently say both "zero" and "oh" while speaking a single CCN. While figures are not available on the proportion of observed CCNs that contain multiple occurrences of the digit zero, it is evident that the percentage of multiple zero calls involving the use of both "zero" and "oh" is considerably greater than nine percent.

When speaking the CCN, customers used words outside the single-

digit vocabulary in only eight percent of all calls. Almost all of these calls involved some combination of three types of multiple-digit utterances. The most common type was a two-digit combination such as "fifty-six," "eighty-eight," "thirteen," etc. The next most common type was the phrase "double zero" or "double oh" for the digit combination 00, usually when the 00 was the leading pair of a segment. The word "hundred," usually used for a terminal 00 in a segment, was the third most common type.

#### *3.1.4 Extraneous vocalizations*

There were relatively few occurrences of extraneous vocalizations from the customer. Less than two percent of all calls included such events. The most commonly used word was "dash," spoken between segments of the CCN. This occurred because some operating companies print the CCN on the credit card with dashes between segments.

#### *3.1.5 Customer correction of errors*

The customer corrected an error in the spoken CCN before any request for repetition from the operator on less than three percent of all calls. The most common error was to leave off the area code when giving the CCN. Since the requirement to give the area code had been in force for less than three months at the time these data were collected, customers were still adjusting to the new CCN format.

#### *3.1.6 Operator requests for repetition*

The operator asked the customer to repeat some or all of the CCN on less than six percent of all calls. Usually the operator requested repetition of all 14 digits. This occurred because the operator can collect any number of digits entered on the TSPS console number pad only by cancelling all entered digits and beginning again. The three most common reasons for requesting repetition were operator keying errors, the operator misunderstanding or not hearing the customer, and the customer forgetting to give the area code.

#### *3.1.7 Customer speaking rates*

Customer speaking rates are of interest only for those calls on which the CCN was entered in a manner otherwise consistent with the constraints likely to be placed on users of any connected-speech, speaker-independent ASR system. Examining these calls allows one to determine if credit card customers were speaking too quickly for ASR systems when their spoken input was acceptable in all other respects. A call was classified as acceptable if it had the following characteristics:

1. Only single-digit words were spoken.

2. No extraneous vocalizations occurred.
3. The customer did not make an error in entry.
4. The operator did not request a repetition.

An operator may request repetition of a CCN, even though the input was acceptable to an ASR system. However, to be conservative, these calls were excluded from the acceptable set, which contained 80 percent of all calls.

As a function of location, the mean speaking time for a CCN is 5.16 seconds in Milwaukee, 5.33 seconds in Louisville, and 5.09 seconds in Boston. Although the variation among these means is small, an analysis of variance shows it to be statistically significant ($F = 11.79$, $p < 0.001$). The mean speaking times for males and females are nearly identical, being 5.17 seconds and 5.19 seconds, respectively.

The performance of connected-speech, speaker-independent ASR systems is likely to fall off rapidly as the rate of speaking connected strings increases beyond about 2.5 words per second.[6] For this reason, speaking rates within connected strings of digits are of more interest than total speaking times for CCNs. To compute a speaking rate for each call, the total speaking time was corrected for the pauses occurring between segments. This correction was made by assuming that for any speaker, the duration of such pauses was roughly equal to that speaker's mean speaking time per digit. Based on the listening experience of the two data collectors, this appears to be a reasonable assumption. The appropriate estimate of speaking rate for a call is therefore given by the following expression:

$$\text{Rate} = (14 + s)/t,$$

where $s$ is the number of intersegment pauses and $t$ is the total speaking time for the CCN. The value of $s$ was based on the segmentation judgment made by each data collector on each call. Calls for which the digits were judged to be run together present problems for this rate estimate. Any call having five or more digits run together should fall in this category, so the number of intersegmental pauses is not constant within the category. For simplicity, it was assumed (based on the data collectors' observations) that the average number of pauses in a call classified as run together was one, so $s$ was set to that value.

The distribution of speaking rates is given in Fig. 1, where, for 94.2 percent of the acceptable calls, the speaking rate was greater than 2.5 digits per second.

### *3.2 Implications for customer behavior*

The above findings indicate two areas where the modal customer-speaking behavior is likely to require modification before customers

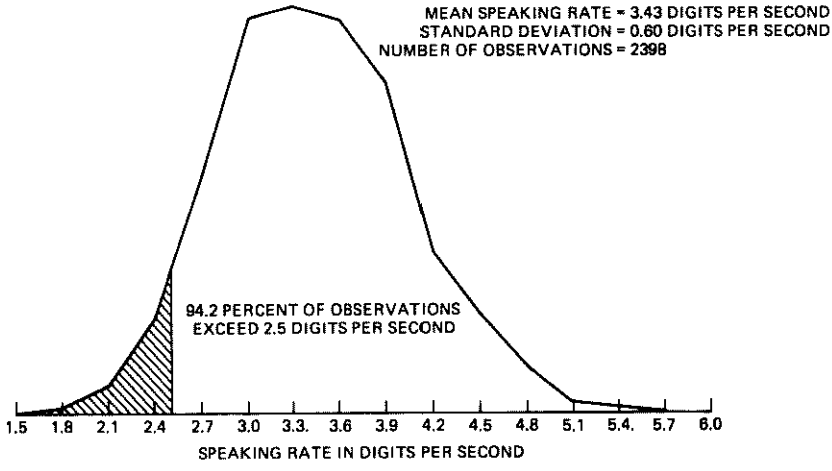



Fig. 1—Distribution of speaking rate for calls classified as acceptable.

could successfully interact with a speaker-independent, connected-speech ASR system. First, assuming that any such system will have considerably more difficulty recognizing "oh" than "zero" (due to the relatively small amount of energy in a spoken "oh" and to coarticulation effects), customers may have to learn to say "zero" in place of "oh". Second, assuming that 2.5 digits per second represent an approximate maximum acceptable speaking rate, most customers will have to lower their rates. Some means is needed of getting all customers under the critical rate without lowering the rate for many of them too much. In other words, both the mean and variance of the speaking rate distribution need to be decreased.

While these data provide some useful baseline information about spoken digit strings in a field environment, generalization of the findings to other situations must be done carefully. The CCN is a long, highly familiar number and credit card customers are experienced callers. Input characteristics may differ for shorter, less familiar digit strings or for a less experienced user population. Also, there are several important questions that could not be addressed in this study. For instance, while there were no important differences among locations on any of the variables investigated, the possibility remains that variation in speaker accent will present critical problems for any ASR system. Questions such as this cannot be answered without testing a system on actual speech samples from a field environment.

Despite its limited scope, this study was useful in guiding subsequent human factors work. The studies reported below followed directly from the questions produced by these data.

## IV. CONTROL OF USER SPEECH

Having determined the speech characteristics of customers giving numbers to operators, we conducted a series of three studies designed to investigate methods of eliciting spoken numbers from users of automated network services that would be acceptable to ASR systems. These studies are summarized below.

### *4.1 Simulation study*

The first study of the series involved a laboratory simulation of a connected-speech, speaker-independent ASR system in CCS. Each of the 30 subjects in the study was given a credit card with a 14-digit number printed with 3 3 4 4 segmentation. Subjects were then instructed to make a series of between 24 and 36 credit card calls that required voice entry of the credit card number to the simulated ASR system.

#### *4.1.1 Prompts and feedback*

After dialing a ten-digit number, most subjects heard a tone followed by a brief pause, then the following prompt:

> Please speak your credit card number in groups of four digits or less. Please wait for the numbers to be repeated back to you before speaking the next group. If the numbers are repeated incorrectly, say the word "error." Then when you hear the tone, repeat the last group of numbers spoken. Listen again for the response and then proceed with the next group of numbers. At the tone, please begin speaking your credit card number.

A similar prompt was used for another group of subjects that was not given feedback of the spoken digit groups. The instructions for this group included the phrase "say zero instead of oh." This phrase was not included when feedback was provided because we wanted to see the effect of feeding back "zero" when the subject said "oh." The subjects did not know that digit recognition was done by a human who keyed in the spoken segments of the number, triggering voice feedback of the spoken digits to those subjects selected to receive feedback.

We attempted to implicitly control speaking rate by means of the rate at which feedback was given; digits were fed back at either 2.50 or 1.25 digits per second. To investigate the effect of system recognition error rate on user input, two different digit recognition error rates of 1.8 percent and 5.4 percent were simulated. Each subject experienced at least two different combinations of feedback rate and error rate.

#### *4.1.2 Summary of results*

Subjects were initially told that once they thought they no longer needed to hear the prompt they could begin to speak after the first

tone. The typical subject would listen to the prompt on the first one or two calls, then begin to speak after the first tone on all subsequent calls. Despite the length of the prompt and the limited exposure to it, subjects had little difficulty following the instructions. All subjects in the feedback condition corrected errors smoothly and consistently. There were very few departures from the digit vocabulary (0.1 percent of all calls) and subjects nearly always used proper segmentation (99.8 percent of all calls).

While all subjects not given feedback followed the instruction to say "zero" instead of "oh," the feedback "zero" less successfully induced subjects who received feedback to say "zero". The proportion of those subjects saying "zero" on the first call was 0.3; this increased to 0.7 by the twelfth call.

The mean speaking rate across all conditions was 2.58 digits per second. While this rate is lower than the rate of 3.43 digits per second observed in the field, the speaking rate did not vary significantly as a function of either feedback (none, slow, fast) or system error rate. Subjects in the feedback conditions did lower their speaking rate slightly (by 0.2 digit per second) when correcting a system recognition error, but then returned to the higher rate on the next call. Thus, under the conditions of this study, user speaking rate was not sensitive to feedback, either in terms of feedback rate or recognition error rate. However, it should be noted that the simulated-recognition error rates in this study were independent of a subject's speaking rate. In a real ASR system, the system error rate would increase with increasing user speaking rate, possibly making users more sensitive to feedback.

### *4.2 Telephone prompt study*

The simulation study showed that some aspects of subjects' speech could be effectively controlled (at least in a laboratory setting) with simple prompting. However, the instructions in that study did not attempt to control speaking rate. Therefore, the next study concentrated on the speaking rate problem. Instead of bringing subjects into a laboratory stimulation to evaluate prompts, we chose to contact subjects by phone in their own work environments and ask for their cooperation in a very brief experiment that required them to simply say their own home phone numbers, including the area codes. Participants heard one of a set of recorded prompts and responded by speaking their numbers. All subjects were employees at Bell Laboratories in Holmdel, New Jersey. Spoken home telephone numbers provided a stronger test of our ability to control speaking rate, since a highly familiar number is likely to be spoken more rapidly than an unfamiliar one.

#### 4.2.1 Candidate prompts

Table I gives the four components from which six prompts were composed. Three of the prompts began with Component 1 in Table I; one of these was completed by adding only Component 2, while the other two were completed by adding both Component 2 and either the first or second sentence of Component 3. The remaining three prompts were the same as the above three, except that Component 1 was omitted.

This set of six prompts allowed separate evaluation of the effect on subjects of knowing that they were speaking to a machine and the effect of specific instruction on how to speak. The first sentence of Component 3 in Table I (rate instruction) was designed simply to lower the speaking rate; the second (isolation instruction) was designed to elicit isolated speech. Absence of either of those two sentences from the prompt provided a control condition. A total of 60 subjects provided data on these prompts, 10 for each of the 6 prompts.

#### 4.2.2 Summary of results

Table II gives the mean speaking rate for each prompt, corrected for intersegment pauses. A $3 \times 2$ analysis of variance showed that informing subjects that they were speaking to machines significantly lowered speaking rates ($F = 6.04$, $p < 0.03$). Also, specific instruction about how to speak significantly affected speaking rate ($F = 7.83$, $p < 0.01$). The interaction between the machine information and specific instruction conditions was not significant ($F = 0.51$).

One other instruction evaluated in an early phase of this study deserves mention. This instruction was worded as follows:

> At the tone, please speak your telephone number. Say "zero" instead of "oh." Speak at the following rate: [Recorded voice speaking, "One, two, three (pause) four, five, six"].

This instruction produced the lowest mean speaking rate (1.31 digits per second) of any prompt evaluated. Prior to playing the prompt, the experimenter told the subjects that they would speak to a machine, instead of including that information as part of the prompt.

As can be seen in Table II, even when no machine information and no specific rate instruction is given, the speaking rate is still considerably lower than that observed in the field study. As in the simulation study, this probably occurred because the subjects knew they were

Table I—Components of telephone study prompts

| |
|---|
| 1. This is a machine which recognizes human speech. |
| 2. At the tone, please speak your telephone number. Say zero instead of oh. |
| 3. Speak slowly and distinctly. |
| OR |
| Pause briefly after each digit. |

Table II—Speaking rates in the telephone prompt study (digits/s)

| | Machine Information | |
|---|---|---|
| Instruction | Absent | Present |
| Control | 2.41 | 2.04 |
| Rate | 2.18 | 1.54 |
| Isolation | 1.56 | 1.36 |

participating in an experiment and were making a special effort to speak clearly. That special effort cannot be expected in an actual service environment. Nonetheless, the results clearly show that simple instructions can considerably lower the speaking rate.

Beyond the fact that the isolation instruction lowered user speaking rates, it is of interest to know if it also succeeded in eliciting isolated speech. Although the speech given in response to the instruction sounded adequately isolated to the listener, the data were not recorded in a form that allowed a more thorough treatment of the isolation question. This question is directly addressed in the next study.

### *4.3 Speech isolation study*

Up to this point, the reported work has focused primarily on issues surrounding connected-speech recognition. For reasons related to a larger, ongoing ASR project, the focus of our human factors work now shifted to the problem of eliciting speech acceptable to an isolated speech recognition system. We wanted to see whether it was possible, through the use of prompts alone, to obtain isolated speech from subjects. While it is possible to force isolation through the use of a pacing cue or feedback after each spoken item, such techniques tend to produce slower input from the user than is required by the ASR system. Particularly in applications involving entry of long digit strings (such as credit card numbers), experienced users may find paced entry tedious. We adapted the procedure used in the previous study to the current needs. Besides attempting to develop in this study an initial prompt that would produce isolated speech, we also investigated the use of a reprompt to be used if the speech given in response to the initial prompt was not adequately isolated. Each of 90 subjects (60 Bell Laboratories employees and 30 from the surrounding community) heard one of a set of prompts over the telephone and responded with their home telephone numbers. Because of the concerns of the larger project, subjects were asked to give their numbers without the area codes.

#### *4.3.1 The initial prompt*

Based on the results of the previous study, we selected two variations

of each of two prompts for evaluation as the initial prompt. The candidates were:

1. At the tone, please say your number. Pause briefly between digits.
2. At the tone, please say your number. Pause briefly after each digit.
3. At the tone, please say your number. Pause between digits, like this. (Recorded voice speaking, "One, two, three")
4. At the tone, please say your number, as follows: (Recorded voice speaking "One, two, three")

The phrase "say zero instead of oh" was not used because reliable recognition of "oh" by an isolated speech system is not as difficult as it would be with a connected speech system, due to the absence of coarticulation effects. Also, in an attempt to keep the prompts as short as possible, we did not include a sentence telling customers that they would speak to a machine.

#### *4.3.2 The reprompt*

To evaluate the effect of a second attempt, a subject whose speaking time did not exceed 5.0 seconds on the first attempt received a second prompt, as follows:

We're sorry, would you please say your number again, but pause longer between digits.

#### *4.3.3 Evaluation of isolation*

Since the central question in this study was whether subjects were producing isolated speech, we needed a definitive test of isolation. This was provided by recording subjects' speech on analog tape and sending the tapes to the Bell Laboratories Acoustics Research Department at Murray Hill, New Jersey, where they were processed for end-point detection. Of primary interest were the number of isolated segments detected in each subject's speech. Since subjects spoke their own seven-digit home telephone numbers, seven segments should be detected on a number spoken with correctly isolated speech.

#### *4.3.4 Summary of results*

After gathering data from 40 subjects, 10 for each prompt, it was evident that Prompt 4 was unacceptable. Four of the ten subjects hearing this prompt responded with three digits or expressed confusion. Prompt 4 was therefore eliminated from further consideration. Data were then collected from an additional 30 subjects, 10 for each remaining prompt. After an initial evaluation of these data, 20 more subjects were added to better discriminate between Prompts 1 and 2.

For Prompt 1, 17 of the 30 subjects took longer than 5.0 seconds to say their telephone numbers on the first attempt and were therefore

not given the reprompt. The end-point detector found seven isolated segments in 14 of those 17 first attempts and six segments in each of the remaining two (one was missing from the tape). Of the 13 subjects given the reprompt, seven segments were detected on the first attempt for two subjects and on the second attempt for 12. The speech level was too low to segment on the second attempt of the remaining subject.

For Prompt 2, 22 of the 30 subjects were not given the reprompt. Of these, 18 achieved perfect isolation as determined by the end-point detector. Five segments were detected for two subjects, six for another, and again one subject was missing from the tape. None of the eight subjects given the reprompt spoke seven isolated segments on the first attempt. Six of the eight spoke perfectly isolated digits after the reprompt. One of the remaining two gave three segments and the other gave four.

For Prompt 3, 19 of the 20 subjects were not given the reprompt. Of these, 14 were determined to have spoken seven isolated digits. One spoke seven digits in six segments and another in five. The end-point detector found five digits in five segments for two subjects, with the speech level on the remaining two digits being too low to segment. For a third subject, there were six isolated digits, plus one with too low a level. The one subject given the reprompt spoke six segments on the first attempt and seven on the second.

The results of this study are an encouraging indication that the combination of an initial prompt with a reprompt can be effective in eliciting isolated speech. The differences in effectiveness among the initial prompts are not large, but are in favor of Prompt 3. However, since that prompt is considerably longer than either of the others, use of one of the comparable shorter prompts is preferable in any real application.

## V. CONCLUSION

The work reported above represents the initial human factors steps toward eventual use of ASR in Bell System network applications. That effort has been concentrated on those human factors questions surrounding the user-system dialog. As indicated in Section II of this paper, there are many other human factors issues that must be faced before any network application of speaker-independent ASR will be possible. However, our work on those remaining issues is beyond the scope of this paper. Also beyond the present scope are those potential telecommunications applications of both speaker-dependent and speaker-independent ASR outside the network. This broader range of applications raises several human factors questions in addition to those considered here.

## VI. ACKNOWLEDGMENTS

Several people deserve thanks for their contributions to the work summarized here. H. Holinka assisted in all of the studies. L. Chapman, a visiting student from the University of Texas at El Paso, helped plan and collect data for the laboratory simulation. T. M. Gruenenfelder was responsible for analysis of the data from that study. M. L. Viets was heavily involved in all phases of the telephone prompt and speech isolation studies. C. J. Karhan served as primary coordinator for the project of which the speech isolation study was a part. I would also like to thank L. R. Rabiner and J. G. Wilpon of the Acoustics Research Department at Bell Laboratories for evaluating the isolation of the recorded speech. Finally, special thanks are due E. A. Youngs, who initiated the human factors effort reported here and has been a constant source of ideas and encouragement.

## REFERENCES

1. T. B. Martin, "Practical Applications of Voice Input to Machines," Proc. IEEE, *64* (April 1976), pp. 487–501.
2. J. M. Baker, "How to Achieve Recognition: A Tutorial/Status Report on Automatic Speech Recognition," Speech Technology, *1*, No. 1 (Fall 1981), pp. 30–43.
3. C. S. Meyers and L. R. Rabiner, "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition," B.S.T.J., *60*, No. 7, Part 2 (September 1981), pp. 1389–1409.
4. A. E. Rosenberg, L. R. Rabiner, and J. G. Wilpon, "Recognition of Spoken Spelled Names for Directory Assistance Using Speaker-Independent Templates," B.S.T.J., *59*, No. 4 (April 1980), pp. 571–92.
5. S. E. Levinson, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 2 (April 1979), pp. 134–41.
6. L. R. Rabiner, private communication.

## AUTHOR

**John E. Holmgren,** B. S. (Psychology), 1965, University of Wisconsin; Ph.D. (Mathematical Psychology), 1970, Stanford University; Bell Laboratories, 1979–1982; American Bell, 1982—. In the Human Factors Department at Bell Laboratories, Mr. Holmgren worked on potential applications of automatic speech recognition. His other major responsibility was in the area of audiographics teleconferencing. In July 1982, Mr. Holmgren transferred to American Bell, where he is a Group Supervisor in AIS Net 1000 Research and Development. Member, Psychonomic Society.

# New Functions for Technology

This section describes the *UNIX*™ Writer's Workbench software, its philosophy (L. T. Frase), the programs (N. H. Macdonald), and people's response to the programs (P. S. Gingrich). The programs do many things that an editor does when proofreading and commenting on a paper, and they have stimulated much public interest. The programs have been discussed on the *Today* show, and in articles in *Time*, *Discover*, and other magazines and newspapers. Why are the programs so interesting? Part of the answer is that they relieve us from the drudgery of proofreading. But part of the answer is also that such programs come close to helping as a human would, without the embarrassment of personal criticism.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# The *UNIX*™ Writer's Workbench Software: Philosophy

By L. T. FRASE*

(Manuscript received December 17, 1981)

Technology has dramatically increased the number and complexity of written documents. The Bell System spends over $100,000,000 annually on technical documents, with much effort devoted to review and revision. The *UNIX*™ Writer's Workbench programs assist documentation by automating copy editing and proofreading tasks. These programs deliver detailed measures and comments about text readability, punctuation, word use, abstractness, and other features. Authors use the system to evaluate draft documents, and many feel the programs improve their writing skills. The programs have been used for quality control and text research, and in writing courses. This paper discusses principles used in system development, introduces the two following papers, and suggests possible uses for automated language analysis aids.

## I. INTRODUCTION

Massive growth in the number of documents and in the sophistication needed to understand technical language has created challenges for many organizations, outside and within the Bell System. The Naval Air Systems Command, for instance, supplies technical manuals

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

for 135 aircraft. There are over 25,000 manuals, totaling 3,000,000 pages. In 1950, the manuals for one aircraft contained fewer than 2,000 pages; today, the manuals for one aircraft contain nearly 300,000 pages.[1] In the Bell System, there are over 100 general categories of documents. Just one of these categories, the Bell System Practices, contains 400 broad document categories, with 35,000 active titles. Current issues are routinely distributed to 45,000 locations. But number of pages is not the only concern. In some states, if documents are relevant to the physical or financial well-being of individuals, laws dictate how difficult the language of a text may be. Recent "plain language" legislation, passed by six states, allows courts to rewrite texts that are unclear.[2]

The telecommunications industry, along with others, is faced with two substantial problems: (1) how to manage a large number of documents, and (2) how to make them understandable. Review and evaluation of technical documents is costly, time-consuming, subjective, and often tedious. Mechanized aids for reviewing documents, although not relieving humans of responsibility for document quality, can simplify and speed the documentation process. Efforts elsewhere in Bell Laboratories address the problem of managing a large number of documents.[3,4] The *UNIX** Writer's Workbench software was designed to help improve quality.

Until recently, document technology has consisted of two parallel, but separate, lines of work. The first consists of experimental research on what makes text easy or difficult to understand. This work, reflected in psychological and linguistic research, includes studies of the effects of text format, wording, and graphic displays.[5] The second line of work includes more intuitive or prescriptive bases for writing and text design. This involves the expertise of text and graphics designers,[6] such as technical editors and layout specialists, as well as rhetorical traditions that specify writing standards.[7] Both knowledge bases, one derived from experimental research and the other from nonexperimental sources, provide useful guidelines for effective text design.

Unfortunately, text researchers and designers have not always worked closely together. Psychological theories of what makes a document difficult to understand often seem oversimplified to designers, and designers' intuitions and prescriptions often seem vague to researchers. Designers have been concerned with creating good documents, while researchers have been trying to understand problems of text comprehension.

The use of computers for language analysis is beginning to stimulate

---

* Trademark of Bell Laboratories.

integration of technical knowledge from research and design areas. McMahon, Cherry, and Morris,[8] for instance, have shown how language analysis, such as vocabulary analysis or assigning parts of speech to words, can be done easily and accurately by computer. Humans find those tasks difficult and tedious. Models of decision processes used by experts have been incorporated into programs that make design recommendations, such as the most suitable column width to use for a particular text.[9] Knowledge-based expert systems,[10] developed by researchers in artificial intelligence, provide additional ways of expressing, in computer-readable language, expertise derived from different disciplines. Development of the Writer's Workbench system has required, and continues to encourage, this contact between different disciplines. The programs recognize the validity of principles derived from research and the validity of principles derived from intuition and practical experience. Hence, they help bring research and application closer together.

The Writer's Workbench system provides a simple set of commands that deliver many assessments needed in documentation work. These include editorial comments on punctuation, word use, spelling, and text abstractness, along with analyses of grammatical parts of speech and calculations of overall text readability (expressed as reading grade levels). The programs combine work done by the Computing Science Research Center and the Human Performance Engineering Department of Bell Laboratories.

Using routines already developed for *UNIX* software, such as spelling and parts of speech analysis programs, developers were able to build programs that expanded and went beyond existing facilities. For portability, the wide distribution of *UNIX* operating systems in Bell Laboratories was an advantage. An early version of the programs was tried on several *UNIX* systems for four months before the first general release within Bell Laboratories. During those months, program efficiency, outputs, and options were modified in response to user feedback. Although current versions of the programs incorporate further enhancements and pruning of esoteric programs, the system architecture has remained the same.

Text to be analyzed must be stored in the computer. Some authors prefer to compose text on line; others have it entered by clerical staff. Text should contain standard formatting commands, for instance, commands that define headings and other special elements. To analyze a text, the user logs in to the system and issues a command name followed by the name of the file that contains the text. The program then executes, prints analyses, and perhaps suggests changes to improve the text. Currently, the programs are appropriate for proofreading and stylistic analyses of expository and descriptive prose, but not

for procedural documents consisting entirely of lists or abbreviated step-by-step directions.

The Writer's Workbench system is limited by the text features that the programs can recognize, and also by our understanding of the relation of text features to reading difficulty. For instance, one program recognizes when the first word in a sentence is not capitalized; it is more difficult to develop a program that can recognize when important content is missing. Currently, the programs do not use linguistic parsing algorithms; they cannot "understand" the subject and object of a sentence, for example, as humans would. Nevertheless, the richness and variety of feedback delivered by the programs are different enough from commonly encountered word and text processing systems that we find it useful to refer to these expanded capabilities as "language processing" functions. Hence, we have used this terminology here and in the following two papers, perhaps as much to remind ourselves of what remains to be done as to characterize the current programs.

## II. DESIGN PRINCIPLES

Six specific principles, reviewed below, guided Writer's Workbench program development. The principles provide a context for the detailed program descriptions in the paper that follows this.[11] To summarize, the principles assume that language analysis programs should be rational, diverse, evaluative, modifiable, specific, and informative.

### *2.1 Expert knowledge base*

The Writer's Workbench programs are *rational* in the sense that they are based on research or expert consensus that justifies the relevance of program information for text comprehension or use. Some program outputs are based on psychological research, showing, for instance, that passive sentences are more difficult to understand than active sentences.[11] Other programs are based on advice from writing and style experts.[7,12] For instance, standard punctuation usage is evaluated by some programs. Knowledge about the features of good writing is scattered among disciplines; hence, no single discipline is entirely adequate for formulating language analysis programs. Currently, we have drawn from rhetorical and psychological literature to provide text assessments that people commonly use in document evaluation. To do more, contributions from artificial intelligence and related fields are required. In addition, since not all features of document design can be clearly supported by research or expert opinion,[13] programs contain messages directing users to information sources that explain certain evaluations.

### *2.2 Program diversity*

Written communication is so complex that single measures, such as an overall index of text readability, can draw an author's attention away from other important text characteristics. Therefore, we wanted to provide *diverse* measures so users would think about a constellation of features. For instance, one program delivers over 40 measures of text characteristics, some measuring word characteristics, others measuring sentence characteristics.

We assume that authors have diverse needs, which cannot be foreseen entirely. Hence, the Writer's Workbench system contains many programs and options that can be used, or not, as an author desires. The diversity of programs works as a structured system to let users explore characteristics of words (such as syllables), sentences (such as length in words), or paragraphs (such as beginning and ending sentences).

### *2.3 Relative judgments*

A text is good or bad depending on whether it is adequate for a particular task or audience. *Evaluative* judgments about a document usually relate to some standard of excellence. The Writer's Workbench system makes such comparisons explicit. Measures from one document are compared to measures obtained from other documents. For instance, evaluative comments about the readability of a text can be based on statistics derived from training or technical memoranda judged to be of high quality. The author is warned when the text being analyzed departs significantly from the standard documents. Ability to compare documents to various standards emulates an important component of human text judgments, and it is a distinctive feature of the Writer's Workbench system.

### *2.4 User modifications*

Since it is not possible to anticipate the documentation needs of all users, programs should be *modifiable*. Three aspects of the programs can be adjusted by the user—standards, input, and output. For instance, one program calculates statistics for variables derived from texts supplied by an author. These statistics can then be used as standards to evaluate features of the author's subsequent texts, such as percentage of passive sentences. Other programs allow adding or deleting words or phrases that are detected by spelling and word usage dictionaries. In addition, input can be modified to include or exclude lists from analyses, and output format and length can be changed by appending simple descriptors to commands.

### *2.5 Text location*

For authors to review exactly where language may have gone awry in a text, *specific* text locales must be identified. For instance, printing an average measure of sentence length is less useful for an author than printing sentences that exceed a specified length. Early work on identifying troublesome text locales for a reader was done by Koether and Coke.[14] Command options in the Writer's Workbench system provide summary statistics or review sentences that violate certain standards, such as length or word use. In the latter case, line numbers show where offending text can be found. This level of specificity is necessary if programs are to speed text revision.

### *2.6 Information resources*

The Writer's Workbench system is intended to be *informative* rather than to make rigid decisions. Text statistics should be interpreted cautiously, and we suspect they will be used most effectively when an author can get advice related to those statistics. The Writer's Workbench system provides guidance, using information files, about word use, punctuation, and other features. Brief explanatory comments are also given in programs, with information about how to obtain more information.

## III. RATIONALE AND DESIGN

"*UNIX*™ Writer's Workbench Software: Rationale and Design,"[11] describes program architecture, and discusses at length the literature behind programs and their options. Current literature on writing is rich in descriptions of text patterns that characterize faulty expression,[15] and contains descriptions of actions that a writer can take to correct those faults. These patterns and actions, which are the bases for program measures and recommended revisions, are described in the first part of the next paper. Next, the individual programs and their options are described. The programs are divided into those that proofread, those that comment on style, and those that comment on organization. Finally, features that help people use the programs—human factors issues—are surveyed.

## IV. RESULTS OF A FIELD STUDY

The Writer's Workbench system has been used in Bell Laboratories for over three years; currently it runs on over 100 *UNIX* systems. Bell Laboratories users have responded favorably to the programs, and many believe the programs improve writing skills.[16]

Of course, there are many environments in which language analysis programs might be used. "*UNIX*™ Writer's Workbench Software:

Results of a Field Study,"[17] describes data collected from program trials conducted outside Bell Laboratories. The two outside locations included one human factors organization and one technical writing organization. The groups were engaged in different writing tasks and the staffs had different computer experience.

According to the field study, authors liked the detailed computer suggestions about how to revise their texts, and they were better able to detect errors with program help than without it.

## V. CONCLUSIONS

The Writer's Workbench programs do not do everything humans do when evaluating text. They cannot determine whether important content is missing, nor do they assess format features or the meaning of texts, as humans would. Nevertheless, the programs speed editing, and they help detect features that authors might otherwise miss.

Language analysis programs have many uses. Components of the Writer's Workbench system have been used for research; for instance, to derive general quantitative descriptions of text style and to relate these styles to text difficulty.[18] They have also been used, by various groups at Bell Laboratories, to help develop technical documents for new software systems and for writing technical memoranda. Programs could be used to obtain management information, for example, to monitor changes in documents across time, and to determine the match between skills of readers and the texts they must read.[19] Finally, the programs can be used for instruction to provide detailed feedback to students about the characteristics of their writing, or by an instructor to evaluate the effects of instruction on student writing.

The Writer's Workbench programs have recently been used as an adjunct to a technical writing course sponsored by the Bell Laboratories Education Center. Such programs could be especially useful to authors whose native language is other than English. Use of language analysis programs in research, development, management, and instruction will no doubt grow as these programs increase in diversity and sophistication.

## VI. ACKNOWLEDGMENTS

Programs created by L. L. Cherry contributed much to the development of the Writer's Workbench system. She also helped with program enhancements. W. Vesterman, of Rutgers University, stimulated early work on stylistic and diction programs. S. A. Keenan made important contributions to program development and documentation, as well as to Writer's Workbench system trials. J. L. Collymore, M. L. Fox, and M. F. Poller also contributed to development, and exchanges with W. F. Fox and J. J. Dever were helpful.

## REFERENCES


1. W. G. Muller, "Useability Research in the Navy," in T. G. Sticht and D. Zapf (Eds.), *Reading and Readability Research in the Armed Services*, Alexandria, VA.: Human Resources Research Organization, 1976.
2. "Plain Language Laws," in "Simply Stated," Newsletter of the Document Design Center, Washington, D.C.: American Institutes for Research, No. 18(July, 1981), p. 1.
3. M. J. Rochkind, "The Source Code Control System," IEEE Trans. Software Eng., *se-1*, No. 4(December 1975), pp. 364–70.
4. M. H. Bianchi, R. J. Glushko, and J. R. Mashey, "A Software/Documentation Development Environment Built from the UNIX Toolkit," Int. Fed. Inform. Processing, Working Conf. Automated Tools Inform. Syst. Design and Development (January 1982), New Orleans, LA.
5. J. Hartley, "Eighty Ways of Improving Instructional Text," IEEE Trans. Professional Commun., *PC-24*, No. 1(March 1981), pp. 17–27.
6. E. S. Ferguson, "The Mind's Eye: Nonverbal Thought in Technology," Science, *197*, No. 4306(August 1977), pp. 827–36.
7. R. A. Lanham, *Revising Prose*, New York: Charles Scribner's Sons, 1979.
8. L. E. McMahon, L. L. Cherry and R. Morris, "UNIX Time-Sharing System: Statistical Text Processing," B.S.T.J., *57*, No. 6(July–August 1978), pp. 2137–54.
9. S. A. Keenan, "Computer Projections of the Cognitive Effects of Text Changes," Amer. Educ. Res. Assn., Boston, MA, April 7–11, 1980.
10. R. O. Duda and J. G. Gaschnig, "Knowledge-Based Expert Systems Come of Age," Byte, *6*, No. 9(September 1981), pp. 238–81.
11. N. H. Macdonald, "*UNIX*™ Writer's Workbench Software: Rationale and Design," B.S.T.J., this issue.
12. W. Strunk, Jr. and E. B. White, *The Elements of Style*, New York: Macmillan, 1972.
13. L. T. Frase, "Ethics of Imperfect Measures," IEEE Trans. Professional Commun., *PC-24*, No. 1(March 1981), pp. 48–50.
14. M. E. Koether and E. U. Coke, "A Scheme for Text Analysis Using FORTRAN," Amer. Educ. Res. Assn., New Orleans, LA, February 25–March 1, 1973.
15. L. Flower, *Problem-Solving Strategies for Writing*, New York: Harcourt Brace Jovanovich, Inc., 1981.
16. L. T. Frase, N. H. Macdonald, P. S. Gingrich, S. A. Keenan and J. L. Collymore, "Computer Aids for Text Assessment and Writing Instruction," N.S.P.I. Journal, (November 1981), pp. 21–4.
17. P. S. Gingrich, "*UNIX*™ Writer's Workbench Software: Results of a Field Study," B.S.T.J., this issue.
18. E. U. Coke, unpublished work.
19. E. U. Coke and M. E. Koether, "A Study of the Match Between Technical Document Difficulty and the Reading Skills of Technical Personnel," B.S.T.J., this issue.


## AUTHOR


**Lawrence T. Frase,** A.B. (Philosophy), 1960, University of Miami; M.S., 1963, Ph.D. (Educational Psychology), 1965, University of Illinois; University of Massachusetts (Psychology Department), 1965–1967; Bell Laboratories, 1967–1976; National Institute of Education, 1967–1968; Bell Laboratories, 1978—. Mr. Frase has published research on human reasoning, reading, writing, and document design. While on leave from Bell Laboratories, he was head of the Learning Division at the National Institute of Education in Washington, D.C. Currently he is working on applications of computers to human information processing. He is a fellow of the American Psychological Association.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July-August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# The *UNIX*™ Writer's Workbench Software: Rationale And Design

By N. H. MACDONALD*

(Manuscript received December 17, 1981)

The *UNIX*™ Writer's Workbench software is a set of computer programs that help with two stages of document production: evaluation and editing. These programs analyze prose documents and suggest improvements. There are several types of programs: those that proofread, analyze style, and reformat the text in new ways, and those that provide information about the English language. This paper first describes the rhetorical and psychological writing principles that underlie the Writer's Workbench programs. It then describes the major Writer's Workbench programs and how they judge writing, based on these writing principles. Finally, it presents the human factors principles used in the design and development of the Writer's Workbench system.

## I. INTRODUCTION

The previous paper in this issue of the *Journal*[1] pointed out the growing need for automated language processing, that is, for tools to help authors write clearly and understandably. Such tools are especially important for technical writing where the content is precise and a reader's failure to understand the text can be costly.[2] This paper

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

describes how the *UNIX** Writer's Workbench software attempts to meet this need.

The first major section of this paper describes the principles of good writing that are incorporated into the Writer's Workbench programs. Many of these principles are supported by research showing that, indeed, different writing styles make a difference to the reader. The next section describes the major Writer's Workbench programs and relates them to the principles. The last major section describes the user considerations that guided the design of the Writer's Workbench programs.

## II. SOME PRINCIPLES OF GOOD WRITING STYLE

This section presents some principles of good writing style. Research suggests that text that violates these principles is more difficult to comprehend. These principles belong to the word, sentence, paragraph, or document level, but of course, some principles overlap categories.

### *2.1 Word level*

There are several word level principles besides the obvious ones such as using and spelling words correctly.

#### *2.1.1 Wordiness*

Style guides usually advise writers to avoid hackneyed, empty, or frequently misused phrases, such as "at this point in time" and "notwithstanding the fact that."

#### *2.1.2 Definite, specific, concrete language*

Strunk and White[3] remark that "the surest way to arouse and hold the attention of the reader is by being specific, definite, and concrete." They illustrate this principle by this pair of sentences, the first vague, the second concrete:

> A period of unfavorable weather set in.
> It rained every day for a week.

Psychological research on memory and the imageability (ability to create an image) of words suggests that texts with many abstract words will be more difficult to remember, and presumably to understand, than texts with concrete words.[4-8]

#### *2.1.3 Word frequency and length*

Coleman reported a high negative correlation between the average

* Trademark of Bell Laboratories.

frequency of content words in a passage and its difficulty to readers.[9] That is, texts made up of many infrequent words are difficult to comprehend. He also found that difficult passages contained more letters, syllables, and morphemes per word. Since frequency and length are highly correlated,[10] these two effects are possibly one effect. In keeping with these findings, Klare recommends Anglo-Saxon-based words over Latin-based words in English since the Latin form is usually longer, as in "go" versus "proceed."[11]

### *2.2 Sentence level*

There are several obvious sentence-level principles, for instance, using correct punctuation and correct grammar. Some of the less obvious or more subjective principles are described below. These are not principles of right or wrong, but rather of better or worse.

#### *2.2.1 Passive voice*

One of the biggest writing problems, particularly in scientific writing, is the overuse of the passive voice. Historically the passive voice was used to indicate the objectivity of science and the scientist. The scientist did not state, "I found that. . .," but rather, "It was found that. . . ." This usage was dogma; as Einstein said, "When a man is talking about scientific subjects, the little word 'I' should play no part in his exposition."[12]

One problem with the passive voice in scientific materials is that its use has spread from obscuring "I" and "we" to many other cases as well, e.g.,

> A variable-gain control is included in this circuit.

Scientific objectivity is still served by stating:

> This circuit includes a variable-gain control.

Perhaps because of a change in scientists' perceptions of their role,[13] or perhaps because of the difficulty of the passive style, many scientists and scientific editors[14,15] now recognize and even promote the use of the active voice, including the use of first-person pronouns.

Why should we avoid the passive voice? The rhetoric books describe it as "dron[ing] like nothing under the sun," wordy and unclear,[16] and less direct and less vigorous[3] than the active voice. It may indeed be all those things, but in addition, psychological research has shown that the active versions of a sentence are recalled better[17] and verified faster.[18] Scientific texts written in the third person passive, as "It was concluded that . . ." are remembered less well and appreciated less than the same content written in the active voice.[19]

Kirkman[20] took samples of scientific papers and rewrote the content in six different styles. He asked scientists and engineers to rate which version they found "most comfortable to read, easiest to grasp and simplest to digest." In three different surveys, he found they preferred "direct, active writing, with a minimum of specialist vocabulary, a judicious mixture of personal and impersonal constructions, short and uncomplicated sentences and liberal paragraphing."

These data do not imply that passive voice should never be used; at times it is preferable to

1. Emphasize the object of the sentence
2. Vary the rhythm of the text
3. Avoid naming an unimportant actor.

EXAMPLE: The mail was delivered.

However, the passive voice should be restricted to the useful and necessary cases, rather than used widely and indiscriminately.

#### *2.2.2 Nominalizations*

Nominalizations are nouns that have been created from verbs. They usually end in "ion," "ment," "ence," or "ance," e.g., "transformation," "establishment," and "admittance." The empirical case against using nominalizations is strong. Coleman[21] found that individual sentences without nominalizations were remembered better than their nominalized forms. A multiple-choice comprehension test on content failed to show a significant difference between the two versions. But on a memory task, subjects required significantly more exposures to memorize sentences containing two nominalizations than to memorize the same content written in active-verb form.

In a later experiment, Coleman[17] investigated ten different types of nominalizations and their active versions. He found that for those pairs of sentences in which the active version was phrased in two clauses and the nominalized version in one, the active was memorized more quickly. For example

ACTIVE: If he discusses the reason for the price-change, it will be appreciated.

NOMINALIZED: His discussion of the reason for the price-change will be appreciated.

Coleman[21] took passages from a psychology text and rewrote all the nominalizations, passive sentences, and adjectivalizations (verbs formed from adjectives) into active sentences. (Coleman[17] found no difference between adjectivalizations and their active forms.) He found

that students answered correctly 25 percent more questions from the rewritten texts than from the originals. In a similar experiment, subjects were asked to write the passages immediately after reading them. Subjects recalled the simplified passages significantly better than the originals.

#### *2.2.3 Expletives*

In grammar, the term expletive refers to a syllable, word, or phrase that adds no information. In particular "expletives" are words such as "it" or "there," which anticipate a later word or phrase. Thus, in "There are three solutions to this puzzle," "There" is an expletive anticipating "solutions."

Many times such expletives can be deleted, e.g., "This puzzle has three solutions," although sometimes they cannot, e.g., "It is raining." Although no research demonstrates that expletives make text more difficult, Brogan[22] argues that when the expletive deemphasizes the main verb inappropriately, the sentence should be changed. For instance,

> It is this necessity that adds to their complexity.

can be changed to

> This necessity adds to their complexity.

making "adds" more salient.

### *2.3 Paragraph level*

#### *2.3.1 Readability*

The readability or reading grade score for a text predicts how many years of schooling a reader would need to understand it. (Units other than years of schooling are sometimes used.) The prediction is usually based on the length of the words in the text and the length of the sentences. Different readability formulas calculate the lengths differently and weight the factors differently.

As we mentioned in the previous section, the length of a word (highly related to its frequency) predicts its difficulty. Sentence length is related to sentence type, with complicated sentences usually containing more words than simple ones. Readability formulas predict reasonably well the difficulty of the text, not because sentence length and word length cause reading difficulty, but because they are highly correlated with features such as complexity and frequency, which do.

As with any predictor, these formulas can be fooled. The Dale-Chall formula[23] takes vocabulary items into consideration, but most formulas

do not and will provide incorrect readability scores for nonsense text. All the formulas will give the same values for text with the sentences input backwards or forwards. But, in general, the formulas give a reasonable prediction of text difficulty when presented with naturally written text.

Unfortunately, research has shown[11] that the comprehensibility of the text is not necessarily improved, although the reading grade score is, by simply shortening words and sentences. The best way to improve the comprehensibility of a text is to rewrite it following the principles of good writing.

#### *2.3.2 Variation*

In writing, as in most fields of endeavor, moderation is best. The previously described principles are not absolutes; some passives and nominalizations are reasonable, and in fact, variation in sentence length, structure, and type is necessary to make writing interesting and keep the reader's attention.[13] There are other more important reasons to vary sentence type, which usually varies length as well.

Writing instructors suggest that less important ideas should be grammatically subordinated to more important ones so that the grammatical structure emphasizes the logical structure. Two simple sentences can be joined by using a "that" clause or an adverb, such as "although," to subordinate one to the other. The less important sentence should be in the subordinate clause after the "that" clause or adverb. For example, the following sentences:

1. The short, simple sentence is the most comprehensible form for an individual sentence.
2. Overusing such sentences may make a document seem disjointed.

can be combined:

> Although the short, simple sentence is the most comprehensible form for an individual sentence, overusing such sentences may make a document seem disjointed.

The combined sentence subordinates sentence (1) to sentence (2), thus emphasizing that the information in sentence (2) is more important than that in sentence (1).

### *2.4 Document level*

#### *2.4.1 Organization*

Most books on writing recommend that the first sentence of each paragraph present the topic of the paragraph or else provide a transition from the previous paragraph into a new topic.[3,13] If, for most paragraphs, the first sentence reflects the topic, then these sentences give the reader a reliable signpost to the meaning. Headings also provide signposts to topics and topic changes. A paper with good

headings and topic sentences can be skimmed easily and quickly, and even the person who reads every word will find it easier to follow.

#### *2.4.2 Audience considerations*

Style books strongly advise writers to know their audience and to write for them. This is particularly important for materials such as instruction manuals, which the reader needs to understand. Writing for the reader extends from questions of vocabulary and sentence structure to content and organization. For content and organization, Flower[24] gives particularly thorough advice.

## III. PROGRAMS

Until recently, students of writing could use only books and teachers to help them. This has slowly been changing with the advent of computer programs to do some of the work. Most programs, however, have focused solely on checking and correcting spelling.[25] Several readability indices have also been automated, but in general, wire services, magazines, newspapers, and businesses still do not analyze their text with the computer, although it is often stored and edited in a computer.

Although not yet in widespread use, several text analysis systems do exist. Besides the Writer's Workbench programs[26,27] to be described here, other systems include EPISTLE,[28] an IBM project still in the research stage; JOURNALISM,[29] a University of Michigan system that provides feedback to journalism students; and CRES,[30] a Navy system to help improve the quality of technical manuals and training materials. The EPISTLE system is a business office system that will abstract the contents of incoming letters but will also correct grammatical errors in outgoing letters. JOURNALISM provides some proofreading, but because it is programmed with specific knowledge about the articles it evaluates, it is able to comment on the organization and content as well. CRES calculates the readability score for the text, flags uncommon and misspelled words and long sentences, and suggests simple replacements for difficult words and phrases.

The rest of this section describes some of the Writer's Workbench programs, focusing on those that most strongly reflect the principles discussed in Section II.

### *3.1 Proofreading: proofr*

The most useful Writer's Workbench programs are in some ways the least interesting. Every writer can use proofreading help, since as we become more familiar with a piece of writing we become poorer at spotting errors in it.

The proofreading program, proofr, invokes five separate programs.

A three-line example of input text and its proofr output are shown in Fig. 1. Each of these five programs can be run individually, but are more conveniently run as a package. When run separately, some of the programs have capabilities not found when they are run as part of proofr. Each separate program will be discussed in turn.

### *3.1.1 Spelling: spellwwb*

The spellwwb program, a spelling checker based on the *UNIX* system spell program,[31] allows users to have a personal dictionary of words.

```
INPUT:    Our report, "The Basic Fundamentals of Organizatonal Complexity",
          is enclosed.  Please send any recomended changes at your
          earliest convenience. thanks.

PROOFR    ********** SPELLING **********
OUTPUT:   Possible spelling errors in examplefile are:

          Organizatonal                recomended

          If any of these words are spelled correctly, later type
                    spelladd word1 word2 ... wordn
          to have them added to your spelldict file.

          ********** PUNCTUATION **********

          The punctuation in examplefile is first described.

          2 double quotes and 0 single quotes
          0 apostrophes
          0 left parentheses and 0 right ones

          The program next prints any sentence that it thinks is incorrectly
          punctuated and follows it by its correction.

          line 1
          OLD: Our report, "The Basic Fundamentals of Organizatonal Complexity",
          NEW: Our report, "The Basic Fundamentals of Organizatonal Complexity,"
          line 3
          OLD: earliest convenience. thanks.
          NEW: earliest convenience. Thanks.

          For more information about punctuation rules, type:

                    punctrules

          ********** DOUBLE WORDS **********

          For file examplefile:

          No double words found

          ********** WORD CHOICE **********

          Sentences with possibly wordy or misused phrases are listed next,
          followed by suggested revisions.

          beginning line 1 examplefile
          Our report, "The *[ Basic Fundamentals]* of Organizatonal Complexity",
          is enclosed.

          beginning line 2 examplefile
          Please send any recomended changes *[ at your earliest convenience]*.

          file examplefile: number of lines 3, number of phrases found 2

          ---------- Table of Substitutions ----------

          PHRASE                    SUBSTITUTION

          at your earliest convenience: use "soon" for " at your earliest convenience"
          basic fundamentals: use "fundamentals" for " basic fundamentals"

          ********** SPLIT INFINITIVES **********

          For file examplefile:

          No split infinitives found
```

Fig. 1—Input to and output from proofr program.

The program searches the input text and prints all words that are not in its dictionary or the user's dictionary.

In addition, the spellwwb program can be used interactively to correct spelling errors. The spellwwb program prompts the user with each possibly misspelled word. Responding to each, the user can:

1. Find all lines on which it appears in the file
2. Invoke another program to try to determine the correct spelling
3. Tell the program how to change the spelling
4. Leave it as it is
5. Save it in a personal dictionary of correct words.

The user can also specify that certain misspellings always be changed. For instance, a poor speller might store the correction "relevant" for "relevent." Then spellwwb makes all specified corrections and updates the user's personal spelling file.

#### *3.1.2 Punctuation: punct*

The punct program searches for simple punctuation errors. It recommends changes to:

1. Move commas and periods to the left of double quotes, and move semicolons and colons to the right of double quotes
2. Capitalize the first letter of sentences
3. Balance double and single quotes and parentheses.

The program enforces straightforward rules, not ones that require judgment, such as deciding whether a comma or semicolon is the appropriate mark. When punct finds an error, it prints the original line, followed by its correction of the line.

Some of the rules punct enforces are unfamiliar to many people, such as the relative position of double quotation marks with other marks of punctuation. Nevertheless, they are accepted standards of American English. The punct program directs a user who has made punctuation errors to use the program punctrules, which provides a list of pertinent punctuation rules. The user, thus, has easy access to reference information, which can be used to decide whether the suggested changes are appropriate.

#### *3.1.3 Consecutive occurrences of the same word: double*

Using context line editors for editing text increases the chance of having the same word twice in a row. The double program locates consecutive occurrences of the same word, which can be split across two lines.

#### *3.1.4 Wordy phrasing: diction*

The diction program, described by Cherry,[32] searches a text file for phrases that writing experts have classified as wordy or frequently

misused. The latest version of its dictionary also contains some phrases that may reflect a sexual bias. The program prints sentences containing such phrases and surrounds them with stars and brackets (*[ ]*). It then recommends substitutions for these phrases. For instance, for the phrase, "bring to a conclusion," it recommends using "conclude," "end," or "finish."

Users should have the flexibility of tailoring the word list to their environment. In general, "end," "stop," or "finish" may be good choices for "terminate," but for many people writing about software, "terminate" is the appropriate word. Such a user can direct the program to stop searching for "terminate" by creating a personal dictionary. Having a private dictionary of extra terms and suppressed terms increases the usefulness of the diction program.

#### *3.1.5 Split infinitives: splitinf*

The splitinf program uses a "parts of speech" analysis program, parts,[33] to find infinitives that are split by adverbs. Such split infinitives as in "to quickly decide" are the most common type. Since users may be unfamiliar with this error, those whose papers contain split infinitives are told to use splitrules, which will print grammatical information about split infinitives.

### *3.2 Stylistic analyses*

The stylistic analysis programs provide information whose interpretation is less concrete than that given by the proofreading programs. Hence, the information is more difficult to use; following the advice faithfully can require a considerable amount of time.

#### *3.2.1 Tabular stylistic information: style*

The style program,[32] based on parts, provides 71 numbers describing the stylistic features of a text. The most important variables it reports are several readability indices (described more fully by Cherry[32]), information on the average length of the words and sentences, the distribution of sentence lengths, the grammatical types of sentences used, e.g., simple and complex, the percentage of verbs in the passive voice, the percentage of nouns that are nominalizations, and the number of sentences that begin with expletives.

The style output for this article, shown in Fig. 2, is more useful for research on the style of documents, however, than for helping inexperienced writers improve their writing style. The style table is difficult for many writers to interpret because

1. Users may not know the meaning of some terms, e.g., "expletive" and "nominalization."
2. Users frequently do not know whether the numerical values

```
readability grades:
        (Kincaid) 11.3  (auto) 12.6  (Coleman-Liau) 13.1  (Flesch) 13.2 (48.8)
sentence info:
        no. sent 240 no. wds 4636
        av sent leng 19.3 av word leng 5.18
        no. questions 1 no. imperatives 0
        no. content wds 2734  59.0%  av leng 6.72
        short sent (<14) 24% (58) long sent (>29)  9% (22)
        longest sent 64 wds at sent 150; shortest sent 4 wds at sent 70
sentence types:
        simple  42% (101) complex  38% (92)
        compound  7% (16) compound-complex  13% (31)
word usage:
        verb types as % of total verbs
        tobe  32% (170) aux  16% (85) inf  17% (89)
        passives as % of non-inf verbs  14% (63)
        types as % of total
        prep 10.5% (487) conj 3.8% (177) adv 4.2% (197)
        noun 28.0% (1296) adj 17.2% (797) pron 4.7% (220)
        nominalizations  2 % (90)
sentence beginnings:
        subject opener: noun (48) pron (28) pos (1) adj (35) art (57) tot  70%
        prep  13% (32) adv  6% (15)
        verb  1% (3)  sub_conj  6% (14) conj  2% (5)
        expletives  1% (2)
```

Fig. 2—Style program output for this paper.

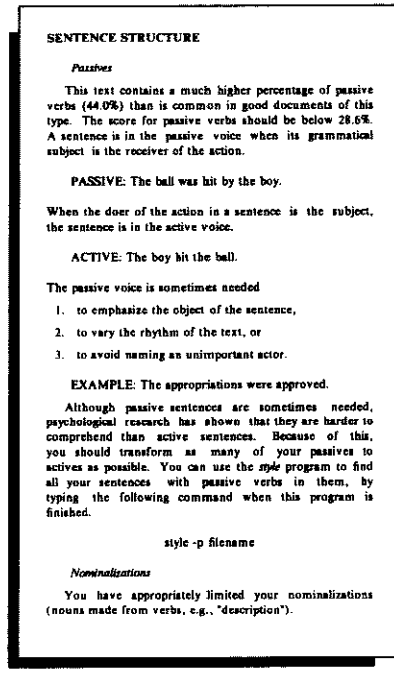
should be high or low, even for terms that are probably familiar, such as "complex sentence."

3. Users who know enough to minimize or maximize the use of some construction still do not know what numerical value is appropriate.

### *3.2.2 Interpreted stylistic analysis: prose*

The prose program goes beyond the style program by providing the style statistics and an interpretation as well. The prose program compares the style values of the user's text against a set of standards and describes the differences in a two-to-three page output written in English sentences. A section of prose output is shown in Fig. 3.

Several sets of standards for comparison are available since texts are written for different types of readers and for different purposes. Users select which set of standards should be used in the interpretation of their text. Currently there are built-in standards for technical papers and prose training materials. These standards were set as follows. Department heads in the Bell Laboratories basic research area were asked to identify the best technical writers in their departments. These people were, in turn, asked to identify their best written technical documents (content aside). This process yielded twenty-eight documents. An editor in a Bell Laboratories training department provided 34 documents that he judged to be particularly well written. The technical and training documents were then run through the style program. The means and standard deviations of each of the style



SENTENCE STRUCTURE

*Passives*

This text contains a much higher percentage of passive verbs (44.0%) than is common in good documents of this type. The score for passive verbs should be below 28.6%. A sentence is in the passive voice when its grammatical subject is the receiver of the action.

PASSIVE: The ball was hit by the boy.

When the doer of the action in a sentence is the subject, the sentence is in the active voice.

ACTIVE: The boy hit the ball.

The passive voice is sometimes needed

1. to emphasize the object of the sentence,
2. to vary the rhythm of the text, or
3. to avoid naming an unimportant actor.

EXAMPLE: The appropriations were approved.

Although passive sentences are sometimes needed, psychological research has shown that they are harder to comprehend than active sentences. Because of this, you should transform as many of your passives to actives as possible. You can use the *style* program to find all your sentences with passive verbs in them, by typing the following command when this program is finished.

style -p filename

*Nominalizations*

You have appropriately limited your nominalizations (nouns made from verbs, e.g., "description").

Fig. 3—Section of prose program output for a poor paper.

variables were computed for each document set. These means and standard deviations now make up the standard for that document type. When the value of a user's variable is more than one standard deviation from the mean of that standard, prose recommends changes. The mkstand program can be used to create additional standards from a set of documents. This flexibility allows any writing group to tailor the standards for its particular audience.

If any of the input text's values is more than one standard deviation from the selected standard, prose explains why this may make the document hard to comprehend and how to rewrite the text to remove the problem. If scores on a variable are acceptable, and the variable is an important one, prose tells the user that the text achieved an appropriate score.

### *3.2.3 Stylistic problems in context: findbe*

Although the style program can isolate individual sentences that contain passive verbs, expletives, or nominalizations, it is frequently difficult to know which sentences to change and how to change them without seeing the surrounding context. The findbe program automates part of a prescription for revision given by Lanham, who tells the

revisor to circle all forms of the verb "to be" and to try to replace them.[34,35] This advice agrees with some of the information presented earlier since "to be" occurs with passives, e.g.,

The difficulty of the passive voice *IS disregarded* by many writers.

with many nominalizations, e.g.,

Coleman's *discussion* of nominalizations *IS* comprehensive.

and with expletives, e.g.,

*There ARE* many ways to avoid expletives.

The findbe program underlines and capitalizes all forms of "to be." The text is then formatted as usual, providing the user with a paper that is easy to read, since all sentences are in context and problem areas are highlighted. This turns out to be a useful way of looking at the first draft of a paper.

#### *3.2.4 Checking text abstractness: abst*

The abst program[36] indexes the conceptual abstractness of the input text. Abstractness is defined here as the percentage of words in the text that also occur on a list of 314 words rated as abstract in psychological research.

When the percentage of abstract words is over 2.3 percent, the program suggests that concrete examples be introduced to make the document more understandable. (This cutoff was derived from the collection of good documents used to develop the prose standards.) The abstract words found in the text are saved in a file for the user to review.

### *3.3 Organization*

#### *3.3.1 Judging organization: org*

The organization of a text is important since an appropriate structure makes it more comprehensible. For a computer program to analyze the organization of a text fully, it would need to abstract the content. Without a parser for English, or some other way of interpreting the meaning of a text, our programs cannot give feedback on the quality of the content and organization.

The org program, however, was designed to give a writer a different perspective on a text as an aid in evaluating its organization. The program formats the text and preserves headings and paragraph

boundaries, but prints only the first and last sentence of each paragraph.

For writers who follow the traditional topic sentence and concluding sentence format for paragraphs, the output can be a good abstract of the paper. Even for writers with a more casual style, seeing the overall structure of a long paper can help to improve it.

## IV. USER CONSIDERATIONS

In discussing human-computer interactions, two issues should be considered: the ease of using the programs and the quality of the computer's responses.[37] This section will describe how the Writer's Workbench system attempts to optimize both aspects of the interaction.

### *4.1 User interactions*

#### *4.1.1 Program power*

Perhaps the most important organizational decision made was to design the Writer's Workbench system hierarchically. One command, wwb, runs the most commonly used programs, the proofreading program, proofr, and the English-language stylistic program, prose. The wwb program is easily remembered as the acronym for the Writer's Workbench system, and so the casual user of the system finds the system simple to access.

For the experienced user of the Writer's Workbench programs, proofr and prose can be used individually, as can the separate components of proofr, thus allowing such users all the power they need. For one program, proofr, there is an alternate spelling, proofer, because based on the pronunciation it was often misspelled this way.

Many of the programs allow users to have their own dictionaries to tailor the output. Rather than requiring the users to type the names of the dictionaries as part of the command line, these programs use a particular file if it exists. Of course, users can still override this when they choose. Furthermore, commands were created to add words to these dictionaries, rather than requiring the users to edit them and keep them in sorted order. Thus, casual users can create personal spelling dictionaries and access them automatically with just two commands.

#### *4.1.2 Documentation*

For new users, there is now a substantial amount of paper documentation that describes the programs, how to use them, and their relation to good writing. But because users frequently do not own the paper documentation or do not have it near them, there are many on-line aids as well.

As with all *UNIX* system commands there are manual pages for each Writer's Workbench program, which are helpful to experienced *UNIX* system users. For the casual user, there is an on-line introductory system, which describes the uses of each Writer's Workbench program command.

Two other commands give on-line help. The wwbinfo command provides a list of all the commands and their functions. The wwbhelp command takes a key word as an argument and lists all the programs that have anything to do with that key word. These are words such as the following: "syllable," "prose," "passive," "sentence," and "organization."

Each Writer's Workbench program can also be run with a "flags" option, which prints that program's format and options. Further, since many of the programs have default options, these options are also echoed back with the user-specified options, as described in Section 4.2.2.

Finally, the output of many of the programs suggests other programs that would be useful. These suggestions are based on the analyses of the input text, and can thus remind users of programs that are useful for that text. For instance, users with spelling errors are told how to add correctly spelled words to their personal spelling dictionary, and the prose output suggests other appropriate programs. For example, users with too many passives are told how to find all sentences with passive verbs by using the style program.

### *4.2 Computer responses*

What users have to remember and what they have to type are certainly important variables to consider when evaluating a system. But the quality of the messages the computer provides is also important.

#### *4.2.1 Output length*

The Writer's Workbench programs have attracted many new *UNIX* system users. For such users, the traditional "silence" of *UNIX* system commands is unfriendly,[38] e.g., if the spell command finds no misspelled words it simply stops, and the user receives a prompt (not a pat on the back). This silence is exactly what the regular *UNIX* system user wants, as McIlroy states, "Canned chitchat, like the plastic announcements on airplanes, may please newcomers, but it annoys old hands."[39] The obvious problem is that regular and casual users share machines, and what is right for one is frustrating for the other.

The first versions of most of the Writer's Workbench programs were verbose, irritatingly so, for experienced *UNIX* system users. We added "-s" options (for short) to most programs, which removed the

chitchat and most of the "pats on the back." Recently, we changed all the programs so that users can specify once what length of output they want in the future. Since expert users are best able to change this default, users who do not specify any length receive the long version.

#### *4.2.2 Feedback*

The Writer's Workbench programs inform the user when the program has started. Even experienced users may want reassurance that everything is proceeding on a heavily loaded system. Most commands echo the command line and include any options that were not chosen by the user but came about through default. This gives users a record of the exact command run; it may also remind them that they do not want a particular default, and it may alert them to options they did not know existed.

The Writer's Workbench programs cannot correct incorrect entries, but they try to provide complete, informative, and accurate error messages.

### *4.3 Needed improvements*

To date, proofr is a major proofreading package, geared toward users who print the output on paper and make any recommended changes themselves. Since the output refers to a text problem by labeling it with the line number in the unformatted file, a user who has someone else type the text and only has the formatted output does not find these line numbers useful. For users with CRT display terminals, the output can be too lengthy to fit in the terminal's memory so that by the end of the program the first part of the output has disappeared. This makes it difficult to notice all the problems and change them.

We are currently designing two new versions of the proofr program. A version for word processing center customers will provide all the proofreading comments superimposed on the formatted output. For CRT users, we plan a completely interactive version, which will move linearly through the file and display possible corrections for the user to accept or ignore.

## V. SUMMARY

This paper described some principles of good writing and some experimental results that show that readers prefer writing that embodies these principles and find it easier to understand. The paper then described a set of computer programs called the *UNIX* Writer's Workbench software. These programs help a writer isolate problems with general style as well as with individual sentences, phrases, and words. The final section of the paper described some of the human

factors principles that guided the design of the programs. The following paper in this journal[40] will describe the reception of the Writer's Workbench programs at Bell Laboratories and at two trial locations.

## REFERENCES


1. L. T. Frase, "The *UNIX*™ Writer's Workbench Software: Philosophy," B.S.T.J., this issue.
2. J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of New Readability Formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease Formula) for Navy Enlisted Personnel," Navy Training Command Research Branch Report 8-75 (February 1975).
3. W. Strunk, Jr. and E. B. White, *The Elements of Style*, New York: The Macmillan Co., 1959.
4. A. Paivio, "A Factor-Analytic Study of Word Attributes and Verbal Learning," J. Verbal Learning and Verbal Behavior, *7*, No. 1 (February 1968), pp. 41–9.
5. G. Frincke, "Word Characteristics, Associative-Relatedness, and the Free-Recall of Nouns," J. Verb. Learn. Verb. Behav., *7*, No. 2 (April 1968), pp. 366–72.
6. W. A. Winnick and K. Kressel, "Tachistoscopic Recognition Thresholds, Paired-Associate Learning, and Immediate Recall as a Function of Abstractness-Concreteness and Word Frequency," J. Exp. Psychol., *70*, No. 2 (August 1965), pp. 163–8.
7. A. M. Gorman, "Recognition Memory for Nouns as a Function of Abstractness and Frequency," J. Exp. Psychol., *61*, No. 1 (January 1961), pp. 23–9.
8. R. C. Anderson and R. W. Kulhavy, "Imagery and Prose Learning," J. Ed. Psychol., *63*, No. 3 (June 1972), pp. 242–3.
9. E. B. Coleman, "Developing a Technology of Written Instruction: Some Determiners of the Complexity of Prose," In E. Z. Rothkopf and P. E. Johnson (Eds.), *Verbal Learning Research and the Technology of Written Instruction*, New York: Teachers College Press, 1971, pp. 155–204.
10. G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Cambridge, MA: Addison-Wesley, 1949.
11. G. R. Klare, *A Manual for Readable Writing*, Glen Burnie, MD: Rem Company, 1953.
12. A. Einstein, *Essays in Science*, New York: The Philosophical Library, 1949, cited in Mills and Walter, p. 29.
13. G. H. Mills and J. A. Walter, *Technical Writing*, 4th ed., New York: Holt, Rinehart, and Winston, 1962.
14. R. A. Day, *How to Write and Publish a Scientific Paper*, Philadelphia: ISI Press, 1979, p. 119.
15. M. O'Connor, *The Scientist as Editor: Guidelines for Editors of Books and Journals*, New York: John Wiley and Sons, Inc., 1979, p. 163.
16. S. Baker, *The Practical Stylist*, New York: Thomas Y. Crowell Co., 1962.
17. E. B. Coleman, "Learning of Prose Written in Four Grammatical Transformations," J. Appl. Psych., *49*, No. 5 (October 1965), pp. 332–41.
18. P. B. Gough, "Grammatical Transformations and Speed of Understanding," J. Verb. Learn. Verb. Behav., *4*, No. 2 (April 1965), pp. 107–11.
19. R. D. Ramsey, "Grammatical Voice and Person in Technical Writing: Results of a Survey," J. Tech. Writing Comm., *10*, No. 2 (1980), pp. 109–13.
20. J. Kirkman, *Good Style for Scientific and Engineering Writing*, London: Pitman, 1980.
21. E. B. Coleman, "The Comprehensibility of Several Grammatical Transformations," J. Appl. Psychol., *48*, No. 3 (June 1964), pp. 186–90.
22. J. A. Brogan, *Clear Technical Writing*, New York: McGraw-Hill, Inc., 1973, pp. 147–8.
23. E. Dale and J. S. Chall, "A Formula for Predicting Readability: Instructions," Educ. Res. Bull., *27*, No. 1 (January 1948), pp. 37–54.
24. L. Flower, *Problem-Solving Strategies for Writing*, New York: Harcourt Brace Jovanovich, Inc., 1981.
25. J. L. Peterson, "Computer Programs for Detecting and Correcting Spelling Errors," Commun. ACM, *23*, No. 12 (December 1980), pp. 676–87.
26. N. H. Macdonald, "Pattern Matching and Language Analyses as Editing Supports," The Amer. Educ. Res. Assn., Boston, April 1980.
27. N. H. Macdonald, L. T. Frase, P. S. Gingrich, and S. A. Keenan, "Writer's

Workbench: Computer Aids for Text Analysis," IEEE Trans. Commun., Special Issue on Communications in the Automated Office, *30,* No. 1 (January 1982), pp. 105–10.
28. G. E. Heidorn, K. Jensen, L. A. Miller, R. J. Byrd, and M. S. Chodorow, "The EPISTLE Text-Critiquing System," IBM System J., *21,* No. 3 (1982), pp. 305–26.
29. R. L. Bishop, "The JOURNALISM Programs: Help for the Weary Writer," Creative Computing, *1,* No. 2 (January/February 1975), pp. 28–30.
30. J. P. Kincaid, J. A. Aagard, J. W. O'Hara, and L. K. Cottrell, "Computer Readability Editing System," IEEE Trans. on Professional Commun. *PC-24,* No. 1 (March 1981), pp. 38–41.
31. M. D. McIlroy, "Development of a Spelling List," IEEE Trans. Commun., Special Issue on Communications in the Automated Office, *30,* No. 1 (January 1982), pp. 91–9.
32. L. L. Cherry, "Writing Tools," IEEE Trans. Commun., Special Issue on Communications in the Automated Office, *30,* No. 1 (January 1982), pp. 100–5.
33. L. L. Cherry, "PARTS—A System for Assigning Word Classes to English Text," Computing Science Technical Report, *81,* Bell Laboratories, Murray Hill, N. J. 07974, 1978.
34. R. A. Lanham, *Revising Prose,* New York: Charles Scribner's Sons, 1979.
35. R. A. Lanham, *Revising Business Prose,* New York: Charles Scribner's Sons, 1981.
36. L. T. Frase, P. S. Gingrich, and S. A. Keenan, "Computer Content Analysis and Writing Instruction," Amer. Educ. Res. Assn., Los Angeles, CA, April 17, 1981.
37. R. W. Bailey, *Human Performance Engineering: A Guide for System Designers,* New York: Prentice Hall, 1982.
38. D. A. Norman, "The Trouble with UNIX," Datamation, *27,* No. 12 (November 1981), pp. 139–50.
39. M. D. McIlroy, unpublished paper.
40. P. S. Gingrich, "The *UNIX*™ Writer's Workbench Software: Results of a Field Study," B.S.T.J., this issue.

## AUTHOR

**Nina H. Macdonald,** A.B. (Linguistics), 1971, A.M. (Linguistics), 1974, and Ph.D. (Linguistics), 1979, University of Michigan; Bell Laboratories, 1976—. Ms. Macdonald joined Bell Laboratories first as a Resident Visitor in the Linguistics and Speech Analysis Department, studying the role of pauses in the perception of sentences. In 1979 she joined the Human Performance Engineering Department, where she has worked on the development of the Writer's Workbench programs. Member, Linguistics Society of America, Association for Computational Linguistics.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

***Human Factors and Behavioral Science:***

# The *UNIX*™ Writer's Workbench Software: Results of a Field Study

By P. S. GINGRICH*

(Manuscript received December 17, 1981)

This paper describes a study in which the *UNIX*™ Writer's Workbench software was used by writers to analyze and revise the texts they wrote. For ten weeks, two groups of writers were observed: those for whom writing was their principal activity, and others for whom writing was only one of their responsibilities. These participants used the Writer's Workbench programs to produce documents related to their jobs. The results indicate that both types of writers can use the programs without changing their accustomed modes of operating. Writers found the programs helpful; they liked the immediate feedback the programs provided and the detailed suggestions on how to revise their texts. In addition, when editing prepared texts, writers found more errors using Writer's Workbench output than when they had no output available. Finally, participants thought they spent less time editing their documents when they used the Writer's Workbench programs, although total time spent on writing and editing did not change.

## I. INTRODUCTION

The *UNIX*† Writer's Workbench software was developed in two major stages. The first version of the system contained 23 programs

* Bell Laboratories.

† Trademark of Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

ranging from those that searched for proofreading errors to those that summarized stylistic features. These programs were used within Bell Laboratories for nearly a year, during which time data were collected on patterns of program use. A second, expanded version was then developed to reflect the experience gained from the first version.

To date, there is little formal evidence about user interaction with Writer's Workbench programs. We monitored the program on eight *UNIX* systems and observed that the programs were used most when they first became available, and then later they were used on a constant, but less frequent basis, amounting to an average of six to seven program runs per week per user. Informal discussions with users indicated that the peaks in use were due to users experimenting with a variety of input texts, good and poor, to evaluate the potential of the programs. Once they understood the potential, use dropped to a more normal level.

The present study examined the way experienced writers at two Bell System locations used the Writer's Workbench programs as a tool in performing their jobs. We were interested in examining how using the programs affected people's attitudes toward the system, the documents they write, and their modes of writing.

The Writer's Workbench programs provide feedback about many characteristics of writing. But do writers want the feedback? Which programs are used most often? Does using the programs make a difference in the way people write or in their attitudes about what constitutes good writing? How might the programs be modified to make them more useful to writers? This study was the first attempt to address these issues systematically.

## II. FIELD STUDY

### *2.1 Background of the participants*

Fourteen participants from each of two Bell System locations took part in the study. The 14 participants in Group 1 were course developers and instructors with a mean of 10.5 years service in the Bell System, and most had been in their current jobs between one and three years. Forty-three percent of the Group 1 participants had completed four years of college. Before the study began, these participants said they spent about 75 percent of their time writing and editing documents.

Of the 14 participants in Group 2, 11 were technical staff and 3 were clerks. For the technical participants, writing documents was only one of many job responsibilities; prior to the study, they reported spending about one-third of their time writing and editing documents. They also helped to design, develop, and assess systems. The three clerks were included because they processed documents for three managers

from whom we collected data. The technical participants had a mean of 13.6 years service in the Bell System, and most had been in their current jobs between one and three years. Sixty percent of these participants had completed four years of college, and half of those had master's degrees. All three managers had completed four years of college; none of the clerks had.

The two groups of writers differed in their approaches to writing documents and their knowledge of the *UNIX* operating system. Group 1 participants were more knowledgeable about the *UNIX* system and more inclined to compose, edit, and complete drafts on the terminal without the services of clerks. Such differences between groups could lead to different patterns of using the Writer's Workbench system.

### *2.2 Design of the study*

#### *2.2.1 Trial procedures*

The study included two days of training, a one-week period for practice using the Writer's Workbench programs, and 10 weeks of using the programs in normal work activities.

During the two-day training sessions, participants completed questionnaires, learned about the Writer's Workbench system, and then used the programs. After we trained the participants, we tested them for mastery of 11 important concepts of the Writer's Workbench system, including characteristics of a document's style, specific program functions, and program options. Our training successfully conveyed the hierarchical arrangement of the major Workbench commands and the most frequently used options. Participants did have difficulty, however, in naming specific commands for given functions, perhaps because they had not yet used all the commands. All the commands, however, were explained during training.

After the training was completed, participants were encouraged to explore the Writer's Workbench programs during one week of warm-up before the official on-line record keeping began. We provided structured exercises that participants could use to run each program, as well as a handbook that contained a tutorial introduction to the Writer's Workbench programs, and documentation describing each program.

For the next 10 weeks participants used the Writer's Workbench programs. We gave them guidelines, which suggested primary programs to run on each document they wrote, and we encouraged them to use additional programs as well.

#### *2.2.2 Data collection instruments*

*2.2.2.1 Questionnaires.* Participants completed questionnaires before the study began, at the midpoint, and after the study ended.

These questionnaires gathered demographic information about the participants, assessed their attitudes toward computers, and determined characteristics of their type of work and mode of writing. Some questions recurred on two or three of the questionnaires to measure changes in attitudes and modes of operating.

*2.2.2.2 On-line data collection.* Every time a Writer's Workbench program was run, the program automatically recorded its name, the date and time, who ran it, and the name of the text file it analyzed. Some programs, such as the prose analysis programs,[1,2] also saved data about the text characteristics. We used this information to describe program use across time and also to note changes in text files.

In addition, an interactive program prompted users to rate the helpfulness of each program's output on a five-point scale. These data were collected to determine users' opinions of how well the programs analyzed each text file.

*2.2.2.3 Revision tasks.* During the study, participants completed four document revision tasks, which involved editing in ten minutes a 300-word passage containing planted errors. Each participant edited the four different passages once. Each passage appeared equally often in the following sessions:

Session 1. Edit one passage before training begins (no Writer's Workbench program output).

Session 2. Edit one passage immediately after training (no Writer's Workbench program output).

Session 3. Edit one passage at the end of the study. (Appropriate Writer's Workbench program output was attached to each passage.)

Session 4. Edit one passage three weeks after the end of the study (no Writer's Workbench program output).

These revision tasks provided a controlled environment in which to compare the participants' ability to improve a text at various stages of the study.

*2.2.2.4 Interviews with participants.* Each participant was interviewed for 25 minutes at the end of the study. We asked 11 general questions to elicit the participants' comments on how they used the Writer's Workbench programs and what they liked most and least about the system.

### *2.3 Results*

#### *2.3.1 User acceptance of the programs*

We examined user acceptance of the Writer's Workbench programs in several ways. Program use was recorded automatically, and following each use, participants rated the helpfulness of the output. In interviews, participants also reported what they liked least and best

about the system. We found no significant difference in the way the two groups of participants used the programs or their acceptance of them. Consequently, their data were combined in most of the following sections.

*2.3.1.1 Program use.* Although the way participants used the programs varied greatly, on the average they ran about six programs a week.

Table I shows the number of times each program was run during each week of the study. The wwb program, along with its two components proofr and prose, and the spellwwb program were run most often. Use of the programs diminished over the course of the study, but use of the informational "help" programs, such as punctrules and wwbinfo, dropped to zero some weeks before the end of the study. It seems that participants used the "help" programs only as they learned about the Writer's Workbench programs.

*2.3.1.2 Helpfulness ratings.* Participants rated the helpfulness of the output each time they ran a Writer's Workbench program and generally found the programs helpful. The average rating was 3.8 on a scale of 1 (not helpful) to 5 (very helpful). The wwb and prose programs, those most frequently used by Group 1 and Group 2, respectively, had average ratings of 4.0 and 3.7 on the helpfulness scale. Table II shows the mean rating for each program by group.

*2.3.1.3 What participants liked best.* During the final interviews and on questionnaires, many participants said that the Writer's Workbench programs were most valuable because they gave immediate, objective criticisms on concrete problems in their text. They liked the advice from the proofr component because they found it accurate and complete.

For the prose component, participants reported that they felt the advice was objective and specific, but they often were not sure of how to make the changes it suggested. Nevertheless, many participants reported that the stylistic information from prose was an improvement over the often vague advice and subjective opinions of human reviewers.

Participants commented on three other general aspects of the Writer's Workbench system. First, it saved time (both by the speed with which it reviewed text and by eliminating delays caused by looking for a colleague to review the document). Second, it was completely private, giving the writer a chance to improve the document before anyone else saw it. Third, participants reported they were more aware of principles of good writing and of how to change their style to suit the purpose and audience of a particular document.

*2.3.1.4 What participants liked least.* Two criticisms of the programs surfaced during final interviews and from questionnaire responses.

Table I—Average individual program use during training, warm-up week, and trial weeks

| Program | Week of Trial | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TR | WU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11† | Total |
| **wwb programs** | | | | | | | | | | | | | | |
| *wwb | 128 | 111 | 52 | 43 | 29 | 22 | 22 | 38 | 22 | 21 | 6 | 9 | 9 | 512 |
| proofr | 11 | 29 | 14 | 15 | 3 | 6 | 7 | 8 | 3 | 1 | 0 | 2 | 0 | 99 |
| spellwwb | 22 | 58 | 11 | 27 | 13 | 10 | 20 | 42 | 20 | 10 | 20 | 5 | 4 | 262 |
| punct | 17 | 10 | 2 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 33 |
| double | 9 | 4 | 0 | 3 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 19 |
| dictplus | 0 | 3 | 2 | 6 | 2 | 0 | 2 | 0 | 1 | 0 | 0 | 1 | 0 | 17 |
| diction | 0 | 4 | 2 | 0 | 0 | 1 | 0 | 0 | 4 | 0 | 0 | 3 | 1 | 15 |
| suggest | 1 | 23 | 21 | 0 | 1 | 1 | 7 | 10 | 0 | 0 | 0 | 0 | 4 | 68 |
| splitinf | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| prose | 44 | 47 | 22 | 25 | 9 | 8 | 31 | 7 | 9 | 13 | 2 | 4 | 0 | 221 |
| parts | 1 | 4 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 12 |
| style | 32 | 27 | 13 | 5 | 9 | 5 | 23 | 5 | 5 | 3 | 3 | 2 | 6 | 138 |
| **Help programs** | | | | | | | | | | | | | | |
| spelltell | 9 | 3 | 2 | 8 | 2 | 1 | 1 | 6 | 8 | 0 | 0 | 0 | 0 | 40 |
| worduse | 14 | 22 | 17 | 6 | 5 | 11 | 16 | 8 | 2 | 1 | 0 | 0 | 5 | 107 |
| punctrules | 6 | 5 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| splitrules | 6 | 3 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 12 |
| wwbhelp | 16 | 8 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 29 |
| wwbinfo | 26 | 4 | 1 | 2 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 38 |
| wwbstand | 10 | 4 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 21 |
| **Other programs** | | | | | | | | | | | | | | |
| *abst | 10 | 12 | 3 | 53 | 12 | 0 | 0 | 13 | 1 | 2 | 0 | 0 | 0 | 106 |
| *acro | 5 | 7 | 2 | 0 | 1 | 3 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 20 |
| chunk | 0 | 4 | 1 | 1 | 0 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 11 |
| dictadd | 9 | 17 | 7 | 2 | 0 | 1 | 0 | 2 | 4 | 0 | 0 | 3 | 0 | 45 |
| *findbe | 16 | 26 | 12 | 12 | 4 | 0 | 3 | 2 | 3 | 2 | 0 | 1 | 2 | 83 |
| match | 4 | 4 | 0 | 2 | 3 | 0 | 3 | 0 | 2 | 1 | 1 | 3 | 1 | 24 |
| org | 2 | 3 | 0 | 1 | 1 | 0 | 5 | 0 | 0 | 1 | 0 | 0 | 2 | 15 |
| *sexist | 10 | 45 | 6 | 22 | 2 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 88 |
| spelladd | 20 | 13 | 6 | 10 | 0 | 28 | 2 | 7 | 1 | 0 | 0 | 0 | 0 | 87 |
| syl | 4 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 10 |
| topic | 0 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 6 |
| wwbmail | 6 | 3 | 5 | 11 | 2 | 1 | 1 | 2 | 4 | 1 | 0 | 0 | 0 | 36 |
| Total | 438 | 508 | 207 | 261 | 105 | 105 | 151 | 155 | 95 | 62 | 32 | 33 | 36 | 2188 |

* Programs whose use was highly recommended in the trial guidelines.
† Week 11 includes only use by Group 1.

Table II—Users' ratings of Writer's Workbench programs

| Program | Group 1 | | Group 2 | |
|---|---|---|---|---|
| | Mean Rating of All Participants | Number of Program Runs | Mean Rating of All Participants | Number of Program Runs |
| wwbhelp | NR* | 0 | 5.0 | 3 |
| punct | 3.0 | 4 | 5.0 | 2 |
| splitrules | 3.5 | 2 | 5.0 | 1 |
| punctrules | NR | 0 | 5.0 | 1 |
| spelladd | 3.6 | 41 | 4.9 | 12 |
| wwbinfo | NR | 0 | 4.7 | 6 |
| sexist | 3.0 | 7 | 4.7 | 33 |
| spelltell | 2.0 | 3 | 4.7 | 23 |
| wwbmail | NR | 0 | 4.6 | 28 |
| spellwwb | 3.9 | 123 | 4.4 | 56 |
| dictplus | NR | 0 | 4.4 | 14 |
| double | NR | 0 | 4.2 | 6 |
| worduse | 3.3 | 43 | 4.1 | 27 |
| wwb | 4.0 | 156 | 3.9 | 92 |
| proofr | 4.0 | 3 | 3.9 | 41 |
| prose | 2.5 | 12 | 3.8 | 94 |
| abst | 2.1 | 16 | 3.7 | 67 |
| style | 4.0 | 42 | 3.7 | 23 |
| wwbstand | NR | 0 | 3.6 | 7 |
| acro | 2.7 | 3 | 3.6 | 5 |
| findbe | 3.1 | 21 | 3.4 | 8 |
| dictadd | 3.4 | 9 | 3.4 | 8 |
| match | 2.5 | 8 | 3.0 | 4 |
| topic | 4.0 | 1 | 3.0 | 2 |
| org | 2.9 | 8 | 3.0 | 2 |
| parts | 2.3 | 4 | 2.7 | 3 |
| chunk | 1.0 | 3 | 2.7 | 3 |
| syl | NR | 0 | 2.3 | 4 |
| diction | 2.4 | 8 | 2.0 | 34 |
| suggest | 2.9 | 42 | NR* | 0 |
| splitinf | NR | 0 | NR | 0 |
| Average (Total) | 3.6 | 559 | 4.0 | 578 |

* NR means no rating.
NOTE: The number of ratings does not agree with the number of programs run shown in Table V because participants did not begin rating program output until after the warm-up week. Also, some end data was not incorporated from the last week.

First, some participants reported that they found it difficult to apply the advice they received from the prose and diction programs. They either were unfamiliar with the principle the program checked or were unsure of how the principle was being evaluated.

Participants also criticized the length of some program outputs. For example, some said that for the wwb command, the default output was too long and wwb − s output was too short. An output of intermediate length was suggested.

### *2.3.2 Effects on documents*

What effects does using the Writer's Workbench programs have on the documents produced? Do writers locate more errors using Writer's Workbench output to edit a document? Does using the programs for

a period of time later enable writers to edit more effectively on their own?

***2.3.2.1 Revision tasks.*** Results from the revision tasks show how the participants' ability to improve a document differed at various stages of the study. The revision tasks were completed: (1) before the study, (2) after training, (3) after 10 weeks of using the Writer's Workbench programs, and (4) three weeks after the study ended. Output from the Writer's Workbench programs was available only in the third session.

Passages used in the revision tasks were adapted from stories in the *Bell Laboratories Record* and traced the development of new Bell System operations. The average length of the passages was 300 words. Table III describes the twenty errors planted in each of the four passages.

Of the nine categories of inserted errors shown in Table III, the Writer's Workbench programs cannot provide feedback on using a word that looks like the correct word (category g), mistyping one word that produces another (category h), and making a mistake in the agreement of subject and verb (category i).

Table IV shows the results of the revision tasks. More errors were detected in Session 3 than in any other session. Having the Workbench output available improved performance compared both with where the user had no experience with it (Sessions 1 and 2) and where the user

Table III—Type and number of errors in each revision passage

| Type | Number of Error(s) | Item |
|---|---|---|
| a | 4 | Spelling errors, which were not legitimate words and would be found by the spell program |
| b | 2 | Punctuation errors that the punct program could identify |
| c | 6 | Wordy phrases included in the diction phrase dictionary |
| d | 1 | Immediate repetition of a word |
| e | 1 | Split infinitive |
| f | 3 | Instances of sexist language |
| g | 1 | Case of a word substituted for another that looked similar |
| h | 1 | Misspelled word that was a proper English word and would not be found by the spell program |
| i | 1 | Case of subject verb disagreement |

Table IV—Proportion of errors detected for each session*

| | Session 1 | Session 2 | Session 3 | Session 4 |
|---|---|---|---|---|
| Mean | 0.271 | 0.275 | 0.408 | 0.309 |
| Standard error | 0.006 | 0.007 | 0.007 | 0.007 |
| Number | 21 | 21 | 21 | 21 |

Session 3 vs Session 1—$t_{(20)} = 3.35, p < 0.005$
Session 3 vs Session 2—$t_{(20)} = 3.41, p < 0.005$
Session 3 vs Session 4—$t_{(20)} = 2.49, p < 0.05$

* Data are reported for those 21 participants who completed all four sessions.

had employed the Writer's Workbench system in the past but did not have its output available while proofreading (Session 4).

The data are broken down by category in Fig. 1. We see from this figure, which shows the percentage of detected errors for each session (bars 1 to 4) for each category, that Session 3 is clearly distinct from the others.

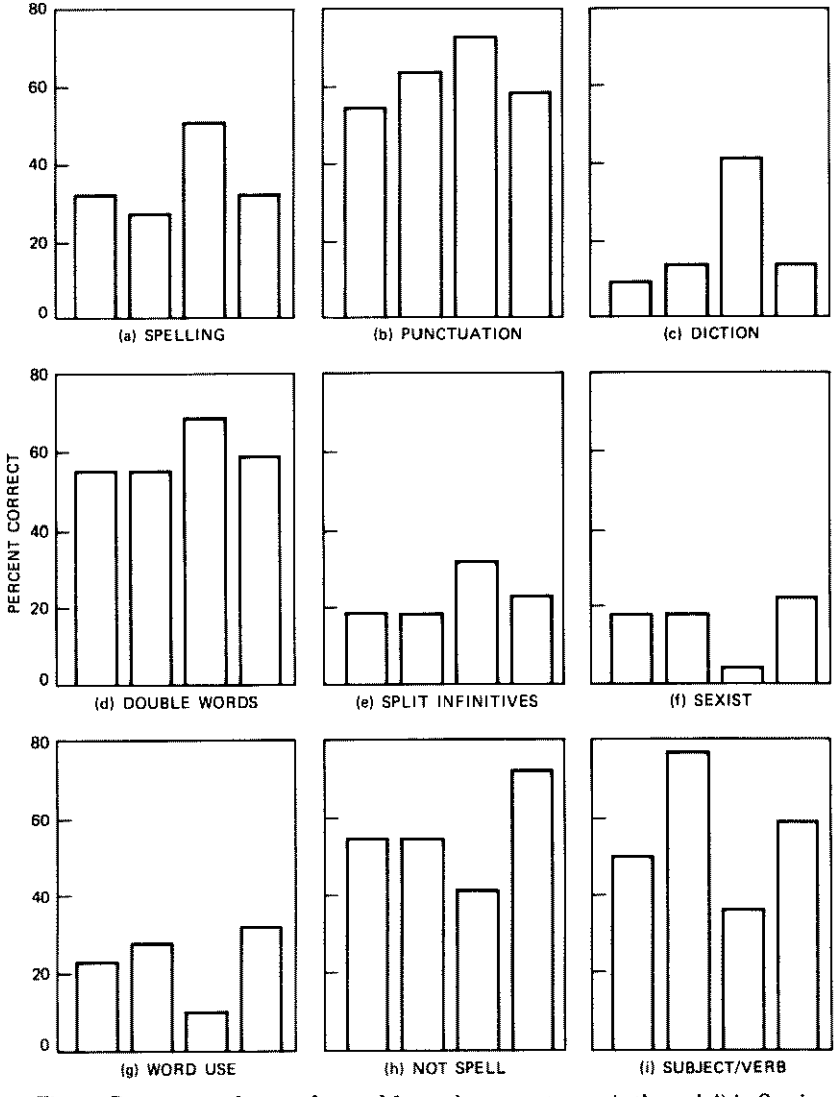



Fig. 1—Percentage of errors detected for each error category (a through i) in Sessions 1 through 4: (a) spelling errors; (b) punctuation errors; (c) wordy diction; (d) double words; (e) split infinitives; (f) sexist language; (g) incorrect word use; (h) spelling errors not found by spell program; and (i) subject/verb disagreement.

We included the errors in categories g, h, and i because we wanted to determine whether the participants had become so dependent on the Writer's Workbench analysis that they would fail in Session 3 to notice the errors that it does not find. The results of Session 3 support this supposition; participants found many of the Writer's Workbench errors and overlooked others.

It appears that participants used the time in Session 3 to find and mark the errors given by the Writer's Workbench programs, but did not have a chance to proofread the paper for other errors. Indeed, several participants remarked that they spent their time reviewing the output and locating the corresponding error in the passage, rather than proofreading the passage as a whole. As a result they may have run out of time. This seems likely in view of the low rate of marking sexist usage, the output for which came last in the collection of Writer's Workbench program outputs given to participants. As Fig. 1f shows, participants marked far fewer sexist usages in Session 3 than in any other session. Even so, overall performance was best in Session 3.

#### *2.3.3 Effects on writers' activities*

The writers completed questionnaires on how the Writer's Workbench programs influenced the way they prepared documents and assessed their stylistic features.

***2.3.3.1 Mode of writing.*** The questionnaires asked participants how they write: Do they write on-line or on paper? Do they make their changes on-line or on paper? As we described above, before the study began participants in Groups 1 and 2 differed in their methods of writing and editing. To test whether their methods changed during the study, we performed a 2 × 3 (group × time) analysis of variance on each of these two items. (The range of responses to both questions represented the degree of involvement with a terminal during the task of either writing or editing. A value of 1 represented the most direct involvement with the terminal and a value of 4 represented the least.) Again, time (pre-, mid-, and post-study) was the within-subject repeated measure; and group (Group 1 or Group 2) was the grouping factor. The results show only a significant group effect for the writing method [$F(1,15) = 7.32$, $p < 0.05$], as well as for the editing method [$F(1,15) = 21.0$, $p < 0.001$]. Participants in Group 1 were less likely to have a clerk do the terminal work for them ($M = 1.43$ for writing and $M = 1.19$ for editing) than were participants in Group 2 ($M = 2.47$ for writing and $M = 2.37$ for editing). There were no significant time effects or time × group interactions.

***2.3.3.2 Estimates of stylistic features.*** We wanted to know whether using the Writer's Workbench programs would affect the way participants evaluated the stylistic errors in their documents. The pre-,

mid-, and post-questionnaires included questions about nine features of writing style that are measured by the Writer's Workbench programs. Participants estimated whether their draft documents typically had

—too few
—the right amount
—too many instances

of each feature, or they indicated that they could not evaluate the amount by saying either

—there was no right or wrong answer, or
—they could not answer the question.

The most marked difference was in how participants evaluated their proportion of passive sentences before and after the study. Before the study, about 85 percent of the Group 1 participants said there was no right or wrong number of passives sentences, or said they could not answer. After the study, no one answered in either of those categories; half thought they had too many passive sentences in their drafts, and half thought they had the right number. Before the study, more than 50 percent of Group 2 said they either had the right number of passive sentences, or they could not answer. By the end of the study, 60 percent of Group 2 thought they either had too many, or too few, passive sentences.

### *2.4 Discussion*

Participants were able to use the programs frequently, liked them, and found the output helpful. By reviewing the results of the different measures together, we come to a clearer understanding of what effects the Writer's Workbench programs have on writers.

First, the data on program use reveal that, on the average, participants ran six Writer's Workbench programs per week. (This figure is comparable to data collected from Bell Laboratories *UNIX* systems we have monitored in the past.) Over time, there was a reduction in the total number of programs run, possibly because the novelty of using the programs wore off, or because of the cyclical nature of the documentation process. Writing a document generally entails collecting information, organizing it, and then writing, editing, and revising. Participants may have confined their use of the Writer's Workbench programs to the first stage of the editing process and to the final version. In addition, documentation efforts vary with the phases of product development; thus, the activities that the Writer's Workbench programs can support are cyclical. Observing program use over a much longer time period, perhaps a year, would give a clearer understanding of the long-term frequency of use, one less subject to seasonal cycles of writing demands.

Even when program use declined over time, the programs were still rated as helpful. The average helpfulness rating was 3.8 on a scale from 1 (not helpful) to 5 (very helpful). The use of programs was more strongly determined by individual needs than by our recommendations. For example, use of the acro program, which locates and prints all acronyms in a document, was highly recommended in the study guidelines. Yet, users in Group 1 did not find this program helpful (2.7 helpfulness rating) because it gave them too much output. Hence, no one used it after the third week of the study. The participants did, however, continue to use the programs that took the tedium out of editing, such as the proofreading programs. That the participants continued to use many of the programs indicates that they found these programs helpful.

The results of the revision tasks clearly show that the programs are helpful in locating errors when the person is under pressures of time. Using the Writer's Workbench system in Session 3, participants found significantly more errors in the same amount of time than in the other sessions. To produce their typical error-free final products, participants would need to spend far more editing time without Writer's Workbench system than with it. The results of the Session 4 revision tasks also show that one must use Writer's Workbench programs continually to get this benefit; it is not enough to have used them in the past. The computer is just much better at finding certain errors than our participants were. However, we do not know whether error types not located by the programs would tend to be missed by the human proofreader more or less often as a result.

When we look at how using Writer's Workbench programs affects a writer's mode of writing, we see few changes. Those who used clerks to type their texts before the study continued to do so during the study. Those who typed their texts themselves continued to do so. Participants did not need to change how they wrote to use the Writer's Workbench programs effectively, nor did it increase or decrease the personal contact individuals had with terminals.

In the current study, we were limited in the degree of control we could impose on the participants. There were no rewards for using the Writer's Workbench programs and incorporating the suggestions into later drafts, nor were there any punishments for not using them. We could not control how much or how little participants wrote during the 10-week trial. We had no independent judges to evaluate the quality of the documents produced with the aid of the Writer's Workbench programs, nor did we have comparable people writing comparable texts without the aid of the programs so we could compare our participants' texts. Thus, we were limited in the issues we could address.

Currently, studies with university English composition students are addressing some of the unanswered questions. In composition classes, students are using the Writer's Workbench program output to edit and revise their essays. Instructors and independent judges will evaluate the essays and compare the quality of these essays to those of control students not using the Writer's Workbench programs. In addition, these new studies include unskilled writers rather than experienced adult writers, which will allow us to determine whether using the system helps students learn how to write better. These findings will enable us to evaluate more fully the effectiveness of the Writer's Workbench programs.

## III. ACKNOWLEDGMENTS

I want to thank the people who contributed to the work of the study reported here. Nina Macdonald conducted the activities for participants in Group 1. Once the study began, Stacey Keenan monitored both sites and analyzed much of the data. Mary Fox summarized responses from hours of taped interviews. Merle Poller assisted in developing the questionnaires. I also want to thank those who participated in the Writer's Workbench system study.

## REFERENCES

1. N. H. Macdonald, "The *UNIX*™ Writer's Workbench Software: Rationale and Design," B.S.T.J., this issue.
2. N. H. Macdonald, L. T. Frase, P. S. Gingrich, and S. A. Keenan, "The Writer's Workbench: Computer Aids for Text Analysis," IEEE Trans. Commun. (Special issue on communication in the automated office), *COM-30*, No. 1 (January 1982), pp. 105–10.

## AUTHOR

**Patricia S. Gingrich,** B. S. (Education), 1970, Pennsylvania State University; M.S. (Education), 1973, Hunter College of the City University of New York; Bell Laboratories, 1979—. Before joining Bell Laboratories, Ms. Gingrich taught at Hunter College. At Bell Laboratories she worked on the development and testing of the Writer's Workbench programs. She is currently a member of the Product Strategy Department, where her work includes product evaluation and planning.

THE BELL SYSTEM
TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 September 1983 Number 7, Part 1

# Stabilized Biasing of Semiconductor Lasers

By R. G. SWARTZ* and B. A. WOOLEY*

(Manuscript received December 10, 1982)

In this paper we analyze the design of biasing and control circuits for semiconductor lasers in a generalized context based on an idealized laser characteristic. In particular, we address three major design considerations: whether to bias the laser above or below threshold, how to stabilize the optical output levels independent of variation in the average output power, and to what degree the output levels can be stabilized relative to various circuit and device parameters. Results of our study indicate that to eliminate from the optical output any dependence on either variation in laser device characteristics or the dc average of the input signal, feedback control of both the prebias and modulation current is necessary.

## I. INTRODUCTION

Within the past few years digital lightwave communication systems have become a practical reality. Several systems have been demonstrated for both interoffice trunk transmission and the subscriber plant.[1-4] In these applications optical fiber systems have the advantages of inherently large bandwidths and electrical isolation.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Copying in printed form for private use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

High-bit-rate lightwave systems commonly use semiconductor laser diodes as the optical sources. The diodes are threshold devices whose characteristics depend on both age and operating temperature. As a consequence, large variations in the light output of the lasers will occur unless special measures are taken to properly bias and modulate these devices.

Several circuits for biasing and digitally modulating injection lasers have been reported. To ensure modulation of the laser output at the highest possible rates, these circuits typically dc bias the laser near its threshold. A modulation current is then superimposed on this bias to switch between the high and low light outputs. The circuits described to date commonly used negative feedback control of the bias current to stabilize the laser light output. In some early circuits the feedback stabilizes the average optical output of the laser. For this method to be successful, the digital input codes must exhibit a fixed on-off ratio (constant average value). More recent laser driver circuits employ balancing compensation of the modulation signal and purport to allow arbitrary on-off ratio digital codes.[5-8]

In this paper we consider the design of laser biasing and control circuits in a generalized context. Within this context we address three major design considerations: the choice of biasing the laser above or below threshold, how to stabilize the output independently of the nature of the laser modulation, and to what degree the laser output levels can be readily stabilized relative to various circuit and device parameters. Initially, we consider the stabilization obtainable by means of the approach adopted in a recently described monolithic laser driver, wherein feedback stabilization of the laser bias current is augmented by a simple balancing compensation of the modulation signal.[7] Following this analysis, we examine the benefits of using modulation current compensation.

## II. FEEDBACK BIAS STABILIZATION

Figure 1 shows the luminosity versus current characteristic assumed for heterojunction lasers in this analysis. This relation can be characterized by three parameters: the threshold current, $I_T$, the subthreshold differential slope efficiency, $\eta_1$, and the above-threshold slope efficiency, $\eta_2$. (The variables used are defined at the back of this paper.) These parameters analytically approximate the characteristic of Fig. 1 by the piecewise linear relationships

$$L = \eta_1 I_L, \qquad I_L \leqslant I_T \tag{1}$$

and

$$L = \eta_1 I_T + \eta_2 (I_L - I_T), \qquad I_L \geqslant I_T, \tag{2}$$

where $L$ is the luminosity (or light output intensity) of the laser.

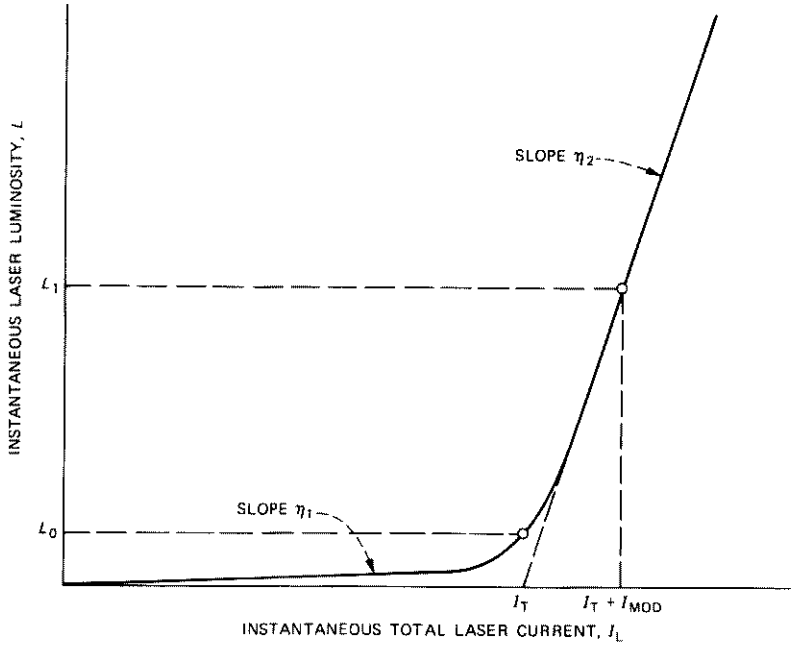



Fig. 1—The luminosity versus current relationship for an injection laser.

Figure 2 is a generalized circuit diagram for a recently reported integrated laser driver employing feedback stabilization.[7] In this circuit the laser is biased near its threshold by a prebias current $I_A$. Added to this bias is a modulation current, $I_M$, which switches the laser between its ZERO and ONE light output levels ($L_0$ and $L_1$). The prebias current is stabilized by a negative feedback loop comprising the laser, a photodetector, a reference current ($I_B$), a low-pass filter ($C_A$), and a current amplifier ($A$). The photodetector generates a current proportional to the optical output of the laser, typically by monitoring the light emitted from its rear face. The photodetector current ($I_D$) is compared to the reference current at the summing node, $S$, and the resulting current difference is then low-pass filtered and amplified to generate the prebias current. Because the modulation current, $I_M$, will alter the dc component of the laser output, the current $I_X$ is added to node $S$ to cancel this influence, as described below.

It is assumed throughout the analysis that:

1. The differential slope efficiency of the laser is much greater above threshold than below, i.e., $\eta_2 \gg \eta_1$.

2. As a consequence of the filtering provided by $C_A$, the response time of the feedback loop is long in comparison with the time constants of modulation-related parameters. It is also required, however, that the feedback loop response time be much shorter than the time

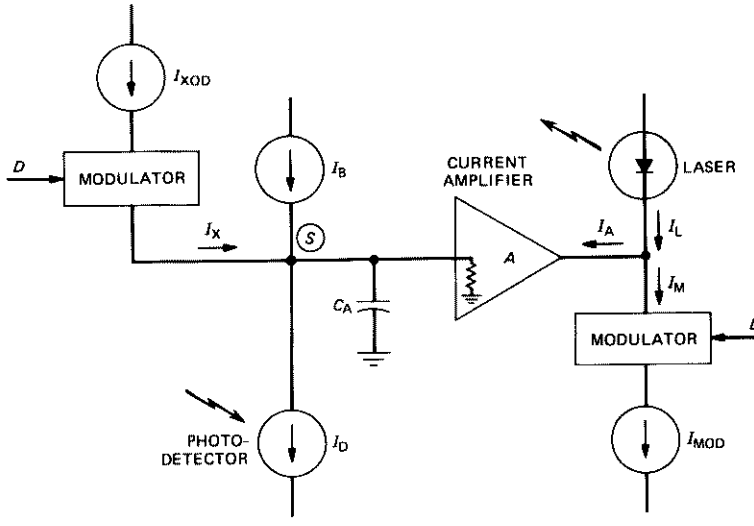



Fig. 2—A laser driver employing feedback stabilization of bias current.

constants of the laser parameter drift that occurs as a consequence of aging or changes in environment.

In the circuit of Fig. 2, the instantaneous laser current, $I_L$, is given by the sum of the prebias current and the instantaneous modulation current

$$I_L = I_A + I_M, \tag{3}$$

where

$$I_M = DI_{MOD} \tag{4}$$

and $D$ is the binary data signal driving the modulator ($D = 0$ or 1). In Fig. 2, the capacitor, $C_A$, serves to average the summation of currents feeding the input to amplifier $A$. Thus,

$$I_A = A_I(I_B + \bar{I}_X - \bar{I}_D) \tag{5}$$

where $A_I$ is the amplifier current gain, and $\bar{I}_X$ and $\bar{I}_D$ are the dc components (averages) of the balance and detector currents, respectively.

The output of the photodetector in Fig. 2 is assumed to be related to the laser light output by a proportionality factor, $f$. Thus, the average detector output current is given in terms of the average laser luminosity by

$$\bar{I}_D = f\bar{L}. \tag{6}$$

The laser light levels $L_0$ and $L_1$ are defined such that $L_0 \triangleq L$ when $D = 0$ and $L_1 \triangleq L$ when $D = 1$. It therefore follows that

$$\bar{L} = \bar{D}(L_1 - L_0) + L_0 \tag{7}$$

and then from (6) and (7) that

$$\bar{I}_D = f[\bar{D}(L_1 - L_0) + L_0]\,. \tag{8}$$

Implicit in this equation is the assumption that time delays in the responses of the laser and photodetector are negligible.

Inserting (8) into (5) and observing that $\bar{I}_X = \bar{D}I_{XOD}$ leads to the relationship

$$I_A = A_I\{I_B + \bar{D}I_{XOD} - f[\bar{D}(L_1 - L_0) + L_0]\}\,. \tag{9}$$

This result, together with the relationship between the laser luminosity ($L$) and current ($I_L$), as represented by (1) and (2), will next be used to determine the laser light output levels, $L_0$ and $L_1$. However, to proceed with this analysis we must first determine whether the laser is prebiased above threshold or below. This distinction, which seems minor at first glance, has important implications for the ultimate stability of the optical output. We first consider the above-threshold case.

### *2.1 Above-threshold prebiasing ($I_{L0} \geq I_T$)*

From eq. (3) and the definition of $L_0$ and $L_1$ it follows that

$$I_{L0} = I_A \tag{10a}$$

and

$$I_{L1} = I_A + I_{MOD}. \tag{10b}$$

For the case where the laser is biased above threshold ($I_{L0} \geq I_T$), it thus follows from (2) that

$$L_0 = \eta_2(I_A - I_T) + \eta_1 I_T \tag{11a}$$

and

$$L_1 = \eta_2(I_A + I_{MOD} - I_T) + \eta_1 I_T\,. \tag{11b}$$

Substitution of (11) into (9) leads to the result

$$I_A = A_I\{I_B + \bar{D}I_{XOD} - f[\eta_2\bar{D}I_{MOD} + \eta_2(I_A - I_T) + \eta_1 I_T]\}\,. \tag{12}$$

This expression can be solved for $I_A$ to obtain

$$I_A = \frac{A_I[I_B + (k_2 - k_1)I_T] + A_I\bar{D}(I_{XOD} - k_2 I_{MOD})}{1 + A_I k_2}\,, \tag{13}$$

where the laser-photodetector current efficiencies $k_1$ and $k_2$ are defined as $k_1 \triangleq f\eta_1$ and $k_2 \triangleq f\eta_2$.

Equations (11) and (13) can be now used to determine the light output levels for the laser:

$$L_0 = \frac{\eta_2 A_I I_B + (\eta_1 - \eta_2) I_T}{1 + A_I k_2} + \frac{A_I \bar{D} \eta_2 (I_{XOD} - k_2 I_{MOD})}{1 + A_I k_2} \tag{14a}$$

and

$$L_1 = L_0 + \eta_2 I_{MOD} . \tag{14b}$$

The quantity $A_I k_2$ represents the loop gain of the bias feedback loop. In a proper design the loop gain is necessarily very large ($A_I k_2 \gg 1$). Under this condition, together with the assumption that $\eta_2 \gg \eta_1$, we can simplify the expressions for the laser light output to

$$L_0 = \frac{1}{f}\left[I_B - \frac{I_T}{A_I}\right] + \frac{\bar{D}}{f}(I_{XOD} - k_2 I_{MOD}) \tag{15a}$$

$$L_1 = L_0 + \eta_2 I_{MOD} . \tag{15b}$$

The balance current, $I_X$, is incorporated in the circuit of Fig. 2 for the purpose of eliminating the dependence of the laser light levels on $\bar{D}$, the dc component of the data input signal. This is accomplished to first order by choosing $I_{XOD} = k_2 I_{MOD}$ so as to eliminate the second term in (15a). However, a dependence on $\bar{D}$ will reappear as a consequence of changes in the above-threshold conversion efficiency, $k_2$, that results from the drift of $\eta_2$ with time. In particular, if $I_{XOD}$ is chosen to balance the circuit at some initial time when $\eta_2 = \eta_2^0$,

$$I_{XOD} = f \eta_2^0 I_{MOD} , \tag{16}$$

and if $\Delta\eta_2$ is defined to represent the subsequent drift in $\eta_2$,

$$\Delta\eta_2 \triangleq \eta_2 - \eta_2^0 , \tag{17}$$

then it follows from eqs. (15) through (17) that the laser output levels can be expressed as

$$L_0 = \frac{1}{f}\left[I_B - \frac{I_T}{A_I}\right] - \bar{D} I_{MOD}(\Delta\eta_2) \tag{18a}$$

and

$$L_1 = L_0 + \eta_2 I_{MOD}. \tag{18b}$$

A principal function of the feedback loop in Fig. 2 is to eliminate the dependence of the optical output on the laser threshold current. However, there remain in (18) terms dependent on $I_T$, and we now consider their relative importance. Assume the laser is biased near threshold so that $I_{L0} \approx I_T$. Then, if the drift in $\eta_2$ is small so that $I_{XOD} = f\eta_2^0 I_{MOD} \approx f\eta_2 I_{MOD}$, it follows from (10a) and (13) that

$$I_{L0} \approx \frac{A_I[I_B + (k_2 - k_1) I_T]}{1 + A_I k_2} \approx I_T . \tag{19}$$

Thus,

$$\frac{I_T}{A_I} \approx \frac{I_B}{1 + A_I k_1}. \tag{20}$$

The quantity $A_I k_1$ is the *subthreshold* loop gain of the feedback loop, and if this gain is large ($A_I k_1 \gg 1$), then

$$\frac{I_T}{A_I} \ll I_B. \tag{21}$$

Therefore, the threshold current dependent terms in (18) are negligible. Thus, under the conditions that the feedback loop gain is large both above and below threshold ($A_I k_2 \gg A_I k_1 \gg 1$), the laser light output levels can be expressed simply as

$$L_0 \approx \frac{I_B}{f} - \bar{D} I_{MOD}(\Delta\eta_2) \tag{22a}$$

$$L_1 \approx \frac{I_B}{f} - \bar{D} I_{MOD}(\Delta\eta_2) + \eta_2 I_{MOD}. \tag{22b}$$

We can draw a number of conclusions with regard to above-threshold biasing from the results expressed in (22):

1. The laser light output levels, $L_0$ and $L_1$, are to first order independent of the subthreshold slope efficiency, $\eta_1$, and the threshold current, $I_T$.

2. If the above-threshold slope efficiency, $\eta_2$, is constant ($\Delta\eta_2 = 0$), the light output levels are independent of the data signal dc component, $\bar{D}$.

3. If $\eta_2$ drifts as a function of time or changes in environment, the light levels $L_0$ and $L_1$ will exhibit some dependence on $\bar{D}$.

4. In a proper design of the circuit represented by Fig. 2, the feedback loop gain must be large below, as well as above, the laser threshold.

### *2.2 Subthreshold prebiasing ($I_{L0} \leq I_T$)*

When the laser is biased below threshold, it is necessarily the case that $I_{L0} = I_A \leq I_T$ and that $I_{L1} = I_A + I_{MOD} > I_T$; it therefore follows from (1), (2), and (10) that

$$L_0 = \eta_1 I_A \tag{23a}$$

and

$$L_1 = \eta_2(I_A + I_{MOD} - I_T) + \eta_1 I_T. \tag{23b}$$

Substitution of (23) into (9) then leads to the expression

$$I_A = A_I(I_B + \bar{D}I_{XOD} - f\{\eta_1 I_A + \bar{D}[\eta_2 I_{MOD} + (\eta_2 - \eta_1)(I_A - I_T)]\}) \quad (24)$$

and this result can be solved for $I_A$ to obtain

$$I_A = \left[\frac{A_I}{1 + A_I k_1 + \bar{D}A_I(k_2 - k_1)}\right] \cdot \{I_B + \bar{D}[I_{XOD} - k_2 I_{MOD} + (k_2 - k_1)I_T]\}, \quad (25)$$

where $k_1 \triangleq f\eta_1$ and $k_2 \triangleq f\eta_2$.

As is the case for above-threshold biasing, the loop gain of the feedback circuit should be large both above and below the laser threshold ($A_I k_2 \gg 1$ and $A_I k_1 \gg 1$). Under these conditions, together with the assumption $\eta_2 \gg \eta_1$, (25) simplifies to

$$I_A \simeq \left(\frac{1}{k_1 + \bar{D}k_2}\right)\{I_B + \bar{D}[I_{XOD} - k_2(I_{MOD} - I_T)]\}. \quad (26)$$

To eliminate the dependence of the optical output levels on $\bar{D}$, $I_{XOD}$ must be chosen so as to remove the dependence of $I_A$ on $\bar{D}$. However, for subthreshold biasing a dependence on $\bar{D}$ will reappear in the event of drift in any of the laser parameters $\eta_2$, $\eta_1$, or $I_T$. If $\eta_1^0$, $\eta_2^0$ and $I_T^0$ denote the values of $\eta_1$, $\eta_2$, and $I_T$ at the time when $I_{XOD}$ is initially adjusted to cancel out the $\bar{D}$ dependence of $I_A$, then from (24) the appropriate value of $I_{XOD}$ is

$$I_{XOD} \simeq k_2^0\left(\frac{I_B}{k_1^0} + I_{MOD} - I_T^0\right), \quad (27)$$

where $k_1^0 = f\eta_1^0$, $k_2^0 = f\eta_2^0$, and we have assumed that $\eta_2^0 \gg \eta_1^0$ and $A_I k_1 \gg 1$. For this choice of $I_{XOD}$ the initial value of $I_A$ is simply

$$I_A^0 \simeq \frac{I_B}{k_1^0}. \quad (28)$$

The expression for $I_A$ at some time following the initialization of $I_{XOD}$ is obtained by substituting (27) into (26), with the result

$$I_A = \left(\frac{1}{k_1 + \bar{D}k_2}\right)\left\{I_B\left(\frac{k_1^0 + \bar{D}k_2^0}{k_1^0}\right) + \bar{D}\,[k_2 I_T - k_2^0 I_T^0 - (k_2 - k_2^0)I_{MOD}]\right\}. \quad (29)$$

It then follows from (23) and (29) that for subthreshold prebiasing, and under the assumption of large feedback loop gain both above and below the laser threshold, the light output levels are given by

$$L_0 = \left(\frac{\eta_1}{\eta_1 + \bar{D}\eta_2}\right)\left\{\frac{I_B}{f}\left(\frac{\eta_1^0 + \bar{D}\eta_2^0}{\eta_1^0}\right)\right.$$
$$\left. + \bar{D}[\eta_2 I_T - \eta_2^0 I_T^0 - (\eta_2 - \eta_2^0) I_{MOD}]\right\} \tag{30a}$$

and

$$L_1 = \left(\frac{\eta_2}{\eta_1 + \bar{D}\eta_2}\right)\left\{\frac{I_B}{f}\left(\frac{\eta_1^0 + \bar{D}\eta_2^0}{\eta_1^0}\right)\right.$$
$$\left. + I_{MOD}(\eta_1 + \bar{D}\eta_2^0) - \eta_1 I_T - \bar{D}\eta_2^0 I_T^0\right\}. \tag{30b}$$

From these complex expressions, as compared to the simple results obtained in (22) for above-threshold biasing, we can draw the following conclusions with respect to subthreshold biasing:

1. The laser light output levels are not stabilized against individual variations in any of the parameters characterizing the laser ($\eta_1$, $\eta_2$, and $I_T$).

2. The light output levels will exhibit a dependence on the data signal average, $\bar{D}$, if changes occur in any one of the laser parameters.

### III. MODULATION CURRENT COMPENSATION

As we demonstrated in Section 2.1, if the laser is prebiased above its threshold current, the circuit of Fig. 2 effectively stabilizes the optical output against variations in both the laser's subthreshold slope efficiency, $\eta_1$, and its threshold current, $I_T$. However, the light output levels remain sensitive to the above-threshold slope efficiency, $\eta_2$, and, as a consequence, to the average value of the input data, $\bar{D}$. One means of eliminating this remaining sensitivity is to compensate for changes in $\eta_2$ through control of the modulation source current, $I_{MOD}$. This approach has, in fact, already been proposed in specific designs.[6,8]

As in Section 2.1 we assume that the laser is prebiased above threshold ($I_{L0} \geq I_T$) so as to eliminate sensitivity of the optical output to $\eta_1$ and $I_T$. It is then apparent from (15) that the sensitivity of the light output levels to $\eta_2$ and $\bar{D}$ can be eliminated if the source current $I_{MOD}$ can be continuously adjusted so as to hold the product $\eta_2 I_{MOD}$ constant. This can be accomplished by deriving a signal proportional to the difference between the ZERO and ONE light levels, and then using negative feedback to control $I_{MOD}$ in a manner that stabilizes this difference. After a signal is obtained proportional to the difference $L_1 - L_0$, $I_{MOD}$ is generated as

$$I_{MOD} = I_{REF} - \gamma(L_1 - L_0), \tag{31}$$

where $\gamma$ is a constant characterizing the feedback loop controlling $I_{MOD}$, and $I_{REF}$ is a modulation reference, or "baseline", current.

From (11) it follows that for above-threshold biasing

$$L_1 - L_0 = \eta_2 I_{MOD}. \tag{32}$$

Upon substituting this expression into (31) and then solving for $I_{MOD}$, we obtain

$$I_{MOD} = \frac{I_{REF}}{1 + \gamma\eta_2} \simeq \frac{I_{REF}}{\gamma\eta_2} \quad \text{for} \quad \gamma\eta_2 \gg 1. \tag{33}$$

In this equation the term $\gamma\eta_2$ represents the loop gain of the negative feedback loop controlling $I_{MOD}$ and should necessarily be much greater than unity.

If we substitute (33) in (14) we obtain the following expressions for the laser light output levels:

$$L_0 = \frac{\eta_2 A_I I_B + (\eta_1 - \eta_2) I_T}{1 + A_I k_2} + \left(\frac{A_I \bar{D} \eta_2}{1 + A_I k_2}\right)\left(I_{XOD} - \frac{k_2 I_{REF}}{\gamma\eta_2}\right) \tag{34a}$$

and

$$L_1 = L_0 + \frac{I_{REF}}{\gamma}. \tag{34b}$$

If, as in Section 2.1, we assume that $\eta_2 \gg \eta_1$, $A_I k_2 \gg 1$ and $A_I k_1 \gg 1$, then, recognizing that $k_2 = f\eta_2$, it follows from (34) that

$$L_0 \approx \frac{I_B}{f} + \frac{\bar{D}}{f}\left(I_{XOD} - \frac{f I_{REF}}{\gamma}\right) \tag{35a}$$

and

$$L_1 = L_0 + \frac{I_{REF}}{\gamma}. \tag{35b}$$

It is apparent from (35) that the sensitivity of $L_0$ and $L_1$ to $\eta_2$ has been successfully eliminated. As in Section 2.1 the remaining dependence on $\bar{D}$ can be removed by the appropriate choice of $I_{XOD}$, namely,

$$I_{XOD} = \frac{f I_{REF}}{\gamma}. \tag{36}$$

The expressions for the ZERO and ONE light output levels then reduce to the very simple form

$$L_0 = \frac{I_B}{f} \tag{37a}$$

and

$$L_1 = \frac{I_B}{f} + \frac{I_{REF}}{\gamma}. \tag{37b}$$

Clearly these levels are now, to first order, independent of the laser parameters $\eta_1$, $\eta_2$, and $I_T$ and the dc component of the data signal, $\bar{D}$.

Compensation of the modulation source current as described below can be implemented as illustrated in Fig. 3. Following the approach of Gruber, et al.,[6] the circuit of Fig. 2 is modified by the inclusion of high-speed buffers (B1 and B2), positive (B3) and negative (B4) peak detectors, and a summing amplifier (B5). The current $I_{MOD}$ is developed at the output of the summing amplifier and is proportional to $L_1 - L_0$. The secondary negative feedback loop controlling $I_{MOD}$ will act to hold $L_1 - L_0$ constant.

The modulation current feedback loop will have a characteristic response time. Consistent with our assumption that $I_{MOD}$ is a parameter that changes slowly with respect to the response of the prebias

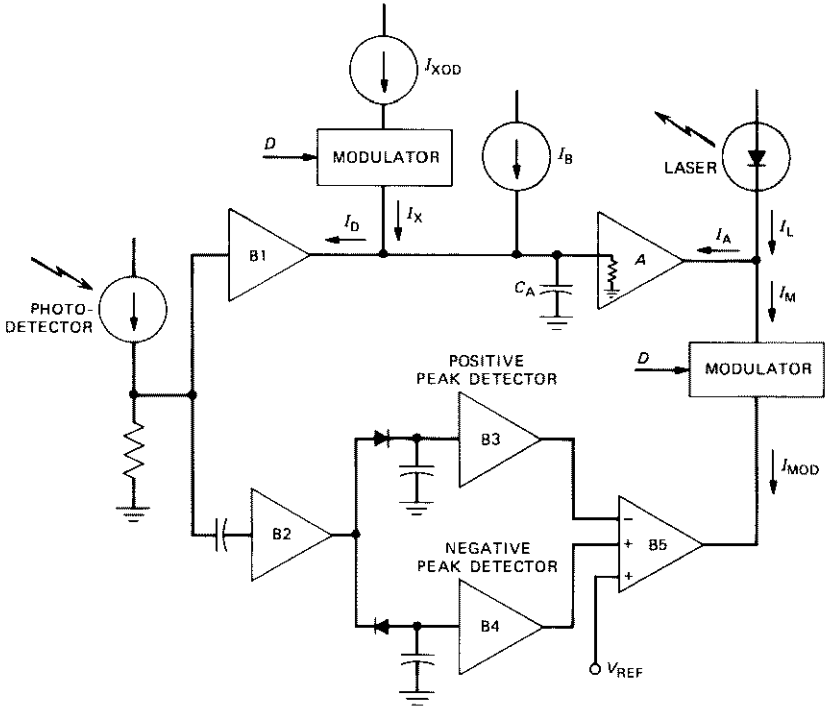



Fig. 3—An improved laser driver incorporating feedback control of both bias and modulation currents. Light output levels are set independently of laser parameters.

current feedback loop, the secondary feedback loop should respond slowly in comparison with the feedback controlling $I_A$.

## IV. CONCLUSION

This paper has presented an analysis of a generalized method of negative feedback stabilized biasing and modulation of semiconductor lasers. Our objective was to evaluate the effectiveness of the stabilization and determine the critical feedback loop parameters. The analysis considered not only the direct influence of variations in $\eta_1$, $\eta_2$, and $I_T$, but also the effect of changes in the average modulation signal and the issue of biasing the laser above or below threshold.

For the more simple bias schemes reported to date, we showed that the laser light levels are susceptible to variation in any of the laser parameters when the laser is dc biased below threshold. When the laser is biased above threshold, only changes in $\eta_2$ affect the light output.

We also analyzed a method of stabilizing the difference $L_1 - L_0$ and thereby fixing laser light output independent of variations in any laser or modulation parameters. To maintain this independence the laser must, of course, be prebiased to remain above threshold under all expected conditions. Moreover, the optical output will still be sensitive to changes in the photodetector light-to-current conversion factor, $f$.

## LIST OF VARIABLES

### *Device parameters*

- $\eta_1$ laser subthreshold differential slope efficiency
- $\eta_2$ laser above-threshold differential slope efficiency
- $I_T$ laser threshold current
- $f$ photodetector light-to-current conversion factor
- $k_1$ laser-photodetector subthreshold conversion efficiency $\triangleq f\eta_1$
- $k_2$ laser-photodetector above-threshold conversion efficiency $\triangleq f\eta_2$

### *Modulation-related (rapidly changing) parameters*

- $D$ digital signal data (ONE or ZERO)
- $I_L$ instantaneous total laser current
- $I_M$ instantaneous modulation current: 0 or $I_{MOD}$
- $I_X$ instantaneous balance current: 0 or $I_{XOD}$
- $I_D$ instantaneous photodetector output current
- $L$ instantaneous laser luminosity

### *Nominally DC (slowly changing) parameters*

- $\bar{D}$ average (dc) value of digital signal data
- $I_{L0}$ logic ZERO laser current

$I_{L1}$ logic ONE laser current
$I_{MOD}$ modulation source current
$I_{XOD}$ balance source current
$L_0$ logic ZERO laser luminosity
$L_1$ logic ONE laser luminosity
$I_B$ bias current
$I_A$ amplifier output current $\triangleq$ laser prebias current $= I_{L0}$
$A_I$ amplifier current gain
$I_{REF}$ modulation reference current
$\gamma$ conversion efficiency of feedback loop controlling the modulation source current, $I_{MOD}$

Notation: For an arbitrary variable $X$, the bar notation $\bar{X}$ signifies the average or dc value of $X$.

## REFERENCES


1. "Atlanta Fiber System Experiment," Special Issue, B.S.T.J., *57*, No. 6 (July–August 1978), pp. 1717–888.
2. M. I. Swartz, "Design and Performance of the FT3 Lightguide Transmission Medium," IEEE Int. Conf. Commun. Conf. Record, *1* (1981), pp. 6.1.1–8.
3. H. Buenning, H. W. Kreutzer, and F. Schmidt, "Subscriber Stations in Service Integrated Optical Broad Band Communications Systems," IEEE Trans. Commun., *COM-30*, No. 9 (September 1982), pp. 2163–71.
4. K. Asatani, R. Watanabe, K. Nosu, T. Matsumoto, and F. Nihei, "A Field Trial of Fiber Optic Subscriber Loop Systems Utilizing Wavelength Division Multiplexers," IEEE Trans. Commun., *COM-30*, No. 9 (September 1982), pp. 2172–84.
5. P. W. Shumate, Jr., F. S. Chen, and P. W. Dorman, "GaAlAs Laser Transmitter for Lightwave Transmission Systems," B.S.T.J., *57*, No. 6 (July–August 1978), pp. 1823–36.
6. J. Gruber, P. Marten, R. Petschacher, and P. Russer, "Electronic Circuits for High Bit Rate Digital Fiber Optic Communication Systems," IEEE Trans. Commun., *COM-26*, No. 7 (July 1978), pp. 1088–98.
7. R. G. Swartz, B. A. Wooley, A. M. Voshchenkov, V. D. Archer, and G. M. Chin, "An Integrated Circuit for Multiplexing and Driving Injection Lasers," IEEE J. Sol. State Circuits, *SC-17*, No. 4 (August 1982), pp. 753–60.
8. F. S. Chen, "Simultaneous Feedback Control of Bias and Modulation Currents for Injection Lasers," Electron. Letts., *16*, No. 1 (January 3, 1980), pp. 7–8.


## AUTHORS


**Robert G. Swartz,** S.B.(EE), 1974, M.I.T.; M.S.(EE), 1975, Ph.D.(EE), 1979, Stanford University; Bell Laboratories, 1979—. Mr. Swartz has done research in a variety of circuit, device, and material areas including silicon molecular beam epitaxy, ultra-short gate MOSFETs, and piezoelectric transducers for medical imaging. Since arriving at Bell Laboratories, he has been particularly interested in circuits for laser control. Member, American Physical Society, IEEE, Sigma Xi; Secretary, 1981 International Solid-State Circuits Conference.

**Bruce A. Wooley,** B.S., 1966, M.S., 1968, and Ph.D., 1970 (Electrical Engineering), University of California, Berkeley; Bell Laboratories, 1970—. Since joining Bell Laboratories, where he is a member of the Microelectronics Research Department, Mr. Wooley's research interests have included monolithic broadband feedback amplifiers, integrated threshold logic circuits, analog-to-digital conversion, and integrated voiceband digital filters. Most re-

cently he has investigated integrated circuits and technology for tactile sensing and for very high speed communication systems. Fellow, IEEE; member, Sigma Xi, Tau Beta Pi, Eta Kappa Nu, IEEE Solid-State Circuits Council, and the Administrative Committee of the IEEE Circuits and Systems Society; Chairman, IEEE Solid-State Circuits and Technology Committee; Chairman, 1981 International Solid-State Circuits Conference; Guest Editor, IEEE Journal of Solid-State Circuits.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# Empirical Evaluation of Profile Variations in an MCVD Optical Waveguide Fiber Using Modal Structure Analysis

By A. CARNEVALE* and U. C. PAEK†

(Manuscript received November 15, 1982)

This paper presents a computer modeling evaluation of the effect on mode propagation of typical perturbations in the refractive index profile produced as a result of the MCVD process. Certain profile variations were eliminated, singly and/or collectively, and the resultant changes in modal structure analyzed. The most important factor in the deterioration of the bandwidth is the presence of a refractive index dip at the core center. Small shifts in the level of dopant were seen to broaden pulse shapes and thus reduce bandwidth. Much less effect on bandwidth is associated with the ripples in the profile, as is also true for dips in refractive index below the index of the cladding near the core-cladding interface. A perpendicular rise, or step, in index of refraction at the core-cladding interface reduces the number of altered modes near the cladding.

## I. INTRODUCTION

In a previous paper[1] we reported on calculation of the modal structure as a function of wavelength ($\lambda$) of an optical waveguide produced by the Modified Chemical Vapor Deposition (MCVD) process using an exact numerical procedure. This technique[2] has been used extensively to investigate various theoretical aspects of optical waveguide propagation.[3-6] We observed excessive splitting and scattering in the

* Bell Laboratories. † Western Electric.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

effective indices ($N_e$) and the group indices ($N_g$), which lead to a reduction of the bandwidth (BW) at any particular $\lambda$.[1] We concluded that the profile variations inherent in the MCVD process were responsible for the reduction of the BW. This paper analyzes in detail the same profile considered there.

The profile is shown in Fig. 1. The data, obtained by the laser beam refraction method,[7] indicates the change in refractive index ($\Delta N_r$) as a function of the unnormalized radius (r). $\Delta N_r = 0$ corresponds to the cladding, in this case, $SiO_2$. The major deviations (imperfections) of this MCVD profile from a power law ($\alpha$) profile are common to most profiles produced by the MCVD process. These include:

1. Region A in the plot, usually referred to as "BURNOFF" or "BURNOUT", which indicates a precipitous reduction in the refractive index at the core center. This reduction is believed to be due to the vaporization of dopant components, such as $GeO_2$, during the high-temperature collapse phase of the process.

2. Region B, which extends over most of the profile, containing what are generally called "RIPPLES", whose width and amplitude

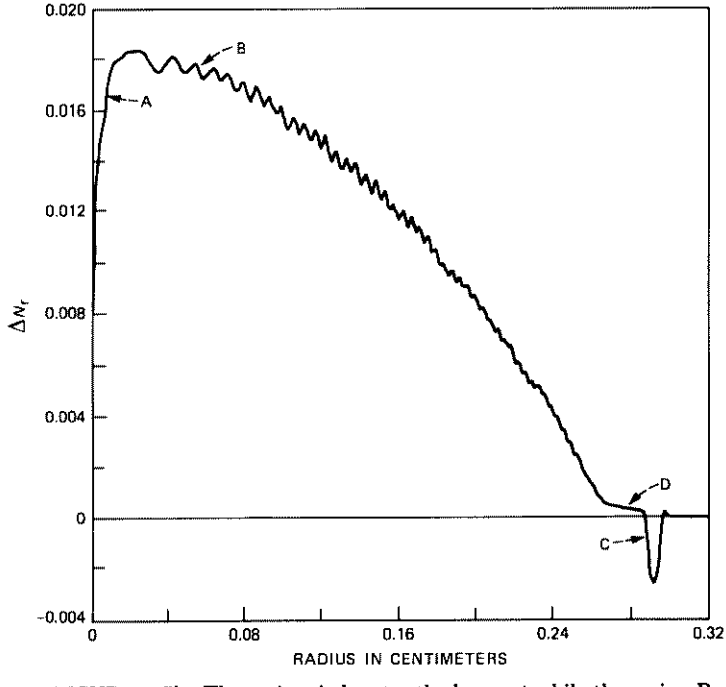



Fig. 1—MCVD profile. The region A denotes the burnout while the region B, which extends over most of the profile, shows the characteristic ripples referred to in the text. Regions C and D refer to the dip and the step, respectively. The radius data are not normalized.

increase toward the core center. These ripples are a direct result of the compositional gradient across the MCVD layer, burnoff of $GeO_2$ at the surface of the layer, and the finite number of discrete layers of changing composition (typically 50).

3. The sharp "DIP", C, near the core-cladding interface caused by the use of an index-reducing dopant in the deposited barrier layer, in this case, boron. In some special cases this "depressed cladding" feature can have a greater width than shown.

4. The "STEP", D, at the cladding.

The above deviations constitute the major obvious perturbations of the MCVD profile from an ideal profile. More subtle deviations, which are usually ignored, will be seen later in this report.

## II. METHOD

It was deemed necessary and sufficient to find the effective indices and the group indices for the:

1. Meridional modes $TE_{0,q}$, and $TM_{0,q}$
2. Helical modes $HE_{1,q}$, and $EH_{1,q}$ over some wavelength region of interest.

The subscripts above denote the angular and radial mode number (m, q) respectively. Thus we conducted the modeling process at 11 specific wavelengths itemized below in Table I.

### *2.1 "As is" case*

Initially, $N_e$ and $N_g$ of the modal groups at $m = 0$ and $m = 1$, were obtained for each of the $\lambda$'s listed in Table I, operating on the MCVD profile as it exists; i.e., on an "as is" basis. Figure 2 shows examples of the distribution of $N_g$ vs $N_e$ for $\lambda = 0.82$ (a), 1.32 (b), and 1.55 (c) $\mu$m, respectively. We note the counterclockwise rotation of the data as a function of increasing $\lambda$; this rotation has been commented on in our early work.[4] The profile was shown to remain constant with $\lambda$, but the group index vs effective index changes in this manner owing to the dependence of bandwidth on $\alpha$.

We further note the clear separation (splitting) of the $HE$ modes ($\Delta$) from all other modes (O, ●); this separation decreases as $\lambda$ increases. The two groups appear to be moving roughly in parallel but

Table I—Wavelengths used for modeling process

| Test No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Wavelength ($\mu$m) | 0.6328* | 0.70 | 0.82* | 0.90 | 1.00 | 1.10 | 1.20 | 1.23 | 1.32* | 1.40† | 1.55* |

* Of current interest in engineering applications. All others listed are solely for the purpose of parameterizing the MCVD deviations from an ideal profile, as a function of wavelength, if possible.

† Water peak.

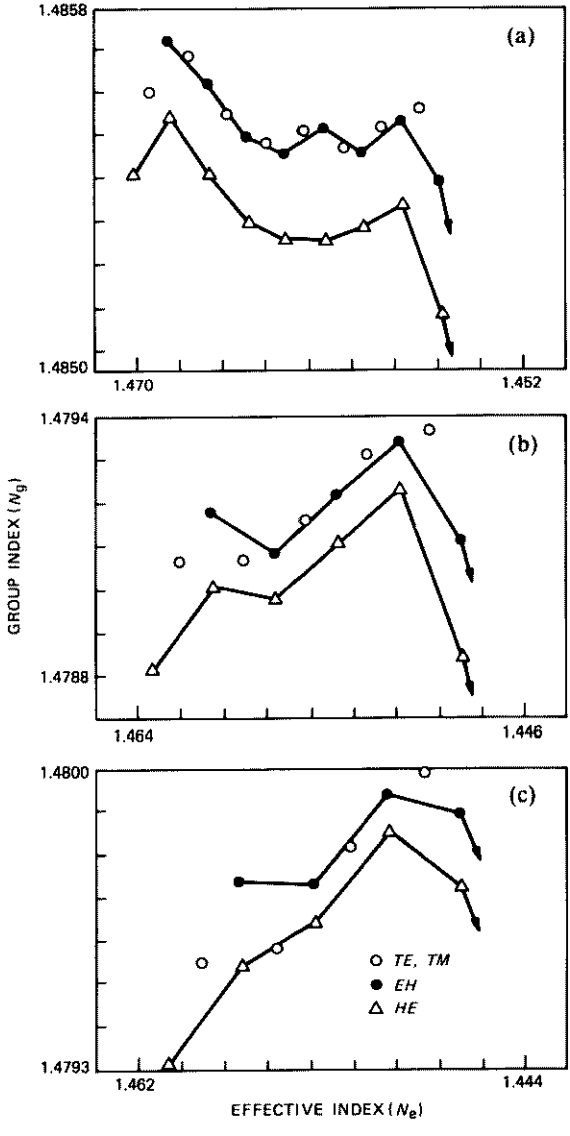



Fig. 2—$N_g$ vs $N_e$—"as is" case. (a) $\lambda = 0.82$ μm. (b) $\lambda = 1.32$ μm. (c) $\lambda = 1.55$ μm.

displaced in time. This can result in two or more peaks, depending on any mode mixing, in the output pulse of the fiber, and a reduction in the BW at any λ. BW reduction would be more severe at short λ's, where the splitting is greater. Last, we observed the drop-off (lower $N_g$) of those high-order modes near the cladding. These modes have been altered because of their proximity to the cladding. The plots of

Fig. 2 can be compared to the ideal case of a power law profile using the computed $\alpha$, $\Delta N$, and $r$ derived from the MCVD profile.[1] A plot of $N_g$ at each $\lambda$ results in a shape similar to those shown in the MCVD plot by (○, ●); i.e., other than the *HE* (Δ) modes.

### 2.2 *"No ripple or dip" case*

To analyze the effect of the MCVD imperfections on the modal structure, each was removed separately. Our first step was to remove the ripple in the section (B) between the burnout (A) and the step (D) in Fig. 1. Past experience in $\alpha$ curve fitting to MCVD profiles indicated that rarely can one fit section $B$ with a single $\alpha$. Because we did not want to destroy any short-term (local) variations in $\Delta N$ and/or $\alpha$, we partitioned the $B$ region into eight equal radial segments, and fitted each segment with a least squares parabola. We then used the fitted curve as the profile in each segment. The index of refraction data shown in Fig. 1 is collected at a number of discrete points and is generally not a continuous curve.

Next, the dip (C) was removed by connecting the end points with a two-point straight line, thus making the step (D) continuous. This correction caused negligible change in the spectrum and so for brevity we eliminated the ripple and dip in one operation.

One example, which is typical of all $\lambda$'s tested, is given in Fig. 3 at $\lambda = 0.6328$ $\mu$m. Figure 3a is the modal display on an as is basis, while Fig. 3b is the result for no ripple or dip. The as is plot is consistent with those $\lambda$'s shown in Fig. 2, i.e., the rotation of the data, the drop-off of the high-order modes, and, most important, the splitting of *HE* from all other modes. The relatively short $\lambda$ of 0.6328 $\mu$m makes the $N_e$ and $N_g$ very sensitive to the short-term variations in the profile. We see the development of some jagged peaks and depressions in the display as we approach the cladding. Since it is the high-order modes that are most affected, imperfections in the profile near the cladding are most likely responsible for the development of these sharp features. It could be the step D, or as seems more likely, a shift in $\alpha$ or $\Delta N$ in the latter part of section B in the profile. Identification of the parameter responsible ($\Delta N$ or $\Delta\alpha$) is difficult. In any case, a change in either results in shifting $N_g$ in the same direction. We can see two pronounced shifts at $r = 18$ and $r = 22$ in Fig. 1. If we examine the fine structure of the profile, we can detect other such shifts over the full range of the radius. Although these shifts appear to be small, they can cause large changes in the eigenvalues and, thereby, in $N_g$. We can therefore associate the ragged appearance of $N_g$ in our plots with very small changes in $\Delta N_r$.

Elimination of the ripple and the dip (Fig. 3b) reduces the scatter of the data and lessens the effect of the shifts in $\Delta N$, probably because

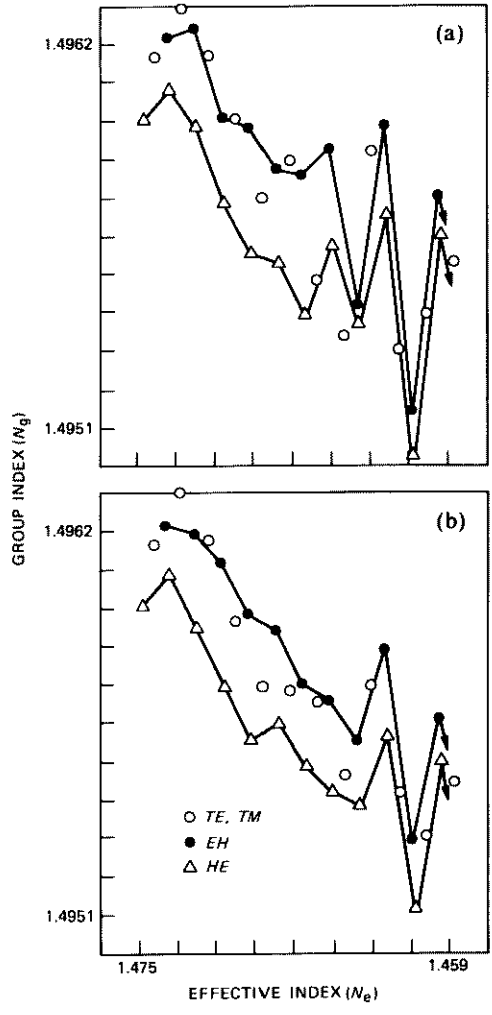



Fig. 3—Plot of $N_g$ vs $N_e$ for $\lambda = 0.6328$ $\mu$m. (a) "As is" case. (b) Ripple and dip removed.

it acts as a smoothing mechanism. However, it does not eliminate or reduce the separation of the *HE* modes and therefore is not the major cause of reduced BW resulting from this profile. The data in Figs. 3a and b appear to have approximately the same slope, but the slope in Fig. 3b is more clearly defined, probably as a result of the smoothing action.

### *2.3 "No burnout" case*

The next MCVD deviation eliminated was the dip in the index of refraction at the core center, i.e., the burnout region (A) in Fig. 1. To

accomplish this, we found the maximum $\Delta N$ in the profile and extended it in a horizontal line to the core center. The rest of the profile was kept intact, thus preserving the original ripples, dip, and step. Figure 4 shows the results at $\lambda = 1.23$ $\mu$m. Figure 4a shows the results on an as is basis, while Fig. 4b shows the case without burnout.

The as is result at this $\lambda$ is again consistent with our earlier results. At this longer $\lambda$, $N_e$ and $N_g$ are much less sensitive to the short-term variations in the profile, which is evident in the relatively smooth plot of $N_g$ vs $N_e$ as compared to the same plot (Fig. 3a) at $\lambda = 0.6328$ $\mu$m.

The elimination of the burnout (Fig. 4b) has removed the displacement in time of the *HE* modes and all modal groups are now traveling together. This remarkable result was observed at all $\lambda$'s (see Table I) tested. Except for considerations of $\alpha$ optimum ($\alpha_{opt}$), the burnout appears to be of paramount significance in the deterioration, or optim-

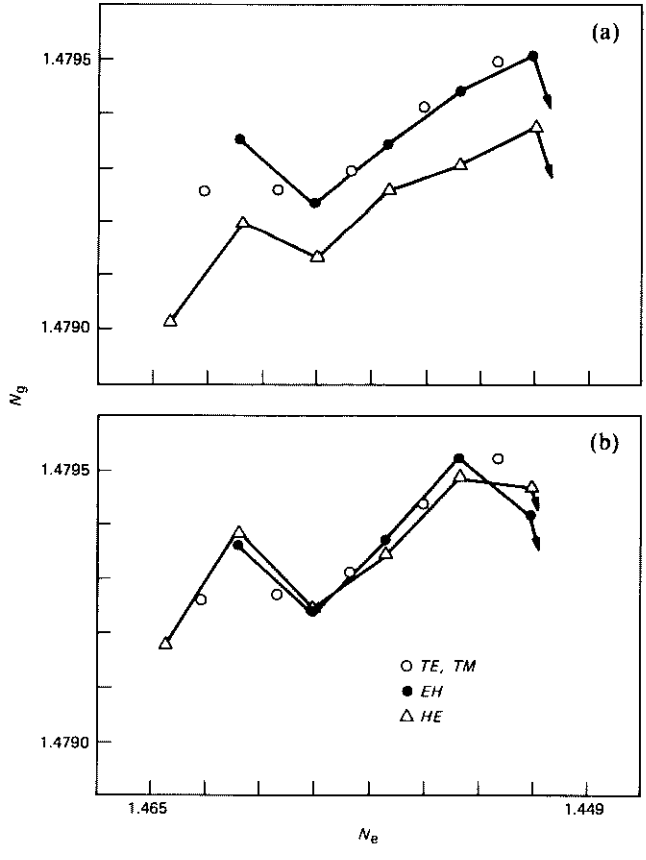



Fig. 4—$N_g$ vs $N_e$ for $\lambda = 1.23$ $\mu$m. (a) "As is" case. (b) "No burnout" case.

ization, of the BW. We should realize that the burnout region in this MCVD profile is much less than that seen in most MCVD profiles.

### *2.4 "Inverted burnout" case*

Because of the previous results we decided to investigate a phenomenon sometimes seen in MCVD profiles. This defect can occur as a result of overcompensation of dopant in attempting to reduce or eliminate burnout. A study of some MCVD profiles[8] in which this defect was present indicated that we could simulate this characteristic by simply inverting the burnout. We found the difference of $\Delta N$ from the maximum $\Delta N$ and added this difference to the maximum, over the burnout region A in Fig. 1.

Figure 5 shows one result at $\lambda = 1.20$ $\mu$m, which was typical of all

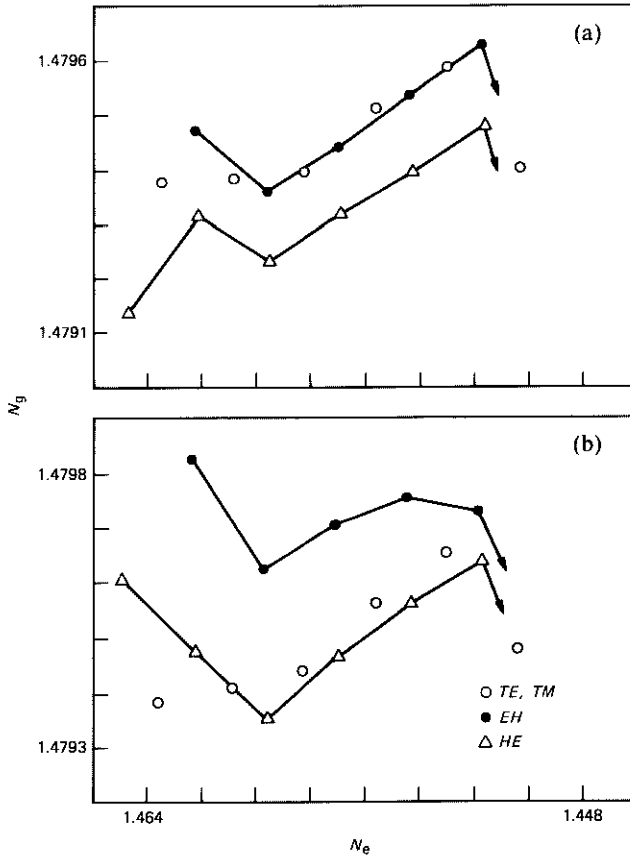



Fig. 5—$N_g$ vs $N_e$ for $\lambda = 1.20$ $\mu$m. (a) "As is" case. (b) "Inverted burnout" case.

λ's tested. Figure 5a is the modal display on an as is basis (which includes the burnout), and Fig. 5b is the result for the inverted burnout (overcompensation).

The as is result is consistent with our previous observations on similar spectra at other λ's.

Inverting the burnout (Fig. 5b) has displaced the *EH* modes from all the rest, and further, this displacement is much greater than the burnout effect, and thus is far more serious than burnout in the decrease of the BW. Our calculations indicate that if both burnout and overcompensation are present simultaneously (which has been observed at times), both splittings would also occur, i.e., the two effects would not be cancelled). Thus we would expect three or more peaks, depending on mode mixing, in the output pulse of the fiber.

### *2.5 "No step" case*

One last perturbation in the MCVD profile remains to be examined, namely, the step (D) near the cladding. We used the least squares fitted curve for the ripple section of the profile nearest the step. We eliminated the step D (and also the dip C) by using the derived equation to extrapolate paired values of $r$, $\Delta N_r$ to the cladding. At $\Delta N_r = 0$, the radius is then renormalized to 25 $\mu$m. One comprehensive result is given in Fig. 6 at $\lambda = 0.70$ $\mu$m, where all plots are on identical scales and are called:

1. As is
2. No ripple or dip
3. No ripple, dip or burnout
4. No ripple, dip, burnout, or step.

The as is modal display (A) is consistent with our previous observations at other λ's on this basis. We see the separation of the *HE* modes due to the burnout and the increased sensitivity to the short-term variations in this profile at this relatively short λ. We note the development of the sharp features in the distribution as we approach the cladding; this is very similar to the comparable spectrum (Fig. 3a) at $\lambda = 0.6328$ $\mu$m.

Removing the ripple and dip (Fig. 6b) provides the same smoothing mechanism previously cited (see Fig. 3b for example).

Figure 6c is the modal distribution realized when the ripple, dip, and burnout have been removed, retaining the step. We see that all modes are now travelling together (no *HE* separation), and that we are now approaching the distribution to be realized for an ideal profile, except for the short-term shifts in $\Delta N_r$ in the profile.

Figure 6d is the modal distribution obtained when the ripple, dip, burnout, and step have been eliminated. This distribution represents the closest approach to an ideal profile. The severe drop-off in $N_g$ of

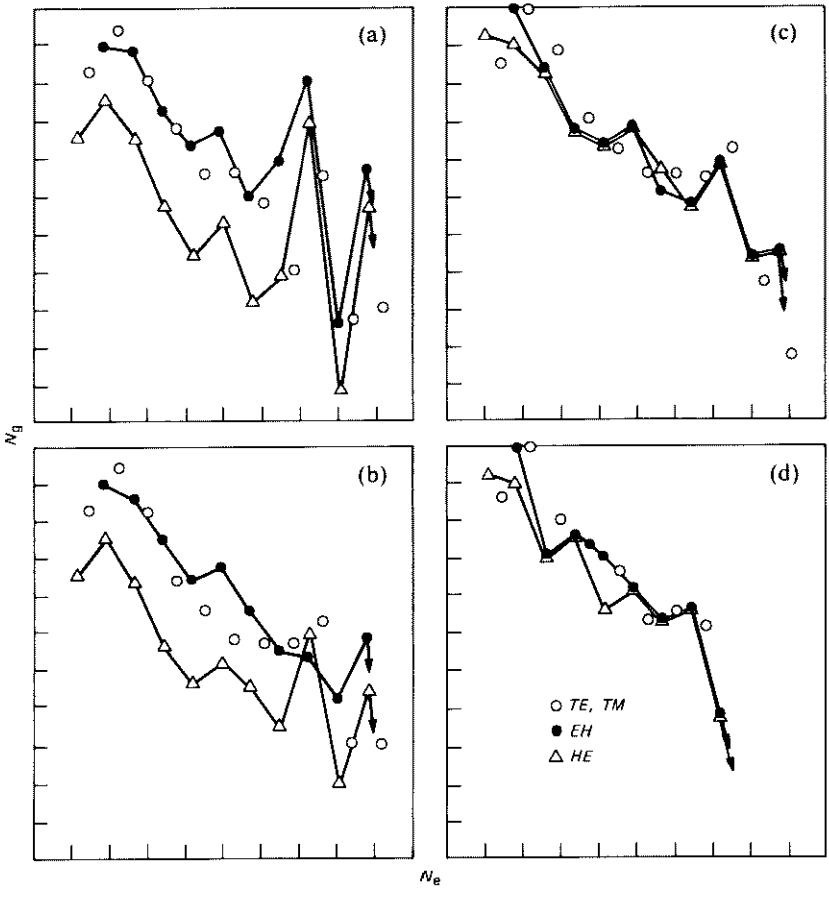



Fig. 6—$N_g$ vs $N_e$ for $\lambda = 0.70$ $\mu$m (Scales are identical in all sections). (a) "As is" case. (b) No ripple or dip case. (c) No ripple, dip, or burnout (step is in). (d) No ripple, dip, burnout, or step.

the high-order modes can lead to a reduction of the BW. Thus we see that the step is beneficial when we are attempting to optimize the BW. Note that there are many more modes in the distribution (Fig. 6c) than there are in our closest approach to an ideal profile (Fig. 6d). In this example, the step will introduce approximately 75 more modes to the distribution as compared to the ideal profile. Figure 7 uses data obtained at $\lambda = 0.6328$ $\mu$m (A) and $\lambda = 1.32$ $\mu$m (B) to show the steps beneficial effect. Each plot represents the results for the MCVD profile with the ripple, dip, burnout, and step removed. The solid line represents $N_g$ values for an ideal profile using the computed values of $\alpha$, $\Delta N$, and $r$ derived from the MCVD profile, at each $\lambda$. The dashed line indicates the path taken by the high-order modes, which have been

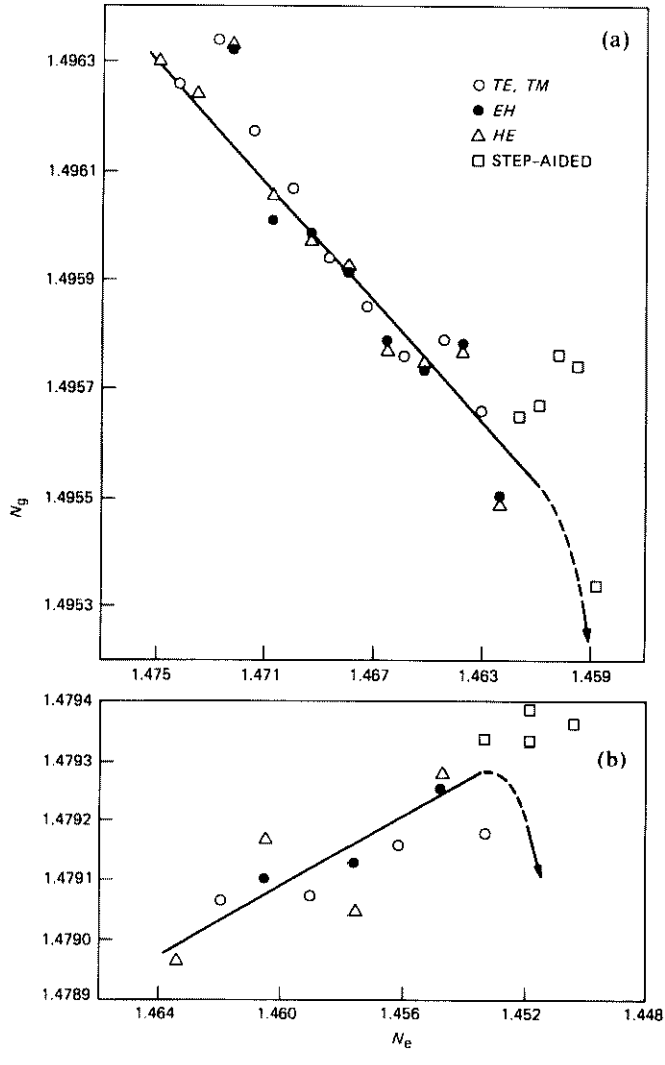



Fig. 7—$N_g$ vs $N_e$. The solid line depicts an ideal profile using the pertinent parameters derived from the MCVD profile. (a) $\lambda = 0.6328$ $\mu$m. (b) $\lambda = 1.32$ $\mu$m.

altered by their proximity to the cladding. The filled squares in each plot show the $N_g$ of some of these altered modes if the step is retained. Since this is the region of high modal density, the squares represent an addition of $\approx$100 modes to the modal distribution at $\lambda = 0.6328$ $\mu$m and $\approx$50 additional modes at $\lambda = 1.32$ $\mu$m. These additional modes can only be helpful when we are trying to optimize BW at any $\lambda$. We can now predict that the BW at these two $\lambda$'s would be a function of the slope of the line.[4]

## III. DISCUSSION

In our earlier work,[4] we have established that for a fixed profile that is a function of $\alpha$, $\Delta N$, and $r$, a least squares linear fit of $N_g$ vs $N_e$, ignoring the high-order modes, will show an increasing slope for increasing $\lambda$. At a given $\lambda$, we have defined $\alpha_{opt}$ as that $\alpha$ which produces a slope of zero. A slope of zero will generate the minimum spread in $N_g$, in the group velocities ($V_g$), and of course, in the delay times. Using that criterion we can see[1] that this MCVD fiber should have an optimum operating wavelength ($\lambda_0$) of about 1.00 $\mu$m. Independent BW measurements[9] have established a BW performance of $\approx$1.5 GHz·km over the range

$$1.06\ \mu\text{m} \leq \lambda \leq 1.35\ \mu\text{m}$$

with a maximum of $\approx$2 GHz·km at $\lambda = 1.23\ \mu$m. We have reproduced the output pulse displays from this work in Fig. 8. In this figure the number in each plot is the $\lambda$ in $\mu$m at which the distribution was obtained. A study of these distributions shows at least two peaks (broadening) at the short $\lambda$'s, which appear to merge as $\lambda$ increases. In all the pulses we see the common characteristic of a trailing tail. This tail is generated by the slowest modes in the modal distribution; at $\lambda = 0.7\ \mu$m (Fig. 6a) these would be the *low-order* modes, while at $\lambda = 1.55\ \mu$m (Fig. 2c) these would be the *high-order* modes. In Fig. 7, we have shown that the step at the core-cladding interface slows down the modes by raising the group indices. We have not attempted to derive the optimal step to optimize the BW for this profile. The reader should bear in mind that *all* modes will also be slowed down by increasing $\alpha$.

In the "as is" section of this report (Section 2.1) we have seen that the burnout displaces the *HE* modes from all other modes and that the two modal groups will travel in parallel but be displaced in time. This will lead to at least two peaks in the output pulse shapes. Further, we have noted that the displacement in time will decrease as a function of increasing $\lambda$. This naturally suggests that the two peaks should merge (depending on instrumental resolution) in the output pulse of $\lambda$ is increased (see Fig. 8).

## IV. REDUCTION OF DATA

We now consider the separation of the various modal groups at several $\lambda$'s (Table I) for the three conditions:

1. As is (includes burnout)
2. Inverted burnout (overcompensation)
3. No burnout,

and, for 2 and 3 above, the rest of the profile remains unaltered. Thus we are concerned here only with the index of refraction dip at the core

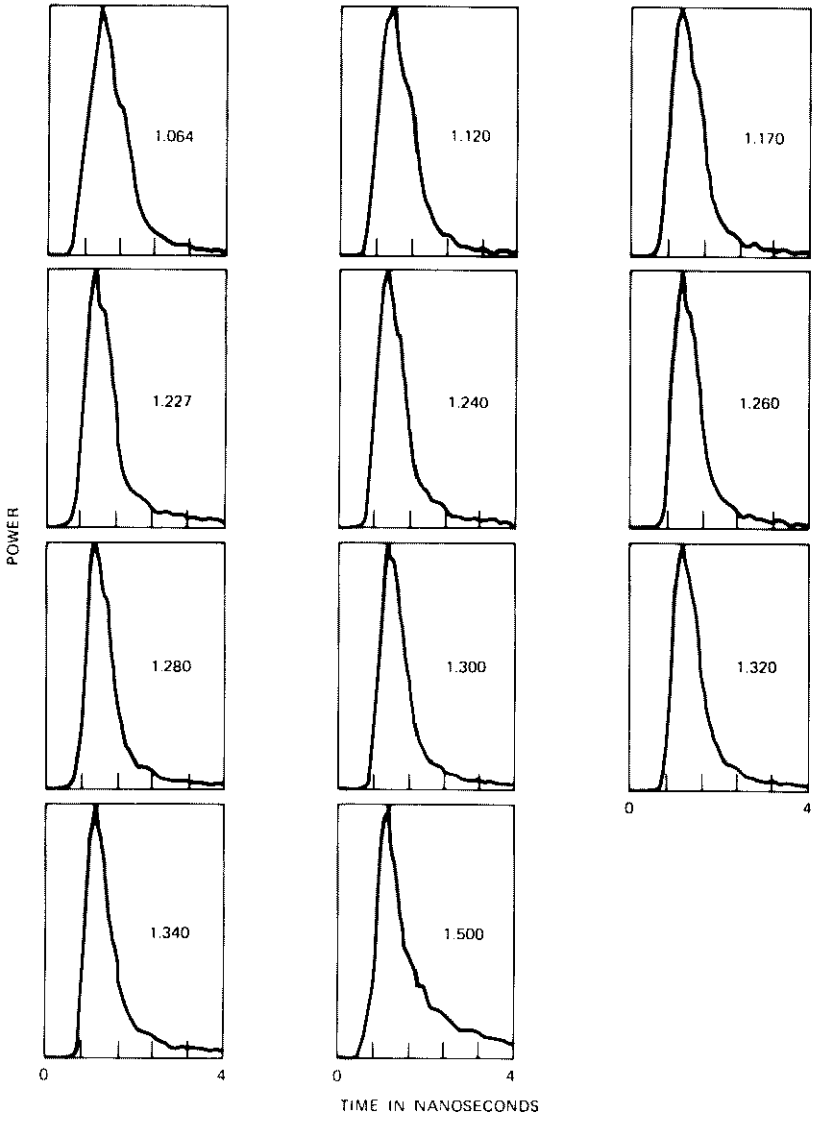



Fig. 8—Power vs time, as a function of the λ (μm) listed in each plot, for the specific MCVD fiber used in this study. Reprinted here, with permission, from Ref. 9.

center. We have previously stated that the separation of the modal groups appears to decrease for increasing λ, and this is true for 1 and 2 above. No such pattern was detected for the no burnout case. To codify these results over the range of λ's listed in Table I, we did the following.

At all λ's tested (Table I, Figs. 2a through 6a) we could generate at

least five distinct *EH-HE* splittings. We therefore found the average splitting ($\overline{\Delta N}_g$) over the first five such splittings at all λ's for each of the three cases listed above. These results are displayed in Fig. 9 as a function of λ. In the figure, the curves are defined as follows:

(A) Inverted burnout—*EH* modes separated from and moving slower than the comparable *HE* modes;

(B) As is—i.e., with burnout. *HE* modes separated from and moving faster than the comparable *EH* modes;

(C) No burnout—no real separation of modal groups—probably a random fluctuation about some relatively small average;

and 30 on the $\overline{\Delta N}_g$ axis is ≈1 ns.

Overcompensation (Curve A) is about 50 percent more deleterious in inducing splitting and is always worse in that respect than the burnout (Curve B). Further, if both phenomena are present, then the

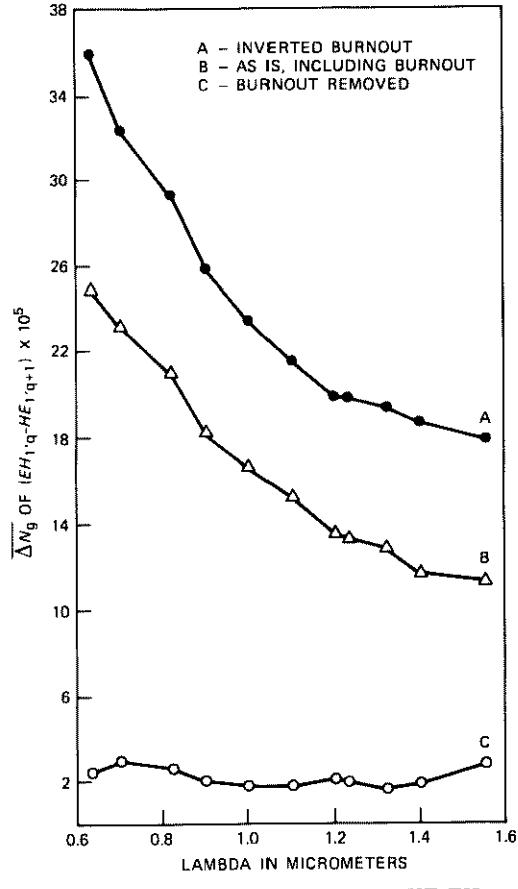



Fig. 9—The average splitting of $N_g$ for the first five *HE-EH* modes, plotted as a function of λ, for the MCVD profile. The measurement 30 on the y axis is ≈1 ns.

total splitting is the sum of the two curves. Three distinct peaks (or more) can be produced in the output pulse of the fiber, and these modal groupings might be, among others:

1. *HE*
2. *EH*
3. *TE, TM*
4. MIXED (*HE, EH*).

In practice, preform makers[8] do not consider preforms that show significant overcompensation for fiber drawing. Experience has shown that this perturbation in the profile yields poor BW in drawn fibers.

Removal of the burnout (Curve C) has reduced the splitting of the *EH-HE* modes to a minimum that is constant over the full range of λ's investigated. This residual splitting can now be attributed to the perturbations still present in the profile, ripple, dip, and short-term $\Delta N_e$, $\Delta\alpha$ variations, etc. If this condition could be realized in practice, i.e., burnout eliminated, the BW at each λ would be a function of $\alpha_{opt}$.[4]

In our computational method,[2] we obtain the eigenvalues ($N_e$) at three closely spaced wavelengths, where

$$\lambda_1 < \lambda_2 < \lambda_3,$$

and $\lambda_2$ is the operating wavelength. The eigenvalues are inserted in the numerical derivative to obtain the group indices for $\lambda_2$ by

$$N_{g2} = N_{e2} - \lambda_2 \left| \frac{N_{e1} - N_{e3}}{\lambda_1 - \lambda_3} \right|.$$

Errors in the determination of $N_e$ can be cancelled or amplified in the above formula. Thus we chose to obtain the average splitting in effective index ($\overline{\Delta}\overline{N}_e$) only at $\lambda_2$. The analogous display to Fig. 9 for $\overline{\Delta}\overline{N}_e$ is given in Fig. 10. In the figure, curves A, B, and C have the same connotation as before. We observe qualitatively the same trends for $\overline{\Delta}\overline{N}_e$ as for $\overline{\Delta}\overline{N}_g$ in Fig. 9. This information may be of some use to those computational methods[10] that calculate $N_g$ directly from $N_e$.

## V. CONCLUSION

We have examined the effect of various perturbations on the BW of an MCVD prepared fiber. We conclude that the primary reason for the deterioration of the BW in the MCVD as is profile is caused by the severe change in refractive index at the core center. This defect produces modal groupings that are displaced in time by as much as 0.5 ns, more or less, depending on the operating wavelength ($\lambda_0$). Short λ's are particularly sensitive to this defect, but even for longer λ's, the burnout will seriously affect attempts to maximize the BW. If burnout is minimized or eliminated in the MCVD process, we believe the next most important goal should be to optimize $\alpha$ at $\lambda_0$. Eliminating local

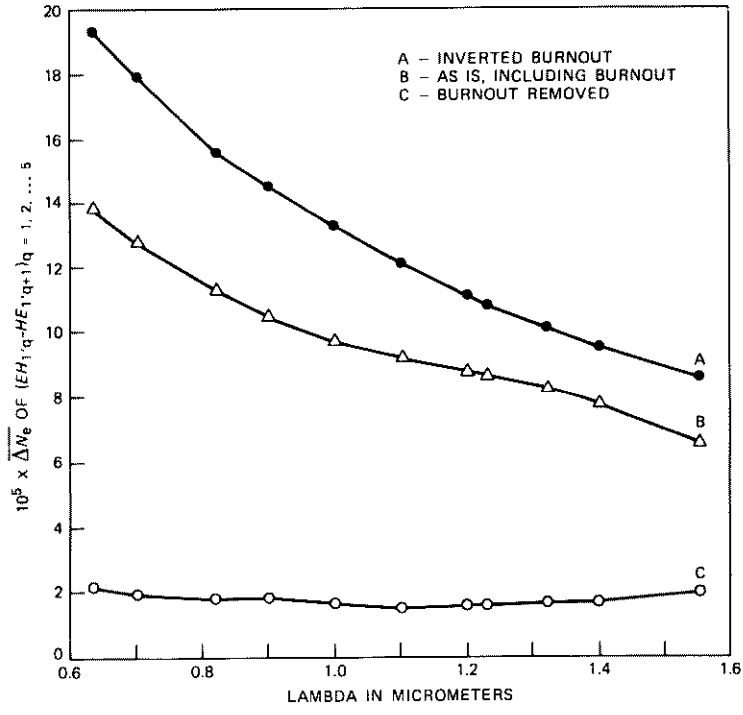



Fig. 10—Same as Fig. 9, except this is for the effective indices at the plotted λ.

shifts in $\Delta N$ would be the next priority, followed by reducing the amplitude of the ripples. This last is probably the least harmful in causing deterioration of BW. The step at the core-cladding interface appears to be beneficial in that it significantly *reduces* the number of high-order modes that have been altered by their proximity to the cladding. The works of Vassel[11] suggest that a perpendicular rise in the index of refraction at the core-cladding interface will have the same effect as the step (D) in Fig. 1.

In theory,[1,4] this MCVD fiber should have a maximum BW at $\lambda_0 \approx 1.00\ \mu m$. The experimental $\lambda_0$ was determined[9] to be 1.23 μm, and this shift to the longer $\lambda_0$ was verified by the statistical study conducted in Ref. 1. In an ideal profile, the governing factor for the optimization of BW is the determination of $\alpha_{opt}$ for $\lambda_0$.[4] In the MCVD profile this criterion has been overridden by the excessive broadening of the output pulse induced by the burnout causing the theoretical $\lambda_0$ to shift longer λ's. Even at longer λ's the effect of the burnout on BW is significant and may preclude the optimization of BW beyond the present state of the art.

Optimization of BW, based on optimum parameters ($\alpha_{opt}$, $\Delta N_{opt}$, etc.) derived from theoretical (ideal) concepts, will have limited appli-

cation to MCVD fibers. This limited applicability can explain to a great extent why workers in the field endeavoring to optimize BW utilizing concepts based on the theory cannot reproduce these bandwidths.

The literature is replete with studies of the effects on BW caused by disturbances in the profile such as burnoff, ripple, and steps in the index of refraction at the core-cladding interface. Most of these studies (including the present work) describe a cause and effect relationship. We are not aware of any study correlating these effects in terms of local changes in the $\alpha$ value, which the burnoff and/or ripples cause, resulting in a deterioration of the BW.

In our most recent work, we have tried to determine the fundamental cause of mode splitting. In this project we believe we have been successful. The fundamental cause of mode splitting is easily applied to the width and depth of the burnoff, and to the width (variable or otherwise), amplitude, and frequency of the ripples. The project is incomplete at this time.

## VI. ACKNOWLEDGMENTS

This work was supported by P. A. Fleury, M. I. Cohen, R. J. Klaiber, and L. S. Watkins. We are encouraged by the enthusiasm of the materials scientists engaged in the lightguide program at Bell Laboratories, Murray Hill and Atlanta, and at the Engineering Research Center (ERC), Western Electric Co., Princeton. We are indebted to the computer personnel at ERC, Western Electric Co., Princeton, without whose strong support and cooperation this task would not have been completed.

## REFERENCES

1. A. Carnevale, and U. C. Paek, "Modal Structure of an MCVD Optical Waveguide Fiber," B.S.T.J., *62*, Part 1 (July-August 1983), pp. 1415–31.
2. G. E. Peterson, A. Carnevale, U. C. Paek, and D. W. Berreman, "An Exact Numerical Solution to Maxwell's Equation for Lightguides," B.S.T.J., *59*, No. 7 (September 1980), pp. 1175–96.
3. G. E. Peterson, A. Carnevale, and U. C. Paek, "Comparison of Vector and Scalar Modes in a Lightguide with a Hyperbolic Secant Index Distribution," B.S.T.J., *59*, No. 9 (November 1980), pp. 1681–91.
4. G. E. Peterson, A. Carnevale, U. C. Paek, and J. W. Fleming, "Numerical Calculation of Optimum $\alpha$ for a Germania-Doped Silica Lightguide," B.S.T.J., *60*, No. 4 (April 1981), pp. 455–70.
5. U. C. Paek, G. E. Peterson, and A. Carnevale, "Dispersionless Single-Mode Lightguides with $\alpha$ Index Profiles," B.S.T.J., *60*, No. 5 (May-June 1981), pp. 583–98.
6. U. C. Paek, G. E. Peterson, and A. Carnevale, "Electromagnetic Fields, Field Confinement, and Energy Flow in Dispersionless Single-Mode Lightguides With Graded-Index Profiles," B.S.T.J., *60*, No. 8 (October 1981), pp. 1727–43.
7. L. S. Watkins, "Laser beam refraction traversely through a graded-index preform to determine refractive index ratio and gradient profile," Applied Optics, *18* (July 1, 1979), pp. 2214–22.
8. S. R. Nagel, K. L. Walker, and S. G. Kosinski, private communication.
9. L. G. Cohen, and S. Jang, unpublished work.

10. T. Lenahan, "Calculation of Modes in an Optical Fiber Using the Finite Element Method and EISPACK," B.S.T.J., *62*, No. 9, Part 1 (November 1983).
11. M. O. Vassel, "Calculation of Propagating Modes in a Graded-Index Optical Fibre," Opto-Electron, *5* (1974), p. 271.

## AUTHORS

**Anthony Carnevale,** B.S. (Physics), 1960, Fairleigh Dickinson University; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Carnevale has been engaged in work on nuclear magnetic resonance, electron paramagnetic resonance, and computer software. For the last four years, his work has been devoted to fiber optics.

**Un-Chul Paek,** B.S. (Engineering), 1957, Korea Merchant Marine Academy, Korea; M.S., 1965, Ph.D., 1969, University of California, Berkeley; Western Electric, 1969—. At the Western Electric Engineering Research Center, Princeton, N. J., Mr. Paek has been engaged primarily in research on laser material interaction phenomena and fiber optics. Member, Optical Society of America, American Ceramic Society, Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# A Comparison of Line Difference Predictions for Time-Frequency Multiplexing of Television Signals

By R. L. SCHMIDT*

(Manuscript received January 27, 1983)

Studies are presented of a scheme utilizing time-frequency multiplexing (TFM) to multiplex two television signals on a microwave radio channel. This scheme transmits alternate lines of the television signal at full bandwidth, along with a differential signal for the remaining lines. The differential signal is derived by subtracting the true value of each picture element from a prediction for that value based on surrounding elements located in the transmitted lines. This differential signal is then placed on a carrier, and frequency multiplexed with the baseband signal. In this way, two television lines of information can be transmitted in one line time, allowing a second similarly constructed signal to be transmitted during the vacated line. Thus, the television signals will be time interleaved on a line-to-line basis. This paper describes the comparison of two different prediction algorithms used to construct the line-differential signal. Time and frequency domain analyses were carried out on the computer and then verified with hardware testing.

## I. INTRODUCTION

Recently, a time-frequency multiplexing (TFM) system was described[1] as a method for multiplexing two National Television System Committee (NTSC) color television signals onto a single microwave radio channel. This system proposes to transmit alternate

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

lines of a television signal at full bandwidth, and the remaining lines by means of a band-limited line-differential signal formed by using a prediction obtained from adjacent, full bandwidth lines. This differential signal is then placed on a carrier and frequency multiplexed with a full bandwidth television line. In this way two lines are sent in one line period, and, by time multiplexing, two pictures can be sent on a single microwave radio channel.

The reconstructed picture in the receiver is then a combination of alternate lines of full bandwidth video, and lines regenerated from the prediction and the band-limited line-differential signal. Consequently, a more accurate prediction would require less differential signal information. Stated another way, better picture quality can be attained for a given bandwidth limitation of the differential signal. The proposed differential signal band limiting for the TFM system is 3 MHz, and, within this constraint, an improved prediction would allow the transmission of more detailed pictures.

The originally proposed system had a prediction based on an average of adjacent picture elements of the same color subcarrier phase, from the previous and upcoming lines (Fig. 1). These elements were chosen because we did not have to take into account the color subcarrier, making this prediction the simplest to implement. However, since the prediction elements had to be of the same color subcarrier phase, they were not the elements closest to the element being predicted. This would create a larger differential signal on sharp vertical edges. This paper refers to this prediction as $XP_1$, and its associated differential signal as $XDIF_1$.

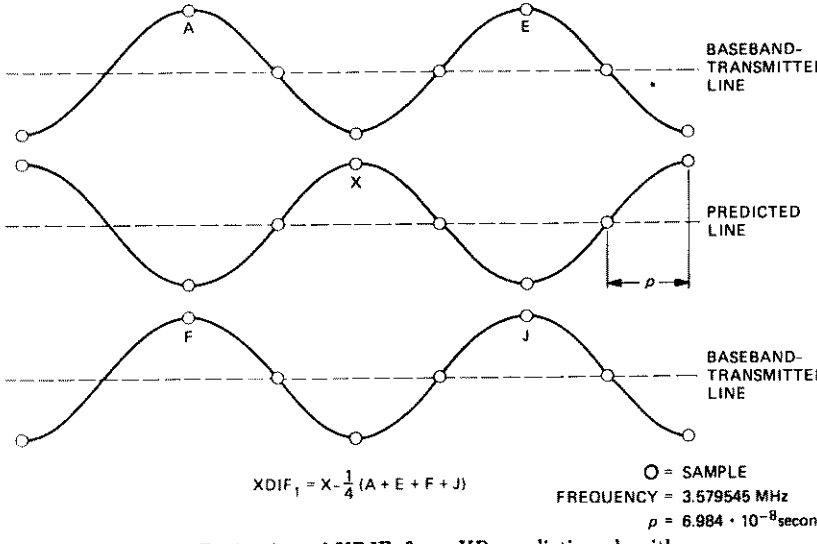



Fig. 1—Derivation of $XDIF_1$ from $XP_1$ prediction algorithm.

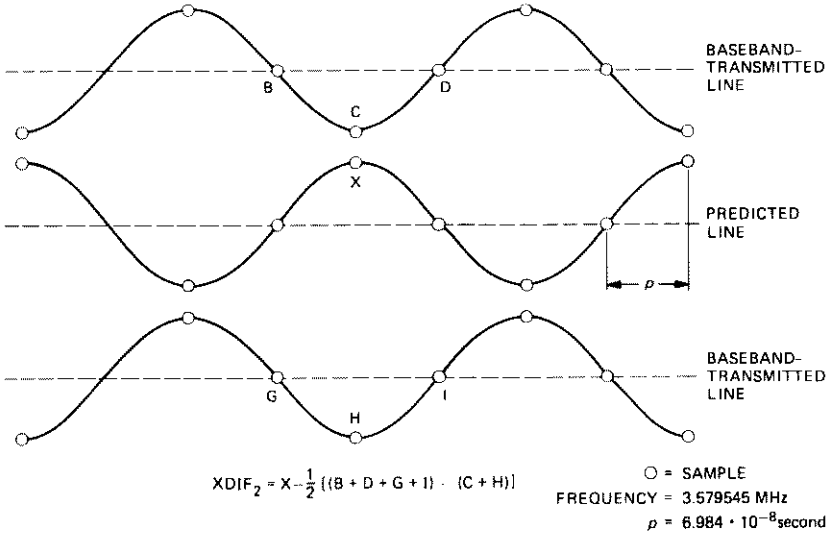



Fig. 2—Derivation of $XDIF_2$ from $XP_2$ prediction algorithm.

It was conjectured that picture quality would improve if we chose a line difference prediction that used the picture elements from the previous and upcoming lines closer to the coded element (Fig. 2), thus improving the resolution of the predicted value on vertical edges. Here we refer to this prediction as $XP_2$ and its associated differential signal as $XDIF_2$.

## II. TIME DOMAIN ANALYSIS

The complexity of the baseband signal, and its quasi-periodic nature, would make analysis of the network response to an average video signal tedious and not very informative. However, since line-to-line correlation is high, some typical waveforms can be looked at and yield useful results.

Since there is a high content of color subcarrier in a typical television waveform, the differential signal should be small for that frequency. This criterion is met for both line difference predictions, and is shown in Appendix A to be zero for any flat field of constant color.

Appendix B shows that if there is color change from one line to the next the worst case differential signal will be the same for both predictions, with the amplitude being an average of the amplitudes of the two color sine waves for the adjacent lines. However, the worst case for each predictor does not occur at the same color subcarrier phase angle.

Along a horizontal line, color changes are almost always accompanied by luminance changes, and, since the bandwidth of the lumi-

nance is much higher than that of the chrominance, the effect of chrominance distortions will be secondary. With regard to luminance changes, three different waveforms are used to demonstrate the difference between the two prediction algorithms. The first waveform considered is the standard $2T$ pulse[2] shown in Fig. 3. This is a sine squared function, and is assumed repetitive from line-to-line. Calculations were done with a computer using the same general approach as shown in Appendices A and B. The results in Fig. 4 show that $XDIF_2$ (dotted line) has a reduced amplitude for both positive and negative directional peaks. In addition, the negative peaks are of shorter duration, resulting in less energy. Although this implies a slightly higher bandwidth, the magnitude of the band-limited signal for $XDIF_2$ is substantially less, indicating a more accurate reconstructed signal in the receiver, with less dependence on the differential signal.

The next waveform considered is the rising edge of the standard bar pulse[2] shown in Fig. 5. This function is represented a follows:

$$f(t) = \begin{cases} 0 & t < 0 \\ 6.4 \cdot 10^6 t & 0 \leq t \leq 156.25 \text{ ns} \\ 1 & t > 156.25 \text{ ns.} \end{cases}$$

Again, line-to-line repetition is assumed, and Fig. 6 shows the response of the two different algorithms. It can be seen that although the amplitudes of both differential signals are the same, the width is less for $XDIF_2$, resulting in less energy. Since the line-differential

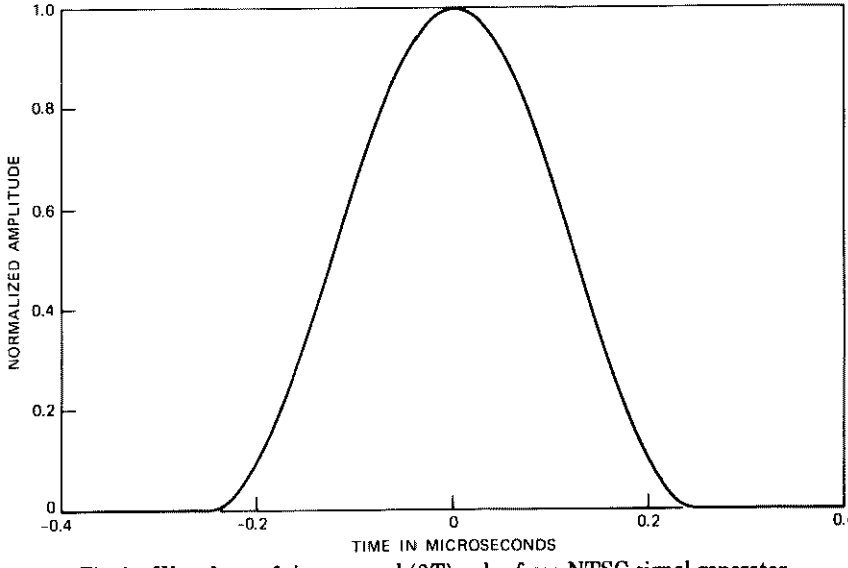



Fig. 3—Waveform of sine squared ($2T$) pulse from NTSC signal generator.

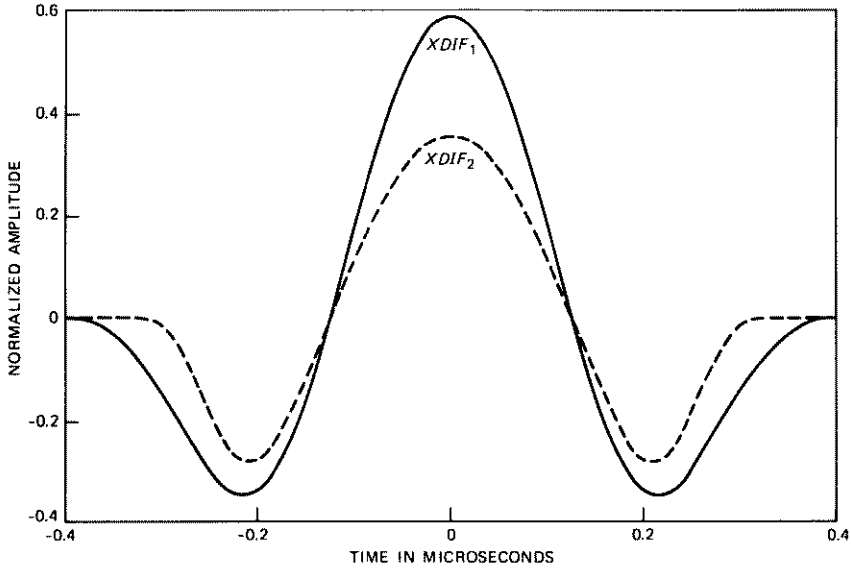



Fig. 4—Differential signal response of both algorithms to sine squared pulse.

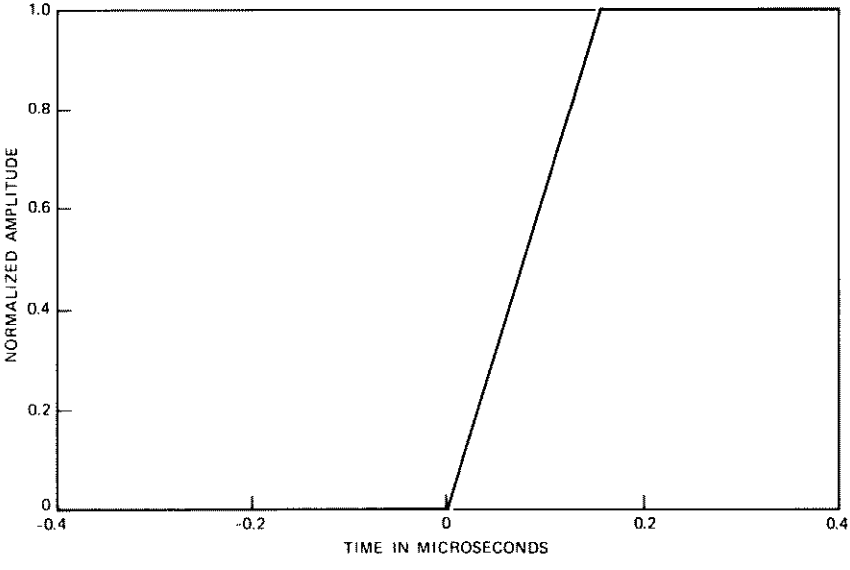



Fig. 5—Rising edge of bar pulse from NTSC signal generator.

signal in the proposed system is band limited to 3 MHz, this will again result in a smaller signal after filtering, with the same advantages as in the previous example.

The next and final waveform evaluated is a raised cosine function

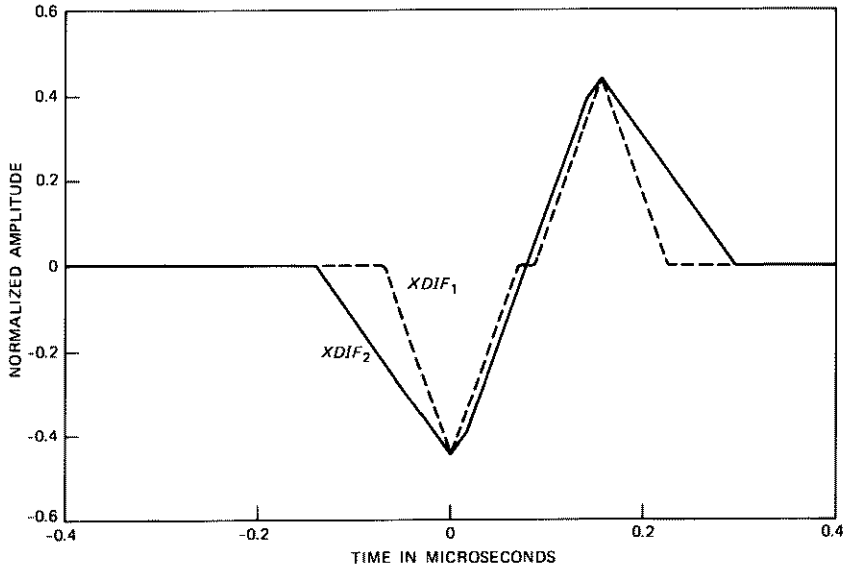



Fig. 6—Differential signal response of both algorithms to rising edge of bar pulse.

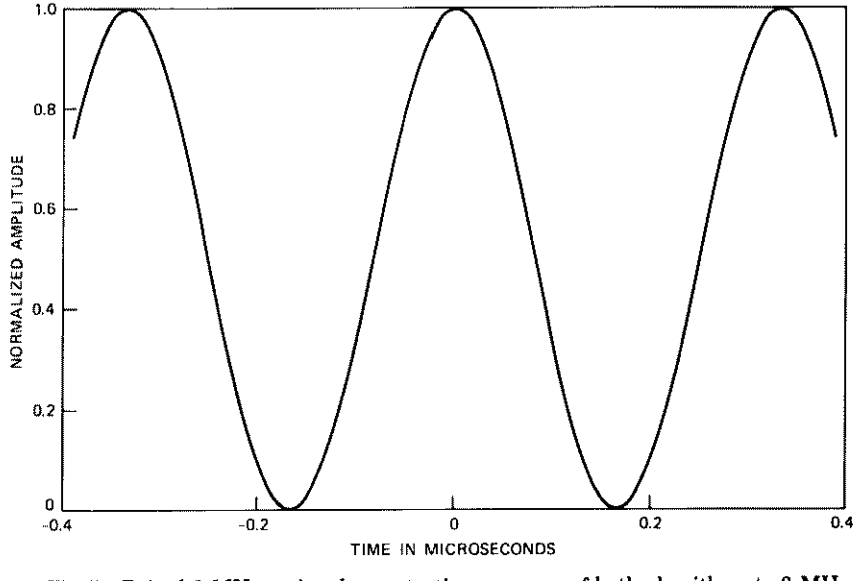



Fig. 7—Raised 3-MHz cosine demonstrating response of both algorithms to 3-MHz section of multiburst test signal.

at a frequency of 3 MHz, shown in Fig. 7. We chose this frequency because it is the intended maximum bandwidth of the differential signal. This would correspond to the fourth burst of the standard multiburst pattern[2] except that, for convenience, the amplitude is set to unity. Again, line-to-line repetition of the signal was assumed, and it can be seen in Fig. 8 that there is a 20-percent decrease in signal amplitude with $XDIF_2$. This again indicates a better reconstruction of the signal in the receiver with the available full bandwidth information.

## III. FREQUENCY DOMAIN ANALYSIS

Before going into the analysis of the two predictors in the frequency domain, it will be useful to look at the frequency spectrum of a typical video signal.

It is well known[3] that the energy of a typical video luminance signal is concentrated on the harmonics of the horizontal or line sweep frequency $F_H$ (≈15734 Hz). This means that most of the luminance energy of the video signal is distributed at multiples of $F_H$ along the frequency axis. The chrominance information also peaks at line frequency intervals, and, by placing the color subcarrier $F_c$ at an odd multiple of half the line frequency, i.e., $F_c = 455/2 \cdot F_H$, the chrominance information will be interleaved with luminance information peaks.

Chrominance information bandwidth is approximately 1.5 MHz, but, with the composite signal band-limited to 4.2 MHz, it results in

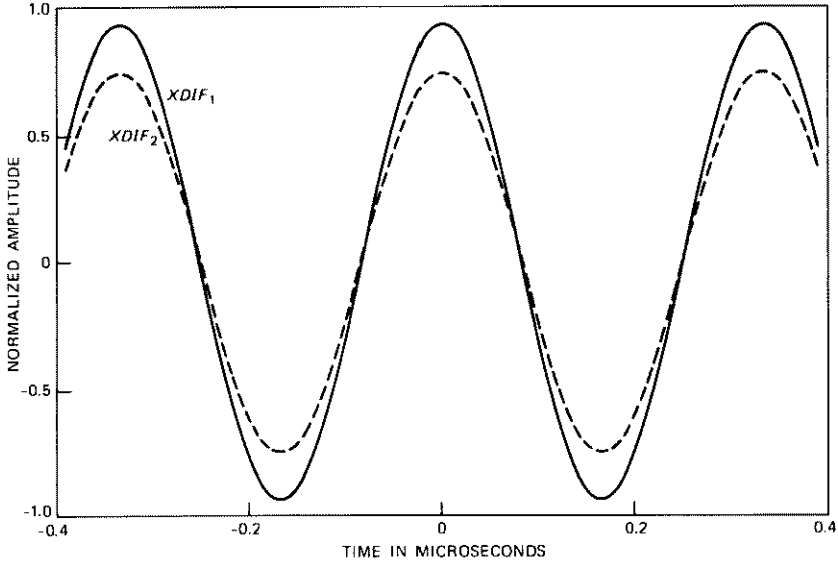



Fig. 8—Differential signal response of both algorithms to 3-MHz raised cosine.

a chrominance bandwidth of 1.5 MHz below the color subcarrier and 0.5 MHz above. Figure 9 shows the spectrum of a chrominance-modulated linear ramp test pattern. Figure 9a is the 0- to 4-MHz spectrum that shows the luminance information at low frequencies and the chrominance centered about $F_c = 3.579 \cdots$ MHz. Figure 9b shows the fundamental and the first nine harmonics of the line frequency, and Fig. 9c shows the color subcarrier and the line multiples on either side of it. It can be seen in this third photograph, also, that the luminance information (which peaks halfway between the chrominance peaks) is much lower than the chrominance at this frequency for this waveform.

The ideal frequency response of the linear system that produces the differential signal would then require that the differential signal be zero at even intervals of half the line frequency for the lower frequencies, and zero at odd intervals of half the line frequency for frequencies around the color subcarrier. Where the transition from odd to even intervals of half the line frequency should occur is not exactly defined, but it should not be lower than the lowest chrominance component, i.e., color subcarrier frequency minus the chrominance bandwidth, or approximately 2.08 MHz. In fact, the ideal value is picture dependent and, more exactly, would depend on the relationship between the luminance high-frequency content and the chrominance information content.

## IV. FREQUENCY DOMAIN COMPARISON

It is well known from circuit theory that, in a linear system with input $x(t)$, output $y(t)$, and impulse response $h(t)$,

$$Y(j\omega) = H(j\omega) \cdot X(j\omega),$$

where uppercase letters designate Fourier Transforms. Since the time relationship of the differential predictors is known, the frequency domain representation of the network $H(j\omega)$ can be found easily, and this process is shown in Appendix C for both the original and the new predictors.

With the results of these simple calculations, a program was written to plot the network response $|H(j\omega)|$. Figures 10 and 11 show the 0- to 100-kHz response of the original and new differential signal networks, respectively. It can be seen that both signals are zero for the line frequency and its harmonics. Since this is where most of the incoming signal energy is concentrated, both of these algorithms should behave well in this regard.

Figures 12 and 13 show the response of the two algorithms in a 100-kHz interval centered around the color subcarrier. It can be seen that both algorithms are zero at the color subcarrier and are again periodic

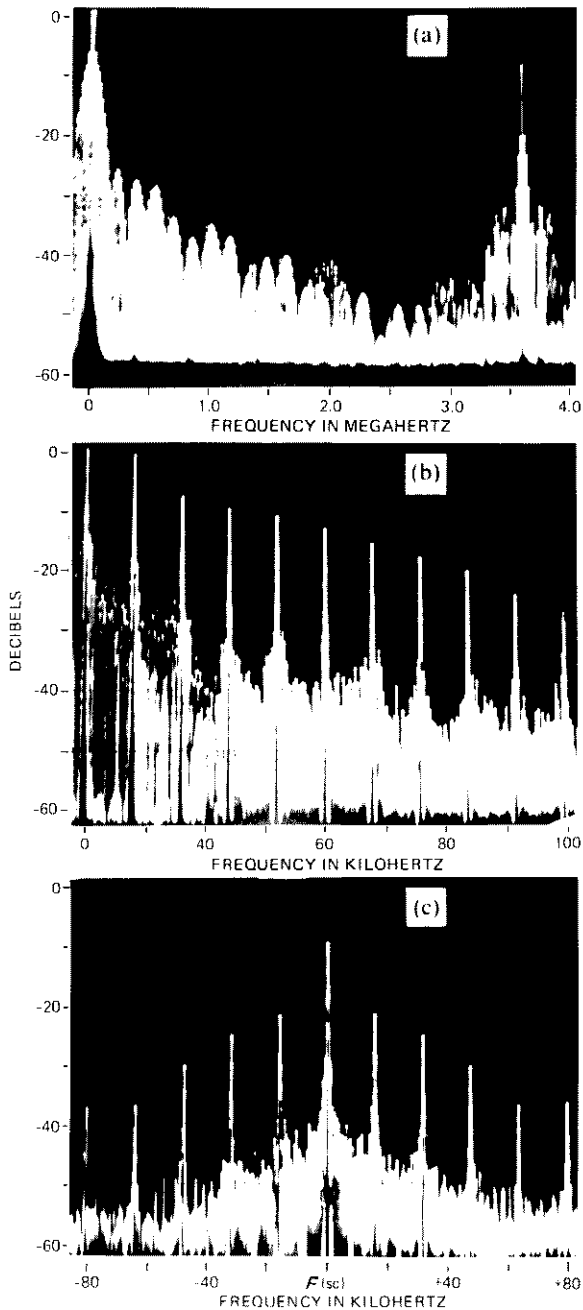



Fig. 9—Spectrum of chrominance-modulated linear ramp test pattern: (a) from 0 to 4.2 MHz; (b) from 0 to 100 kHz, with peaks at even multiples of half the line frequency; and (c) over a 100-kHz band centered around color subcarrier, with peaks at odd multiples of half the line frequency.

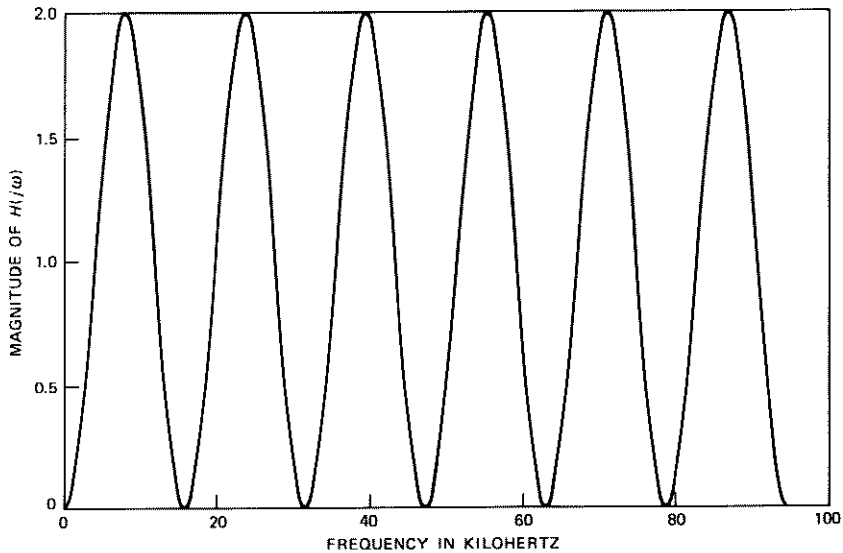



Fig. 10—Calculated differential signal spectrum of $XP_1$ algorithm from 0 to 100 kHz.

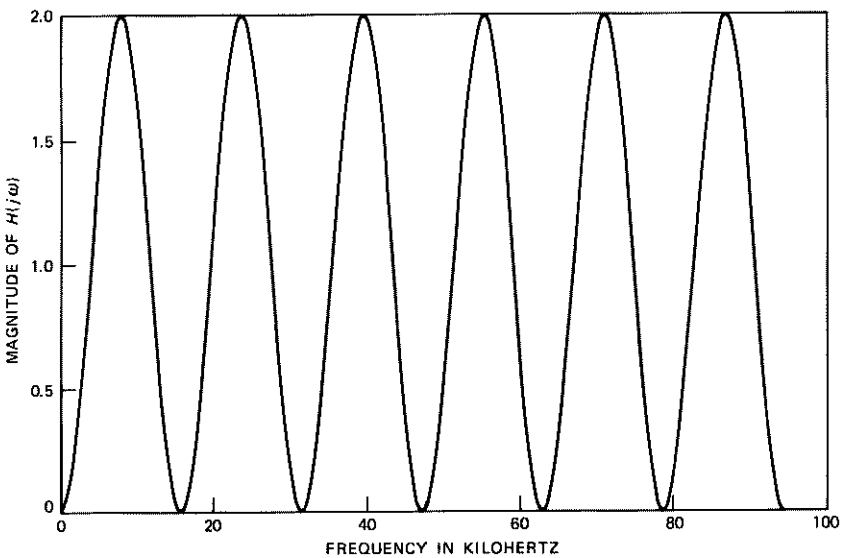



Fig. 11—Calculated differential signal spectrum of $XP_2$ algorithm from 0 to 100 kHz.

at a line rate. Since this corresponds to the high-energy chrominance portions of the incoming signal for this part of the spectrum, both algorithms should again behave well.

The difference in the response of the two algorithms can be seen,

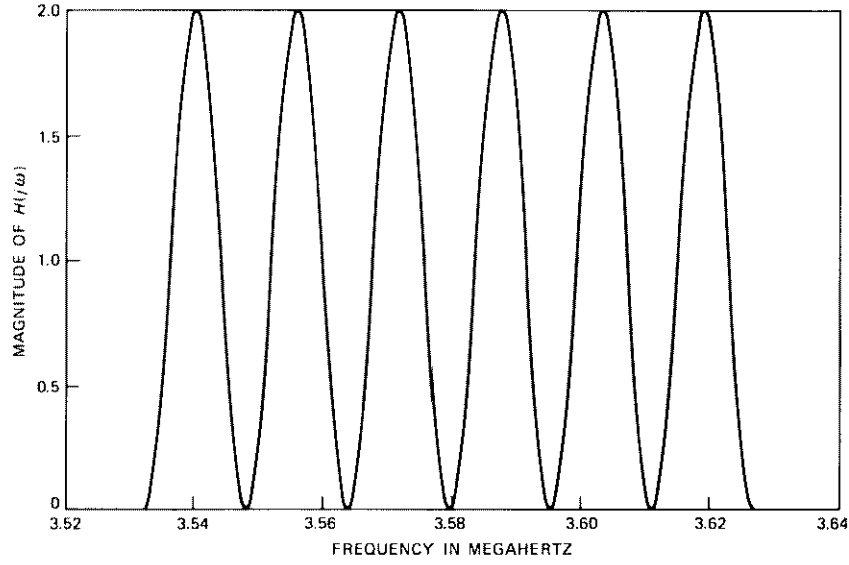



Fig. 12—Calculated differential signal spectrum of $XP_1$ algorithm for 100-kHz window centered around color subcarrier.

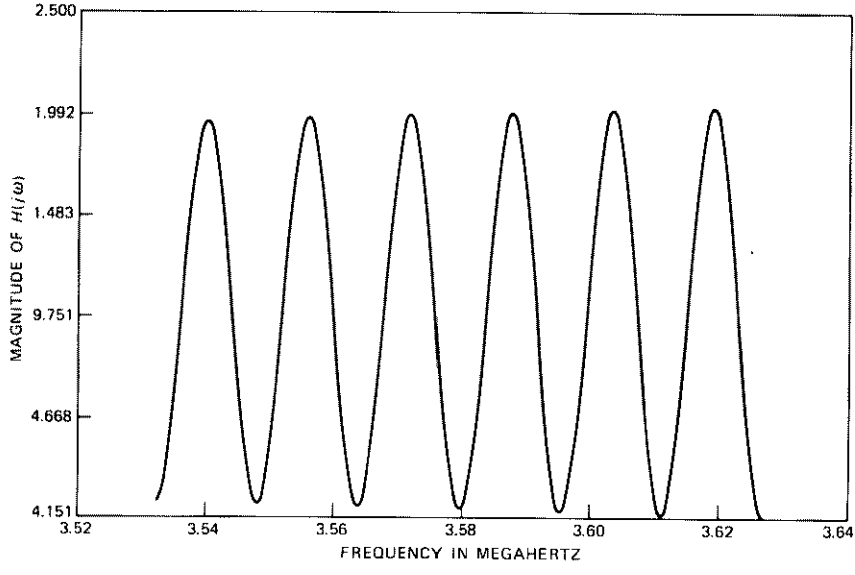



Fig. 13—Calculated differential signal spectrum of $XP_2$ algorithm for 100-kHz window centered around color subcarrier.

however, in Figs. 14 and 15. In these figures the differential signal response is plotted from 0 to 3 MHz, and the oscillatory response pattern is shown with the sine wave minimum values falling on even multiples of half the line frequency in the lower part of the spectrum, and at odd multiples of half the line frequency in the upper part.

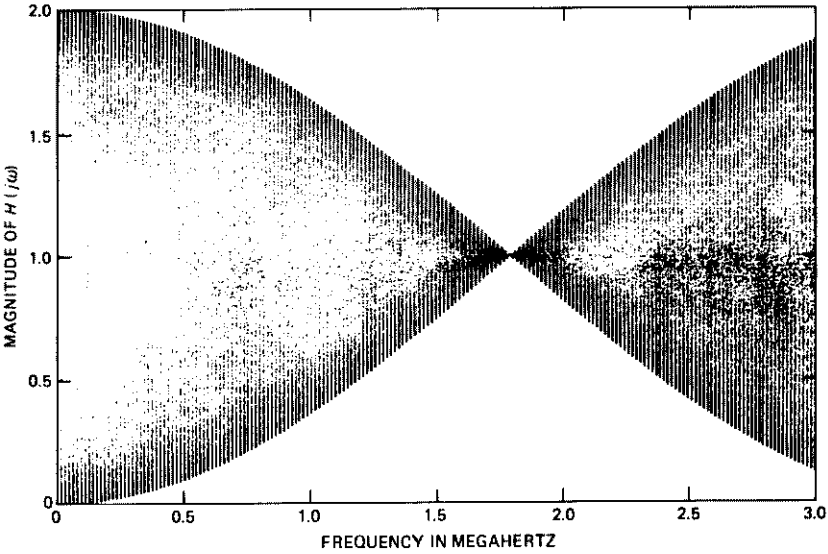



Fig. 14—Calculated differential signal spectrum of $XP_1$ algorithm from 0 to 3 MHz.

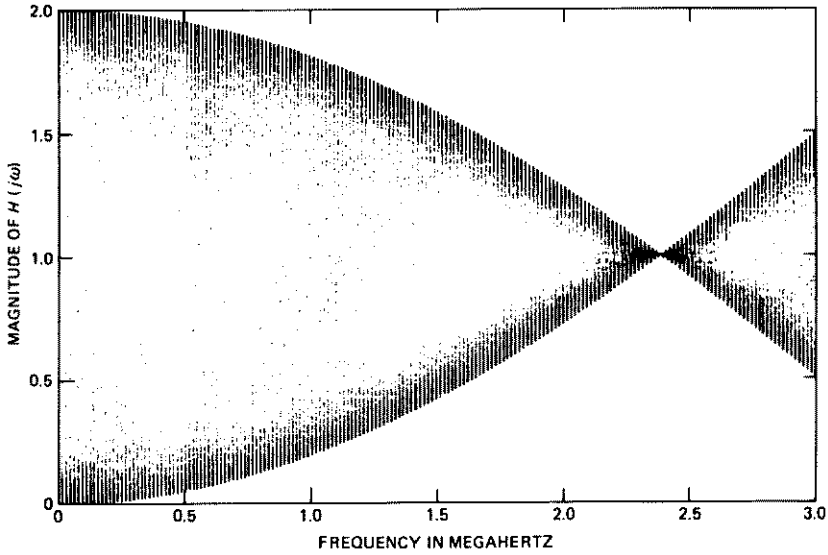



Fig. 15—Calculated differential signal spectrum of $XP_2$ algorithm from 0 to 3 MHz.

Because the lower limit of the chrominance information bandwidth is 2.08 MHz, we can see that the $XDIF_1$ transition is too soon (at about 1.8 MHz). This results in a greater than unity gain for the luminance information in this region and, therefore, a poorer prediction with no resulting savings in chrominance information because it is out of band. The $XDIF_2$ signal, however, is below unity for luminance response until 2.38 MHz, giving a better luminance prediction over an additional 500 kHz of bandwidth. This gives a poorer line difference prediction for the upper 300 kHz of the chrominance band, but since a fast chrominance change is almost always accompanied by a luminance change, the loss should be offset by an improvement in luminance response. Figure 16 plots the envelope for the luminance response and shows that the $XDIF_2$ signal will be lower throughout this whole region, indicating a better line difference prediction. These plots cross over at the color subcarrier, but that is of no concern since the differential signal will be band limited to 3 MHz in the proposed system.

## V. HARDWARE VERIFICATION OF RESULTS

To verify these results, a Tektronix test signal generator was modified so that a sweep generator signal could be added to the flat field test pattern. In this way the system could be analyzed with a full field sine wave input. The differential signal was then put into a spectrum

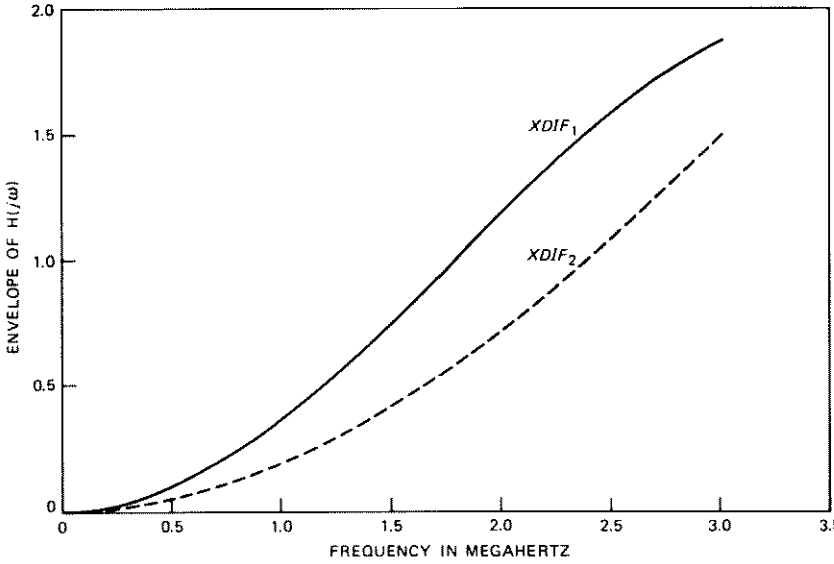



Fig. 16—Envelope of differential signal frequency response for $XP_1$ and $XP_2$, with samples taken at even multiples of half the line frequency.

analyzer. Figure 17 shows photographs of the 0- to 100-kHz spectrum and the 0- to 3-MHz spectrum. The response of the network is identical to the computed values shown in Figs. 10 through 16.

Time domain measurements were also made to verify the results of the analysis of this paper. The measurements were made with the full field pulse bar test pattern from the NTSC test signal generator. The differential signal output was band limited to 3 MHz, which resulted in some deviation from the calculated ideal results. Figures 18a and 18b show the response of the two algorithms to the sine squared pulse. A slight spreading of the signal caused by filtering can be seen, but otherwise the results are quite close to the ideal calculations. Figures 18c and 18d then show the response to the 156.25-ns ramp (bar pulse). Here again the high resolution of the ideal waveforms is lost because of filtering, but the predicted decrease in amplitude is clear.

Subjective tests were then carried out to determine if any degradation in picture quality was visible with either line difference prediction

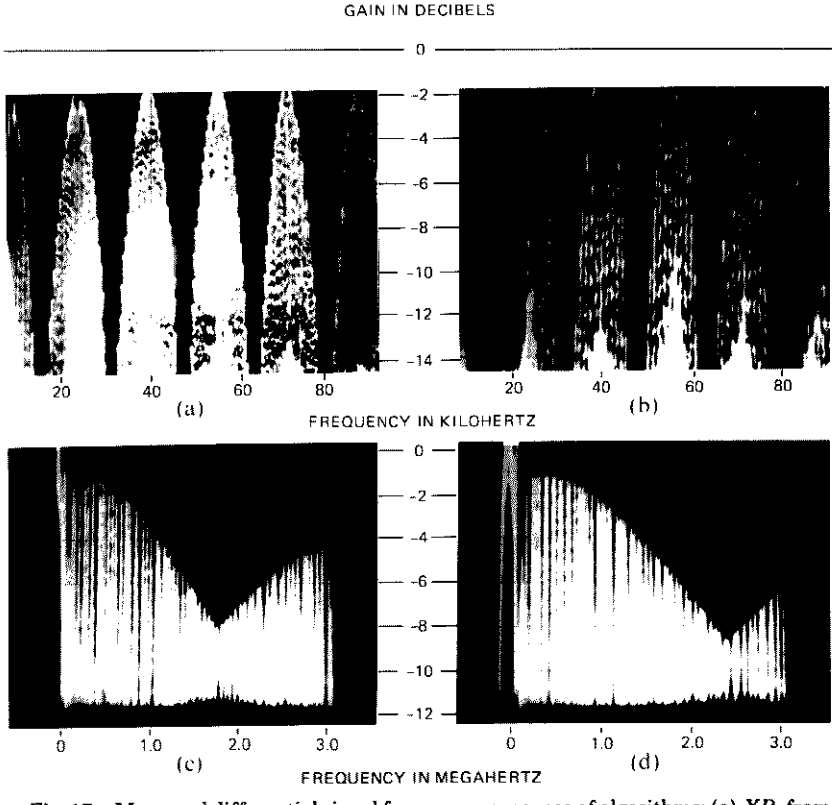



Fig. 17—Measured differential signal frequency responses of algorithms: (a) $XP_1$ from 0 to 100 kHz, (b) $XP_2$ from 0 to 100 kHz, (c) $XP_1$ from 0 to 3 MHz, and (d) $XP_2$ from 0 to 3 MHz.

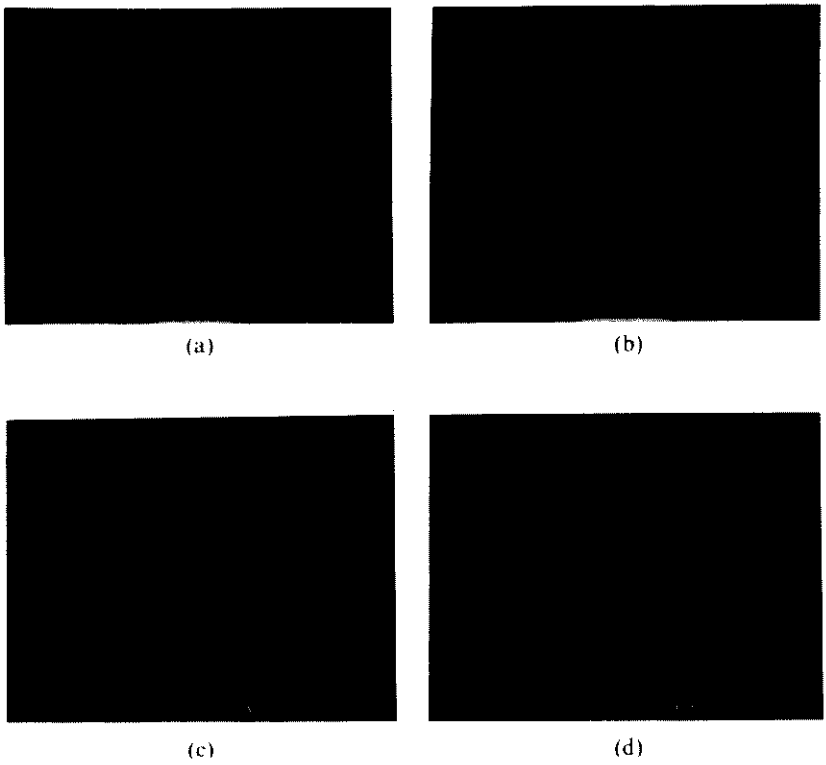
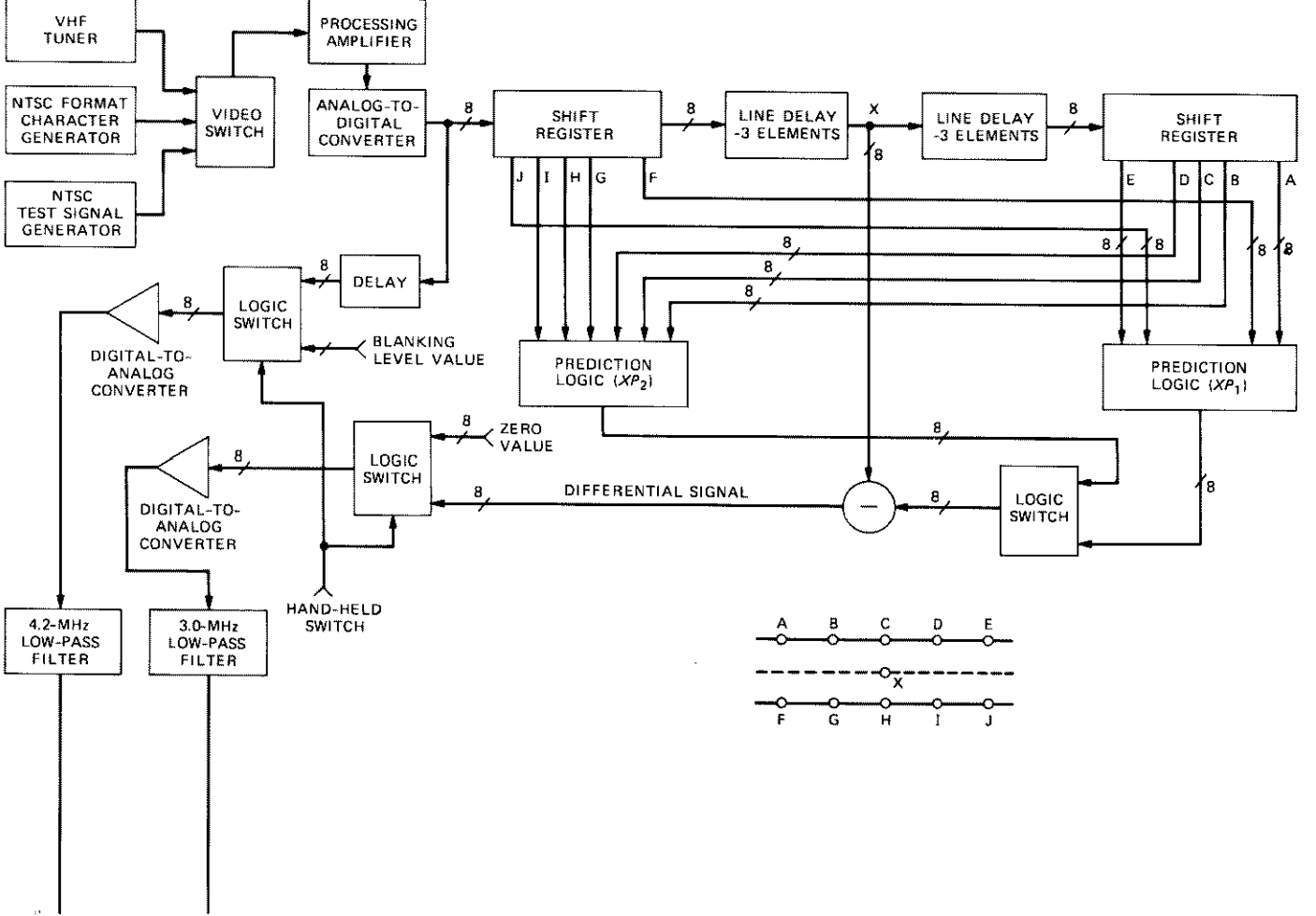



Fig. 18—Measured differential signal time domain responses of both algorithms to the sine for (a) $XP_1$ and (b) $XP_2$. Same responses to rising edge of the bar pulse for (c) $XP_1$ and (d) $XP_2$.

algorithm. To minimize distortion contributions from other hardware sources, so that contributions from the algorithm would be the primary cause of degradation, the TFM hardware was simplified. The band-limited differential signal and the time-multiplexed baseband signals from the transmitter were sent on individual coaxial cables to the receiver. This avoided signal degradations from the frequency multiplexing operation, and also transmission system degradations, leaving only the analog baseband amplifiers and two analog-to-digital-to-analog conversions as additional sources of degradation. Both the transmitter and the receiver use eight-bit pulse code modulation (PCM) for their digital processing, so that these additional contributions should be small in comparison with the differential signal band limiting. A block diagram of the test configuration is shown in Fig. 19.

Several video signals were evaluated using a stringent comparison between the reconstructed video signals and one which is unimpaired except for the two analog-to-digital conversions. The signals were switched on the same television monitor during the vertical blanking


VHF TUNER
PROCESSING AMPLIFIER
NTSC FORMAT CHARACTER GENERATOR
VIDEO SWITCH
ANALOG-TO-DIGITAL CONVERTER
SHIFT REGISTER
LINE DELAY -3 ELEMENTS
LINE DELAY -3 ELEMENTS
SHIFT REGISTER
NTSC TEST SIGNAL GENERATOR
J I H G F
E D C B A
DELAY
LOGIC SWITCH
BLANKING LEVEL VALUE
DIGITAL-TO-ANALOG CONVERTER
PREDICTION LOGIC ($XP_2$)
PREDICTION LOGIC ($XP_1$)
ZERO VALUE
LOGIC SWITCH
DIFFERENTIAL SIGNAL
LOGIC SWITCH
DIGITAL-TO-ANALOG CONVERTER
HAND-HELD SWITCH
4.2-MHz LOW-PASS FILTER
3.0-MHz LOW-PASS FILTER
A B C D E
X
F G H I J

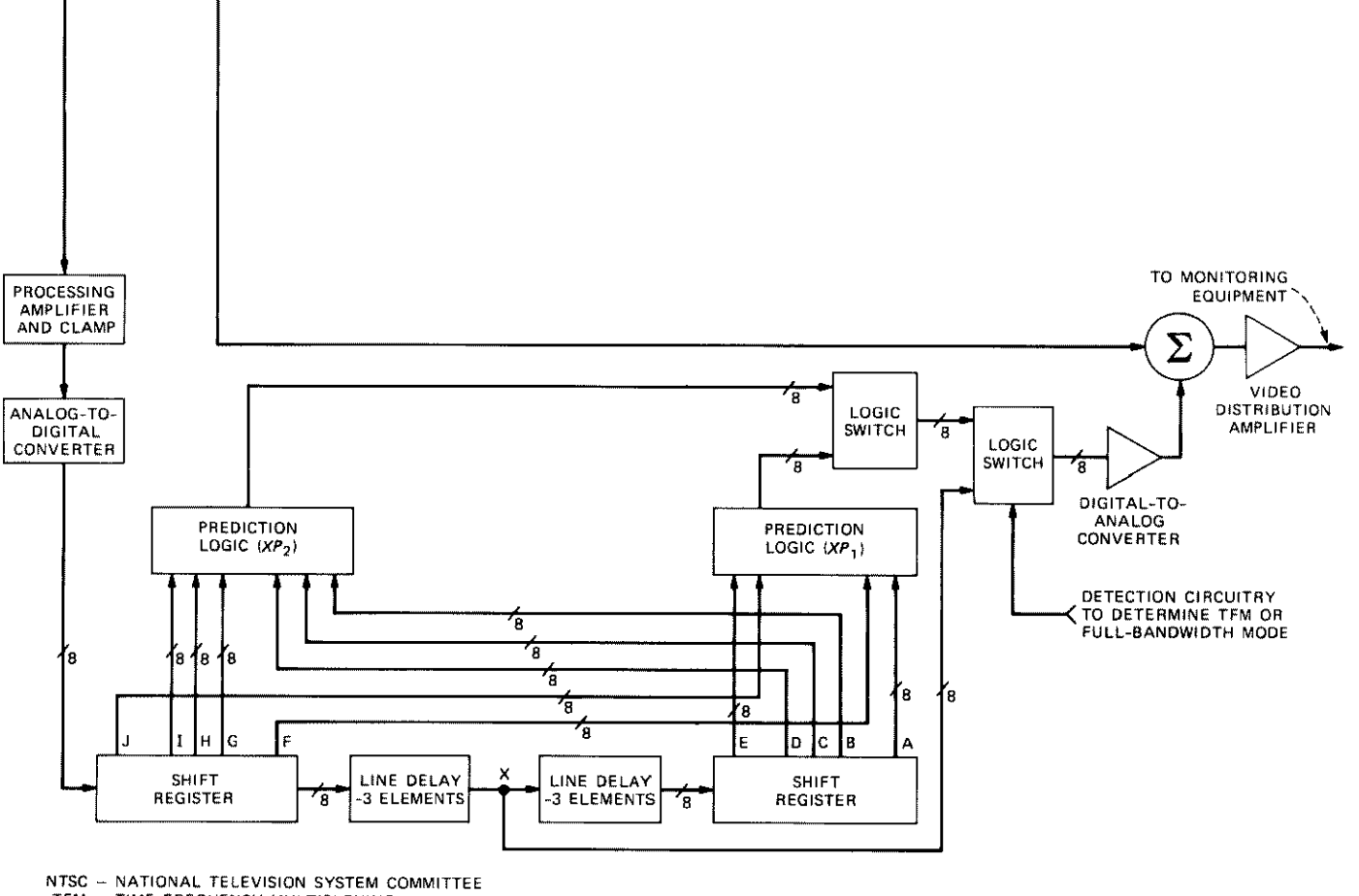



Fig. 19—Block diagram of test configuration used to elevate prediction algorithms.

intervals so that switching transitions were only visible when coding caused degradation. A toggle switch in both the transmitter and receiver allowed selection of either algorithm. The television monitor used for this evaluation did not contain a comb filter, so there is a possibility that some degradation caused by coding was masked by the luminance component band limiting. However, since both algorithms were evaluated with the same constraints, this did not appear to be a problem.

Results showed that with normal off-the-air video signals both algorithms performed well, and neither one produced any visible degradation. However, with electronically generated test patterns and text, the degradation of high-frequency signals, such as the $2T$ pulse, the multiburst pattern, and character edges, was less severe with the new algorithm.

## VI. CONCLUSIONS

Subjective tests verified the results of this analysis. Thus, the quality of a picture using the band-limited differential signal is improved with the new algorithm. The implementation is somewhat more difficult than the original one and requires only the addition of two more arithmetic stages. It is, therefore, a practical and desirable change to make in the system.

## REFERENCES


1. B. G. Haskell, "Time-Frequency Multiplexing (TFM) of Two NTSC Color TV Signals—Simulation Results," B.S.T.J., *60*, No. 5 (May–June 1981), pp. 643–660.
2. *NTC Report No. 7, Video Facility Testing Technical Performance Objectives*, June 1975, revised January 1976. Published by Public Broadcasting Service.
3. D. Fink, *Television Engineering Handbook*, New York: McGraw-Hill, 1957.


## APPENDIX A

### *Comparison of Differential Signal Responses to a Constant Color, Flat Field Luminance Video Signal*

Consider a uniform flat field of some color for the $XP_1$ algorithm shown in Fig. 1:

$$XDIF_1 = X - 1/4(A + E + F + J).$$

Let $X = U \cos(2\pi F_c t)$, where $F_c$ = the color subcarrier frequency. Then, since it is a uniform flat field,

$$A = E = F = J = U \cos(2\pi F_c t) = X.$$

Therefore,

$$XDIF_1 = 0 \qquad \text{for all } t.$$

For the $XP_2$ prediction shown in Fig. 2 with the same $X$ as above,

$$XDIF_2 = X - [1/2(B + D + G + I) - 1/2(C + H)].$$

Letting $\omega_c = 2\pi F_c$ we have

$$B = G = U \cos\left(\omega_c t + \frac{\pi}{2}\right),$$

$$D = I = U \cos\left(\omega_c t - \frac{\pi}{2}\right),$$

and

$$C = H = U \cos(\omega_c t + \pi);$$

therefore

$$\begin{aligned} XDIF_2 &= U \cos(\omega_c t) - \frac{U}{2}\left[2 \cos\left(\omega_c t + \frac{\pi}{2}\right)\right. \\ &\qquad \left. + 2 \cos\left(\omega_c t - \frac{\pi}{2}\right) + 2 \cos(\omega_c t)\right] \\ &= U \cos(\omega_c t) - U[-\sin(\omega_c t) + \sin(\omega_c t) + \cos(\omega_c t)] \\ &= U \cos(\omega_c t) - U \cos(\omega_c t) \\ &= 0 \qquad \text{for all } t. \end{aligned}$$

## APPENDIX B

### *Comparison of Differential Signal Responses to an Adjacent Line Color Change, Flat Field Luminance Video Signal*

Now consider a color change with no corresponding luminance change from one line to the next. The change will be considered between lines two and three of the prediction area, although calculations would be identical for a change from line one to two. Let $U$ = magnitude of the first color, $V$ = magnitude of the second color, and $\omega_c = 2\pi F_c$. Then for the $XP_1$ predictor,

$$\begin{aligned} XDIF_1 &= U \cos(\omega_c t) - \left[\frac{U}{2} \cos(\omega_c t) + \frac{V}{2} \cos(\omega_c t + \phi)\right] \\ &= \frac{U}{2} \cos(\omega_c t) - \frac{V}{2} \cos(\omega_c t + \phi), \end{aligned}$$

where the worst case would be for $\phi = \pi$, giving

$$XDIF_1 = \frac{U + V}{2} \cos(\omega_c t).$$

For the $XP_2$ predictor,

$$XDIF_2 = U\cos(\omega_c t) - \frac{1}{2}\left[U\cos\left(\omega_c t + \frac{\pi}{2}\right) + U\cos\left(\omega_c t + \frac{\pi}{2}\right) + V\cos(\omega_c t + \phi) + V\cos(\omega_c t + \pi + \phi) - U\cos(\omega_c t + \pi) - V\cos\left(\omega_c t + \frac{\pi}{2} + \phi\right)\right]$$

$$= U\cos(\omega_c t) - \frac{1}{2}[V\cos(\omega_c t + \phi) - V\cos(\omega_c t + \phi) + U\cos(\omega_c t) + V\sin(\omega_c t + \phi)]$$

$$= U\cos(\omega_c t) - \frac{U}{2}\cos(\omega_c t) - \frac{V}{2}\sin(\omega_c t + \phi)$$

$$= \frac{U}{2}\cos(\omega_c t) - \frac{V}{s}\sin(\omega_c t + \phi),$$

where the worst case will occur when $\sin(\omega_c t + \phi)$ equal $-\cos(\omega_c t)$ or at $\phi$ equal to $-\frac{\pi}{2}$, giving

$$XDIF_2 = \frac{U + V}{2}\cos(\omega_c t),$$

which is the same as for $XDIF_1$.

## APPENDIX C

### *Transfer Function Derivation for Both Prediction Algorithms*

Knowing the time relationship of the picture elements, we can find the Fourier Transform of the impulse responses of the differential signal algorithms. Let $p = 1$ picture element with a duration of $6.98412 \cdot 10^{-8}$ second. Then $910p = 1$ line $= 6.3555 \cdot 10^{-5}$ second.

For the $XP_1$ predictor:

$$XDIF_1(t) = X(t) - \frac{1}{4}[X(t + 908p) + X(t + 912p) + X(t - 908p) + X(t - 912p)],$$

$$XDIF_1(j\omega) = X(j\omega) - \frac{1}{4}[X(j\omega)e^{-j\omega 908p} + X(j\omega)e^{-j\omega 912p} + X(j\omega)e^{+j\omega 908p} + X(j\omega)e^{j\omega 912p}]$$

$$= X(j\omega)\left\{1 - \frac{1}{4}\left[e^{908pj\omega} + e^{-908pj\omega} + e^{912pj\omega} + e^{-912pj\omega}\right]\right\}$$

$$= X(j\omega)\left\{1 - \frac{1}{2}\left[\cos(908p\omega) + \cos(912p\omega)\right]\right\}$$

$$= X(j\omega)\cdot H(j\omega),$$

with

$$H(j\omega) = 1 - \tfrac{1}{2}[\cos(908p\omega) + \cos(912p\omega)].$$

Now for the $XP_2$ predictor:

$$\begin{aligned} XDIF_2(t) = X(t) &- \{1/2[X(t-911p) + X(t-909p) + X(t+909p) \\ &+ X(t+911p)] - 1/2[X(t-910p) + X(t+910p)]\}, \\ XDIF_2(j\omega) = X(j\omega) &- \{1/2[X(j\omega)e^{+911pj\omega} + X(j\omega)e^{+909pj\omega} \\ &+ X(j\omega)e^{-909pj\omega} + X(j\omega)e^{-911pj\omega}] \\ &- [X(j\omega)e^{910pj\omega} + X(j\omega)e^{-910pj\omega}]\} \\ = X(j\omega)\{1 &- [(e^{909pj\omega} + e^{-909pj\omega}/2) + (e^{911pj\omega} + e^{-911pj\omega}/2) \\ &- (e^{910pj\omega} + e^{-910pj\omega}/2)]\} \\ = X(j\omega)\{1 &- [\cos(909pj\omega) + \cos(911pj\omega) - \cos(910pj\omega)]\}. \end{aligned}$$

Therefore,

$$H(j\omega) = 1 + \cos(910pj\omega) - \cos(909pj\omega) - \cos(911pj\omega).$$

## AUTHOR

**Robert L. Schmidt,** B.S.E.E., 1982, Monmouth College; Bell Laboratories, 1972—. Mr. Schmidt is currently a member of the Radio Communications Research Department, where he has been exploring various techniques in combining multiple television signals onto a single analog radio channel. He has researched television signal encoding, bit rate reduction techniques, and software-controlled coding systems. Member, Eta Kappa Nu.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# On the Recognition of Isolated Digits From a Large Telephone Customer Population

By J. G. WILPON* and L. R. RABINER*

(Manuscript received February 25, 1983)

A field study was initiated to learn about the effects of various telephone transmission and switching conditions on the algorithms currently used in the Bell Laboratories, Linear Predictive Coding (LPC)-based, isolated word recognizer. Digit recordings were obtained from customers over a variety of transmission facilities. During a 23-day recording period a total of 11,035 isolated digits were recorded. For each recording, statistics were recorded about the line condition, the background environment, and the customer's ability to speak his/her telephone number as a sequence of isolated digits. Also recorded was information about the ability of the automatic word endpoint detector to find each spoken digit and to accurately determine the correct endpoints. The results of several recognition tests are presented—one using a previously defined set of laboratory-created digit reference templates, and several others using new sets of reference templates from a subset of the recorded digits. The performance of the recognizer is poor (average digit accuracy of 77.4 percent) using the laboratory template set, but improves substantially (average digit accuracy of 93.1 percent) for a template set created from the field recordings. The reasons for this improvement in digit recognition accuracy are presented, along with their implications to future work in isolated word recognition.

## I. INTRODUCTION

Research on the problems involved in speech recognition has been carried out at Bell Laboratories for close to a decade.[1-7] In all these

*Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

studies the speech database consisted of utterances recorded under laboratory conditions (i.e., cooperative subjects, soundproof booths, and subject prompting) using dialed-up lines over a local Private Branch Exchange (PBX). Peak signal-to-noise ratios ranged from 40 to 60 dB under these conditions. The recognition systems previously studied involved either a user training phase (speaker-dependent systems) or no training phase (speaker-independent systems). The vocabulary sizes ranged from as few as 10 words,[2] to as many as 1109 words.[7] Our past studies of speaker-independent systems involved a relatively small number of subjects, typically 100 for training and 10 to 40 for test calculations. Our current recognition systems performed very well given these conditions.[8]

To test the viability of speaker-independent, isolated word recognition systems for large user populations, it was necessary to conduct an experiment under "real world" conditions. Such an experiment involves using noncooperative telephone customers speaking in an uncontrolled environment over a set of randomly dialed telephone lines. This paper presents such an experiment and its implications of future speech recognition work. During the course of the experiment, a speech database was collected over a 60-day period in a Bell System environment (i.e., recordings obtained directly at a Bell System switching office) from over 3100 subjects. The vocabulary chosen for this study was the 10 digits, zero through nine (the digit zero was generally pronounced "oh"). Since we wanted a system that could handle a large number of users with the least amount of burden to the user we chose to make the system speaker independent.

There are several very important recognition issues that need to be resolved, and only by using a very large speech database, such as the one we obtained, can these issues be addressed. The most important issues involve training the recognizer. Since we have required that the recognition system be speaker independent, several questions arise as to how one obtains the necessary training data. Should training tokens be used from only the "best" speakers over the cleanest telephone lines, or should all the training samples be randomized, i.e., from any talker over any quality transmission line? Another issue involves the number of training tokens needed to adequately represent an extremely large number of potential users. In past studies we have used at most 100 representations for each vocabulary item. Do we necessarily need more tokens? In this paper we investigate these issues among others.

Our results indicate a distinct number of "real world" problems that must be considered when implementing a speech recognition system with a widespread applicability. These include properly treating highly variable background conditions, devising procedures for handling au-

tomatic endpoint detector failures, and problems associated with obtaining isolated speech input. Methods of handling these problems will also be discussed in this paper.

In Section II we describe how we obtained the speech recordings. Section III presents an evaluation of our current speech recognition system, on the basis of a series of recognition experiments. A discussion of the overall results and their implications is given in Section IV.

## II. RECORDING PROCEDURE

Figure 1 depicts the overall recording setup used in this study. Recordings were made at a Bell System switching office in Portland, Maine, and then transmitted back to the Murray Hill Laboratory for analysis. The sequence of operations to record a single user's speech was as follows:

1. A site observer (SO) issued a prerecorded spoken message (a prompt) requesting that the user speak his or her telephone number as a sequence of isolated digits. After the first three digits were spoken,* the observer then initiated recording (i.e., digitizing the spoken data) of the spoken number sequence. As each of the digits was spoken, the observer entered it on a keyboard. Four digits were nominally recorded (digitally at a 6.67-kHz rate) for each subject.

2. The observer determined if the digit sequence was not spoken in an isolated format (i.e., spoken without sufficient pauses). If so, the observer initiated another prerecorded spoken message (a reprompt) requesting the user to repeat the number with a longer pause between digits. After the subject completed speaking the number, he or she was given a prerecorded "Thank you" message.

During this first phase an observer at the Murray Hill Laboratory (MHO) had also been monitoring the call and now took over handling the call, performing the following operations:

1. The MHO also determined (based on listening) whether the digit sequence was spoken in an isolated manner. If the MHO decided that the speech was unacceptable (either because it was spoken in a connected manner or because of unacceptably bad telephone line conditions), a reject code was entered and the entire procedure was terminated for the current call.

2. If the call was acceptable, the MHO entered the sequence of spoken digits heard. This sequence was compared with that entered by the SO and any discrepancies (errors) were noted and fixed by listening to the recorded digits.

---

* For reasons of customer privacy we were not allowed to record all seven digits of the telephone number.

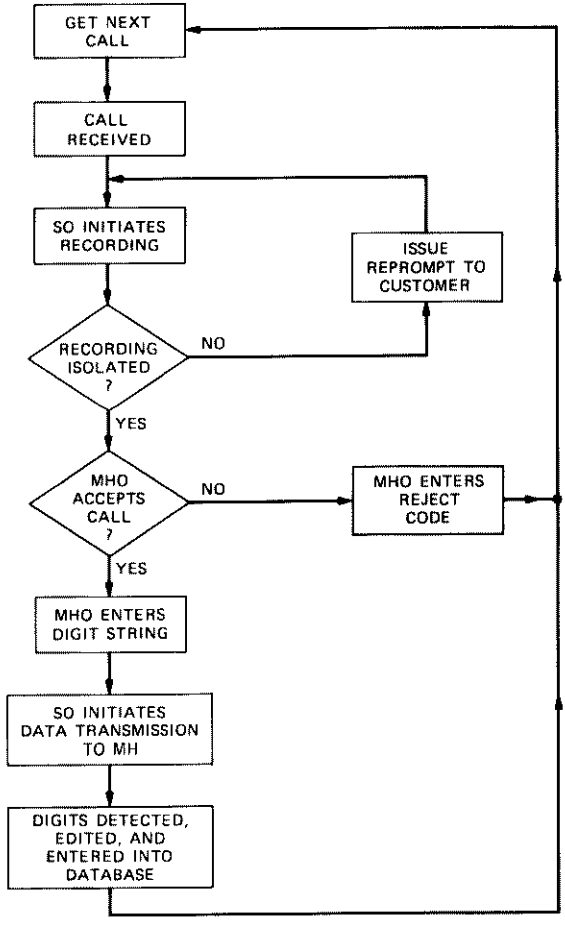



Fig. 1—The overall digit recording system.

3. At this point the MHO initiated digital transmission of the digitized speech from Portland to the MH laboratory.

Once the transfer was completed, an eighth-order Linear Predictive Coding (LPC) analysis, and automatic endpoint detection were performed.[9] The log energy of the waveform was displayed to the MHO, along with the automatically determined sets of endpoints indicating where in the recording interval the isolated words could be found. At this point the MHO had the option of modifying any or all sets of endpoints computed. The segmented speech was then entered into the database for later examination.

Using this procedure we recorded approximately 11,000 digits from

3100 subjects over a 23-day period. During the first 11 days no reprompting was used and recordings were taken for about 8 hours each day. Beginning wih the 12th day the reprompt procedure was instituted and we began recording for 12 hours each day. Figures 2 through 5 show the breakdown of the data recorded. Figure 2 shows a plot of number of digits recorded on a daily (session) basis, for males, females, and the total. For the first several days, only about 200 to 300 digits were recorded because both observers were learning. The dip in recordings around day 10 was due to a major snowstorm which curtailed recording. Figure 3 shows a histogram of the number of utterances recorded per digit for males, females, and total. The digits 2 and 3 had the highest number of occurrences, and the digits 9 and 0 had the fewest number of occurrences. This phenomenon is due to the fact that the digits 0,9 (and sometimes 1) are reserved in some positions for pay phones, businesses, etc. The general falloff in number of occurrences from 2 to 8 is due to the manner in which the telephone numbers were assigned in the Portland area.

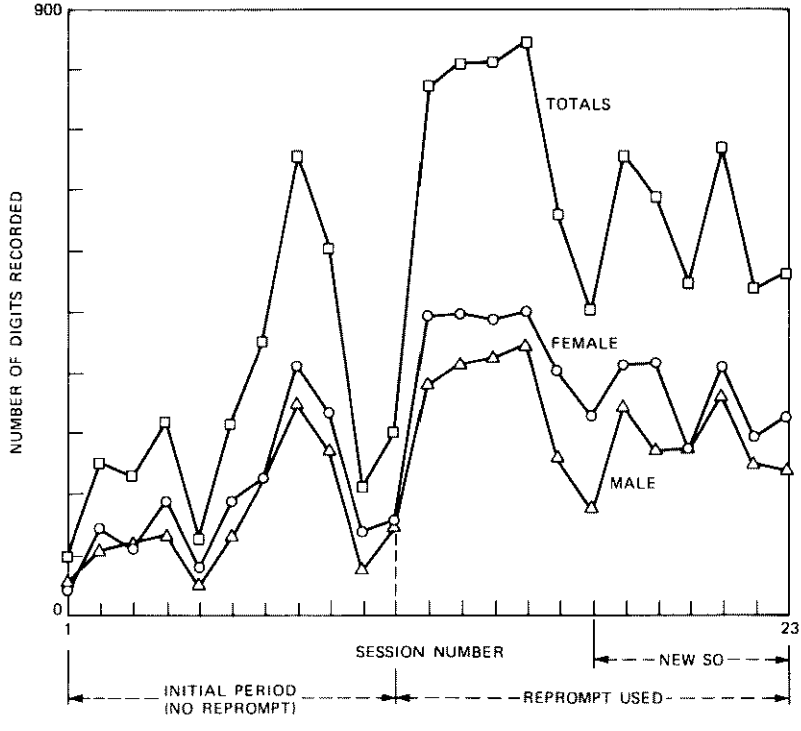



Fig. 2—The number of digits recorded as a function of session number for males, females, and combined (totals).

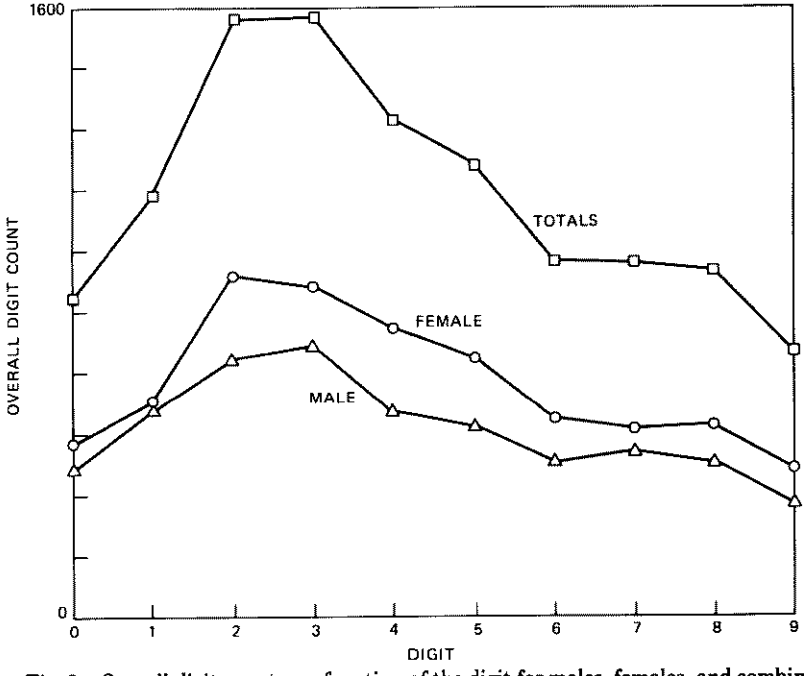



Fig. 3—Overall digit count as a function of the digit for males, females, and combined (totals).

There were several problems that occurred during the recording phase. These problems were classified as being in one of two groups. The first group contained problems associated with the telephone transmission conditions. Artifacts such as clicks, tones, and hum were often superimposed on the subject's speech. Resulting peak signal-to-noise ratios varied from as little as 10 dB to as much as 60 dB. The second group consisted of problems related to the talker and the environment in which he or she spoke. These included nonisolation of speech (i.e., the digits were connected) and the presence of extraneous background speech, such as people talking in the background or a television set being played at an audible level at the handset. Most of the user failures were severe enough to warrant elimination of the customer's speech from the database.

Figure 4 shows a plot of the percentage of calls accepted (i.e., at least 1 digit was extracted from the call) as a function of the session number. The rejected calls are a sum of both SO and MHO rejections. We can see that only 53 percent of calls yielded at least 1 digit. There were several reasons for such a low yield. The main reason for rejection was nonisolation of speech. This took one of two forms; either the subjects spoke in pairs of digits (e.g., 43 followed by 27), or they connected all four digits (e.g., 4327). Figure 5 shows a plot of the

percentage of these two types of rejections, as a function of total rejections, on a per-session basis. During the initial phase of recording, digit pair rejections accounted for about 30 percent of all rejections. Similarly, fully connected speech accounted for approximately 27

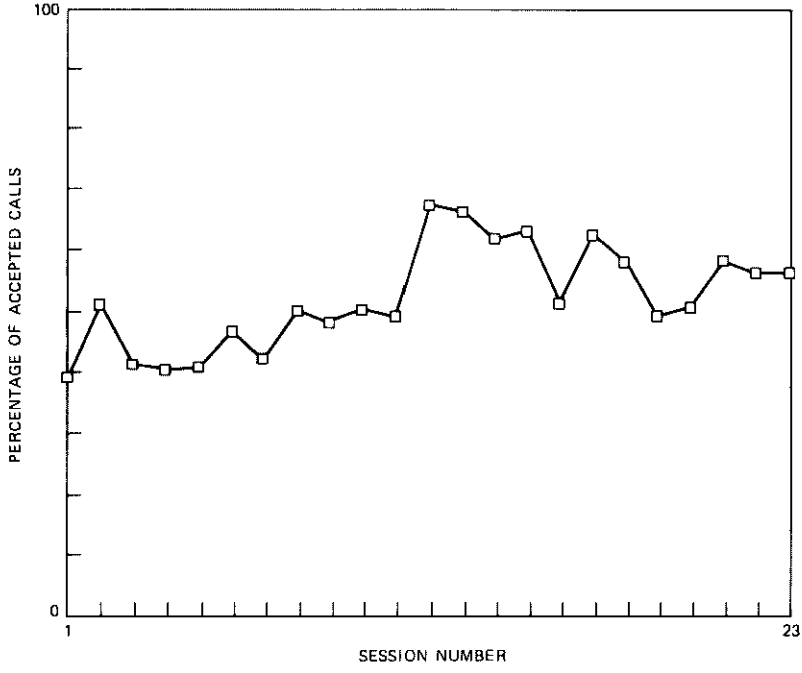



Fig. 4—The percentage of handled calls that were accepted as a function of session number.

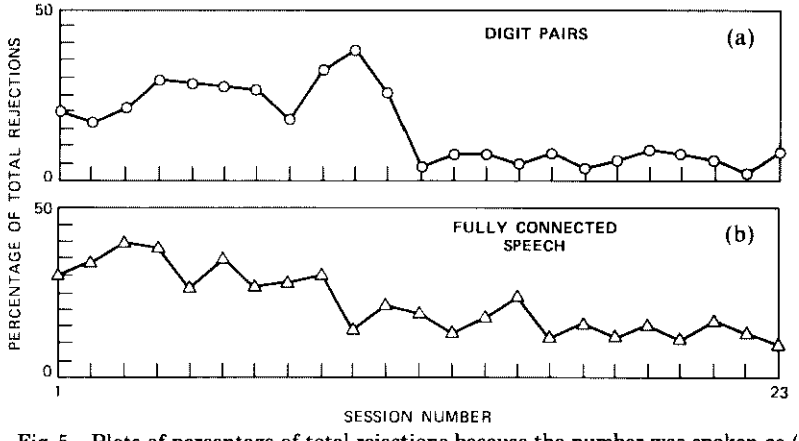



Fig. 5—Plots of percentage of total rejections because the number was spoken as (a) a set of digit pairs (b) as fully connected speech as a function of session number.

percent of total rejections. The plots show that a strong decrease in the number of these rejections occurred after the reprompting phase was initiated (Session 12), resulting in rejections rates of 9 and 15 percent for digit pairs and connected speech, respectively.

Another problem encountered during the recording process was the failure of the automatic endpoint detector to segment all the words properly. This algorithm[9] used the log energy contour of the speech and, after normalizing for background noise level, found as many digits as possible in the recorded string. For "ideal" transmission conditions and correct speaking of the number (i.e., as a sequence of isolated digits) the task of digit detection is relatively straightforward and essentially makes no errors.[9] Figure 6a illustrates such a case for the digit string /5946/. This figure shows the log energy contour of the recording. The dashed vertical lines indicate beginning and ending frames for automatically detected digits. For this example the peak-signal-to-average-background-noise ratio was about 54 dB (i.e., the average signal-to-noise ratio was close to 40 dB). Furthermore, the background noise was fairly stationary and at a low level.

Unfortunately, most of the recordings differed substantially from that of Fig. 6a. Figures 6b through 6e illustrate some problems that made automatic reliable detection of the digits very difficult. Figure 6b shows a log energy contour of the digit sequence /5282/ in which the signal level (during talking) was fairly low (peak-signal-to-background-noise ratio of 26 dB), and to further complicate matters, the background consisted of a mixture of noise and switching transients (clicks) generated within the telephone plant. For this sequence only three of the digits were automatically detected; the second digit was missed.

Figure 6c illustrates a case where the first three digits of the string /2383/ were spoken without a pause between the digits (i.e., in a connected format) and thus only one isolated digit could be used from this recording.

Figure 6d illustrates a case where the voice signal was corrupted by continuous signaling tones throughout the recording interval. The tones were set at a sufficiently high level so that the peak signal-to-tone ratio was only about 35 dB. Although, for this example, the locations of the major portions of each of the four spoken digits (4672) were properly detected, the parameterization of the signal (used later to recognize the digits) was greatly distorted by the tones present while the digits were being spoken. Furthermore, only the initial vowel portion of the digit six was located. The ending frication was lost in the tonal background.

Figure 6e shows an extreme case in which the background level of the line consisted of high-level noise and other extraneous sounds (i.e.,

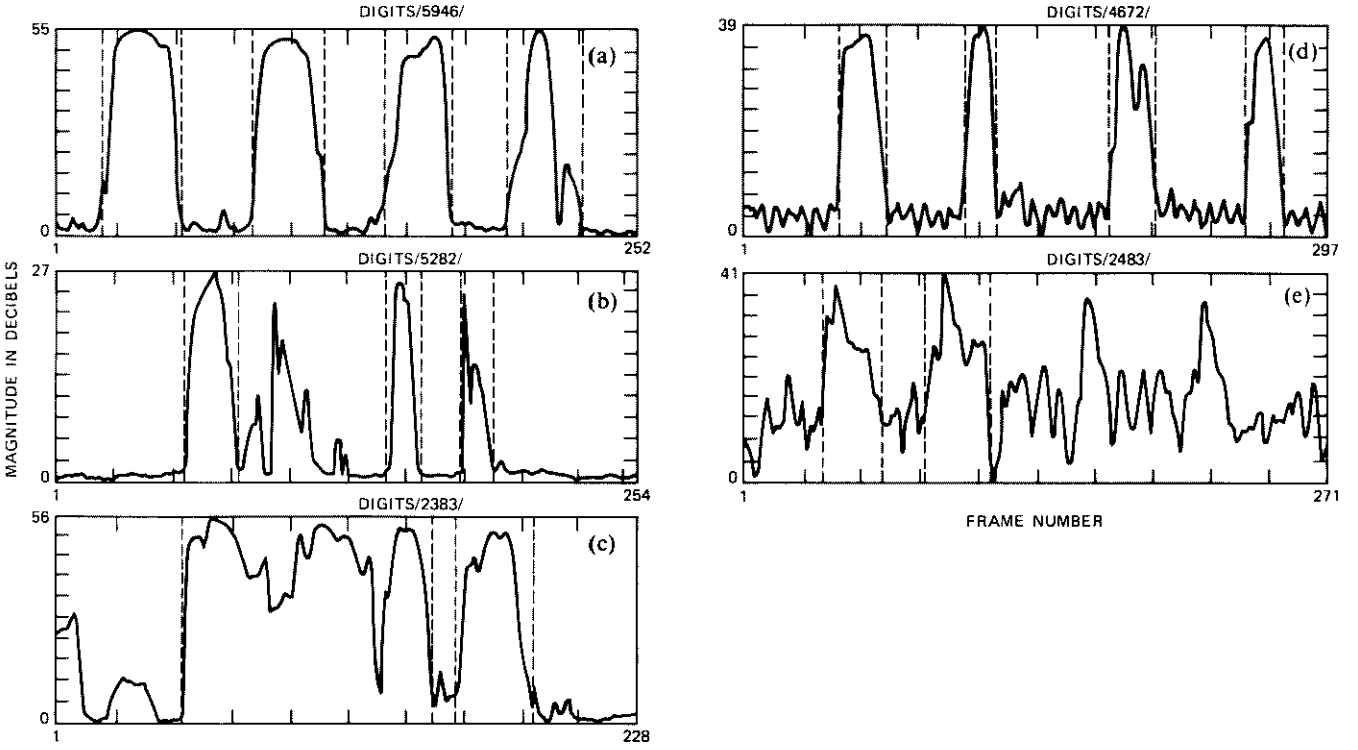



Fig. 6—Log energy contour for: (a) a high-quality recording of a 4-digit sequence; (b) a poor quality recording of a 4-digit sequence with low signal level and transmission clicks in the background; (c) a 4-digit sequence where the first three digits were not spoken in isolation; (d) a 4-digit sequence corrupted by continuous signaling tones throughout the recording interval; (e) a 4-digit sequence corrupted by high background noise and other extraneous sounds.

a very poor line). For this case it was impossible to detect accurate beginning and ending locations for any of the digits in the spoken string (/2483/).

The examples in Figs. 6d and 6e justify the importance of having the MHO check the accuracy of the automatic endpoint detector. In cases in which reliable endpoints are obtained automatically (no errors) the MHO allows the program to store the new digits in the database and update the relevant statistics. In all other cases the MHO can change endpoints or eliminate digits entirely. In this manner recordings with fewer than four isolated digits can provide one or more digits to the database and therefore the call is not entirely wasted. A discussion of the endpoint accuracy will be given in a later section.

An indication of how well the automatic endpoint detector performed was how often the MHO had to modify the endpoint sets. Figure 7 shows a plot of the percentage of calls requiring 0, 1, 2, 3, or 4 endpoint changes as a function of session number. This figure indicates that before using the reprompt, about 45 percent of the accepted calls needed no changes in word endpoints—i.e., no endpoint errors were made. After the reprompt was introduced the percentage rose to about 55 percent—i.e., a gain in endpoint accuracy of about 10 percent caused by the reprompt. This figure also shows that one set

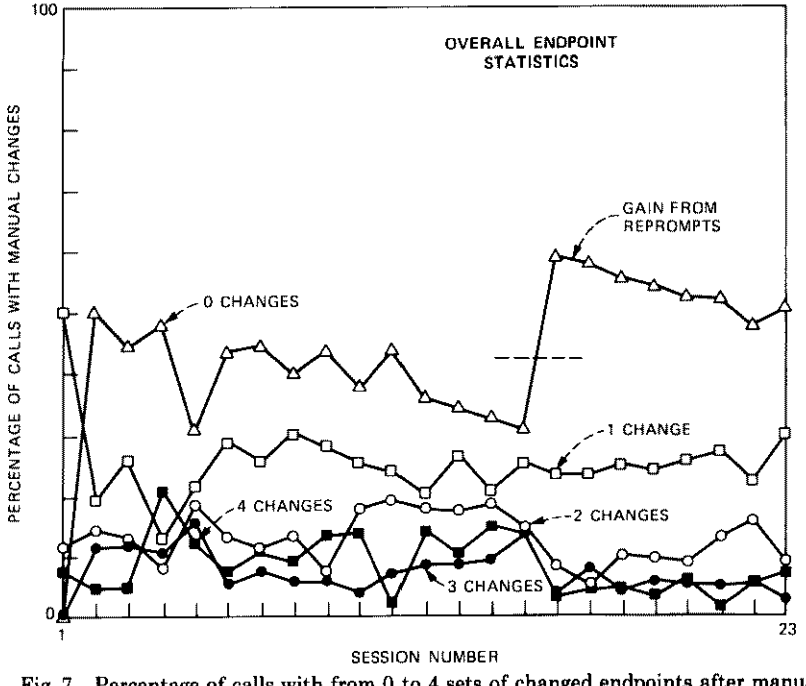



Fig. 7—Percentage of calls with from 0 to 4 sets of changed endpoints after manual corrections, as a function of session number.

of word endpoints needed to be modified about 25 percent of the time, and either three or four sets of endpoint modifications had to be made about 18 percent of the time. These results show clearly the necessity of improving the word endpoint detection. We are currently investigating into new methods of endpoint detection based on the types of problems encountered in this study.

## III. ISOLATED DIGIT RECOGNITION EXPERIMENTS

### *3.1 Description of final database*

After 23 days of recording over a two-month period, the final database contained 11,035 isolated digits taken from 3153 customers spoken during a wide range of telephone transmission conditions. Before any recognition experiments were performed, the full 11,035-digit database was listened to, and each digit was subjectively classified according to a set of background noise conditions to see if any one particular type of recording condition was harder for the recognition system to handle than another. Table I shows the individual types of conditions and the distribution of digits in the eight categories that were used. This table shows that 38.6 percent of the digits were classified as "acceptable". This meant that the background noise on the line was low (a signal-to-noise ratio in the range of 40 to 60 dB), and the given subject spoke in a clear, articulate voice, implying that the endpoint detector would have little problem with these utterances. These words were judged to be the "best" tokens, as close to laboratory data as was possible. The second category had similar signal-to-noise ratios, except here the callers did not speak in a normal fashion (presumably due to the novelty of being asked to speak in an isolated digit format). In such cases the customers dragged out words, or pronounced them in an abnormal way. This accounted for 10.3 percent of the digits. The remaining categories were used to describe the type of background noise present on the line. (If no background noise was present classes 1 and 2 were used.) About 12.5 percent of the digits came from strings that had a loud "crackling" noise superimposed on

Table I—Distribution of digits into categories based on background characteristics

| Condition/Code | Count | Percent |
|---|---|---|
| Acceptable/1 | 4257 | 38.6 |
| Customer-Related Problems/2 | 1133 | 10.3 |
| Crackling Noise/3 | 1379 | 12.5 |
| Pops, Clicks/4 | 1741 | 15.8 |
| Tones, Whines/5 | 1386 | 12.6 |
| Hum/6 | 752 | 6.8 |
| Whirring/7 | 129 | 1.2 |
| Background Speech/8 | 258 | 2.3 |
| Totals | 11035 | 100 |

the speech. Another 15.8 percent had pops and/or clicks throughout the recording. About 12.6 percent of the digits had loud tones present, mostly at 2600 Hz; and another 6.8 percent had loud "humming" noises superimposed with the speech from 200 to 400 Hz. We classified about 1.2 percent of digits as having a noise between "crackling" and "hum". (After we listened further we realized that this category could have been eliminated and its members classified as "crackling" noise.) The final 2.3 percent of the digits had background speech present—either other people's conversations, or television or radio sounds.

Table II shows the distribution of the 3153 calls, categorized by the number of digits (1, 2, 3, or 4) that were obtained from the calls. We see that for 68.8 percent of the calls four digits were actually obtained from the subject's spoken input. The average string length was 3.5 digits.

### *3.2 Description of recognition experiments*

To determine how well we could recognize digits from this database, eight isolated word speech recognition experiments were performed. In the first experiment, the template set consisted of a set of 12 speaker-independent templates for each of the 10 digits plus the word "oh" (since the majority of talkers used "oh" instead of "zero" for the digit 0). The template set (MH templates) was obtained several years earlier from a clustering analysis of the speech of 100 talkers (50 male, 50 female) with recordings having been made over a local, dialed-up telephone line, providing from 40 to 60 dB peak-signal-to-noise ratios for all recordings.[2] The recognizer was the LPC-based recognizer, which has been in use in the Acoustics Research Department for several years.[1,2]

Figure 8 shows the results of this first recognition experiment. Figure 8a shows plots of the recognition accuracy for the top word candidate and the top two-word candidates as a function of the session number. The overall average word accuracy for the top candidate was 77.4 percent and was basically steady (to within statistical variations) with time.

Figure 8b shows a plot of the overall individual variations in digit accuracy. The digit 8 attained an accuracy of close to 95 percent, whereas the digit 4 was only 50 percent accurate. The major problem

Table II—Distribution of calls yielding 1, 2, 3, or 4 digits

| | No. of Digits in String* | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | Total |
| Count | 109 | 375 | 501 | 2168 | 3153 |
| Percent | 3.4 | 11.9 | 15.9 | 68.8 | 100 |

* Average string length −3.5 digits.

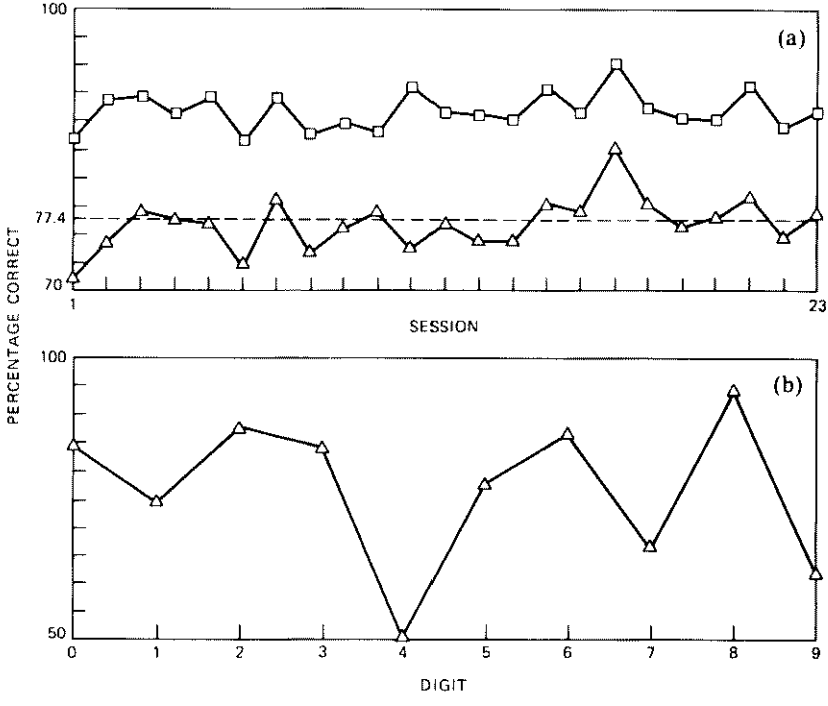



Fig. 8—Average digit recognition accuracy from: (a) MH templates for the top candidate and the top two candidates as a function of session number; (b) MH templates as a function of the spoken digit.

with the digit 4 was that about half the talkers pronounced it as /foe/, rather than /fore/, as represented in the template set. Hence for all such cases the digit 4 was recognized as 0 (i.e., the templates for "oh" provided the best match). Similarly, a modest number of confusions were found between the digits 5 and 9, and 1 and 9.

There was a basic problem with the training set of the first experiment. The pronunciations of the digits, as were prevalent in the Portland, Maine, area, were not well represented within the templates, and the difficult noise recording conditions led to large analysis degradations. Therefore, a second recognition experiment was run in which about 35 percent of the database was selected at random on a per-digit basis as a training set from which a new set of word reference templates [Portland (PO) random templates] was created. Reference template sets were generated using the UWA clustering algorithm of Rabiner and Wilpon,[6] yielding 12-, 20-, 25-, and 30-template-per-word sets, with the 30-template-per-word set yielding the best recognition results. The entire database of 11,035 digits was again used as a test set and the recognition results are given in Figs. 9 through 11 for the 30-template-per-word set. Figure 9a shows plots of the average word

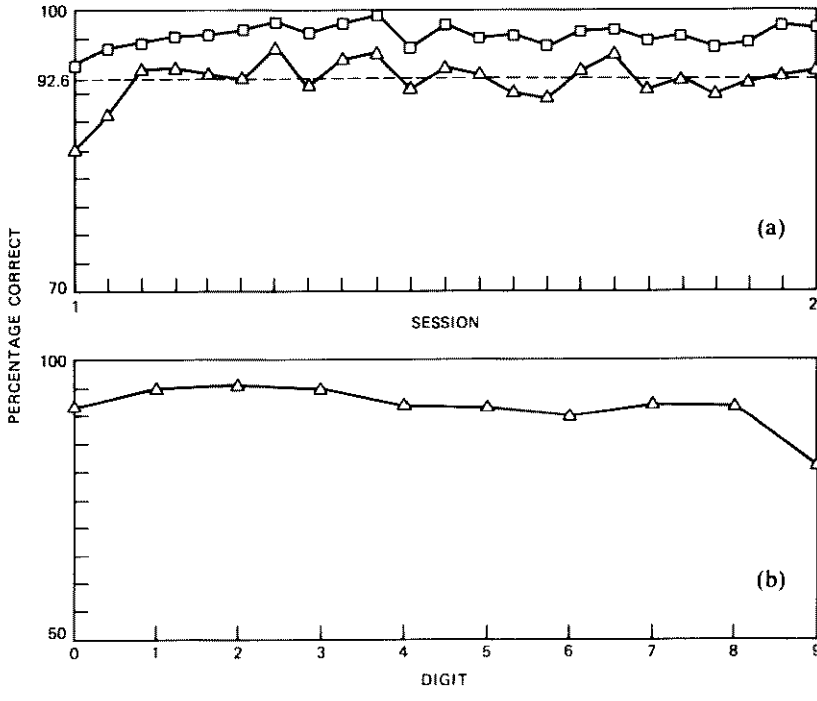



Fig. 9—Average digit recognition accuracy from: (a) PO random templates for the top candidate and the top two candidates as a function of session number; (b) PO random templates as a function of the spoken digit. (The same scale is used in Fig. 8b for comparison.)

recognition accuracy as a function of session number for both the top and the top two word candidates. The overall average word recognition accuracy for the top candidate was 92.6 percent and the individual session scores were fairly constant in time.

Figure 9b shows the overall recognition rates for the individual digits. We now see that all digits, except 9, were recognized with greater than 90 percent accuracy. Interestingly, the digit 9 was the one with the fewest tokens in the training set; hence improved recognition on 9 might result from a larger training set.

Figure 10 shows a plot of the overall word recognition accuracy as a function of the number of templates used per digit. The recognition rate is essentially flat for about 18 or more templates per word; hence only small reductions in accuracy would result from reducing the computation by almost one half.

Finally, Fig. 11 shows the individual accuracy scores for each digit as a function of the number of templates per digit. For some digits, e.g., 4, 6, 8, the recognition accuracy saturates for a small number of templates per digit, whereas for other digits, e.g., 0, 5, 7, 8, a large

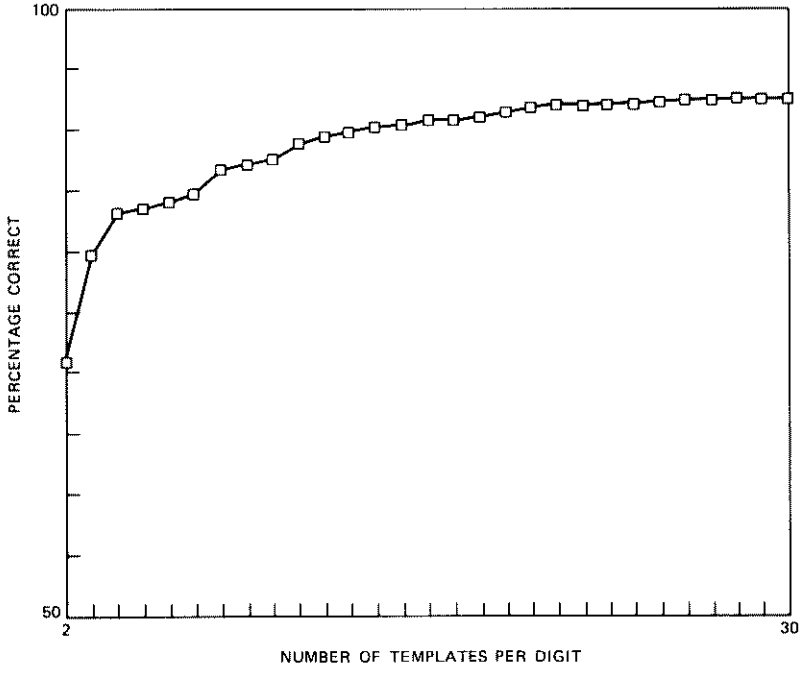



Fig. 10—Average digit recognition accuracy from PO random templates as a function of the number of templates used per digit.

number of templates per digit are needed. These results suggest that a reference set with a variable number of templates per digit could conceivably perform as well as the 30-template-per-digit set.

A problem in using a random set of utterances to train the system is the wide variability of transmission conditions present in the training tokens. The clustering procedure used to generate the templates tries to split the different spoken versions of a word into "similar" groups. If we now add an independent component to the speech, namely transmission variability, one would expect clusters also to be formed based on similarities in background conditions. Therefore we proposed the following recognition test. We used all the data that had been classified as "acceptable" (about 35 percent of the entire database) as a training set, from which we obtained a set of word reference tokens (PO clean templates). Since we were interested in a comparison with the PO random template experiment, a total of 30 reference templates were created for each digit. Again, the entire 11,035-word database was used as a testing set. The recognition results are summarized in Figs. 12 and 13. Figure 12a shows plots of recognition accuracy as a function of session number for the top and the top two word candidates. The average recognition accuracy over all days was

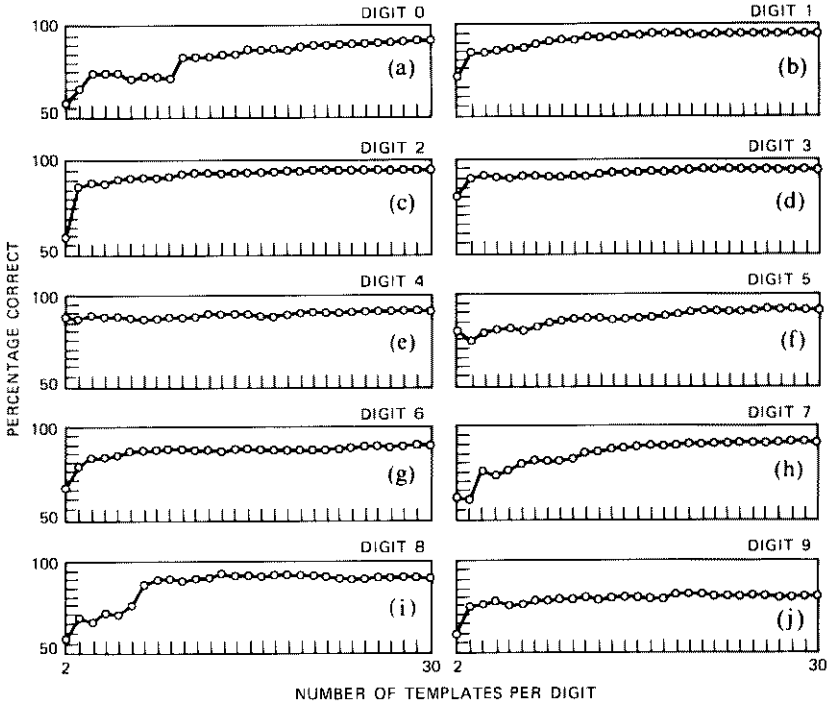



Fig. 11—Average recognition accuracy from PO random templates, for each individual digit, as a function of the number of templates used per digit.

93.1 percent. Compared with the PO random templates the difference in recognition accuracy was only 0.6 percent. Several inferences can be made from this result. First, it may not be necessary to go through the very time-consuming job of listening to several thousand spoken digit sequences and extracting only the "best" for use in training. Second, this result may indicate that the population of approximately 3900 training tokens was large enough to encompass all speaker variations (for most digits) and also all types of transmission system degradations.

Figure 12b shows a plot of the individual digit recognition scores. We see that the digit 9 still has a lower accuracy than the other digits. Figure 13a shows digit recognition accuracy as a function of condition code for the condition codes described previously. An accuracy of 96.6 percent was obtained when the testing set was also from the acceptable category (i.e., the same as the training data). The worst recognition results (86.7 percent) came from the digits categorized as having a very pronounced "humming" sound present during recording. Figure 13b shows string recognition accuracy as a function of condition code. The highest accuracy was 89.1 percent for "acceptable" data and the

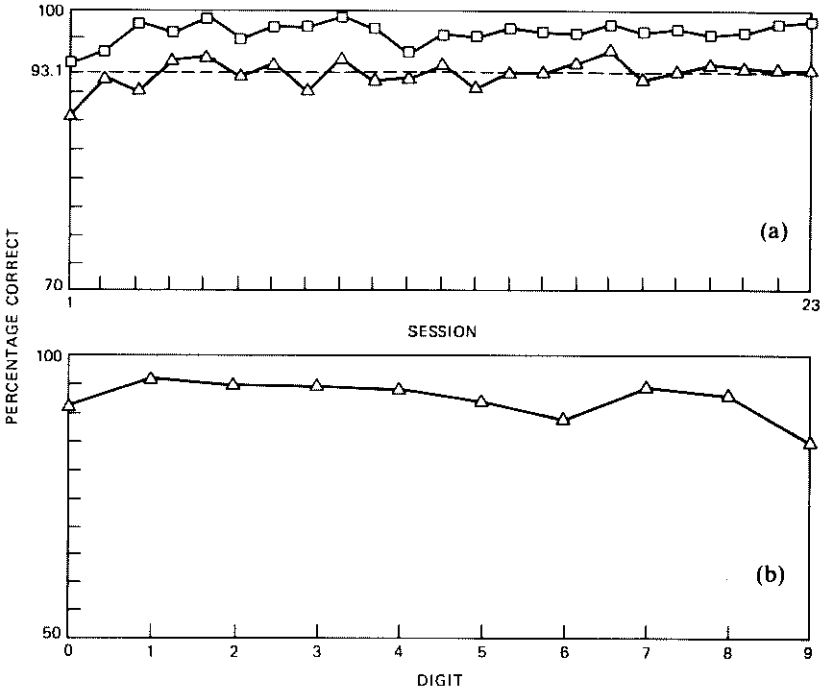



Fig. 12—Average digit recognition accuracy from the (a) PO clean template set for the top and top two candidates as a function of session number; (b) as a function of the spoken digit.

lowest accuracy was 64.3 percent for strings within the "hum" classification. The average string accuracy over all conditions was 80.8 percent. It was clear from these figures that the recognition errors were not distributed uniformly over all recording conditions. As a result, we were interested in determining whether digit errors were independent within each classification. Dependency might indicate that some sort of syntax could be applied to improve overall string accuracy.

To test the assumption that digit errors were independent within each string and classification category, a simple Bernoulli model was assumed in which the probability of error of a single digit was called $\alpha$. Under these conditions, the probability, $P$, of four correct recognitions within a string of four digits is

$$\begin{aligned} P(4 \text{ correct}) &= 1 - P(\text{single error}) + P(\text{double error}) \\ &\quad - P(\text{triple error}) + P(\text{quadruple error}) \\ &= 1 - 4\alpha + 6\alpha^2 - 4\alpha^3 + \alpha^4. \end{aligned} \tag{1}$$

For small $\alpha$'s eq. (1) can be approximated as

$$P(4 \text{ correct}) \approx 1 - 4\alpha. \tag{2}$$

Given that the average string length over all strings recorded was 3.5 digits

$$P(\text{string correct}) \approx 1 - 3.5\alpha. \tag{3}$$

In testing this hypothesis if we look at Fig. 13a, and multiply the digit error rates by 3.5 for each condition, we immediately see that eq. (3) above does indeed hold (at least to a reasonable level of accuracy). This indicates that within each classification the errors were probably distributed uniformly and randomly.

The next series of recognition experiments was carried out to determine whether we were using too many training tokens and therefore could reduce the amount of training data collected in future studies. Reference sets were created using one-third, one-half, and two-thirds of the data used in creating the 30-template-per-word PO clean template set (about 11.5, 17.2, and 23 percent of the entire database, respectively). The results are given in Tables III and IV. Table III shows the number of tokens in each training set or each digit, and Table IV shows the per-digit minimum, maximum, and

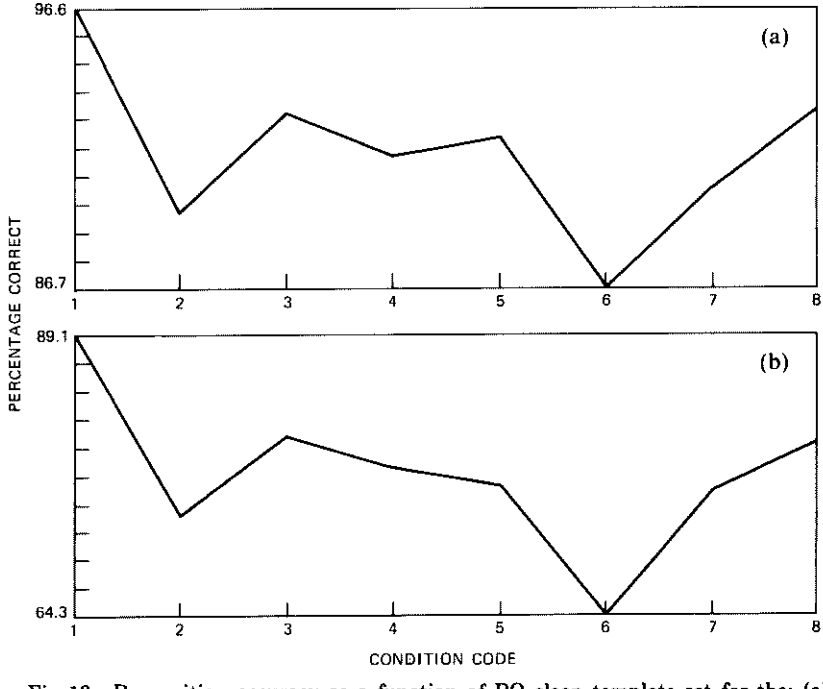



Fig. 13—Recognition accuracy as a function of PO clean template set for the: (a) acoustic classification on a per-digit basis; (b) acoustic classification on a string basis.

average accuracies for each experiment performed. The average per-digit recognition accuracy for the 12-template-per-word PO clean (one-third) template set was 89.7 percent for the 20-template-per-word PO clean (one-half) template set it was 91.4 percent; and for the 25-template-per-word PO clean (two-thirds) template set it was 92.2 percent.

The results show that the digit 9 consistently had the worst recognition scores. One possible explanation for this could be that in all cases the digit 9 had the smallest training set. The next recognition experiment was run to test this hypothesis. In this experiment we used a fixed number of training tokens per word. We were interested in seeing whether the 9 scores improved relative to the other digits. (Obviously we did not expect the overall accuracy to improve.) The results of this experiment are given in Table IV. We can see that the recognition accuracy for 9 has not changed (compared to the PO clean

Table III—Distribution of training tokens for individual words for each recognition experiment

| | MH | PO Random | PO Clean | PO Clean (1/3) | PO Clean (1/2) | PO Clean (2/3) | PO Clean (Fixed) | PO Clean Variable |
|---|---|---|---|---|---|---|---|---|
| 0 | 100 | 306 | 280 | 93 | 140 | 186 | 233 | 280 |
| 1 | 100 | 450 | 391 | 130 | 195 | 260 | 233 | 391 |
| 2 | 100 | 576 | 519 | 173 | 260 | 346 | 233 | 519 |
| 3 | 100 | 540 | 562 | 187 | 281 | 374 | 233 | 562 |
| 4 | 100 | 432 | 462 | 154 | 231 | 208 | 233 | 462 |
| 5 | 100 | 360 | 415 | 138 | 208 | 272 | 233 | 415 |
| 6 | 100 | 306 | 301 | 100 | 150 | 200 | 233 | 301 |
| 7 | 100 | 360 | 325 | 108 | 162 | 216 | 233 | 325 |
| 8 | 100 | 306 | 314 | 104 | 157 | 208 | 233 | 314 |
| 9 | 100 | 234 | 233 | 77 | 166 | 154 | 233 | 233 |

Table IV—Summary of recognition accuracies for the eight recognition experiments

| | MH | PO Random | PO Clean | PO Clean (1/3) | PO Clean (1/2) | PO Clean (2/3) | PO Clean Fixed | PO Clean Variable |
|---|---|---|---|---|---|---|---|---|
| 0 | 84.6 | 91.5 | 91.1 | 85.9 | 89.3 | 90.0 | 92.5 | 93.3 |
| 1 | 75.0 | 94.9 | 96.0 | 90.0 | 94.7 | 95.0 | 95.5 | 94.8 |
| 2 | 87.8 | 95.8 | 94.9 | 95.0 | 94.5 | 95.5 | 94.4 | 93.1 |
| 3 | 84.3 | 95.2 | 94.7 | 94.5 | 95.2 | 95.5 | 94.6 | 95.0 |
| 4 | 50.5 | 92.3 | 94.3 | 91.6 | 91.1 | 90.8 | 89.7 | 93.2 |
| 5 | 78.0 | 91.9 | 92.2 | 86.8 | 89.3 | 90.8 | 90.2 | 91.5 |
| 6 | 86.8 | 90.3 | 89.1 | 93.9 | 90.0 | 89.0 | 89.0 | 88.8 |
| 7 | 66.9 | 92.4 | 94.7 | 86.0 | 91.3 | 92.7 | 92.3 | 95.3 |
| 8 | 94.6 | 92.1 | 93.6 | 92.1 | 88.7 | 94.7 | 93.1 | 93.5 |
| 9 | 62.3 | 81.6 | 85.3 | 82.3 | 83.3 | 80.9 | 85.6 | 84.7 |
| Average | 77.4 | 92.6 | 93.1 | 89.7 | 91.4 | 92.2 | 92.1 | 92.6 |
| Minimum (Word) | 50.5(4) | 81.6(9) | 85.3(9) | 82.3(9) | 83.3(9) | 80.9(9) | 85.6(9) | 84.7(9) |
| Maximum (Word) | 94.6(8) | 95.8(2) | 96.0(1) | 95.0(2) | 95.2(3) | 95.5(2) | 95.5(1) | 95.3(7) |

template set), while the accuracies for the other digits have fallen. These results imply that some digits, e.g., 2, 3, need significantly fewer training tokens then do other digits, like 9.

In the final recognition experiment, we investigated the use of a variable number of templates per word. The number of templates per word was chosen on the basis of the curves of Fig. 11 at the point where the recognition accuracy leveled off. The training set for this experiment was the same as used in the PO clean template set, namely, approximately 35 percent of the entire database for each of the 10 digits. The average recognition accuracy using this new template set was 92.6 percent. This was only 0.5 percent worse than the recognition accuracy of the PO clean template set, but with 100 fewer templates (i.e., about two-thirds of the computation).

### *3.3 Imposing a rejection threshold*

Figures 14 through 16 demonstrate the effects of imposing a threshold on the recognition task. A recognition distance score above the

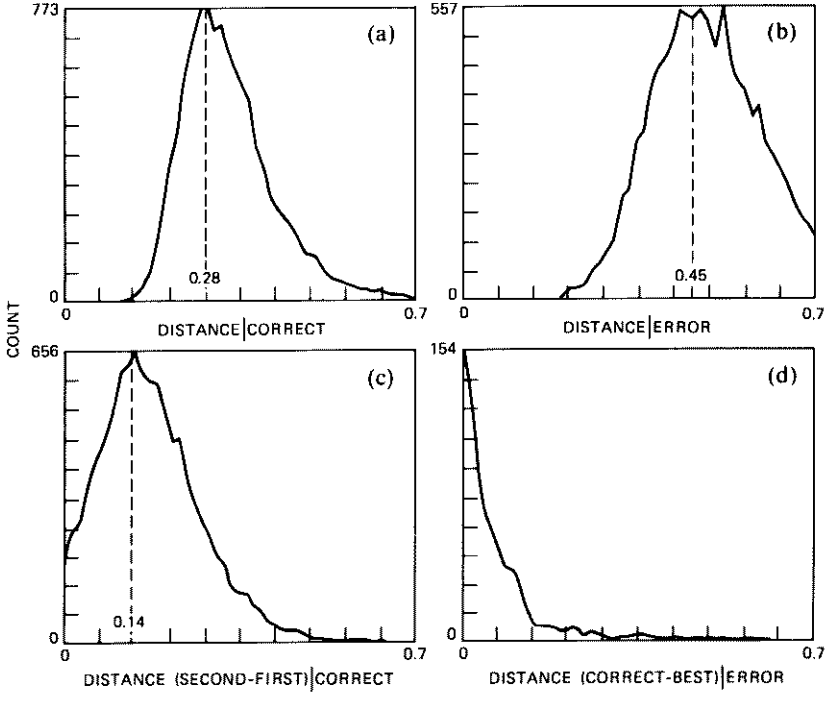



Fig. 14—Plot of histograms of LPC distances for (a) the correct word; (b) the closest incorrect word; (c) the difference between the best and second best choice, given the best choice is correct; (d) the difference between the correct word and the top recognition candidate, given the top candidate is not the correct word.

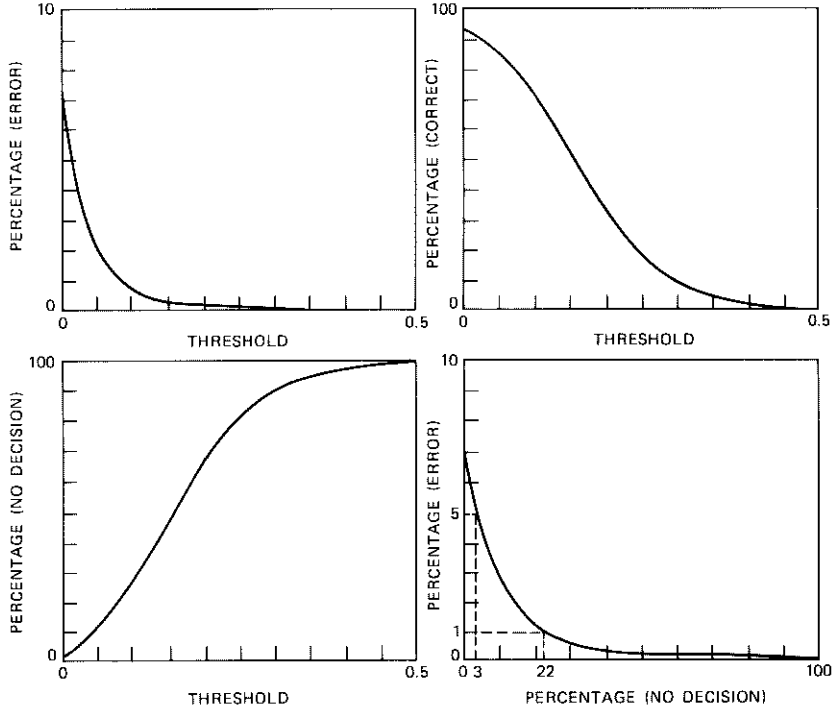



Fig. 15—Plot showing trade-off between no-decision rate and error rate on a per-digit basis, given a thresholding scheme imposed on the recognizer.

threshold would lead to a result of no decision by the recognizer. Figure 14a shows a histogram of LPC distances for the correct word (i.e., all 11,035 digits), with a mean correct distance of 0.28. Figure 14b shows a histogram of the scores for the closest incorrect word, with a mean distance of 0.45. Figure 14c shows a histogram of scores of the difference between the best choice and the second best choice, given the best choice is correct. Its mean of 0.14 indicates that when a word is recognized correctly the next closest word will on average have a distance score about 50 percent greater. Since LPC distances are on a log scale, this difference is a relatively large one. Figure 14d shows a histogram of the difference between the correct word and the top recognition candidate given the top candidate is not the correct word. This plot shows that when a word is misrecognized, the correct word has an LPC distance very close to that of the best choice. These results imply that a thresholding scheme can be applied to the recognition system and yield a good trade-off between accuracy and no-decision choices. Figure 15 shows the results of implementing such a thresholding technique. This plot shows the percent no decisions versus percent error rate, for the PO clean template set. We see that

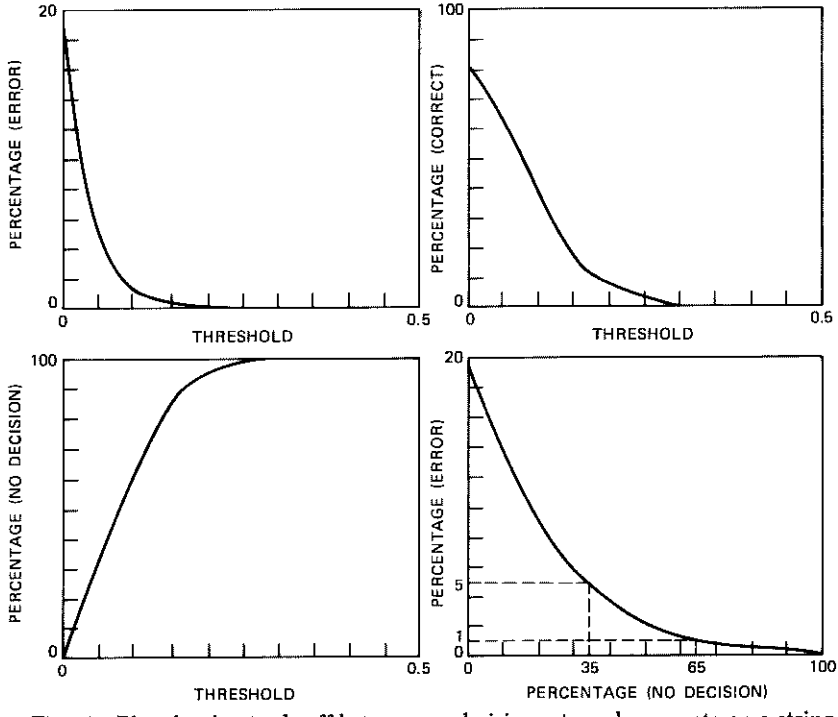



Fig. 16—Plot showing trade-off between no-decision rate and error rate on a string basis, given a thresholding scheme imposed on the recognizer.

if the task for which this recognition system is to be used can only tolerate a 1-percent per-digit error rate, a no-decision rate of 22 percent must also be accepted. However, a 5-percent probability of error yielded only 3 percent no decisions. Figure 16 shows a similar set of trade-off curves for full digit strings. For a digit string with an average of 3.5 digits, a 1-percent string error rate leads to a 65-percent no-decision rate and a 5-percent string error rate leads to a 35-percent no-decison rate. These results suggest that even though thresholding can be used to reduce error rate, other methods that do not lend themselves to such high no-decision rates should be investigated.

## IV. DISCUSSION

The results presented in Sections III show that:

1. Significant problems exist in getting casual telephone customers to speak telephone numbers as a series of isolated digits. These are related to human factors issues (people don't normally speak in isolated word sequences) and problems induced by a wide variety of transmission and switching conditions.

2. The ability to detect words automatically in noisy or nonideal

environments is not adequate for about 50 percent of the recordings. There are some obvious ways of improving the current algorithm for finding words; and the database we collected in this test is currently being used to test various modifications of the algorithm.

3. The recognition results given in Table IV show that the accuracy of the recognizer, even when trained on a subset of the test data, is at best marginally acceptable with a maximum accuracy of 93.1 percent. However, some digits had recognition accuracies well over 93.1 percent, while others (i.e., the digit 9) had scores around 85 percent. More detailed analyses of the type of errors and their causes must be undertaken and some improvements in the training and recognition procedures must be made to approach the accuracies obtained for laboratory recordings (close to 98 percent).

Each of the above problems will be investigated carefully to improve all aspects of automatic recording, detection, and recognition of isolated words.

Another data collection exercise will begin soon at another site. Such issues as the effects of regional dialect on the reference templates will be studied. In addition, further testing of an improved endpoint detector will be performed.

## V. SUMMARY

In this paper we have presented recognition results obtained from a speech database, consisting of 11,035 isolated digits, collected in an actual telephone environment from 3100 nonsolicited subjects. Several different reference template sets were used with a maximum recognition accuracy reported at 93.1 percent.

## VI. ACKNOWLEDGMENTS

The authors would like to acknowledge the many individuals involved in collecting the data described in this paper. In particular, Eddie Youngs, Cindy Karhan, Marty Viets and Helen Hollinka provided the interface to the Bell System and were responsible for direct recording of the spoken digit sequences. Frank Pirz and Keith Bauer designed, built, and programmed the special-purpose recording hardware. Finally, Don Bock and Sandy MacNeil aided in the collection, processing, and analysis of the customer data at the Murray Hill laboratory. To each of these individuals we express our thanks for their help.

## REFERENCES

1. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-23*, No. 1 (February 1975), pp. 67–72.
2. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker

Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 4 (August 1979), pp. 336–49.
3. L. R. Rabiner and J. G. Wilpon, "Speaker Independent, Isolated Word Recognition for a Moderate Size (54 word) Vocabulary," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27*, No. 6 (December 1979), pp. 583–7.
4. J. G. Wilpon, L. R. Rabiner, and A. F. Bergh, "Speaker Independent Isolated Word Recognition Using a 129-Word Airline Vocabulary," J. Acoust. Soc. Amer., *72*, No. 2 (August 1982), pp. 390–6.
5. L. R. Rabiner and J. G. Wilpon, "A Simplified, Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," J. Acoust. Soc. Amer., *68*, No. 5 (October 1980), pp. 1069–70.
6. L. R. Rabiner and J. G. Wilpon, "Considerations in Applying Clustering Techniques to Speaker Independent Word Recognition," J. Acoust. Soc. Amer., *66*, No. 3 (September 1979), pp. 663–73.
7. L. R. Rabiner, A. E. Rosenberg, J. G. Wilpon, and W. J. Keilin, "Isolated Word Recognition for Large Vocabularies," B.S.T.J., *61*, No. 10, Part 1, (December 1982), pp. 2989–3005.
8. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Recognition—Theory and Selected Applications," IEEE Trans. Commun., *29*, No. 5 (May 1981), pp. 621–59.
9. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-29*, (August 1981), pp. 777–85.

## AUTHORS

**Jay G. Wilpon,** B.S., A.B. (cum laude) in Mathematics and Economics, respectively, 1977, Lafayette College, Easton, Pa; M.S. (Electrical Engineering/Computer Science), 1982, Stevens Institute of Technology, Hoboken, N.J.; Bell Laboratories, 1977—. Since June 1977 Mr. Wilpon has been with the Acoustics Research Department at Bell Laboratories, Murray Hill, N.J., where he is a Member of the Technical Staff. He has been engaged in speech communications research and is presently concentrating on problems of speech recognition.

**Lawrence R. Rabiner,** S.B. and S.M., 1964, Ph.D., 1967 (Electrical Engineering), The Massachusetts Institute of Technology; Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication of the Acoustical Society, ASSP Technical Committee on Speech Communication; Member, IEEE Proceedings Editorial Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# Comparing Batch Delays and Customer Delays

By W. WHITT*

(Manuscript received January 25, 1983)

For a large class of queueing systems in which customers arrive in batches, Halfin (1983) showed that the delay distribution of the last customer in a batch to enter service coincides with the delay distribution of an arbitrary customer when the batch-size distribution is geometric. Halfin's result can be applied to study the performance of complicated communication systems in which messages are divided into packets for transmission. Then packets are customers and the delay of a message is the delay of the last customer in a batch to enter service. If the assumptions are satisfied and if packet delays are easier to analyze, then packet delays can be used to calculate message delays. In this paper, we show that these two delay distributions are stochastically ordered when the batch-size distribution is NBUE or NWUE (new better or worse than used in expectation). The delays of arbitrary customers tend to be less (more) when the batch-size distribution is NBUE (NWUE). In addition to the bounds provided by the stochastic ordering, we also suggest an approximation for the relation between the two expected delays based on known results for the $M^B/G/1$ queue having a batch-Poisson arrival process.

## I. INTRODUCTION

Halfin[1] recently showed that for a large class of queueing systems in which customers arrive in batches the delay distribution of the last customer in a batch to enter service equals the delay distribution of an arbitrary customer when the batch size has a geometric distribution. Halfin's result is important when we study the performance of communication systems in which messages are divided into packets for

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

transmission. Then packets are customers and the delay of a message is the delay of the last customer in a batch to enter service. Halfin's result is useful to approximately describe message delays in complicated communication systems, e.g., with contention-resolving schemes such as Carrier Sense Multiple Access (CSMA).[1] In many of these systems message delay is more important but more difficult to analyze than packet delay. Halfin provides conditions under which packet delay results directly yield message delay results.

In this paper we identify conditions on the batch-size distribution under which the delay distribution of the last customer to enter service is stochastically greater than or equal to the delay distribution of an arbitrary customer. We show that it suffices for the batch-size distribution to be NBUE (new better than used in expectation). The ordering is reversed for batch-size distributions that are NWUE (new worse than used in expectation). The batch size $B$ has an NBUE distribution if its mean is greater than all conditional means given tail events, i.e., if $EB \geq E(B - n \mid B \geq n)$ for all $n$. A distribution is NBUE (NWUE) if it has increasing (decreasing) failure rate. The NBUE and NWUE properties are now quite standard for stochastic comparisons. For further discussion, see Barlow and Proschan,[2] and Whitt.[3]

As Halfin observed, his result about batch arrivals can be viewed as a special case of a discrete analog of Poisson Arrivals See Time Averages (PASTA).[4] In the same way, our stochastic comparison results parallel those for customer-stationary and time-stationary characteristics of queues.[5-7] Random quantities associated with batches (e.g., delays of the last customer in a batch) constitute an embedded sequence in the sequence of random quantities associated with all customers (e.g., the delays of arbitrary customers), just as customer arrival points or departure points constitute embedded sequences in continuous time.

Our approach is similar to that of the East German School (Franken et al.) because we begin in Section II with a stationary version, what we call an "equilibrium batch." In this setting, we easily obtain a representation of the expected average associated with an arbitrary customer that makes the desired comparisons and Halfin's result transparent. The representation involves the stationary-excess distribution of the batch-size distribution. (From the theory of stationary point processes, as contained in Franken et al., this can also be viewed as a consequence of the Palm theory.)

The relation between what an arbitrary customer sees and what the last customer in a batch sees can be explained in part as follows. One customer in each batch is the last customer in the batch, but there are more customers in big batches than in small batches. Hence, the distribution of the batch size containing a last customer is the ordinary

batch-size distribution, but the distribution of the position in a batch of an arbitrary customer is the batch-size stationary-excess distribution (see Section III.2 of Cohen,[8] Burke,[9] and Section 5.10 of Cooper[10]). Hence, the relation between the two kinds of delays reduces to the relation between the batch-size distribution and its associated stationary-excess distribution.

These stochastic comparison results are only qualitative. With the methods we use, we are unable to obtain related quantitative results. Moreover, the difference between the expected delays obviously will depend on the context, whereas the qualitative results here generally hold true. Following Halfin's example, we make very few assumptions for our stochastic comparisons. However, in Section III we examine quantitative results by considering the special case of the $M^B/G/1$ queueing model, which has a batch-Poisson arrival process. We get some idea about what to expect in more complicated situations by examining existing quantitative results for this special case. We use these results to obtain an approximate quantitative relationship between expected batch delays and expected customer delays as a function of the first two moments of the batch-size distribution.

In Section IV we briefly show how our equilibrium batch can be viewed as a time-average limit. Throughout we talk about batch arrivals to a queue, but as in Wolff[4] the results apply more generally. The model need not be a queue, and for a queueing model the process need not be an arrival process; for an arrival process the customers in a batch need not arrive together. The batch is simply a label assigned to random variables, as we explain in Section II.

## II. THE EQUILIBRIUM BATCH

We consider a batch in equilibrium, defined in terms of a positive-integer-valued random variable $B$ and a sequence of random variables $\{X_k, k \geq 1\}$, all defined on a common probability space. The variable $B$ represents the size of the batch and the variable $X_j$ is a random quantity associated with the $j$th customer in the batch. For example, $X_j$ might be a function of the delay of the $j$th customer to enter service. We are only interested in $X_j$ for $j \leq B$. In fact, we can consider $X_j$ defined only on the subset $\{B \geq j\}$. Hence, we can speak of the basic data as the random vector $(B, X_1, \cdots, X_B)$.

The variables $X_1, X_2, \cdots$ are typically dependent. Our *basic assumption* is that the events $\{B = j\}$ for $j \geq k$ are independent of the random vector $(X_1, \cdots, X_k)$ for $k \geq 1$. This corresponds to the Lack of Anticipation Assumption (LAA) in Wolff.[4]

Let the probability mass function (pmf) of $B$ be defined as

$$p_n = P(B = n), \qquad n = 1, 2, \cdots, \tag{1}$$

and let $p_n^*$ be the *stationary-excess* pmf associated with $p_n$, defined by

$$p_n^* = \bar{p}_n \Big/ \sum_{k=1}^{\infty} \bar{p}_k, \qquad n \geq 1, \tag{2}$$

where

$$\bar{p}_n = \sum_{k=n}^{\infty} p_k, \qquad n \geq 1, \tag{3}$$

and

$$\sum_{k=1}^{\infty} \bar{p}_k = \sum_{k=1}^{\infty} k p_k = EB < \infty. \tag{4}$$

We are interested in the relationship between the random quantities associated with an *arbitrary* customer and the *last* customer in a batch so we can compare the expected values in these two cases. We interpret the expected value associated with an arbitrary customer as the expected value for all customers in the batch divided by the expected number of customers in a batch. (A way to justify this interpretation is described in Section IV). Because of our basic assumptions, these quantities are

$$A = \frac{\sum_{n=1}^{\infty} p_n(EX_1 + \cdots + EX_n)}{\sum_{n=1}^{\infty} np_n} \tag{5}$$

and

$$L = EX_B = \sum_{n=1}^{\infty} p_n EX_n. \tag{6}$$

The desired relationships between $A$ and $L$ are derived from the following alternate representation for $A$.

*Theorem 1*: $A = \sum_{n=1}^{\infty} p_n^* EX_n$.

*Proof*: Change the order of summation in (5) and apply (2) to obtain

$$A = \sum_{n=1}^{\infty} EX_n \left( \frac{\sum_{k=n}^{\infty} p_k}{\sum_{k=1}^{\infty} k p_k} \right) = \sum_{n=1}^{\infty} EX_n \left( \frac{\bar{p}_n}{\sum_{k=1}^{\infty} \bar{p}_k} \right) = \sum_{n=1}^{\infty} EX_n p_n^*.$$

*Corollary 1*: (*Halfin*[1]) *$A = L$ for all $\{EX_n\}$ if and only if $p_n = p_n^*$ for all $n$ or, equivalently, if $p_n$ is geometric, i.e.,*

$$p_n = (1 - p)p^{n-1}, \qquad n = 1, 2, \cdots \tag{7}$$

*for some $p$, $0 \leq p \leq 1$.*

*Proof*: Sufficiency is immediate. It is well known that $p_n$ is geometric if and only if $p_n = p_n^*$ for all $n$ (e.g., see Corollary 3.3 of Whitt[3]). Necessity is almost as easy: Choose different sequences $\{EX_n\}$, e.g., $EX_n = 1$ if $n = k$, and 0 otherwise.

*Remark*: Theorem 1 and its corollaries apply immediately to distributions as well as means. For example, our original sequence $X_n$ can be replaced by $f(X_n)$ where $f(x) = I_{(-\infty,x)}$, so that $A$ is the expected proportion of customers for which $X_n \le x$ and $L = P(X_B \le x)$.

We now establish inequalities between $A$ and $L$ as a function of the shape of the batch-size pmf $p_n$. These follow immediately from known stochastic-order relations between $p_n$ and $p_n^*$.

For two pmf's $p_n^1$ and $p_n^2$ on the positive integers, we define stochastic-order relations $p_n^1 \le_{st} p_n^2$ ($p_n^1 \le_{ic} p_n^2$) to hold if

$$\sum_{k=1}^{\infty} f(k)p_k^1 \le \sum_{k=1}^{\infty} f(k)p_k^2 \tag{8}$$

for all nondecreasing (nondecreasing and convex) real-valued functions $f$ on the positive integers for which the sums converge.

A pmf $p_n$ is NBUE (new better than used in expectation) if

$$EB = \sum_{k=1}^{\infty} \bar{p}_k \ge \sum_{k=n}^{\infty} \bar{p}_k/\bar{p}_n = E(B - n \mid B \ge n), \qquad n \ge 1, \tag{9}$$

and NWUE with the inequality in (9) reversed.

From Theorem 1, we obtain:

*Corollary 2: $A \le L$ for all nondecreasing sequences $\{EX_n\}$ if and only if $p_n^* \le_{st} p_n$ or, equivalently, if $p_n$ is NBUE.*

*Proof*: From (6), (8), and Theorem 1, $A \le L$ for all nondecreasing $\{EX_n\}$ if and only if $p_n^* \le_{st} p_n$. It is well known that $p_n^* \le_{st} p_n$ if and only if $p_n$ is NBUE [e.g., see Theorem 3.2 (iii) of Whitt[3]]. For necessity, choose $EX_n = 1$ for $n \ge k$, and 0 otherwise.

*Remarks*: (1) Corollary 2 remains valid with $p_n^* \ge_{st} p_n$ instead of $\le_{st}$, which is equivalent to $p$ being NWUE, if either $A \ge L$ or $\{EX_n\}$ is nonincreasing (but not both). (2) For Halfin's problem involving delays, note that the delays of the successive customers in any batch to begin service are nondecreasing with probability one.

We obtain another corollary using the stochastic-order relation $\le_{ic}$ defined in (8).

*Corollary 3: $A \le L$ for all nondecreasing convex sequences $\{EX_n\}$ if and only if $p_n^* \le_{ic} p_n$ or, equivalently, if $p_n$ new is better than $p_n^*$ used in expectation, i.e., if*

$$\sum_{k=1}^{\infty} \bar{p}_k \ge \sum_{k=n}^{\infty} \bar{p}_k^*/\bar{p}_n^*, \qquad n \ge 1.$$

*Proof*: Follow the proof of Corollary 2 using the convexity to treat $\leq_{ic}$. The characterization of $p_n^* \leq_{ic} p_n$ follows easily from the fact that $p_n^1 \leq_{ic} p_n^2$ if and only if $\sum_{k=n}^{\infty} \bar{p}_k^1 \leq \sum_{k=n}^{\infty} \bar{p}_k^2$; see Definition 3.1(iv) and Theorem 3.2(iv) of Whitt.[3]

## III. The $M^B/G/1$ QUEUE

To illustrate the qualitative results and obtain some related quantitative results, we now consider the special case of the $M^B/G/1$ queue having a batch-Poisson arrival process (see Section 5.10 of Cooper[10]). Let $W_F$, $W_A$, and $W_L$ be the equilibrium delay (waiting time before beginning service) of the first customer in a batch, an arbitrary customer, and the last customer in a batch. Let $\tau$ and $\sigma^2$ be the mean and variance of the service time; let $m$ and $\hat{\sigma}^2$ be the mean and variance of the batch size $B$; and let $B^*$ have the batch-size stationary-excess distribution $p_n^*$ in (2), which has mean

$$EB^* = \sum_{n=1}^{\infty} np_n^* = (\hat{\sigma}^2 + m^2 + m)/2m. \tag{10}$$

Let $\lambda$ be the rate of the Poisson process and, for stability, assume that $\lambda m\tau < 1$.

From Cooper, we obtain

$$E(W_F) = \frac{\lambda m^2\tau^2}{2(1 - \lambda m\tau)}\left(1 + \frac{m\sigma^2 + \tau^2\hat{\sigma}^2}{m^2\tau^2}\right). \tag{11}$$

From Theorem 1 and (10), we obtain

$$\begin{aligned} E(W_A) &= E(W_F) + \sum_{n=1}^{\infty} p_n^*(n-1)\tau \\ &= E(W_F) + [E(B^*) - 1]\tau \\ &= E(W_F) + \frac{(m-1)\tau}{2} + \frac{\hat{\sigma}^2\tau}{2m}. \end{aligned} \tag{12}$$

From (6), we obtain

$$\begin{aligned} E(W_L) &= E(W_F) + (EB - 1)\tau \\ &= E(W_F) + (m-1)\tau. \end{aligned} \tag{13}$$

Let $c_B^2$ be the squared coefficient of variation of the batch-size $B$, i.e., the variance of $B$ divided by the square of the mean. We can rewrite (12) as

$$E(W_A) = E(W_F) + [m(c_B^2 + 1) - 1]\tau/2, \tag{14}$$

so that $EW_A \leq EW_L$ if and only if

$$mc_B^2 + 1 \leq m, \tag{15}$$

which is consistent with Corollaries 1 through 3 in Section II because $m = 1/p$ and $c_B^2 = 1 - p$ for the geometric distribution in (7).

In applying Theorem 1, we should observe that the lack of anticipation (LAA) assumption holds for the $M^B/G/1$ queue. Moreover, the Poisson property is only used to get (11); (12) and (13) remain valid provided that the LAA assumption holds. For example, the LAA assumption holds if the intervals between batch arrivals are a stationary sequence independent of the successive batch sizes.

Formulas (11) through (14) suggest an approximation for more general systems:

$$\frac{E(W_A) - E(W_F)}{E(W_L) - E(W_F)} = \frac{m(c_B^2 + 1) - 1}{2(m - 1)}. \tag{16}$$

## IV. LIMITING AVERAGES

Instead of using the framework defined in Section II, we could also begin with a sequence of random vectors $\{(B_k, X_{k1}, \cdots, X_{kB_k}), k \geq 1\}$, where $B_k$ represents the size of the $k$th batch, indexed in order of arrival, and $X_{kj}$ represents the random quantity of interest (delay, etc.) associated with the $j$th customer to enter service in the $k$th batch. Let $B_k$ take values on the positive integers and let $X_{kj}$ be nonnegative for each $k$ and $j$. Let $Y_k$ be the random quantity associated with the $k$th customer indexed, first according to the batch and then according to the order of entering service, defined by

$$Y_k = X_{jm}, \qquad B_1 + \cdots + B_{j-1} + m = k \leq B_1 + \cdots + B_j$$

for $k \geq 1$.

The limiting average value of $X_{kj}$ over *all* customers is naturally defined by

$$\bar{A} = \lim_{n\to\infty} \frac{\sum_{k=1}^{n} Y_k}{n}. \tag{17}$$

We shall work with the related quantity $\hat{A}$ defined by

$$\hat{A} = \lim_{n\to\infty} \frac{\sum_{k=1}^{n} \sum_{j=1}^{B_k} X_{kj}}{\sum_{k=1}^{n} B_k}. \tag{18}$$

In most situations, the limits $\bar{A}$ and $A$ exist and are equal.

We are interested in the relation between $A$ and the limiting average value of $X_{kj}$ over the *last* customer in each batch, defined by

$$\hat{L} = \lim_{n\to\infty} \frac{\sum_{k=1}^{n} X_{kB_k}}{n}. \tag{19}$$

We assume that there exists a random vector $(B, X_1, \cdots, X_B)$ such that

$$\lim_{n\to\infty} \frac{\sum_{k=1}^{n} B_k}{n} = EB < \infty \tag{20}$$

and

$$\lim_{n\to\infty} \frac{\sum_{k=1}^{n} \sum_{j=1}^{B_k} X_{kj}}{n} = E \sum_{j=1}^{B} X_j < \infty, \tag{21}$$

so that

$$\hat{A} = \frac{E \sum_{j=1}^{B} X_j}{EB}. \tag{22}$$

We also assume that

$$\lim_{n\to\infty} \frac{\sum_{k=1}^{n} X_{kj} I_{\{B_k = j\}}}{n} = P(B = j)EX_j. \tag{23}$$

To obtain (23) it is natural to assume that the basic independence assumption holds for each $k$, i.e., $\{B_k = j\}$ is independent of $(X_{k1}, \cdots, X_{kn})$ for all $j$, $j \geq n$. From (23), we have

$$\hat{L} = EX_B. \tag{24}$$

For eqs. (20) through (24) to be valid, in addition to the basic independence assumption, it suffices for the basic sequence $\{(B_k, X_{k1}, \cdots, X_{kB_k}), k \geq 1\}$ to be stationary and ergodic.

With (22) and (24) we obtain the framework defined in Section II.

## V. ACKNOWLEDGMENTS

I am grateful to Shlomo Halfin for introducing me to this problem and to Moshe Segal for making helpful comments.

## REFERENCES

1. S. Halfin, "Batch Delays Versus Customer Delays," B.S.T.J., this issue.
2. R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*, New York: Holt, Rinehart and Winston, 1975.

3. W. Whitt, unpublished work.
4. R. W. Wolff, "Poisson Arrivals See Time Averages," Oper. Res., *30*, No. 2 (March–April 1982), pp. 223–31.
5. D. König and V. Schmidt, "Stochastic Inequalities Between Customer-Stationary and Time-Stationary Characteristics of Queueing Systems With Point Processes," J. Appl. Prob., *17*, No. 3 (September 1980), pp. 768–77.
6. P. Franken, D. König, U. Arndt, and V. Schmidt, *Queues and Point Processes*, Berlin: Akademie-Verlag, 1981.
7. S. Niu, unpublished work.
8. J. W. Cohen, *The Single Server Queue*, Amsterdam: North Holland, 1969.
9. P. J. Burke, "Delays in Single-Server Queues With Batch Input," Oper. Res., *23*, No. 4 (July–August 1975), pp. 830–3.
10. R. B. Cooper, *Introduction to Queueing Theory*, Second Edition, New York: North Holland, 1981.

## AUTHOR

**Ward Whitt**, A.B. (Mathematics), 1964, Dartmouth College; Ph.D. (Operations Research), 1968, Cornell University; Stanford University, 1968–1969; Yale University, 1969–1977; Bell Laboratories, 1977—. At Yale University, from 1973 through 1977, Mr. Whitt was Associate Professor in the departments of Administrative Sciences and Statistics. At Bell Laboratories he is in the Operations Research Department. His work focuses on stochastic processes and stochastic models in operations research.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# Batch Delays Versus Customer Delays

By S. HALFIN*

(Manuscript received January 6, 1982)

For a large class of contention schemes with messages transmitted by subdividing them into packets, we show that the delay distribution of a message is the same as that for an individual packet. We conclude this from analyzing queueing systems with batch arrivals, where batch sizes have a geometric distribution and the queue discipline is indifferent to batch sizes and service times. There we prove that the customer (packet) delay distribution is the same as the batch (message) delay distribution, where delay is defined to be the delay of the *last served* customer in the batch. The proof is based on the discrete-time analog of the Poisson Arrivals See Time Averages (PASTA) theorem. We conclude that, in many cases, we can obtain message delays by calculating or measuring the packet delays, which is usually an easier task.

## I. INTRODUCTION

Consider a queueing system with batch arrivals. We define the delay of batch to be the maximal delay of the customers in the batch. For example, consider the case where data messages arrive at a node of a network and await transmission. Each message is subdivided into packets that may be transmitted individually. Here the customers correspond to the packets and the batch to a message, and it is natural to say that the message is delayed as long as at least one of its packets is delayed.

Next, assume that the number of customers in a batch has a geometric distribution:

$$\Pr(\text{batch size} = n) = p(1-p)^{n-1}, \qquad n = 1, 2, \cdots.$$

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

In our example, if the message lengths (in bits) are exponentially distributed, and each message $m$ is "chopped" into $i(m)$ packets whose lengths (in bits) are independent and identically distributed and independent of the message length, then $i(m)$ has a geometric distribution. [Note that the $i(m)$th packet usually will not be full.]

We prove that for a wide class of such systems the batch delay in equilibrium has the same distribution as the individual customer delay. Here delay is the time from arrival to start of service. In Section II we discuss the various queue disciplines for which this result holds. A discete-time analog of the Poisson Arrivals See Time Averages (PASTA) theorem is introduced in Section III. The main result is stated and proved in Section IV. Section V contains some additional comments, and conclusions are stated in Section VI.

## II. QUEUE DISCIPLINES

The stated result does not hold for all queue disciplines. Consider the case where customers are selected for service independent of their service times. Then it is well known that the expected delay of a customer is independent of the queue discipline.[1] On the other hand, the expected batch delay may depend on the discipline. For instance, if one always chooses the next customer from a batch with the smallest number of remaining customers, the expected batch delay will be smaller than if the next batch is chosen randomly, or in a preassigned order. This can be proved by arguments similar to those showing that giving preferential treatment to customers with small service time reduces the expected waiting time.[2] Thus, to prove equality of the delay distributions, such disciplines must be excluded.

Next, we characterize disciplines for which the result holds:

*Definition:* A queueing discipline will be called *impartial* if it selects customers independently of their services times, and selects batches independently of their sizes.

The following are examples of impartial disciplines:

- First in, first out (FIFO) for batches and for customers within batches.
- Last in, first out (LIFO) for batches and for customers within batches.
- Random choice of a batch, and then a random choice of a customer from that batch (but not a random choice among all waiting customers because this will favor large batches).
- Random choice of a batch, and then serving all the customers of that batch in FIFO, LIFO, or random order.
- Contention schemes: each batch chooses a candidate customer for next service (independent of its service time) and the contention

between the candidates is resolved independent of the sizes of their respective parent batches.

The last family includes disciplines that are applicable to the case of transmitting messages by subdividing them into packets. Here the candidate packet for transmission in each message is the first packet not yet transmitted. Examples of relevant contention-resolving methods are (1) the round robin (token) scheme, and (2) Carrier Sense Multiple Access (CSMA) scheme.[3-5]

## III. DISCRETE-TIME ANALOG OF PASTA

The following result will be needed later. Let $X_n$, $n = 1, 2, \cdots$ be a sequence of random variables, and let $B$ be a set in the value space of $X$. Let $U_n$ be the indicator function of the event $X_n \in B$, for all $n$. Let $A_n$, $n = 1, 2, \cdots$ be a sequence of i.i.d. Bernoulli random variables defined on the same probability space as the $X_n$'s.

$$\text{Let} \quad V_n = n^{-1} \sum_{i=1}^{n} U_i, \quad Y_n = \sum_{i=1}^{n} U_i A_i, \quad \text{and} \quad Z_n = Y_n \left( \sum_{i=1}^{n} A_n \right)^{-1}.$$

Then the following hold:

*Theorem 1: If for every n the set of random variables $X_1, X_2, \cdots, X_n$ is independent of the set $A_n, A_{n+1}, \cdots$ then: $V_n \to V$ w.p. 1 if and only if $Z_n \to V$ w.p. 1, as $n \to \infty$.*

*Remark:* This theorem is a discrete-time analog of the PASTA theorem,[6] which is stated in terms of a continuous time stochastic process $X(t)$, and a Poisson process $A(t)$. The assumption in the theorem is called by Wolff 'Lack of Anticipation Assumption' (LAA). The name PASTA comes from the special case where $A(t)$ is an arrival process to a queue, and $X(t)$ is the number of customers in the system at time $t$. In that case the theorem states that the long-term proportion of time for which $X(t) = m$ (arbitrary nonnegative integer) is equal to the long-term proportion of arriving customers that find the system in state $m$.

*Proof:* Wolff's Lemma 1, Lemma 2, and Theorem 1[6] can be easily stated and verified in the discrete-time case, and thus the conclusion holds.

## IV. THE RESULT AND ITS PROOF

*Theorem 2: Given a queueing system with batch arrivals where:*

1. *The batch sizes are i.i.d. with a geometric distribution, and independent of the arrival times process*
2. *The service times are independent of the arrival times and batch sizes*

3. *The queue discipline is impartial, then the customer delay distribution is equal to the batch delay distribution.*

*Remark:* We interpret the conclusion of the theorem in a time average sense. Thus, for any $x \geq 0$ we compare the proportions of customers and batches that are delayed $x$ time units or less. We prove that if one of these proportions has a long-term limit, so does the other, and they are equal. In the common case, when the queueing system is ergodic, the conclusion can be also interpreted in terms of delay distributions of individual customers and batches.

*Proof:* Let $X_1, X_2, \cdots$ be the sequence of delays of customers arranged in the order in which they go into service. Let $x \geq 0$ be fixed, and let $B = [0, x]$. Let $A_n = 1$, if the $n$th customer in the above order is the last to go to service in its batch, and $A_n = 0$ otherwise. Next, observe that the LAA assumption holds for this setup because for any $n$, $X_1, \cdots, X_n$ are determined by the arrival process, service times of the first $n - 1$ served customers, and queue discipline, all of which are independent of the batch size of the $n$th customer. (Note that if the discipline is not blind to service time, the above service times could depend on the batch sizes. Also note that the delays do depend in general on the *number* of present batches, and thus $A_n$ and $X_{n+1}$ would typically be dependent. For instance, if $A_n = 1$, the probability that no customer remains waiting, implying $X_{n+1} = 0$, increases.) The proof is now completed by applying Theorem 1.

## V. FURTHER COMMENTS

The assumptions of Theorem 2 are quite weak. The theorem is true for a queue with many servers, even with different service rates, as long as the assignment of customers to servers is again independent of the present batch sizes.

We do not require that a service begins immediately when a server becomes free, even if customers are waiting. There may be a "dead period", as is the case in some contention schemes. However, the length of this dead period has to be independent of the present batch sizes.

If we want to derive a similar theorem for *time in system* (rather than delay), then it will not hold in general in the multiserver case. For example, it will not be true for an infinite number of servers, and nonconstant service time. (Ward Whitt[7] provided this example.)

Although the last customer to enter service in a given batch has the same distribution of time in system as a typical customer, its time in system may be different from the batch's time in system. This is so because, in the multiserver case, when that customer completes service, other customers of its batch may still be in service. If we apply

Theorem 2 to the setup where we order the customers by the time they leave the system and let the $X_n$'s be the corresponding times in system, the application will fail because the LAA assumption will no longer be valid.

## VI. CONCLUSION

We have shown that for many queueing systems with batch arrivals having geometrically distributed sizes, the individual customers experience the same delay distribution as the batches themselves. This seems to affect the case of transmitting messages by packets, since in many cases it is possible to obtain packet delays, while from a performance point of view one is more interested in the message delays.


## VII. ACKNOWLEDGMENT

I would like to thank Ward Whitt for his useful comments and suggestions.


## REFERENCES


1. L. Kleinrock, *Queueing Systems*, Vol. 2, New York: Wiley, 1976, p. 113.
2. R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, New York: Addison-Wesley, 1967.
3. L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I—Carrier Sense Multiple Access Modes and Their Throughput—Delay Characteristics," IEEE Trans. Commun., *23* (1975), pp. 1400–16.
4. F. A. Tobagi and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," Proc. LACN Symp., May 1979, pp. 217–44.
5. D. P. Heyman, "An Analysis of the Carrier Sense Multiple Access Protocol," B.S.T.J., *61*, No. 8 (October 1982), pp. 2023–51.
6. R. W. Wolff, "Poisson Arrivals See Time Averages," Op. Res., *30*, No. 2 (1982), pp. 223–31.
7. W. Whitt, private communication.


## AUTHOR


**Shlomo Halfin,** M.Sc., 1958, and Ph.D., 1962 (Mathematics), The Hebrew University of Jerusalem; Bell Laboratories, 1968—. Mr. Halfin's areas of interest are stochastic processes and mathematical programming. He currently works on applications of these methods to communication networks. Member, SIAM, ORSA.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# Combined Source and Channel Coding for Variable-Bit-Rate Speech Transmission

By D. J. GOODMAN* and C.-E. SUNDBERG†

(Manuscript received January 19, 1983)

Motivated by potential applications to mobile radio, we studied variable-bit-rate speech communication through Gaussian-noise and Rayleigh-fading channels. For convenience we used a constant signaling rate of 32 kb/s and adjusted the source-coding and channel-coding rates in response to changing transmission quality. When the channel quality was good enough, we used all 32 kb/s for speech transmission. When the channel quality was lower, we reduced the source rate to 24 or 16 kb/s and introduced channel coding to control distortion due to transmission errors. We concentrated on specific source and channel codes that could be implemented with hardware of modest complexity. The source code was embedded differential pulse code modulation, which is amenable to variable-bit-rate operation and economical to implement. For error control we introduced punctured convolutional codes and a Viterbi decoder with only 16 states. Although the source/channel codec was simple, it offered good performance. Speech quality was at the level of normal telephony when the channel was good; the error-correcting codes extended by up to 13 dB the range of channel signal-to-noise ratios that support adequate quality. Our performance estimates were based on a new analysis of transmission errors in embedded differential pulse code modulation and on computer simulations of speech transmission through fixed and fading channels.

## I. INTRODUCTION

### *1.1 Motivation*

Every communication channel has an optimum rate for digital transmission of speech. If the actual rate is below the optimum, speech

* Bell Laboratories. † University of Lund, Sweden.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

quality is impaired by unnecessary quantizing distortion. If the rate is higher than optimum, the distortion due to transmission errors is excessive. These observations are embodied in rate-distortion theory, which provides theoretical bounds on performance. Implicitly, they influence practical systems in which design decisions are compromises between goals of quality, bandwidth efficiency, and equipment economy.

When the properties of the communication channel are predictable and invariant, one may explore this compromise and arrive at an acceptable design. However, in many instances (for example, in switched telephony) prior knowledge of the channel is only statistical, and in others (fading microwaves) the channel changes drastically with time. With these channels, performance is characterized statistically by the fraction of users experiencing each possible level of quality or by the fraction of time a single user experiences each quality level.

Both spatial and temporal channel fluctuations are inherent in mobile telephony. Because channel characteristics depend on the position of the mobile unit, all users have, at any time, different channels and, because mobile units are moving, their channels change with time. The conventional approach to this statistical nature of the channel is to set design goals of acceptable quality for a large fraction (say 90 percent) of users or to set "outage" limits, i.e., small fractions of time (say 10 percent) when quality is allowed to fall below a certain specification. In meeting these statistical requirements with a fixed coding and modulation scheme, we have a system in which performance is worse than necessary for most users most of the time. The common deficiency is too much quantizing noise, because 90 percent of the channels are better than the marginal one for which the system is designed. The other 10 percent of the users have too much distortion because of transmission errors.

In this paper we investigate variable-bit-rate transmission, with the information rate adjusted according to the properties of the channel. To facilitate implementation, we maintain a constant signaling rate through the channel and reciprocally adjust the information rate of the source and the amount of channel coding for forward error correction. The result, relative to fixed-rate transmission, is improved grade of service or, for the same grade of service, the ability to serve more users within each geographical area.[1]

We limited the content of this paper to transmission components of a variable-bit-rate system: the source and channel codecs and their performance in two types of channel. Control components, which involve measurements of channel quality and two-way communication between user pairs, require further study.

### 1.2 Source and channel codes

The most general variable-bit-rate scheme requires control information, an adjustable voice coder/decoder (codec), an adjustable channel codec that provides forward error correction to combat channel impairments, and perhaps an adjustable modem (modulator/demodulator), as indicated in Fig. 1a. In our studies, we have focused on specific source and channel codes that are simple to implement and are especially well suited to variable-bit-rate operation. We do not claim that these codes are optimum, even within a constraint of limited hardware complexity. However, they are strong candidates for practical implementation and this study of their performance provides valuable insights into general principles of variable-bit-rate transmission.

The source codes are embedded Differential Pulse Code Modulation (DPCM),[2] which can be implemented with fixed analog-to-digital and digital-to-analog converters and simple digital control of the number of bits transmitted. The channel codes are convolutional codes, with rates 1/2, 2/3, and 3/4, that provide significant coding gains with maximum likelihood decoding. At rates 2/3 and 3/4, "punctured" code[3] realizations simplify the decoder and lend themselves to variable-bit-

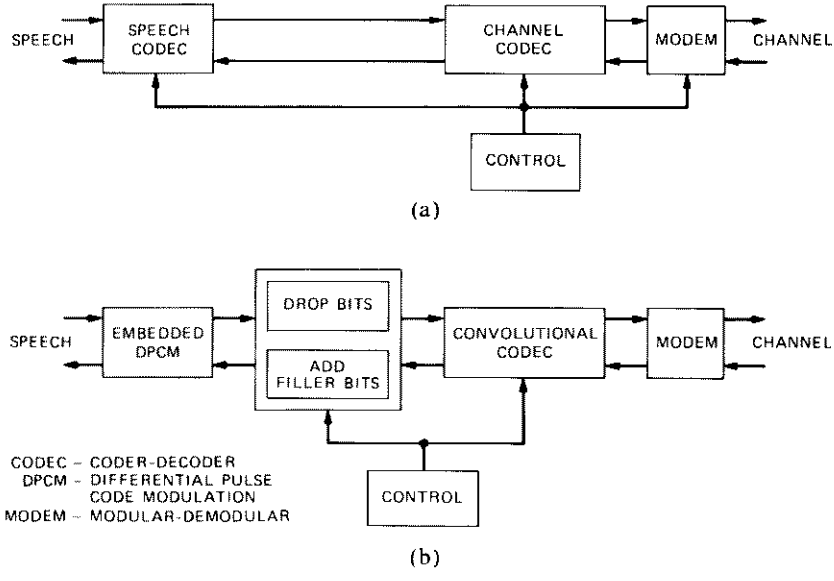



Fig. 1—Elements of a variable-bit-rate system. (a) Adjustable source codec, channel codec, and modem. (b) Fixed embedded source codec and modem. The control of the source rate is simple.

rate operation because the encoder and decoder structures are the same as for the rate 1/2 code. In fact, the punctured codes are rate 1/2 codes with a fraction (1 out of 4 or 2 out of 6) of the channel bits deleted.

In the combined source/channel codec the channel signaling rate is constant; when the channel deteriorates, the least significant speech bits are deleted and the most significant speech bits are protected by a convolutional code. Thus the variable-rate encoding and decoding can be performed, as indicated in Fig. 1b, with a fixed speech codec, a fixed modem, and modest additional hardware relative to fixed-rate operation. Furthermore, the performance characteristics of the embedded source code and the punctured channel code are virtually as good as their conventional counterparts, which require complicated adjustments in a variable-bit-rate environment.

To explore the principle of combining embedded DPCM and punctured convolutional codes we have studied channels with signaling rates of 32 kb/s. We consider four speech transmission formats:

1. All 32 kb/s used for speech transmission
2. 24 kb/s speech transmission, all speech bits protected by a rate 3/4 code
3. 24 kb/s speech transmission, the most significant 2 bits of every 3-bit code word protected by a rate 1/2 code
4. 16 kb/s speech transmission, all speech bits protected by a rate 1/2 code.

These transmission formats, listed in order of increasing resistance to channel impairments, are summarized in Table I, which also presents properties of the convolutional codes discussed in Section III.

Table I—Source and channel code formats, convolutional code properties

| | Format 1 | Format 2 | Format 3 | Format 4 |
|---|---|---|---|---|
| 1. Source code | | | | |
| a) bits/sample | 4 | 3 | 3 | 2 |
| b) bits/s | 32k | 24k | 24k | 16k |
| c) bits/sample protected | 0 | 3 | 2 | 2 |
| 2. Channel code | | | | |
| a) rate | no code | 3/4 | 2/3 | 1/2 |
| b) $U_0 - U_4$ | encoder, Fig. 4 | 10101 | 11001 | 11101 |
| c) $D_0 - D_4$ | | 11111 | 11011 | 01011 |
| d) switching pattern | | UDDD | UDU | UD |
| e) free distance, $d$ | error properties | 4 | 5 | 7 |
| f) weight $w_d$ | | 22 | 25 | 4 |
| $w_{d+1}$ | | 0 | 112 | 12 |
| $w_{d+2}$ | | 1687 | 357 | 20 |
| $w_{d+3}$ | | 0 | 1858 | 72 |
| $w_{d+4}$ | | 66964 | 8406 | 225 |

### *1.3 An example of performance*

Figure 2 pertains to phase-shift-keying modulation in a Rayleigh-fading channel. The curves show speech quality, expressed as segmental signal-to-noise ratio (s/n) (Section 2.4.2), as a function of channel s/n. This is defined as the ratio of average energy per channel symbol to noise power per Hertz. Format 1 (32 kb/s speech transmission) is the best choice when the channel is very good (s/n > 22 dB), but it is also the most vulnerable to transmission impairments. By contrast Format 4 (16 kb/s speech), with substantially more quantizing distortion, can pass through very poor channels with no added degradation. Adaptive DPCM at 16 kb/s is highly intelligible, though somewhat fuzzy. Assuming that the threshold of "adequate" performance is 0.5 dB less than the s/n of 16 kb/s error-free transmission, we see in Fig. 2 that Format 4 extends the useful range of channel quality by 13 dB relative to Format 1.

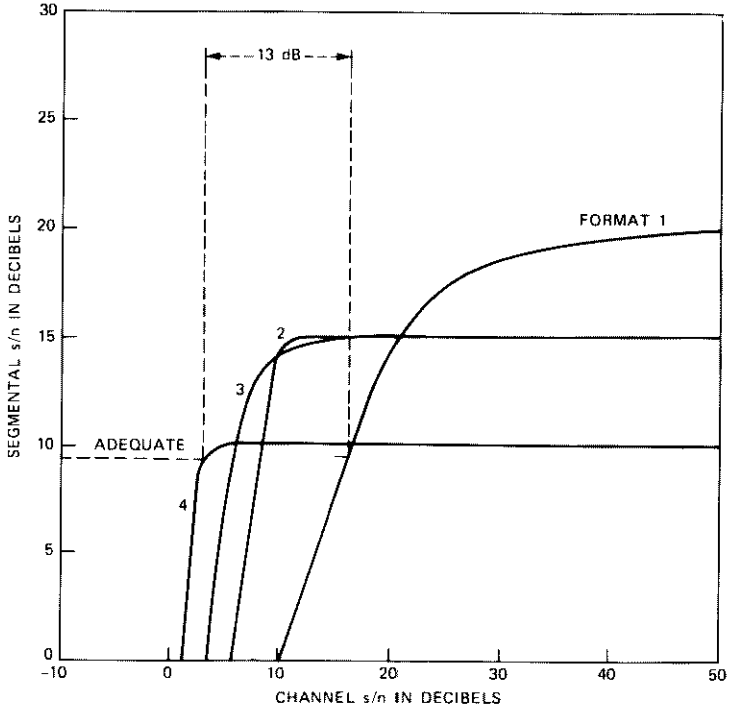



Fig. 2—Performance of the four code formats as a function of the s/n of a Rayleigh-fading channel. If a segmental s/n of 9.5 dB represents "adequate" performance, the variable-rate mechanism extends the threshold of usable channel s/n's from 16 dB (Format 1) to 3 dB (Format 4).

### 1.4 Organization of this paper

In the following three sections we describe the source and channel codes and present theoretical performance bounds and measures obtained from computer simulations of speech communication over idealized channels. The theory of the effects of transmission errors in embedded DPCM will be published in a separate paper,[4] which extends to embedded DPCM Sundberg and Rydbeck's analysis[5–7] of Pulse Code Modulation (PCM) s/n in noisy channels. Our approach to combined source coding and channel coding is similar in spirit to the work of Modestino and Daut on image transmission.[8–10] Section V lists several issues to be addressed in evaluating applications of these techniques to mobile radio communication.

## II. EMBEDDED DIFFERENTIAL PULSE CODE MODULATION

### 2.1 Encoder and decoder

Within the bit stream of an embedded code is a slower bit stream that represents the analog signal source with reasonable quality. Embedded coding simplifies variable-bit-rate communication. The analog-to-digital and the digital-to-analog converters always operate at the same high rate. If necessary, the transmission system deletes bits from the source code and adds filler bits prior to decoding. Although PCM is an embedded code, differential PCM, which is more efficient for speech transmission, is not. On the other hand, minor modifications of the conventional DPCM encoder and decoder do produce an embedded code. In the embedded codec, the integrators in the encoder and decoder process a low-bit-rate representation of the error signal. Then, when bits are deleted in transmission, the same signal arrives at both integrators and large errors are avoided.

Figure 3 models the embedded DPCM encoder and decoder as the combination of two separate codecs: a "minimal" DPCM codec operating with 2 bits/sample, and a "supplemental" PCM codec, which transmits the quantization error of the minimal encoder. The supplemental codec operates at 2 bits/sample and the combination of minimal and supplemental quantizers can be viewed as a two-stage, successive-approximation realization of a 4-bit/sample quantizer. The 2 bits/sample of the minimal codec are always transmitted, while one or both bits of the supplemental codec can be deleted to reduce the transmission rate from 32 kb/s (4 bits/sample) to 24 or 16 kb/s (3 or 2 bits/sample). The codec functions properly because the encoder and decoder predictors operate on the same signal regardless of the number of bits deleted prior to transmission.

Relative to conventional DPCM, the feedback loop of the embedded DPCM encoder operates with reduced resolution when more than 2

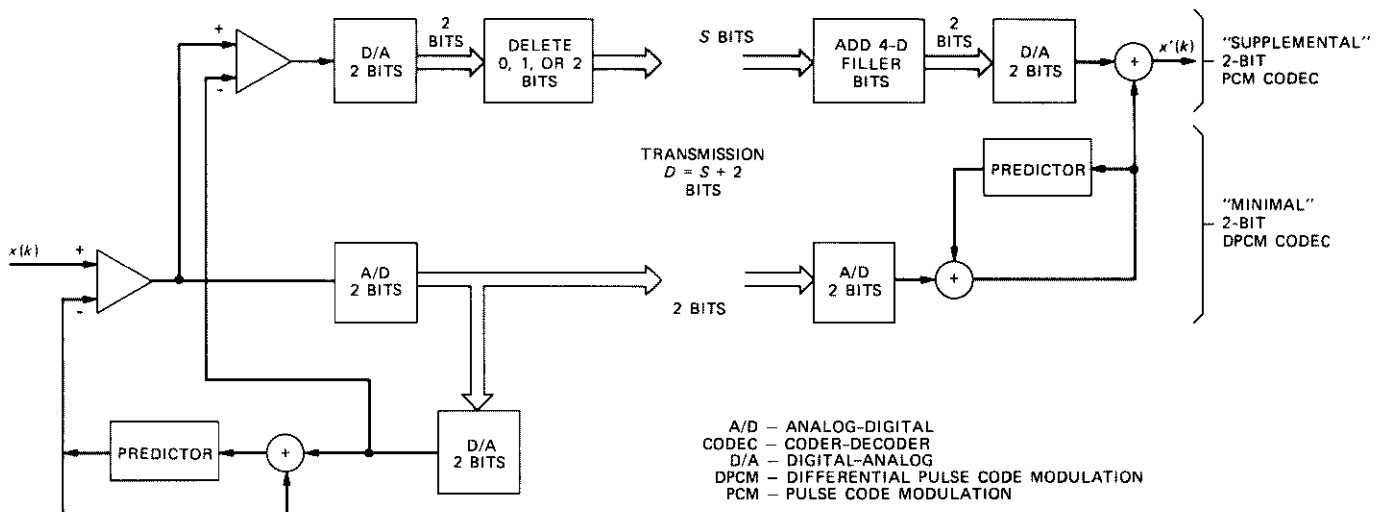



Fig. 3—The embedded DPCM codec modeled as the combination of a "minimal" DPCM codec and a "supplemental" PCM codec. Bits from the supplemental codec can be deleted without severely degrading performance because this deletion has no effect on the ability of the predictor at the decoder to follow the predictor at the encoder.

bits/sample are transmitted. Reference 2 shows that the impact on performance due to this reduced resolution is small, causing no more than a 0.7 dB decrease in s/n in the single-integration codec studied here. (On the other hand, conventional DPCM is unsuited to variable-bit-rate operation. If the transmission system deletes 1 or 2 bits/sample there is a 5.5-dB s/n penalty.)

### 2.2 Effects of transmission errors

Reference 4 presents a detailed analysis of transmission errors in embedded DPCM. Specialized to a single-integration codec with prediction coefficient $a$, a key result of that analysis is the formula for the error in the $k$th sample:

$$x'(k) - x(k) = n_q(k) + e_A(k) + \sum_{i=1}^{\infty} a^i e_M(k - i), \tag{1}$$

in which $x(k)$ and $x'(k)$ are the encoder input and decoder output, respectively. On the right side of (1), $n_q(k)$ is the quantizing distortion of the 2, 3, or 4 transmitted bits/sample, $e_A(k)$ is the noise due to binary errors in all of the transmitted bits, and $e_M(k)$ is the noise due to binary errors in the 2 bits/sample of the minimal quantizer. Equation (1) shows that the errors in these two bits are amplified by the integrator in the decoder.

### 2.3 Audio signal-to-noise ratio

Our analytic tools[4] enable us to compute the mean-square value of (1) under the condition that the quantizer is in a granular state (output and input differ by no more than half a quantizing step). The derived signal-to-noise ratio, therefore, fails to reflect overload distortion. This apparent deficiency does not limit the value of the granular mean-square error as a predictor of speech quality. Subjective tests show that granular noise and overload are quite different perceptually and that adding their squares is often misleading.

A similar issue is raised by the addition in (1) of a granular-noise term ($n_q$) to $e_A$ and $e_M$, which result from transmission errors. Is the mean-square sum a meaningful measure of signal quality? We are optimistic that it is because, contrary to overload, which is signal-dependent, granular noise and noise due to transmission errors are essentially uncorrelated with the input. With this rationale we have derived signal-to-noise ratio formulas of the form[4]

$$\text{s/n} = \frac{E[x(k)]^2}{E[x'(k) - x(k)]^2} = \frac{C}{\sigma_q^2 + \sigma_{qt}^2}, \tag{2}$$

in which $C$ depends on the codec configuration and the source statistics, $\sigma_q^2$ is the quantizing noise power, and $\sigma_{qt}^2$ is the distortion due to

transmission errors, including the effects of correlation between quantizing noise and noise due to transmission errors.

With single-integration DPCM and a 2-bit minimal quantizer

$$C = \frac{1 - a^2L^2/48}{L^2(1 - 2a\rho + a^2)} \tag{3}$$

and

$$\sigma_q^2 \approx 2^{-2D}/3, \tag{4}$$

where $D$ (2, 3, or 4) is the number of bits transmitted, $a$ is the predictor coefficient of the DPCM feedback loop, $\rho$ is the adjacent-sample autocorrelation coefficient of the encoder input signal, $x(k)$, and $L$ is the quantizer load factor: the ratio of overload point to rms quantizer input.

The other quantity in (2), $\sigma_{qt}^2$, depends on the binary representation of quantizer output signals and on the transmission conditions including the modulator and demodulator, the channel, and the encoder and decoder (if they are included) for forward error correction. Table II contains the formulas for $\sigma_{qt}^2$ for the four transmission formats studied in this paper. The formulas pertain to a sign-magnitude binary representation and a quantizer with "3.16-sigma loading", i.e., the quantizer overload point is $\sqrt{10}$ times the rms value of the quantizer input.

Table II—Formulas for computing audio signal-to-noise ratio

*General formula*

$$s/n = \frac{(1 - a^2L^2/48)}{L^2(1 - 2a\rho + a^2)} \cdot \frac{1}{(\sigma_{qt}^2 + 2^{-2D}/3)}$$

*Coder parameters*

Predictor coefficient $a = 0.85$

Quantizer load factor $L = \sqrt{10}$

*Input signal*

Correlation coefficient $\rho = 0.85$

*Transmission effects, sign-magnitude representation*

$P$: binary error probability of unprotected bits

$P_e$: binary error probability of convolutional code

*Format 1*: $D = 4$ (no channel coding)

$$\sigma_{qt}^2 = 0.723P + 0.271P^2 + \frac{a^2}{1 - a^2}(0.725P + 0.275P^2)$$

*Format 2*: $D = 3$ (rate 3/4 code)

$$\sigma_{qt}^2 = 0.25P_e\left(3.37 + \frac{a^2}{1 - a^2}1.73\right)$$

*Format 3*: $D = 3$ (rate 2/3 code)

$$\sigma_{qt}^2 = 0.5P_e\left(1.52 + \frac{a^2}{1 - a^2}1.73\right) + 0.0625P$$

*Format 4*: $D = 2$ (rate 1/2 code)

$$\sigma_{qt}^2 = P_e\left(1.56 + \frac{a^2}{1 - a^2}1.73\right)$$

### 2.4 Speech transmission

#### 2.4.1 Adaptive DPCM

Practical DPCM codecs have time-varying step sizes to accommodate the wide dynamic range of speech sounds. In this study, we have used a robust adaptive quantizer[11] in which the step size of the encoder changes with each sample according to the rule

$$\delta_{k+1} = M\delta_k^{\beta}, \tag{5}$$

where $\delta_k$ is the step size used to encode the $k$th sample, $M$ is a multiplicative factor that depends on the most recent encoder output, and $\beta$ is a leakage factor that helps the codec work properly in the presence of transmission errors.

At the decoder the step size is $\delta_k'$ with

$$\delta_{k+1}' = M'(\delta_k')^{\beta}, \tag{6}$$

where $M'$ depends on the received code word. It can differ from $M$ if there has been an error in transmitting the $k$th code word.

To make the adaptive quantizer work with embedded DPCM, we restrict $M$ to one of two possible values:

$$M = M_1 < 1$$

if the input is in the lower half of the quantizer range, and

$$M = M_2 > 1$$

otherwise. Then with four-bit code words represented in a sign-magnitude format, $M$ and $M'$ depend only on the most significant magnitude bit. As a consequence the decoder step size can track the encoder step size when one or both of the two least significant bits are deleted.

In the simulation studies reported in this paper the adaptive quantizer constants were $M_1 = 0.85$, $M_2 = 1.5$, and $\beta = 0.98$. We selected these values because they offer a good compromise between performance over an ideal channel and tolerance of transmission errors.

#### 2.4.2 Segmental signal-to-noise ratio

In addition to the limitations discussed in Section 2.3, the s/n formulas of Table II are of limited value in evaluating adaptive DPCM coding of speech because they do not account for errors due to different step sizes at the transmitter and receiver. On the other hand, segmental s/n is a quality measure that is reasonably well correlated with listener reports of the quality of speech transmitted by adaptive DPCM over noisy channels. It is defined as the average of the s/n measured in decibels over short segments of the signal, and it can be calculated in a straightforward manner in computer simulations of speech trans-

mission.[12] In our simulations we measure s/n over 16-ms segments (128 samples) and exclude from the average segments detected as silent (power at least 55 dB below the saturation level of the quantizer). We also limit the s/n of each segment to the range −10 to 80 dB.

## III. CONVOLUTIONAL CODING AND DECODING, PSK MODULATION

Our design, analysis, and implementation of the rate 3/4, 2/3, and 1/2 coders and decoders for the most part follow Chapter 6 of Ref. 13. The punctured codes[3] (rate 1/2 codes with a fraction of the channel symbols deleted) are especially well suited to variable-bit-rate operation because the encoder and decoder retain their basic structures as the code format changes; the things that do change are a set of constants (code generators) and the switching patterns that govern the output of channel symbols at the encoder and the input of channel symbols at the decoder.

### *3.1 Code configuration*

In our calculations and simulations, we use codes generated according to Fig. 4. With a five-stage shift register at the encoder, the codes have 4-bit memory (16 unique nodes in the code trellis). This number provides an attractive compromise between performance (enhanced by long memory) and decoder simplicity (short memory). When the rate 1/2 code is employed, the DPCM signal arrives at 16 kb/s, and for each input bit, the switch at the encoder gets one output bit from the top branch of Fig. 4 and one output bit from the bottom branch. When the rate is 2/3, the two most significant bits in each 3-bit DPCM code word stimulate three output bits from Fig. 4. The first of the two DPCM bits results in two output bits—one from the upper branch of Fig. 4 and one from the lower branch, as in the rate 1/2 code. When the second DPCM bit arrives, the encoder releases one bit from the upper branch. Along with these three bits from the convolutional encoder the least significant DPCM bit is transmitted without protection. When the rate is 3/4, the first member of every block of three input bits results in two output bits (from upper and lower branch of Fig. 4); the remaining two input bits in the block stimulate one output bit each (both from the lower branch). The code generators and switching patterns are listed in Table I.

The receiver is a maximum-likelihood decoder, implementing the Viterbi Algorithm. A signal processor produces a measurement (between 0 and 1) for each received bit. This measurement is near 0 if there is a high likelihood that the bit was transmitted as zero and near 1 if there is a high likelihood that one was transmitted. With each block of two (rate 1/2), three (rate 2/3), or four (rate 3/4) received bits, the Viterbi decoder accepts combined measurements of the first

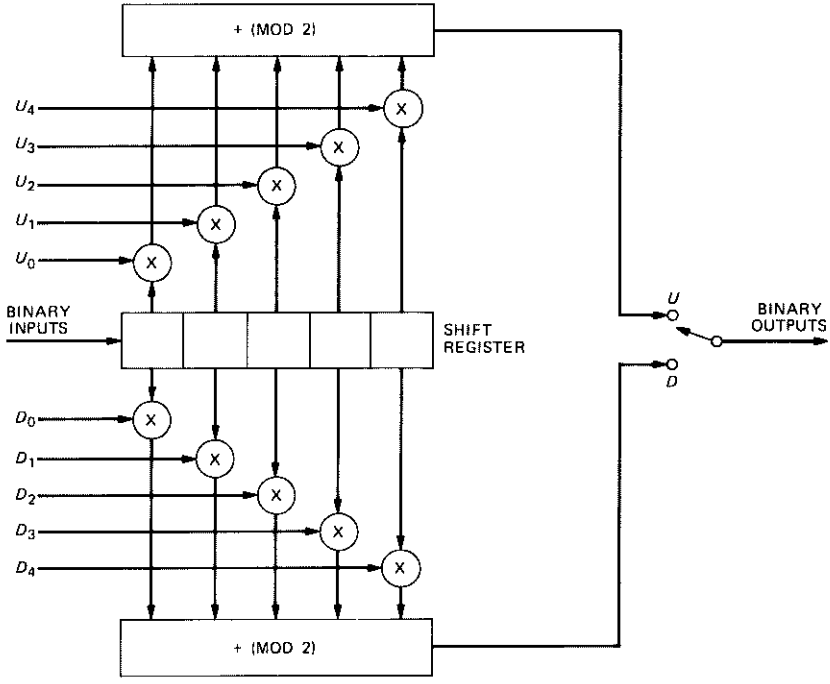



Fig. 4—Structure of the convolutional encoder. $U_i$ and $D_i$ are binary coefficients. For rate 1/2 code there are two output bits ($U$ and $D$) for each input bit. For rate 2/3 there are two outputs ($U$ and $D$) for one input and one output ($U$) for the next input. For rate 3/4 there are two outputs ($U$ and $D$) for the first input and one output ($D$) for each of the next two input bits.

two bits and updates the metrics and path memories associated with the 16 possible decoder states. The decoder then releases one bit corresponding to an input bit delayed by the length of the decoder path memory. (In the simulations reported here the path memory was 30 bits for all codes.) If the rate is 2/3 or 3/4, the measurement of each remaining bit in the block causes the decoder to update all metrics and path memories and to release one additional bit.

### *3.2 PSK modulation, likelihood measurements*

To study the performance of variable-bit-rate transmission with embedded DPCM and convolutional codes we have confined our attention to Binary Phase-Shift Keying (BPSK) modulation with coherent detection. In a Gaussian-noise channel the demodulator output for each channel symbol is either

$$r = -A + n \quad \text{or} \quad r = A + n,$$

depending on whether "zero" or "one" was transmitted. The constant,

$A$, is the received sinewave amplitude and the noise, $n$, is a sample of a zero-mean normal random variable with variance $\sigma^2$. The symbol s/n is $\rho = A^2/2\sigma^2$. In our computer simulations, the likelihood measurement for each symbol is $m = 0.5 + r/\text{const}$, where the constant is large enough to ensure $0 < m < 1$. (With $m$ effectively unquantized, as in our simulations, the constant is otherwise arbitrary. With $m$ quantized to a few bits the constant is a compromise between dynamic range and resolution.)

To update the state metrics of the Viterbi decoder we add $m$ or $1 - m$, depending on whether the relevant branch of the code tree is associated with zero or one transmitted. For a constant-amplitude, white-Gaussian-noise channel, this procedure produces a precise likelihood metric. In a fading channel, which is of principal interest in mobile radio, this metric is only an approximation to the likelihood function.[14] A true likelihood metric involves elaborate computations that would probably be prohibitively complicated in practice.

### 3.3 Probability of error

The theory of convolutional coding provides upper (union) bounds on bit-error probability as a function of s/n. These bounds are infinite series that are truncated for the purpose of numerical computation. In our calculations we have used sums of five terms to estimate the error probabilities of the convolutional codes:

$$P_e = \frac{1}{b} \sum_{n=d}^{d+4} w_n f_{n,C}(\rho), \tag{7}$$

where $d$ is the free distance of the code; $b$ is the number of source bits per code word (1, 2, 3 for rate 1/2, 2/3, 3/4); and $f_{n,C}(\rho)$ is the probability that an incorrect path, $n$ bits removed from the correct path, has a lower metric than the correct path. This probability is a function of the type of channel, ($C = g$ for a Gaussian channel, $C = R$ for Rayleigh fading), the modulation technique, and the s/n $\rho$.

Table I contains the free distances, $d$, and the weights, $w_n$, of the three convolutional codes we have studied. For BPSK with coherent detection we have the precise formula for a nonfading channel,

$$f_{n,g}(\rho) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2n\rho}}^{\infty} \exp\left[-\frac{x^2}{2}\right] dx. \tag{8}$$

For independent Rayleigh fading signals in white Gaussian noise, we do not have a precise formula for $f_{n,R}(\rho)$. Instead we use the upper bound[14]

$$f_{n,R}(\rho) < \left[\exp(\beta/\rho)\, \frac{\sqrt{1 + 2/\beta} - 1}{\sqrt{1 + 2/\beta} + 1}\right]^n, \tag{9}$$

$$\beta = \sqrt{\rho^2 + 1} - 1. \tag{10}$$

Figures 5 and 6 show estimates of binary error rates computed from (7) through (10) and the results of computer simulations of random data transmitted through a convolutional coder, a random channel, and a Viterbi decoder with path memory 30 bits. In Fig. 5 the simulations and calculations for the nonfading channel agree very closely. On the other hand Fig. 6 suggests that the bound in (9) is quite loose over the range of conditions of interest to us here; the estimated channel s/n corresponding to a given error rate is 3 to 4 dB lower in the simulations than in the computed curves. Figures 5 and

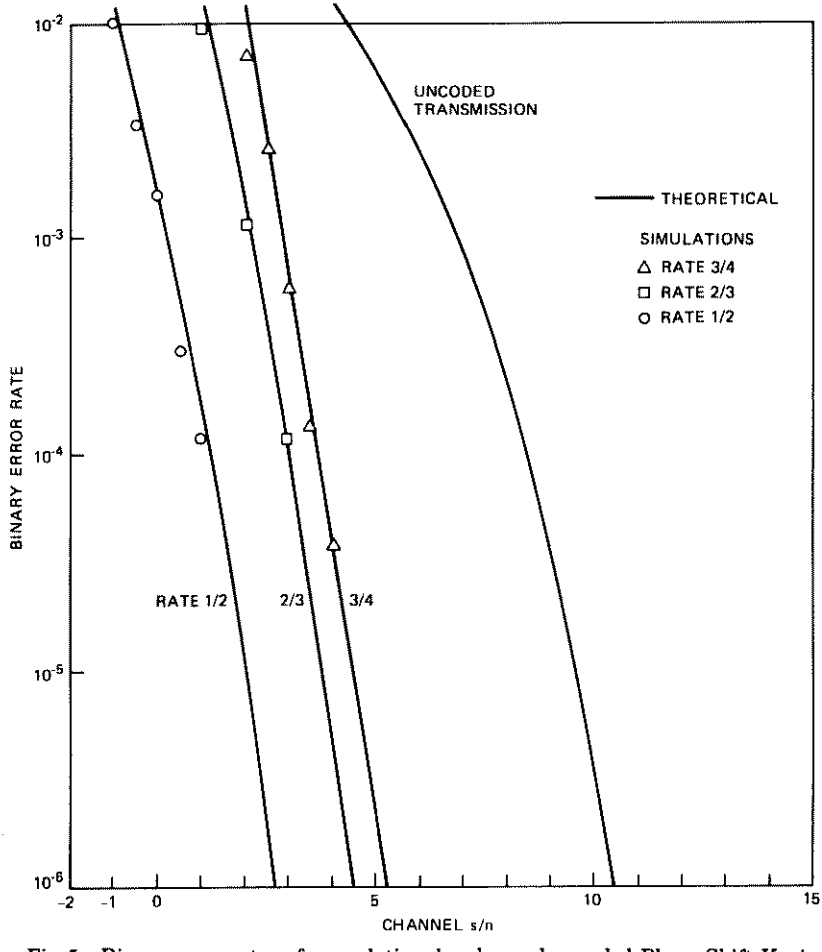



Fig. 5—Binary error rates of convolutional codes and uncoded Phase-Shift Keying (PSK) in a nonfading channel. The theoretical curves and the results of computer simulations are in close agreement.

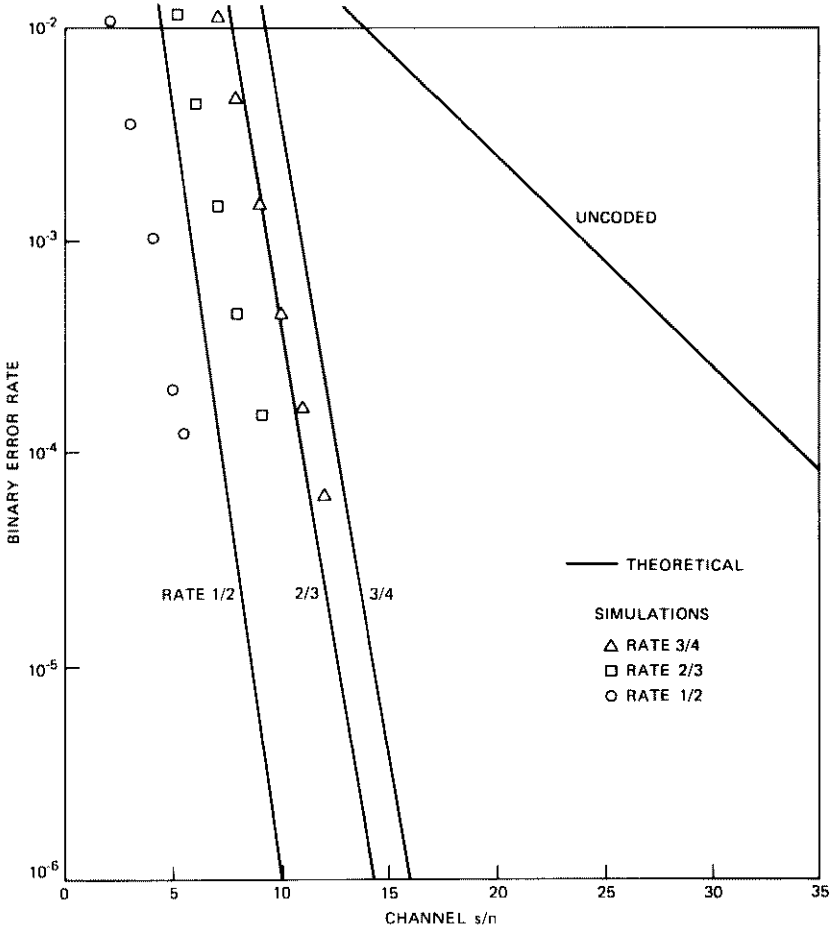



Fig. 6—Binary error rates of convolutional codes and uncoded PSK in a Rayleigh-fading channel. The theoretical curves, based on upper bounds on path error probabilities, are loose over the range of error rates ($10^{-4}$ to $10^{-2}$) of interest to us here.

6 also show $P$, the binary error rates for uncoded transmission, which are $P = f_{1,g}(\rho)$ in (8) for the nonfading channel and

$$P = \frac{1}{2}\left[1 - \sqrt{\frac{\rho}{1+\rho}}\right] \tag{11}$$

with Rayleigh fading.[15]

Note that the independent variable in Figs. 5 and 6 is the s/n of *channel symbols* rather the s/n of *information bits*, which is often plotted in comparisons of coding schemes for digital data transmission.

For our purpose the channel s/n is an appropriate measure because it remains constant as we change code formats.

## IV. PERFORMANCE

To evaluate the potential effectiveness of variable-bit-rate transmission we have computed audio s/n as a function of channel s/n using the formulas in Table II and (7) through (10). Figure 7 applies to a nonfading Gaussian channel and Fig. 8 applies to independent Rayleigh fading signals (the s/n's of all channel symbols are mutually independent) in Gaussian noise. All curves pertain to a sign-magnitude representation of source symbols, which is generally more tolerant of transmission errors than the natural-binary representation.

We assume that an audio s/n within 0.5 dB of the s/n of error-free 16 kb/s transmission provides "adequate" voice quality; therefore, in Fig. 7 we estimate that relative to conventional 32 kb/s transmission, variable-bit-rate operation extends the range of useful channel conditions by 4.3 dB in a nonfading channel. (Without channel coding

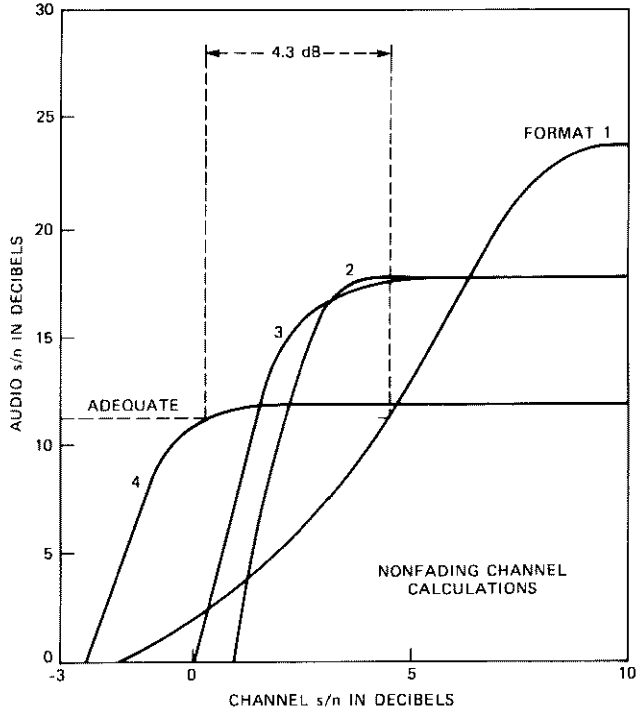



Fig. 7—Calculated performance of the four code formats as a function of the s/n of a nonfading channel. Convolutional coding extends the range of channel s/n's that offer adequate communication by 4.3 dB.

voice quality is inadequate when the channel s/n is less than 4.5 dB. With coding this threshold is 0.2 dB.) Figure 8 suggests that with Rayleigh fading the coding gain is 9.2 dB.

The theoretical formulas, though simple, rely on many bounds, unverified assumptions, and idealizations of a practical communications environment. To obtain more realistic estimates of performance we have resorted to computer simulations of adaptive DPCM transmission of a 2.5-second speech sample. The simulated Viterbi decoder has a path memory of 30 bits. Performance measurements, with speech quality measured as segmental s/n (see Section 2.4.2), are shown in Fig. 9 for a nonfading channel and in Fig. 2 for the Rayleigh-fading channel.

We see that the approximations in Figs. 7 and 8 provide the same qualitative information as the speech simulations in Figs. 9 and 2. For the nonfading channel, simulations and theory lead to nearly equal estimates (4.3 dB and 4.7 dB) of coding gain. In the case of Rayleigh fading the theory underestimates this gain (by 3.8 dB) relative to

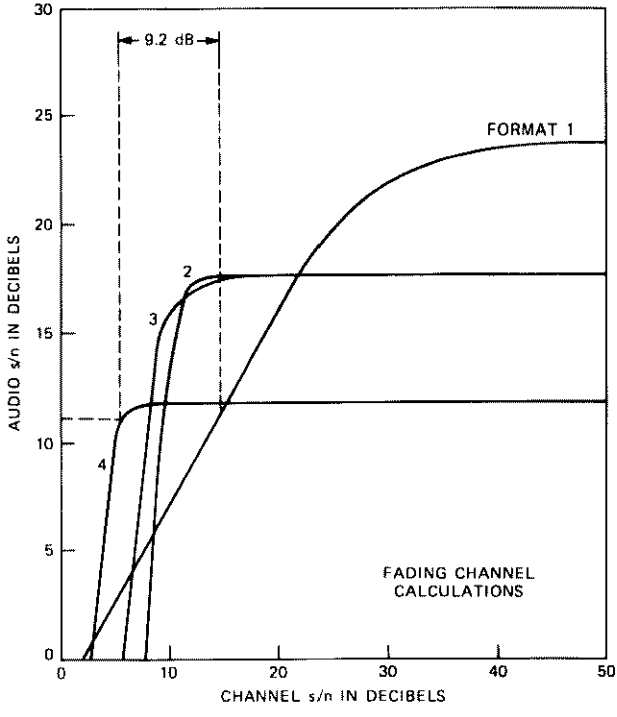



Fig. 8—Calculated performance of the four code formats in a fading channel. The estimate of 9.2-dB coding gain is a lower bound because the computed error probabilities of the Viterbi decoder are loose upper bounds.

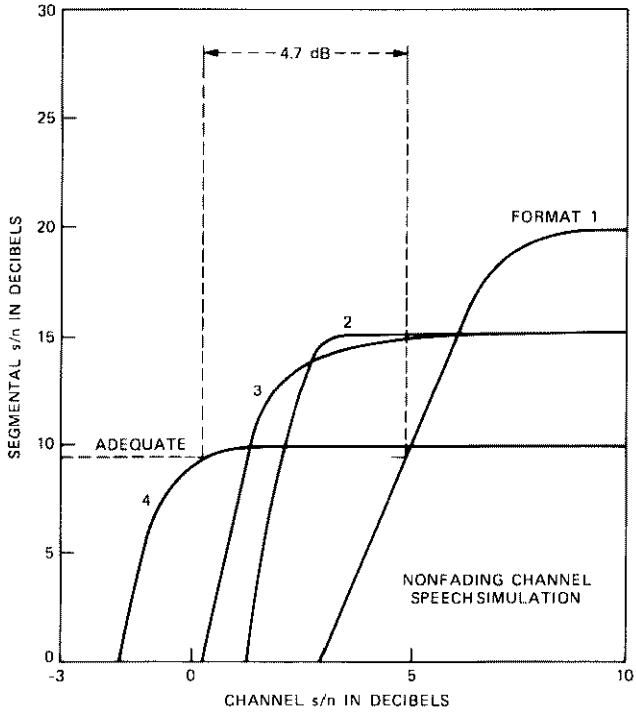



Fig. 9—Performance of the four code formats in simulated speech transmission over a nonfading channel. Coding lowers by 4.7 dB the channel s/n required for adequate quality.

simulation. This discrepancy is largely due to the loose upper bound on the binary error rate of the Viterbi decoder in a fading channel. (See Fig. 6 and Section 3.3.)

All of the performance curves show that transmission Format 2 (24 kb/s speech, all bits protected) is of very limited value. When the coded bits are essentially error free, Format 2 is only slightly better than Format 3 (24 kb/s, 2 of 3 bits/sample protected). In this condition, the transmission noise of Format 3 is due to errors in the third, uncoded, bit, which have a small effect on the quality of embedded DPCM because the errors are not enhanced by the receiver integrator. (In nonembedded DPCM Format 2 would be more effective than Format 3 over a wide range of channel conditions.[4]) In a practical application, therefore, Format 2 would be omitted and the system would switch among Formats 1, 3, and 4.

## V. CONCLUSIONS: APPLICATIONS TO MOBILE RADIO

We have described the key elements of a combined source/channel

codec and evaluated its ability to communicate speech over idealized channels. While this approach has other possible applications, we were motivated to study its potential for enhancing mobile radiotelephony. A detailed study of the codec in a mobile-radio environment is currently under investigation. We conclude this paper by reviewing the features of the scheme and listing important issues to be addressed in assessing its value in a mobile-radio context.

A principal advantage of the source codes and channel codes is their simplicity. The embedded DPCM coder and decoder can be realized on a single-chip microcomputer[16] and the Viterbi decoder, with only 16 states, is within the state-of-the-art of special-purpose integrated circuits.[17] Because the channel signaling rate is constant, no special modem is required. Notwithstanding this simplicity, the speech quality for good channels is comparable to that of conventional telephony and the extension of the useful range of channel conditions (4.5 dB in a nonfading channel, 13 dB with independent Rayleigh fading) is substantial.

To extend this idealized study to the context of variable-bit-rate communication over mobile-radio channels, we are led to study the:

1. Control mechanism for altering code formats
2. Channel characteristics, including the temporal nature of the Rayleigh fading and the geographical distribution of signal-to-interference ratio in a mobile-radio service area
3. Alternative modulation techniques to binary-phase-shift keying
4. Possibility of diversity reception.

## REFERENCES


1. D. J. Goodman and C.-E. Sundberg, "Quality of Service and Bandwidth Efficiency of Cellular Mobile Radio with Variable-Bit-Rate Speech Transmission," Trans. Veh. Technol., *VT-32*, No. 3 (August 1983).
2. D. J. Goodman, "Embedded DPCM for Variable Bit Rate Transmission," IEEE Trans. Commun., *COM-28*, No. 7 (July 1980), pp. 1040–6.
3. J. B. Cain, G. C. Clark, and J. M. Geist, "Punctured Convolutional Codes of Rate $(n-1)/n$ and Simplified Maximum Likelihood Decoding," IEEE Trans. Inf. Theory, *IT-25* (January 1979), pp. 97–100.
4. D. J. Goodman and C.-E. Sundberg, "Transmission Errors and Forward Error Correction in Embedded Differential Pulse Code Modulation," B.S.T.J., *62*, No. 9, Part 1 (November 1983).
5. N. Rydbeck and C.-E. Sundberg, "Analysis of Digital Errors in Non-Linear PCM Systems," IEEE Trans. Commun., *COM-24*, No. 1 (January 1976), pp. 59–65.
6. C.-E. Sundberg and N. Rydbeck, "Pulse Code Modulation with Error-Correcting Codes for TDMA Satellite Communication Systems," Ericsson Technics, *32*, No. 1 (1976), pp. 1–56.
7. N. Rydbeck and C.-E. Sundberg, "PCM/TDMA Satellite Communication Systems with Error Correcting and Error Detecting Codes," Ericsson Technics, *32*, No. 3 (1976), pp. 195–247.
8. J. W. Modestino and D. G. Daut, "Combined Source and Channel Coding of Images," IEEE Trans. Commun., *COM-27*, No. 11 (November 1979), pp. 1644–59.
9. J. W. Modestino, D. G. Daut, and A. L. Vickers, "Combined Source-Channel Coding of Images Using the Block Cosine Transform," IEEE Trans. Commun., *COM-29*, No. 9 (September 1981), pp. 1261–74.

10. D. G. Daut and J. W. Modestino, "Two-Dimensional DPCM Image Transmission Over Fading Channels," IEEE Trans. Commun., *62*, No. 9, Part 1 (November 1983).
11. D. J. Goodman and R. W. Wilkinson, "A Robust Adaptive Quantizer," IEEE Trans. Commun., *COM-23*, No. 11 (November 1975), pp. 1362–5.
12. C. Scagliola, "Evaluation of Adaptive Speech Coders Under Noisy Channel Conditions," B.S.T.J., *58*, No. 6, Part 2 (July–August 1979), pp. 1369–94.
13. G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*, New York: Plenum Press, 1981, Chapter 6, pp. 227–66.
14. J. Hagenauer, "Viterbi Decoding of Convolutional Codes for Fading and Burst Channels," Proc. 1980 Int. Zurich Seminar (March 1980) pp. G2.1–7.
15. M. Schwartz, W. R. Bennett, and S. Stein, *Communication Systems and Techniques*, New York: McGraw Hill, 1966, p. 407.
16. J. R. Boddie, et al., "Adaptive Differential Pulse-Code-Modulation Coding," B.S.T.J., *60*, No. 7, Part 2 (September 1981), pp. 1547–61.
17. R. M. Orndorff, et al., "Viterbi Decoder VLSI Integrated Circuit for Bit Error Correction," Nat. Telecommun. Conf., New Orleans, Louisiana, December 1981, Conference Record, pp. E1.7.1–4.

## AUTHORS

**David J. Goodman,** B.E.E., 1960, Rensselaer Polytechnic Institute; M.E.E., 1962, New York University; Ph.D. (Electrical Engineering), 1967, Imperial College, London; Bell Laboratories, 1967—. Mr. Goodman has studied various aspects of digital communications including analog-to-digital conversion, digital signal processing, subjective assessment of voiceband codecs, and spread spectrum modulation for mobile radio. He is Head, Communications Methods Research Department. In 1974 and 1975 he was a Senior Research Fellow and in 1983 a Visiting Professor at Imperial College, London, England. Member, IEEE.

**Carl-Erik W. Sundberg,** M.S.E.E., 1966, and Dr. Techn., 1975, Lund Institute of Technology, University of Lund, Sweden; Bell Laboratories, 1981–1982. Mr. Sundberg is an Associate Professor in the Department of Telecommunication Theory, University of Lund, and a consultant in his field. He is Director of the consulting company SUNCOM, Lund. During 1976 he was with the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands, as an ESA Research Fellow. He has been a Consulting Scientist at LM Ericsson and SAAB-SCANIA, Sweden, and at Bell Laboratories. His research interests include source coding, channel coding (especially decoding techniques), digital modulation methods, fault-tolerant systems, digital mobile radio systems, spread spectrum systems, and digital satellite communication systems. He has published a large number of papers in these areas during the last few years. Senior Member, IEEE; member, SER, Sveriges Elektroingenjörers Riksförening.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# Alternative Cell Configurations for Digital Mobile Radio Systems

By C.-E. SUNDBERG*

(Manuscript received January 19, 1982)

This paper introduces a novel class of antenna configurations and applies it to cellular digital mobile radio systems with frequency reuse. Directional antennas are used extensively. Cooperation between more base stations than are in the conventional three-corner directional antenna scheme is required for the alternative cells. The paper studies the relationship between signal-to-cochannel interference and trunking efficiency (availability of channels) and compares conventional systems with the same base-station locations. We concluded that the new antenna configurations can significantly improve trunking efficiency. Time-division retransmission systems with space diversity are considered for some cases. Furthermore, we show how transmitter power weighting (i.e., certain transmitters have higher output power than others) can improve the signal-to-cochannel-interference ratio for the novel cellular systems.

## I. INTRODUCTION

Good spectral efficiency in digital mobile radio systems is obtained through frequency reuse. That is, each cell in the cellular system is assigned a number of channels in a frequency-division system, and each channel (frequency) is reused at a cell further away. The closer this second cell is, the higher the system capacity. On the other hand, cochannel interference increases when the interfering cells are too close. Omnidirectional and directional antenna arrangements have

* Bell Laboratories. Currently at the University of Lund, Sweden.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

been considered in cochannel interference reduction. The concept of centrally located base stations with omnidirectional antennas is one possibility. Another suggested scheme is cooperating base stations with 120-degree directional antennas in three corners in a hexagonal cell. In this paper we generalize these antenna configuration ideas. Cooperation between more base stations and extensive use of directional antennas reduce cochannel interference. For example, all channels allocated to three cells in a conventional three-corner scheme can be made available in one "supercell" with the area three times the initial cell. The local availability of channels is now significantly improved. This leads to improved trunking efficiency.

As we will see, working with cells can sometimes be confusing. However, all comparisons below will be made with the same number of base stations per unit area. As a matter of fact, the base-station locations are always the same.

Choice of modulation scheme also affects spectral efficiency. In this paper we will, however, only deal with the frequency-reuse issue for a given modulation scheme, e.g., Quadriphase Shift Keying (QPSK).

This paper will discuss two main ideas. The first is the design of supercells with cooperation between a large number of directional base-station transmitters. The second idea is base-station transmitter power weighting. Since cochannel interference is affected differently by different transmitters, it can be reduced by choosing proper power levels for the base-station transmitters in cellular systems composed by supercells.

The rest of this paper is organized as follows. Section II contains background material on conventional cellular systems, time-division retransmission, and propagation and interference models used in the analysis. Section III contains the new antenna configurations, and Section 3.1 the calculations of signal-to-cochannel-interference ratios for cellular systems with these configurations. Section 3.2 presents the transmitter power weighting analysis. Section IV contains a discussion and conclusions.

## II. BACKGROUND MATERIAL

Before we discuss cellular systems in any detail, we will give some brief background information about conventional cellular arrangement, the time-division retransmission method, and signal propagation and interference models for the fading land mobile radio channel.

### *2.1 Conventional cellular arrangement*

We will discuss frequency reuse in cellular systems for digital mobile radio systems. Figure 1 shows an example of a cellular system with three hexagonal cells ($N = 3$) per cluster. Frequencies are assigned as

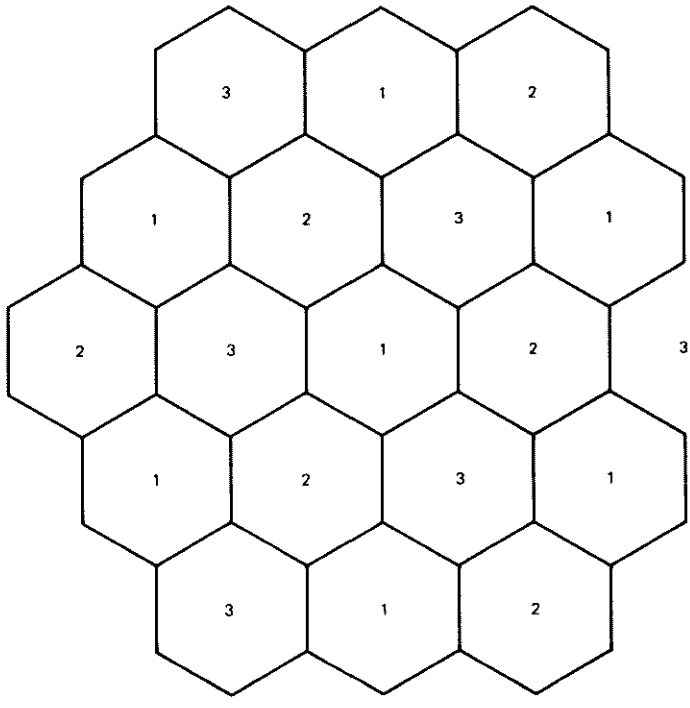



Fig. 1—Example of a cellular system. Each cell is a hexagon of equal area. The frequencies are reused. Cells marked with the same digit use the same frequency channel set. In a cluster of $N = 3$ cells, all channels are used.

in Fig. 2. The channel bandwidth is denoted $B_c$ and the total system bandwidth is denoted $B_T$. Note that for a fixed total number of channels, a fixed antenna configuration, and a fixed cell size, a large frequency reuse factor, $N$, gives good cochannel interference (the interfering cells get further away) but a lower number of channels per base station (a low number of channels available at any particular location). This number is inversely proportional to $N$.

The antennas in each cell in Fig. 1 might be omnidirectional and then located in the center of the cell or 120-degree directional located in each of three corners.[1,2]

### *2.2 Time-division retransmission*

The time-division retransmission (TDR) concept is described in Refs. 3 and 4. The basic ideas are the following. The fading channel changes "slowly" (during one burst or package). Communication in both directions takes place in packages transmitted in the same frequency band: first mobile-to-base and then base-to-mobile. During mobile-to-base transmission, the channel is estimated for maximal-

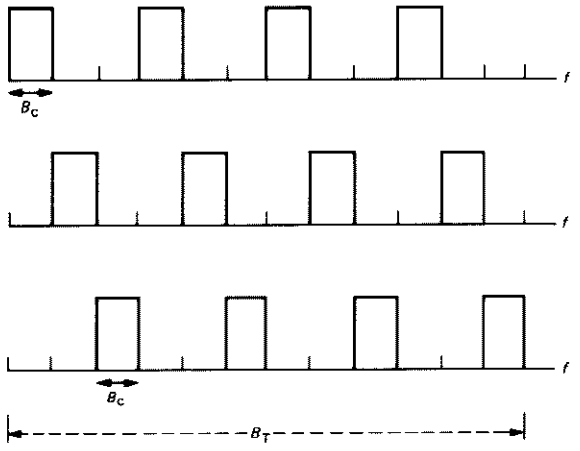



Fig. 2—Frequency plan for the cellular system with $N = 3$, shown in Fig. 6. The channel set marked 1 is used in each cell marked 1, etc. By channel we will consistently mean a two-way channel, including overhead for synchronization (when required).

ratio, space-diversity combining in both directions. Several schemes[3] have been proposed for performing the required co-phasing, for transmission from and reception at the base station. Because all diversity combining takes place at the base stations, the mobile equipment is relatively simple. As a consequence, it is feasible to use more than two branches of diversity with TDR.

We mention the concept of TDR because some of the cellular systems might require diversity because of short-frequency reuse distance and thus high cochannel interference.

### *2.3 Signal propagation and interference*

It is assumed that mobile radio reception in an urban environment is characterized by

$$P(\bar{r}) = |\bar{r}|^{-\alpha} S(\bar{r}) \cdot R^2(\bar{r}), \tag{1}$$

where $P(\bar{r})$ is the received signal power at location $\bar{r}$ (position vector relative to a transmitter).[1,3,4] The first factor $|\bar{r}|^{-\alpha}$ is a reduction factor due to the distance between the mobile unit and the transmitter and $\alpha$ is the propagation constant. It is normally assumed that $\alpha$ is in the range three to four in the urban environment.[1,3] In free space, $\alpha = 2$.

The second factor, $S(\bar{r})$, represents shadow fading,[1,3,4] and the third factor, $R^2(\bar{r})$, represents Rayleigh fading.[4,5] $R$ is the envelope of the received signal. It is modeled as a random variable with the density function

$$p(R) = 2Re^{-R^2}, \tag{2}$$

with $E\{R^2\} = 1$ (see Refs. 4 and 5). In general, $R$ varies with vehicle location and signal frequency.

We will consider propagation and interference in cellular systems with frequency reuse, and use the same basic assumptions as in Refs. 1 and 3.

It is assumed that the cochannel interference is formed by the incoherent sum of contributions from many interfering sources. This sum is assumed to be equivalent to stationary Gaussian noise.[1,3] It is assumed that the shadow and Rayleigh fading of the total interference is negligible compared to the fading of the signal.[1,3,4]

It is also assumed that cochannel interference is the main source of additive signal degradation. The additive thermal background noise is assumed to be negliglible compared to the cochannel interference. Thus, the transmitter power and the cell sizes are assumed to be such that alien background noise from sources other than mobile units and transmitters in the cellular system is negligble compared to cochannel and adjacent-channel interference.

Below we will use the same technique as in Refs. 1 and 3 for calculating cochannel interference. The signal-to-interference ratio is defined as the ratio of the signal power to the total interference power, based on the $|\bar{r}|^{-\alpha}$ propagation law and averaged over shadow and Rayleigh fading. It is assumed that the fading is flat over the band of each channel.

## III. NOVEL ANTENNA CONFIGURATIONS

In this section we will present some new methods of organizing antennas covering the hexagons in a cellular system. Analysis of signal-to-cochannel interference for some of these schemes is presented in Section 3.1. Adjacent channel interference is calculated in Ref. 6. These calculations are carried out based on the propagation assumptions made in the introduction and in Refs. 1 and 3.

Figure 3 illustrates the key idea in this paper. Assume base stations in three of the corners in Fig. 1. Form "supercells" by grouping the three cells with C/3 different channels in each into a new "cell" and let the total number of channels, i.e., C, be available throughout the cell. A new cellular system is formed, where the building block now is a cell with a centerbase station and six 120-degree corner stations.

Consequently, the cochannel interference properties of the cell in Fig. 3 are merely fair. By simply rotating the 120-degree corner antennas 30 degrees, one obtains a new larger hexagon (see Fig. 4). The radius of this cell is $r\sqrt{3}$, where $r$ is the radius of the basic hexagonal cell in the three-corner system. The large hexagon is shown solid in Fig. 4.

It is clear from Fig. 4 that the area of the large hexagon ("super

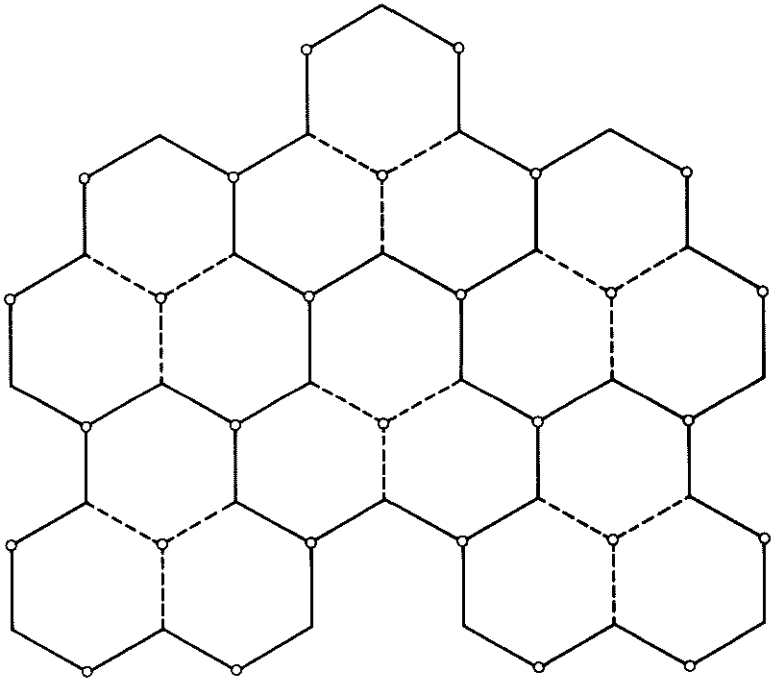



Fig. 3—A cellular system where "supercells" are formed by merging three conventional cells.

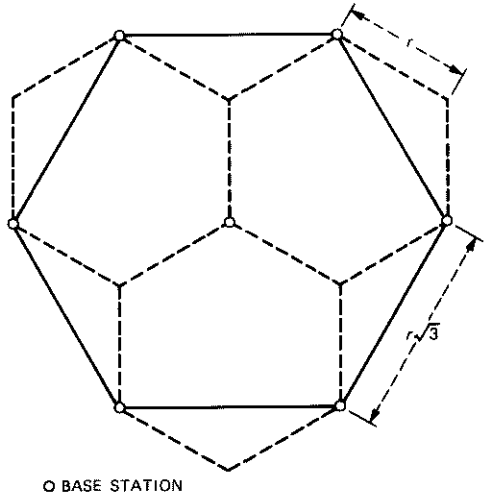



Fig. 4—Comparison between three conventional hexagonal cells (dashed) and a "superhexagon" (solid).

hexagon") is the same as that of three basic hexagonal cells. It is also clear that the cellular system based on the superhexagon has its base stations at exactly the same locations as the basic system with which we started.

Below, we will consider systems based on the superhexagon (and variations thereof). Above, we said that C/3 channels from each of the three basic cells were combined into C channels available throughout the supercell. This will mean not only increased channel availability but also somewhat increased cochannel interference, as we will see. The supercell will also be used with C/3 different channels available throughout the cell. This will mean decreased channel availability and decreased cochannel interference.

Figure 5 shows a number of ways to organize the antennas to cover a cell (hexagon). All the cells in Fig. 5 are shown with equal base-station locations. The shortest distance between base stations is $r\sqrt{3}$, where $r$ is the radius of the basic cell $a$. It is assumed that the cellular system consists of a number of cells with equal antenna arrangements in each cell. Different cellular systems can be constructed based on the cells shown in Fig. 5.

By a cell we will mean the basic building block in a cellular system, like those in Fig. 5. Sometimes we will refer to a cell as a supercell. For these cases, this cell can be thought of as a combination of a group of smaller cells. Throughout the paper, comparisons will only be made between cellular systems where the base stations have identical locations.

Other cell types (triangular, square, etc.) are, of course, also conceivable (see e.g., Ref. 7). Groups of such basic cells can also be combined into "supercells," much the same way as we have done here for hexagons. The discussion in this paper will, however, be confined to the cells in Fig. 5.

Cases a, b, and c in Fig. 5 are considered in Refs. 1 and 3. In Fig. 5a the antenna is centrally located in each cell and is omnidirectional. In Fig. 5b the antenna is also centrally located in each cell, but the base station now consists of three 120-degree antennas. One of them serves a particular mobile unit at a particular moment. Because of the directivity of the antenna, interference is reduced during mobile-to-base transmission. Further improvements are obtained by placing 120-degree antennas in three cell corners (see Fig. 5c). This arrangement improves signal-to-interference ratios in both directions. The reason is that the distance from a base station to the desired mobile unit is improved relative to the distance to interference sources [see eq. (1) above]. The maximum distance from the desired mobile unit to the closest base station is the same in cells a, b, and c. In cell c it is assumed that the corner base station with the best signal-to-interfer-

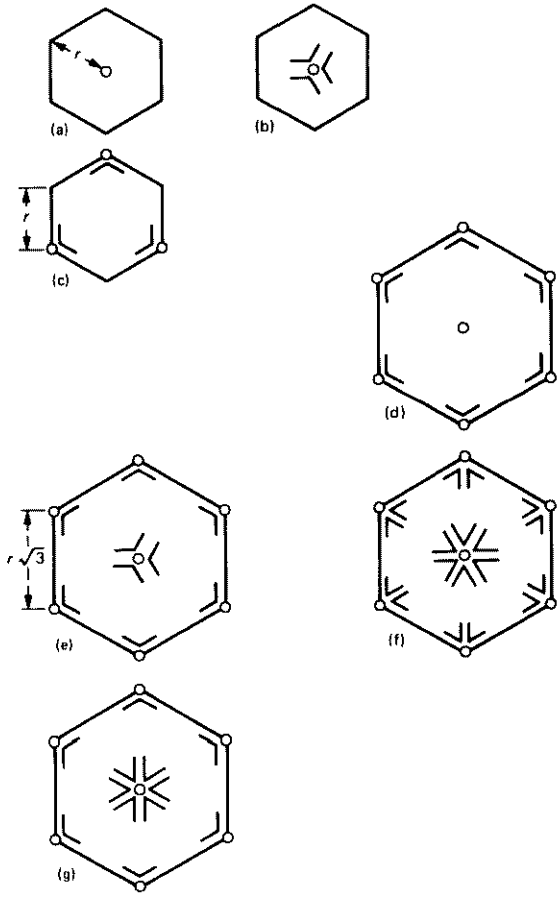



Fig. 5—Different ways of arranging antennas in a cell. Cell a has an omnidirectional antenna in the center of the cell. Cell b has 120-degree antennas in the center of the cell. Cell c has 120-degree antennas in three corners. Cells d, e, f, and g have antennas in all corners and in the center of the cell. Various combinations of 120-degree antennas, 60-degree antennas, and omnidirectional antennas are considered. The cells above will be referred to as cell a to cell g in the text. The cell radius is denoted $r$ for the cells a to c. The radius is $r\sqrt{3}$ for cells d to g. The base-station locations are the same for all cells.

ence ratio serves the mobile unit. The three base stations require coordination so that the desired mobile unit is served by the base station with the best signal-to-interference ratio.

Figures 5d to 5g show arrangements with antennas placed in each cell corner and in the center of the cell. Figure 5e (cell type e) is the one we arrived at in the example above in Fig. 4. Figure 5d shows 120-degree antennas in the corners and an omnidirectional antenna in the center. Figure 5c shows 120-degree antennas in all places. Figure 5f shows the corresponding case with 60-degree antennas. Finally, Fig.

5g shows a hybrid between e and f. Below, we will concentrate on case e and f. Analysis will also be presented for some cases with cell d. For simplicity, the cells will be referred to as cells a to g in the following.

It is evident from Fig. 5 that the maximum distance from a desired mobile unit to the closest base station, $d_0$, is unchanged in cells d, e, f, and g, compared to cells a, b, and c. Figures 6 and 7 show cells e and f in more detail. It is easily seen that the maximum distance to the closest serving base station is $d_0 = r$ for all cells in Fig. 5.

Figure 6 shows the service areas for the different antennas for the cell in Fig. 5e, assuming all base stations are transmitting at equal power level. It is clear that the worst-case locations (in terms of

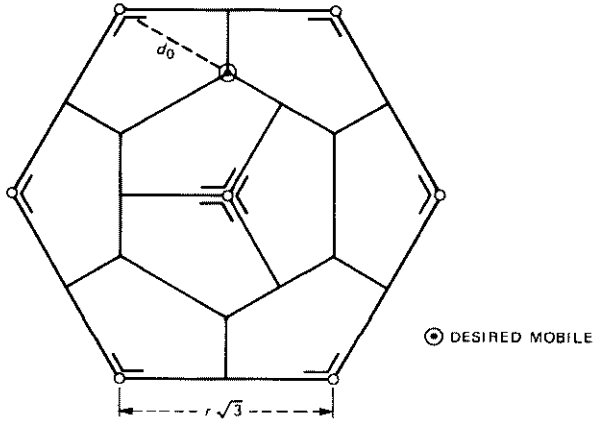



Fig. 6—Coverage areas for the 120-degree antennas in cell e. The example shows the location of a desired mobile unit with maximum distance to a base station.

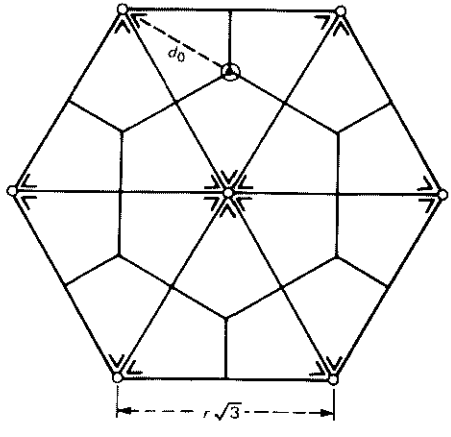



Fig. 7—Same as Fig. 6 for cell f.

maximum distance to the closest base station) for the desired mobile unit is on distance $d_0 = r$ from three base stations. Figure 6 shows that seven different antenna locations have to be coordinated so that a mobile unit within a particular cell is served by the base station with the best signal-to-interference ratio. Typically, only three stations are involved in this comparison for one particular location.

Figure 7 shows the service areas of the 60-degree antennas for the cell in Fig. 5f. Cell f has large similarities to cell e. The signal-to-interference ratio is improved for some cases, though (see tables below). This is, of course, due to the increased directivity of the base stations.

### *3.1 Cochannel-interference analysis for novel antenna configurations*

Tables I to III summarize the results of some of the cochannel-interference calculations based on the assumptions presented above. For background, see Refs. 1 and 3. The cells are defined in Fig. 5. Column 1 also gives the number of base stations per cell (3/3 indicates 3 120-degree stations, 12/6 indicates 12 60-degree stations, etc.). The total number of base stations per unit area and the base-station locations are the same for all schemes (cells). For details of the calculations for cell e, i.e., 120-degree base stations in each of the six corners of the hexagon and in the center (see the appendix). Note that the signal-to-cochannel-interference ratio is not dependent on a particular modulation scheme or on a particular cell size, $r$. It is purely given by the relationships of distances from the desired base station to the desired mobile unit, and from the interfering base stations or mobile units.

It is assumed in the calculations that the background noise from other sources is negligible. The calculated signal-to-interference ratio determines which modulation method can be used (in terms of required detection efficiency) and how many branches of diversity ($M$) are required.[1,8]

Table I gives the results for various cell types for a cluster of one cell ($N = 1$), i.e., all frequency channels are used in each cell. The closest (and largest) cochannel interference comes from adjacent cells. The propagation exponent is assumed to be $\alpha = 3$. Some results for $N = 1$, $\alpha = 4$ are summarized in Table II.

Table I gives both average and worst-case signal-to-interference ratios for both mobile-to-base transmission and base-to-mobile transmission. For all cases, it is assumed that the desired mobile unit is in the least favorable location, with respect to the base stations in the cell where it is served. Worst-case mobile-to-base interference means that all mobile units in the interfering cells are in such positions that their contribution to the total interference is maximum. Average

Table I—Signal-to-cochannel interference ratio, $N = 1$, $\alpha = 3$

| Base-Station Configuration | | Mobile-to-Base | | | | | | Base-to-Mobile | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Worst Case | | Average | | | | Worst Case | | Average | |
| Cell | No./Type of Stations per Cell | Center | Corner | Center, | $M$ | Corner, | $M$ | | $M$ | | $M$ |
| a | 1 | −10 | | −6.9 | 24 | | | −4.8 | 20 | −4.8 | 20 |
| c | 3/3 | | −4.3 | | | −1.8 | 12 | −1.8 | 12 | >−1.8 | 12 |
| d | 6/3+1 | −2.8 | 2.9 | 0.3 | 8 | 5.4 | 4 | 4.4 | 4 | >4.4 | |
| e | 6/3+3/3 | 2.0 | 2.9 | 5.1 | 4 | 5.4 | 4 | 4.4 | 4 | 10.6 | 3 |
| f | 12/6+6/6 | 5.0 | 5.9 | 8.1 | 3 | 8.4 | 3 | 4.4 | 4 | >10.6 | |

* (Cell type, see Fig. 5.)

Table II—Signal-to-cochannel interference ratio, $N = 1$, $\alpha = 4$

| Base-Station Configuration | | Mobile-to-Base | | | | | | Base-to-Mobile | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Worst Case | | Average | | | | Worst Case | | Average | |
| Cell | No./Type of Stations per Cell | Center | Corner | Center, | $M$ | Corner, | $M$ | | $M$ | | $M$ |
| a | 1 | −10.5 | | | | | | −3.8 | | −3.8 | |
| c | 3/3 | | −3.6 | | | | | −0.8 | | >−0.8 | |
| d | 6/3+1 | −1.5 | 6.1 | 4.2 | | | | 8.3 | | | |
| e | 6/3+3/3 | | | 9.0 | 3 | | | 8.3 | 3 | >8.3 | |
| f | 12/6+6/6 | | | 12.0 | 2 | | | 8.3 | 3 | >8.3 | |

* (Cell type, see Fig. 5.)

mobile-to-base interference means that the desired mobile unit still is in its worst possible location, but the interfering mobile units are at equally probable locations within their respective cells. The average is formed as described in Ref. 1.

The base-to-mobile signal-to-interference ratio for the worst case occurs when the desired mobile unit is in its least favorable position with respect to any serving base station in the cell, and when the interfering base stations are those which are as close as possible to the desired mobile unit. With one omnidirectional centrally located base station, there is only one case—worst case and average case are the same. When several 120-degree antennas serve the mobile units from various directions, as in cell e, for example, the worst case is a very pessimistic assumption (see the appendix for details).

Tables I to III also contain columns for the number of diversity branches ($M$) required with space diversity and ideal maximal-ratio

Table III—Signal-to-cochannel interference ratio, $N = 3$, $\alpha = 3$

| Base-Station Configuration | | Mobile-to-Base | | | | | | Base-to-Mobile | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Worst Case | | Average | | | | Worst Case | | Average | |
| Cell | No./Type of Stations per Cell | Center | Corner | Center, | $M$ | Corner, | $M$ | | $M$ | | $M$ |
| a | 1 | 0.6 | | 2.8 | | | | | | | |
| c | 3/3 | | | | | 7.5 | 3 | 8.0 | 3 | | |
| d | 6/3+1 | | | 10.0 | | 14.7 | | 12.6 | | | |
| e | 6/3+3/3 | | | 14.8 | 2 | 14.7 | 2 | 12.6 | 2 | 17.9 | 2 |
| f | 12/6+6/6 | | | 17.8 | 2 | 17.7 | 2 | 12.6 | 2 | >17.9 | 2 |

* (Cell type, see Fig. 5.)

combining and coherent ideal Binary Phase Shift Keying (BPSK) modulation to achieve the bit error probability of $10^{-3}$. This is to give an idea of what the calculated signal-to-interference ratios mean in a cellular system.

For details about the calculations of the signal-to-interference ratios in Tables I to III, see the appendix. Cells a and c have been considered before in Refs. 1 and 3. Some of the signal-to-interference ratios from Tables I to III are from these references. From the signal-to-interference results in Table I, we note that the cells a and c give extremely low values. Many branches of space diversity are required. With cells e and f, more "reasonable" numbers of $M$ are required. Note that cell $d$ is not attractive, since the center base station is very sensitive to interference during mobile-to-base transmission.

Table III gives the signal-to-interference results for clusters consisting of $N = 3$ cells and for a propagation constant of $\alpha = 3$. The results for cells a and c are from Refs. 1 and 3. Note the significant improvements by using cells e and f. For this case, two branches of diversity ($M = 2$) are sufficient for several modulation schemes.

Table III contains results for $N = 3$, $\alpha = 3$. The corresponding results for cell c for $\alpha = 4$ are given in Refs. 1 and 3. For cells d, e, and f, these $\alpha = 4$ results will, of course, be better than the corresponding $\alpha = 3$ results in Table III. They can easily be obtained using the same technique as for the $N = 1$, $\alpha = 4$ case and for the $N = 3$, $\alpha = 3$ case.

Figure 8 shows a detailed comparison of cells e and c. The comparison in Fig. 8 is as before at equal $d_0$ and at equal base-station locations.

The base-station transmitter power is also equal for the two schemes in Fig. 8. Thus, when comparing the schemes in Fig. 8, we observe the following:

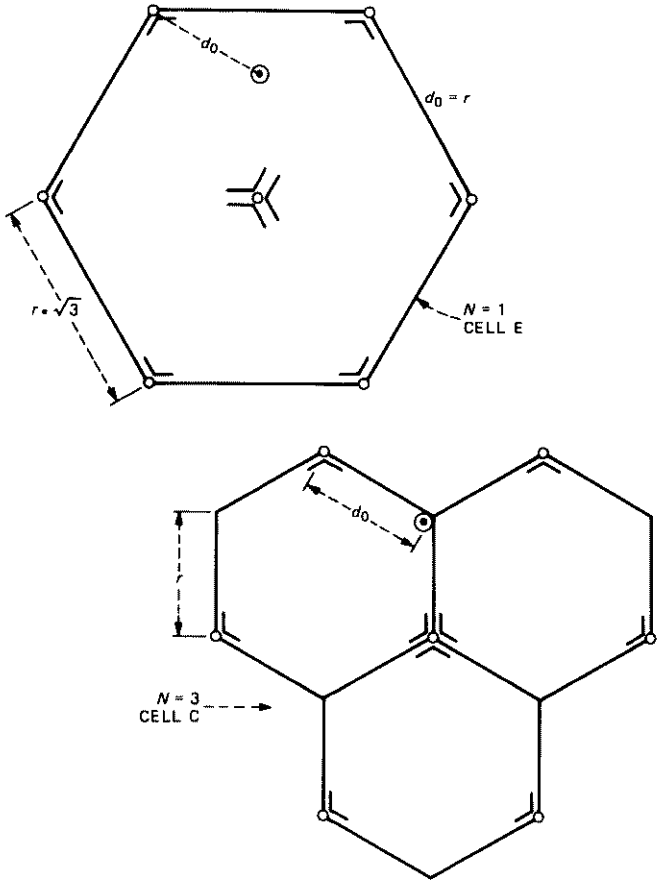



Fig. 8—Comparisons between $N = 3$, cell c radius $r$ and $N = 1$, cell e, radius $r\sqrt{3}$.

1. The number of base stations per unit area is *the same.*
2. The locations of the base stations are the same.
3. The average cochannel interference is comparable (see Table I).
4. The total number of channels served per cell ($N = 1$, cell e or f) and per cluster of cells ($N = 3$, cell c) is the same.
5. The number of channels available at any point of location is three times higher in cells e and f than in cell c. This leads to improved trunking efficiency (see Section 3.3).
6. The number of channels per 120-degree transmitter is three times larger in cells e and f than in cell c.
7. The adjacent-channel-interference problem is worse in cells e and f than in cell c.

It should also be pointed out that, at the worst-case location for

mobile-to-base transmission, there are two base stations within the same distance from the mobile unit for cell c, and sometimes three for cell e (see Fig. 8).

Note that the comparison made in Fig. 8 for $N = 1$, cells e and f and $N = 3$, cell c, can be extended to other $N$'s with similar conclusions. For example, the case with $N = 3$, cells e, and f should be compared to $N = 9$, cell c for the same number of base stations per unit area and with an increase of a factor of three of the number of locally available channels at any particular point of location (see Fig. 9). The adjacent-channel-interference problem is now less serious.

We can also compare two systems based on different cells with the same number of cells ($N$) per cluster. As an example, consider $N = 3$, cell c and e (or f) with the same number of base stations per unit area and the same base-station locations. Table III gives cochannel-interference results. For this situation we can conclude that:

1. The number of base stations per unit area is the same (same locations).
2. The average cochannel interference is better with cells e and f than with c.
3. The total number of channels served per cell is the same. The total number of channels per three cells of type c is three times that of the number of channels available in one cell of type e, f.
4. The total number of channels available at any specific location is the same in the two cases.
5. The number of channels per 120-degree transmitter is the same.
6. The adjacent-channel-interference problems are worse for cell c than e, f.

### *3.2 Transmitter power weighting*

All the transmitters in systems based on cells a, b, and c (see Fig. 5) have the same properties. Except for the edges in the whole cellular system, the interference situation is the same around each transmitter. All transmitters above are assumed to be transmitting at the same power levels.

The situation is different for cells of type e, f. The center transmitters clearly have a different environment than the corner transmitters. It is not immediately clear why these transmitter power levels should be the same, as we assumed in the analysis above. On the contrary, there are possible improvements in base-to-mobile cochannel interference through reduction of the transmitter power levels for the center stations compared to the corner stations. The consequences of this are that the worst-case location for base-to-mobile and mobile-to-base transmission cochannel interference might be different and that the

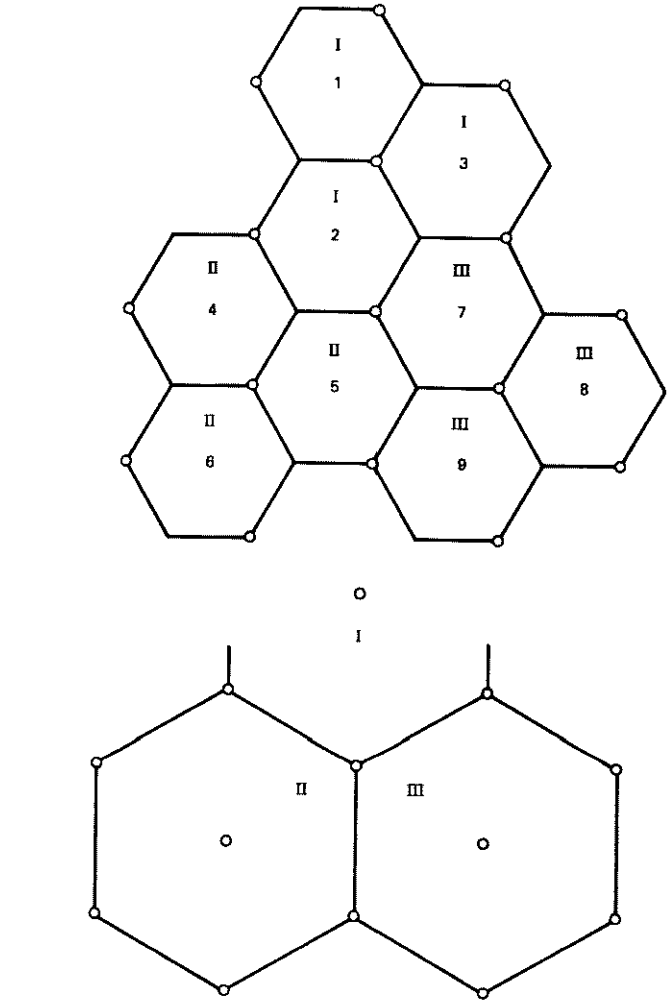



Fig. 9—Comparison between a conventional cellular system with $N = 9$ cells in a cluster and a system based on cell e with $N = 3$ cells in a cluster.

base-to-mobile interference can be improved, for both worst case and average. Mobile-to-base transmission is not affected.

We will now give two examples of what the weighting should be for improving the worst-case components among all the contributions to cochannel interference for the base-to-mobile transmission.

Figure 10 shows the $N = 1$ case with cell e. Assume for simplicity that the radius is one. Assume that the mobile unit is in position M. Furthermore, assume that the worst-case cochannel-interference components will occur somewhere along a straight line from the center

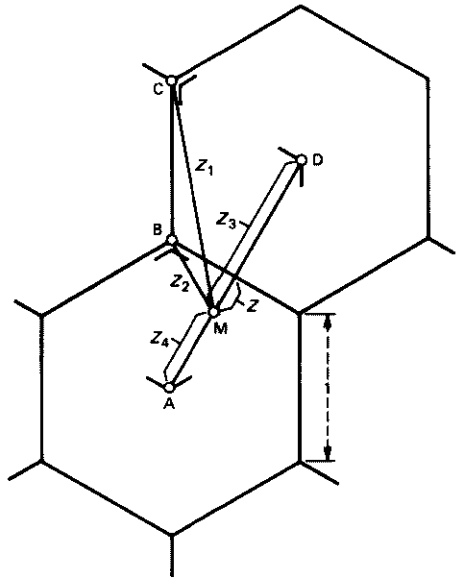



Fig. 10—Notations for analysis of worst-case cochannel interference for cell e, $N = 1$ with transmitter-power weighting.

stations A and D (due to symmetry). The two closest interfering stations are D and C and the two closest base stations serving the mobile unit $M$ are A and B.

Assume that the transmitter power of the corner transmitters is $P$. The center station has the transmitter power $\beta P$, where $\beta$ is the weighting factor. The dominating components in the cochannel-interference ratio are:

1. For a mobile unit served from corner station B, the signal-to-cochannel-interference ratio is

$$\left(\frac{Z_1}{Z_2}\right)^{\alpha} = \left[\frac{1 + \left(\frac{\sqrt{3}}{2} + Z\right)^2}{\frac{1}{4} + Z^2}\right]^{\alpha/2}, \tag{3}$$

when the interferer is station C, and is

$$\frac{1}{\beta}\left(\frac{Z_3}{Z_2}\right)^{\alpha} = \frac{1}{\beta}\left[\frac{\left(\frac{\sqrt{3}}{2} + Z\right)^2}{\frac{1}{4} + Z^2}\right]^{\alpha/2}, \tag{4}$$

when the interferer is station D. $Z$ is the distance from the mobile unit to the cell boundary (see Fig. 10).

2. For a mobile unit served from the center station A, the signal-to-cochannel-interference ratio is

$$\beta\left(\frac{Z_1}{Z_4}\right)^{\alpha} = \beta\left[\frac{1+\left(\frac{\sqrt{3}}{2}+Z\right)^2}{\left(\frac{\sqrt{3}}{2}-Z\right)^2}\right]^{\alpha/2}, \tag{5}$$

when the interferer is station C, and is

$$\left(\frac{Z_3}{Z_4}\right)^{\alpha} = \left[\frac{\left(\frac{\sqrt{3}}{2}+Z\right)^2}{\left(\frac{\sqrt{3}}{2}-Z\right)^2}\right]^{\alpha/2}, \tag{6}$$

when the interferer is station D.

What is the best choice for $\beta$? Select $\beta$ so that the worst-case location ($M$) has as large a signal-to-interference ratio as possible. This optimization depends on the propagation exponent, $\alpha$, i.e., $\beta$ is a function of $\alpha$. We found that the best weighting parameter is $\beta = 0.23$, with $Z = 0.416$ for $\alpha = 4$.

This should be compared with the nonweighted case where $\beta = 1$. Here the worst-case term for the cochannel-interference occurs for $Z = 0$. Thus, the improvement of the worst-case cochannel-interference component with weighting is 6.4 dB for $\alpha = 4$.

It can also be expected that the average base-to-mobile cochannel interference is improved significantly by means of weighting. The worst contributors to the noise are the center base stations, and they are now reduced in power. The optimum $\beta$ with this criterion might differ from the optimum $\beta$ derived above.

More terms than the worst one, of course, have to be taken into account in a complete analysis. There seems to be room for signficant improvements by means of weighting, however.

Transmitter-power weighting can also be employed for cell e, Fig. 5, with $N = 3$ cells in a cluster. The definitions of locations and parameters are equivalent to the above (see Fig. 11). We assume that the worst-case mobile location for the base-to-mobile cochannel-interference contribution occurs on the straight line from transmitter location A to E (see Fig. 11).

Using the same technique as in the appendix and above, we have the following worst-case cochannel-interference components:

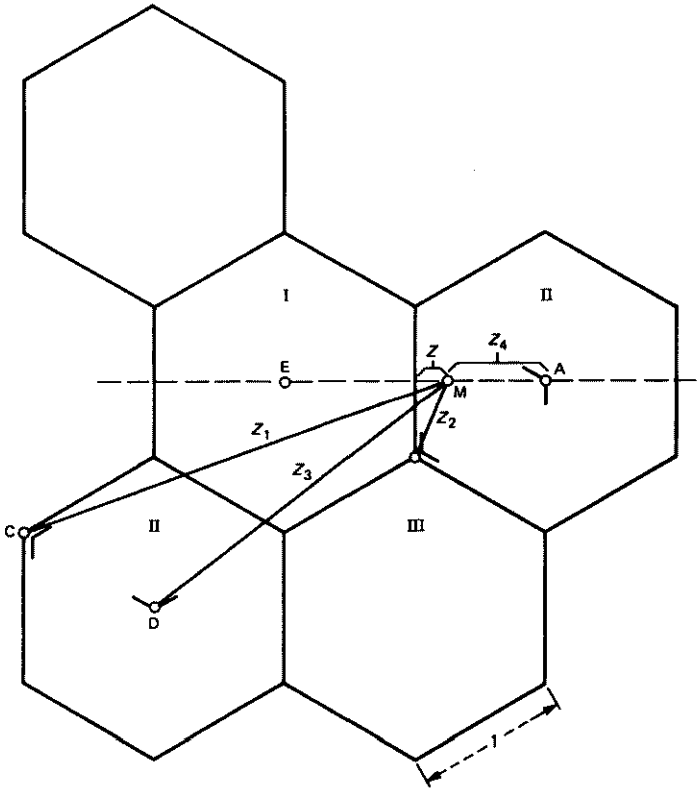



Fig. 11—Same as Fig. 10 with $N = 3$.

1. For a mobile unit served from corner station B, the signal-to-cochannel-interference ratio is

$$\left(\frac{Z_1}{Z_2}\right)^{\alpha} = \left[\frac{1 + \left(\frac{3\sqrt{3}}{2} + Z\right)^2}{\frac{1}{4} + Z^2}\right]^{\alpha/2}, \tag{7}$$

when the interferer is station C, and is

$$\frac{1}{\beta}\left(\frac{Z_3}{Z_2}\right)^{\alpha} = \left[\frac{\left(\frac{3}{2}\right)^2 + (\sqrt{3} + Z)^2}{\frac{1}{4} + Z^2}\right]^{\alpha/2}, \tag{8}$$

when the interferer is station D.

2. For a mobile unit served from the center station A, the signal-to-cochannel-interference ratio is

$$\beta\left(\frac{Z_1}{Z_4}\right)^{\alpha} = \left[\frac{1 + \left(\frac{3\sqrt{3}}{2} + Z\right)^2}{\left(\frac{\sqrt{3}}{2} - Z\right)^2}\right]^{\alpha/2}, \tag{9}$$

when the interferer is station C, and is

$$\left(\frac{Z_3}{Z_4}\right)^{\alpha} = \left[\frac{\left(\frac{3}{2}\right)^2 + (\sqrt{3} + Z)^2}{\left(\frac{\sqrt{3}}{2} - Z\right)^2}\right]^{\alpha/2}, \tag{10}$$

when the interferer is station D. As in the previous example, the transmitter power of all center stations is $\beta P$, where $P$ is the transmitter power of the corner stations. The parameter $\beta$ is the weighting.

The minimum worst-case cochannel-interference components [eqs. (7) to (10)] occur for the parameters in Table IV. This table shows optimum weighting for minimizing the worst-case contribution to the signal-to-cochannel interference ratio for $N = 3$, cell e.

The gain in the table above is the improvement of the worst-case cochannel-interference contribution with optimum weighting compared with no weighting. For the no-weighting case ($\beta = 1$), the worst case is $Z = 1/2\sqrt{3} \approx 0.289$.

The average cochannel interference will also be improved for the $N = 3$ case with proper weighting. The optimum $\beta$ has to be found, however. The optimization above was only carried out for the worst-case components.

The extreme case of center base-station power weighting is $\beta = 0$. For this case, there are no center stations at all. This cell is shown in Fig. 12. One third of the base stations can be saved. However, the cochannel interference increases compared to cell e, Fig. 5, particularly mobile-to-base. The mobile transmitter power must also be increased

Table IV—Optimum weighting

| $\alpha$ | $\beta$ | $Z$ | Gain |
|---|---|---|---|
| 4 | 0.276 | 0.401 | 5.6 dB |
| 3.7 | 0.303 | 0.401 | 5.2 dB |
| 3 | 0.381 | 0.401 | 4.2 dB |

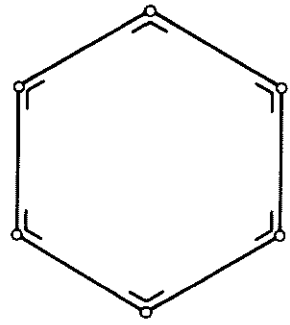


Fig. 12—A cell with one-third fewer base stations than cell e, Fig. 5.

somewhat. The distance to the nearest transmitter is significantly increased for the cell in Fig. 12 compared with cell e.

### *3.3 Trunking efficiency*

We saw above in Section 3.1 that the local availability of channels with cell e is three times that with a cluster of three conventional cells of type c. This three-fold increase in availability of channels leads to improved trunking efficiency with cell e. To illustrate this gain, we give the following example.

Compare a system based on the "superhexagon" cell e with a total of 45 available channels and a system based on the conventional cell c. In the latter case, we assume that 15 channels are available at each station. Thus, in a cluster of 3 cells, type c, there are a total of 45 channels, as with cell e. However, with cell e, the 45 channels are available at all locations throughout the cell e, while with the cluster of 3 cells of type c, 15 channels are available in each cell.

The traffic behavior of Mobile Radio-Telephone Systems might be modeled by Erlang C tables. (The calls are delayed, but eventually placed. The calls are not rerouted over alternate facilities, nor do they go away.)

Assume a blocking probability of two percent using Erlang C tables; the traffic carried by the system in cell e is 32.03 erlangs, while the system with cell c carries 8.03 erlangs in each cell, or 24.1 erlangs in a cluster of three cells. Thus, the system based on cell e carries 7.93 erlangs more per area corresponding to one cell e (3 cells of type c) than the conventional system.

Instead, using Erlang B tables, the traffic carried by the system with cell e is 35.61 erlangs, while with cell c, the traffic in each cell is 9.01 erlangs and in a cluster of three cells about 27.0 erlangs. Thus, the system based on cell e carries 8.6 erlangs more per area corresponding to one cell e (3 cells of type c) than the conventional system.

The best model is perhaps a modified Erlang B—something between Erlang B and C. Some percentage of the calls may never be made or may be rerouted. In any case, the basic conclusion is the same: A significant improvement of the trunking efficiency is achieved with cell e.

### *3.4 Relationship with dynamic channel assignment schemes*

In the example above, it is evident that the trunking efficiency is improved by using cell e instead of a cluster of three cells of type c. The increased local availability of channels should also improve the capability of matching nonuniform geographic traffic patterns.

With conventional cells, this problem is dealt with by using dynamic channel assignment schemes.[4,9] The novel cell types, e.g., cell e, are somewhat like this. The control algorithm is fixed with cell e, however. Further work, especially simulations, is required for evaluating the above relationship.

## IV. DISCUSSION AND CONCLUSIONS

Novel antenna configurations for cellular digital radio systems are introduced and analyzed in this paper. Two key ideas are presented. The first is designing new cells by keeping the base-station locations in conventional cellular systems and rearranging the antenna patterns. Basically, merging conventional cells into larger, novel cells improves trunking efficiency. The local availability of channels is increased. The second idea is transmitter-power weighting. The base-station transmitters in the novel cells have different roles depending on the exact location. The center stations and the corner stations contribute differently to the cochannel interference. Thus, the signal-to-cochannel interference ratio can be improved by proper weighting of the transmitter-power levels.

Many unsolved problems remain. The analysis above (as that in Refs. 1 and 3) is based on idealized assumptions about the selection of the serving base station and on very simple and idealized channel and interference models. More refined analysis and simulations will be necessary. The analysis above was carried out under the idealized assumptions of flat fading. Uniform transmission conditions were assumed for all cells. No delay spread was considered. Perfect timing and synchronization was assumed with coherent detection and ideal maximal-ratio combining. It was furthermore assumed that perfect synchronization for the time-division retransmission scheme was established. The analysis was confined to local-mean values of signal and interference at isolated points. Consequently, the effects of shadow fading were not taken into account, and no results were obtained for overall signal-to-interference statistics throughout entire cellular

areas. The effect of channel occupancy (the fraction of time that a channel is in use) on interference was not considered. All of the above problems and others have to be taken into account in a refined system analysis.

Some of the aspects of choosing a modulation scheme for digital cellular mobile radio systems are dealt with in Refs. 6 and 8. Adjacent-channel interference in cellular systems is calculated in Ref. 6 for some bandwidth-efficient constant amplitude modulation schemes.

Further work should be devoted to the traffic-carrying aspects of cellular systems based on the novel cells. Detailed comparisons with dynamic channel assignment schemes with conventional (and novel) cells should be made.

## V. ACKNOWLEDGMENTS

Thanks are due to the reviewers, who pointed out the trunking efficiency aspects of the novel cells and the relationship with dynamic channel assignment.

## REFERENCES

1. Y. S. Yeh and D. O. Reudink, "Efficient Spectrum Utilization for Mobile Radio Systems Using Space Diversity," IEEE Trans. Commun., *COM-30*, No. 3 (March 1982), pp. 447–55.
2. V. H. MacDonald, "The Cellular Concept," B.S.T.J., *58*, No. 1 (January 1979), pp. 15–41.
3. P. S. Henry and B. S. Glance, "A New Approach to High Capacity Digital Mobile Radio," B.S.T.J., *60*, No. 8 (October 1981), pp. 1891–904.
4. W. C. Jakes, Jr., *Microwave Mobile Communications*, New York: Wiley, 1974.
5. M. Schwartz, W. R. Bennett, and S. Stein, *Communication Systems and Techniques*, New York: McGraw-Hill, 1966.
6. C.-E. Sundberg, "On Continuous Phase Modulation in Cellular Digital Mobile Radio Systems," B.S.T.J., this issue.
7. D. C. Cox, "Cochannel Interference Considerations in Frequency Reuse Small-Coverage-Area Radio Systems," IEEE Trans. Commun., *COM-30*, No. 1 (January 1982), pp. 135–42.
8. C.-E. Sundberg, "Error Probability of Partial Response Continuous Phase Modulation with Coherent MSK-type Receiver Diversity and Slow Rayleigh Fading in Gaussian Noise," B.S.T.J., *61*, No. 8 (October 1982), pp. 1933–63.
9. T. J. Kahwa and N. D. Georganas, "A Hybrid Channel Assignment Scheme in Large-Scale, Cellular-Structured Mobile Communication Systems," IEEE Trans. Commun., *COM-26*, No. 4 (April 1978), pp. 432–8.

## APPENDIX

### *Details of Cochannel-Interference Calculations*

This appendix contains some details about the calculations of the signal-to-cochannel-interference ratios given in Tables I to III in Section 3.1 above.

Figure 13 shows the worst-case location for the desired mobile unit for the case of transmission from base to mobile with $N = 1$, $\alpha = 3$, and 120-degree base stations in all corners and in the center (cell e).

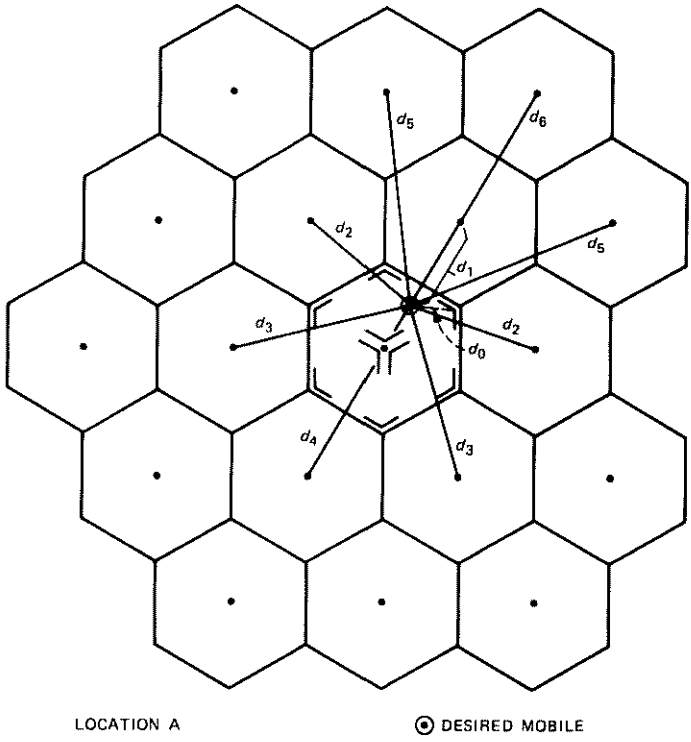



Fig. 13—Worst-case location for the desired mobile unit for worst-case base-to-mobile transmission $N = 1$, $\alpha = 3$, cell e. This location of the desired mobile unit is referred to as location A.

Assume that the base stations in nearby cells are serving mobile units in interfering frequency slots in such a way that the base station in a particular cell is as close as possible to the desired mobile unit, thus contributing maximally to the cochannel interference. Thus, the worst-case interference power to signal power is given by the distance relationships[1,3]

$$\frac{P_I}{P_S} = \left(\frac{d_0}{d_1}\right)^3 + 2\left(\frac{d_0}{d_2}\right)^3 + 2\left(\frac{d_0}{d_3}\right)^3 + 2\left(\frac{d_0}{d_5}\right)^3 + \left(\frac{d_0}{d_4}\right)^3 + \left(\frac{d_0}{d_6}\right)^3 + \text{terms from cells further away.} \quad (11)$$

The worst-case position above is given by the location where the mobile unit is as far away as possible from the nearest base station in the cell. Thus $d_0 = r_0\sqrt{3}$, where $r_0$ is the cell radius (of cell e) (see Fig. 8).

The dominating term in eq. (11) is

$$\left(\frac{d_0}{d_1}\right)^3 = \left(\frac{r_0/\sqrt{3}}{2r_0/\sqrt{3}}\right)^3 = \left(\frac{1}{2}\right)^3. \tag{12}$$

Compare the dominating term in the $N = 1$ schemes with centrally located base stations or corner stations only in Refs. 1 and 3. In this case, this term is 1, independent of $\alpha$. For the above case, the cochannel-interference noise suppression improves with the propagation constant $\alpha$.

Taking into account contributing terms from the 23 interfering cells closest to the cell with the desired mobile unit we have (see Table I)

$$\frac{P_S}{P_I} \approx 4.4 \text{ dB}. \tag{13}$$

This is the signal-to-interference ratio for base-to-mobile transmission with the worst-case location of the desired mobile unit. All the interfering base stations are assumed to be in a worst-case mode of operation too. The propagation exponent is $\alpha = 3$, and the frequency plan is such that all cells use the total number of frequency slots, $N = 1$. We have assumed that each cell has 120-degree base stations in each corner and in the center, cell e.

Figure 14 shows the location of the desired mobile unit in the cell where the worst-case relationship is between the distance to the nearest transmitter in the cell and the distance to the nearest interfering base station in the adjacent cells. In this case, the interference-to-signal ratio is

$$\frac{P_I}{P_S} = \left(\frac{d_0}{d_1}\right)^3 + 2\left(\frac{d_0}{d_2}\right)^3 + 4\left(\frac{d_0}{d_3}\right)^3 + 2\left(\frac{d_0}{d_4}\right)^3 + \text{terms from interfering cells further away.} \tag{14}$$

The dominating term is now given by $d_0 = r_0/2$, $d_1 = r_0\sqrt{3}/2$, thus

$$\left(\frac{d_0}{d_1}\right)^3 = \left(\frac{1}{\sqrt{3}}\right)^3. \tag{15}$$

This term is, of course, larger than eq. (12). However, the other terms in eq. (14) are smaller than their counterparts in eq. (11) because $d_0$ is smaller in Fig. 14 than in Fig. 13. Thus, it is not immediately evident which location, A or B, is generally the worst-case location for the desired mobile unit.

Carrying out the calculations in eq. (14) with the same number of terms as in eq. (11), we have the signal-to-distortion ratio

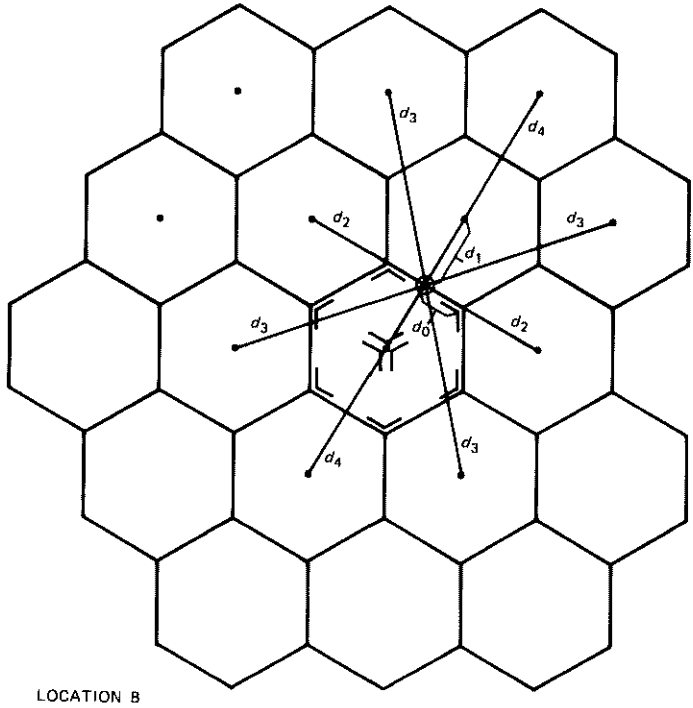



Fig. 14—Location of the desired mobile unit where $(d_0/d_1)$ is maximum. Otherwise, same case as Fig. 13. This location of the desired mobile unit is referred to as location B.

$$\frac{P_S}{P_I} \cong 4.5 \text{ dB} \tag{16}$$

for the worst case in Fig. 14, for $\alpha = 3$. Thus, the location in Fig. 13 is slightly worse for the $N = 1$, $\alpha = 3$ case.

It is even conceivable that some location between that in Figs. 13 and 14 is the worst case for $\alpha = 3$. We do not expect more than a very small (if any) deviation from eq. (13), however.

We have also carried out the calculations of the base-to-mobile worst-case signal-to-cochannel-interference ratio for the cases in Figs. 13 and 14 for the propagation exponent $\alpha = 4$. For this case, the dominating term for location A is

$$\left(\frac{d_0}{d_1}\right)^4 = \left(\frac{1}{2}\right)^4, \tag{17}$$

and for location B,

$$\left(\frac{d_0}{d_1}\right)^4 = \left(\frac{1}{\sqrt{3}}\right)^4. \tag{18}$$

Carrying out the calculation of eqs. (11) and (14) for $\alpha = 4$ for the locations A and B we have, for $\alpha = 4$,

$$\frac{P_S}{P_I} \approx 9.2 \text{ dB location A,}$$

$$\frac{P_S}{P_I} = 8.3 \text{ dB location B.}$$

Thus, in this case ($\alpha = 4$), location B is the worst case. This is what might be expected from the formulas above. For large $\alpha$'s, the first term [in generalizations of eqs. (11) and (14)] will play an increasing role.

Above, we have calculated the signal-to-interference ratio for the worst-case base-to-mobile transmission where we assumed worst-case conditions, i.e., that the desired mobile unit is in the worst location and that the mobile units in neighboring cells are all in such positions that the interference from all cells is maximum. This is, of course, a very pessimistic assumption. Next, we will calculate the average interference power when it is assumed that all positions of a particular mobile unit in a cell are equally probable. Thus, the probability that a mobile unit is served by a particular 120-degree base station is 1/5 in configuration e (see Fig. 5e). Note that it is still assumed that the desired mobile unit is in its worst location. Other positions for the desired mobile unit will yield better signal-to-interference ratios. With the assumptions above, we have the contributions to the average interference-to-signal power ratio for the closest cell,

$$\frac{P_I}{P_S} = \frac{1}{9}\left(\frac{d_0}{d_1}\right)^3 + \text{other smaller terms within the cell}$$

$$+ \text{ terms from other interfering cells.} \tag{19}$$

Note that the contribution from each base station is scaled with either 1/9, when it is pointed towards the desired mobile unit, or 0, when it is pointed in a different direction. The worst-case term $(d_0/d_1)^3$ is now scaled down with a factor of 1/9. Continuing the calculation in eq. (19) and including all the cells included in the calculation of eq. (11), we have the approximate average signal-to-cochannel-distortion ratio with the desired mobile unit in the worst location A for the $\alpha = 3$, $N = 1$ case with 120-degree base stations in all corners and in the center of the cell,

$$\frac{P_S}{P_I} \approx 10.6 \text{ dB.} \tag{20}$$

This is significantly better than the worst case [eq. (13)] of 4.4 dB for the worst-case location for the desired mobile unit. The large improvement is, of course, due to the fact that the surrounding interfering cells only use the interfering antennas that are closest to the desired mobile unit part of the time. Sometimes, there is no interference from a particular cell, because the mobile unit served on the same frequency channel in that particular cell is served by an antenna that is not pointed in the direction of the desired mobile unit.

Figure 15 shows the worst-case mobile-to-base cochannel interference for the 120-degree corner base station in a cell with 120-degree base stations in all corners and in the center. The desired mobile unit is assumed to be in its worst position, A, on distance $d_0 = r\sqrt{3}$ from the serving base station (see Fig. 13). Mobile units on interfering frequency channels are all assumed to be in the least favorable position in their cells (see Fig. 15). With straightforward calculations like those earlier in this appendix, we arrive at the signal-to-interference ratio of approximately 2.9 dB for $\alpha = 3$ (see Table I).

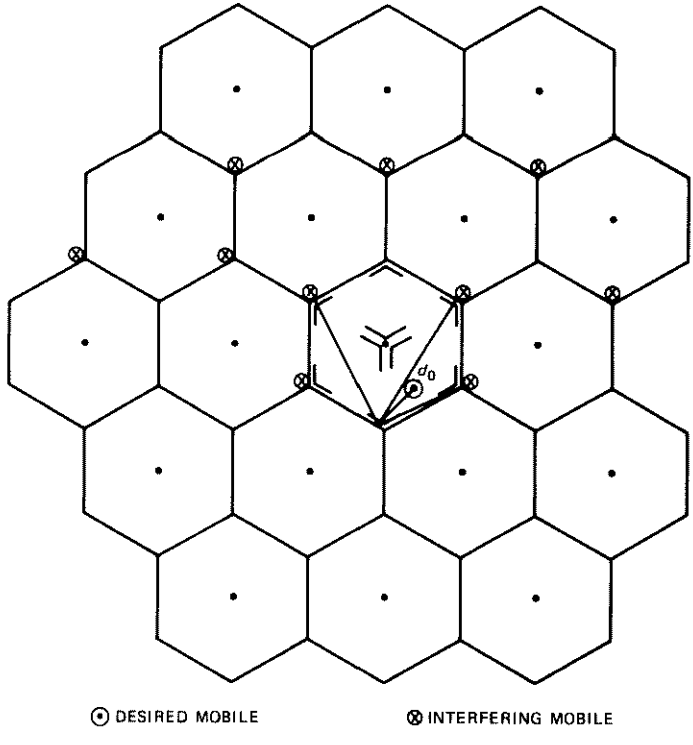



Fig. 15—$N = 1$, $\alpha = 3$, cell e, desired mobile unit in location A. Worst-case mobile-to-base signal-to-interference ratio for a corner station.

The center base station is more sensitive to worst-case mobile-to-base cochannel interference that the corner base stations in Fig. 15. Figure 16 shows this. The worst locations for interfering mobile units are closer to the base station serving the desired mobile unit in this case. The signal-to-interference ratio for this worst case is approximately 2.0 dB.

It is clear from Fig. 16 that an ominidirectional center antenna is very sensitive to mobile-to-base cochannel interference. The lack of directivity forces it to receive interference from all directions, while the desired mobile unit is in a particular direction. Thus, it is advantageous to use 120-degree antennas rather than omnidirectional antennas. Further improvements are obtained by using 60-degree antennas (see Fig. 7 and Table I).

The worst-case base-to-mobile cochannel interference is the same for omnidirectional, 120- or 60-degree center antennas (see Table I).

The assumption that all interfering mobile units are in their worst positions in their respective cells is, of course, a very pessimistic one. It is more realistic to calculate an average interference, where the mobile units are assumed, with equal probability, to be in any position

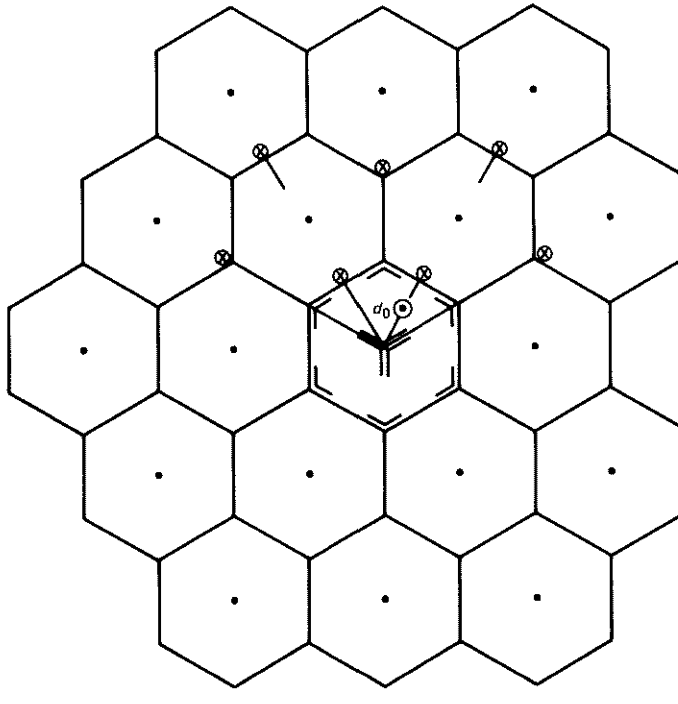



Fig. 16—Same as Fig. 15 for a center station.

within a cell. Using the averaging technique, described in detail in Ref. 1, we obtain the average signal-to-interference ratios for mobile-to-base transmission in Tables I to III in Section 3.1.

By using 60-degree antennas, the interference above is reduced by a factor of two. Since the center base station is the most sensitive one, it is, of course, conceivable with a hybrid scheme with 120-degree antennas in all the corners and 60-degree antennas in the center locations (see Fig. 5g).

The cochannel-interference calculations behind the remaining signal-to-cochannel-interferences ratios in Tables I to III above are carried out with the same techniques as above. The averaging technique for the mobile-to-base cochannel interference in Ref. 1 is easily extended to the propagation exponent $\alpha = 4$. Location A is the worst case for $N = 3$, $\alpha = 3$ with cell e.

## AUTHOR

**Carl-Erik W. Sundberg,** M.S.E.E and Dr. Technology, Lund Institute of Technology, University of Lund, Lund, Sweden, in 1966 and 1975, respectively. Currently, Mr. Sundberg is a Research Professor in the Department of Telecommunication Theory, University of Lund. He is also Director of the consulting company SUNCOM, Lund. During 1976 he was with the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands, as an ESA Research Fellow. He has held positions as Consulting Scientist at LM Ericsson and SAAB-SCANIA, Sweden, and at Bell Telephone Laboratories, Crawford Hill, N. J. His research interests include source coding, channel coding, especially decoding techniques, digital modulation methods, fault-tolerant systems, digital mobile radio systems, spread spectrum systems, and digital satellite communication systems. He has published a large number of papers in these areas during the last couple of years. Dr. Sundberg holds several U.S. and European patents. Senior Member, IEEE; member, SER (Sveriges Elektroingenjörers Riksförening).

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# On Continuous Phase Modulation in Cellular Digital Mobile Radio Systems

By C.-E. SUNDBERG*

(Manuscript received January 19, 1982)

This paper considers the application of constant-amplitude, partial-response, continuous-phase modulation with simple near-optimum receivers to cellular digital mobile radio systems. These smoothed modulation schemes have low spectral sidelobes and narrow main lobes. A combiner for time-division retransmission systems with space diversity is given for continuous-phase modulation. Cochannel interference and adjacent-channel interference are calculated for continuous-phase modulation in cellular systems with frequency reuse. The efficiency of 3RC and 4RC modulations with space diversity is evaluated for conventional cellular systems.

## I. INTRODUCTION

This paper considers a class of constant-amplitude digital modulation schemes with smoothed partial-response, continuous-phase modulation. This gives a bandwidth-efficient digital modulation scheme. With the baseband quadrature combiner described below, these modulation schemes can also be used with time-division retransmission. The concept of time-division retransmission is briefly described below. This scheme makes it possible to use simple equipment at the mobile unit, and yet use several branches of space diversity, assuring a reasonable error rate on the fading channel. Most of the signal processing and all diversity combining is performed at the base stations.

To obtain good spectral efficiency in digital mobile radio systems,

* Bell Laboratories. Currently at the University of Lund, Sweden.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

frequency reuse is employed.[1-4] Each cell in the cellular system is assigned a number of channels in a frequency division system. Each channel (frequency) is reused at a cell further away. The closer this cell is, the larger the number of available channels. On the other hand, the signal-to-cochannel interference increases when the interfering cells are too close. The interference is also affected by the way the base-station antennas are arranged. Omnidirectional and directional antennas have been considered. Furthermore, the location of the base station in the cell is important for the signal-to-interference ratio. In this paper we consider two schemes: Centrally located base stations with omnidirectional antennas, and cooperating base stations with 120-degree directional antennas in three alternate corners in a hexagonal cell. Below, we will see how cochannel and adjacent-channel interference is calculated for continuous-phase modulation in a cellular environment.

The rest of this paper is organized as follows. Section II contains background material on modulation methods, conventional cellular systems, time-division retransmission, and propagation and interference models used in the analysis. Section III presents a baseband maximal-ratio combiner for time-division retransmission with constant-amplitude modulations. Section 4.1 contains the calculations of signal-to-cochannel interference ratios for cellular systems with continuous-phase modulation. Section 4.2 gives some results on adjacent-channel interference. Section V contains a discussion and conclusions.

## II. BACKGROUND MATERIAL

Before the cellular systems are discussed in any detail, we will give some brief background information about the class of constant-amplitude modulation systems considered in this paper, conventional cellular arrangement, the time-division retransmission method, and signal propagation and interference models for the fading land mobile radio channel.

### *2.1 Constant-amplitude modulations*

Constant-amplitude modulation schemes are considered for application to cellular digital mobile radio systems. The transmitted signal is of the form

$$s(t) = \sqrt{\frac{2E}{T}} \cos[\omega_c t + \phi(t)], \tag{1}$$

where the information-carrying phase is

$$\phi(t) = 2\pi h \sum_i \alpha_i q(t - iT), \tag{2}$$

with

$$q(t) = \int_0^t g(\tau)d\tau. \tag{3}$$

The pulse shape $g(t)$ (instantaneous frequency) determines the spectral behavior of the modulation scheme, while $h$ is the modulation index, and the $\alpha_i$'s are data symbols. For detailed background on these schemes, see Refs. 5–11. In this paper, we will mainly concern ourselves with schemes having modulation index $h = 1/2$ and with $g(t)$ in the form of raised cosine pulses of length $LT$, where $T$ is the bit time. This pulse shape is defined by

$$g(t) = \begin{cases} \dfrac{1}{2LT}\left[1 - \cos\left(\dfrac{2\pi t}{T}\right)\right] & 0 \le t \le LT \\ 0 & t \le 0,\ t \ge LT. \end{cases} \tag{4}$$

For $L > 1$, the pulses are overlapping in time, thus creating controlled intersymbol interference. In this paper, only binary transmission $\alpha_i = \pm 1$ will be considered. The reason is that binary schemes with modulation index $h = 1/2$ have simple, near-optimum receivers, i.e., "minimum shift keying (MSK)-type" receivers (see Refs. 7–9).

Figures 1 to 3 show examples of power spectra of (unfiltered)

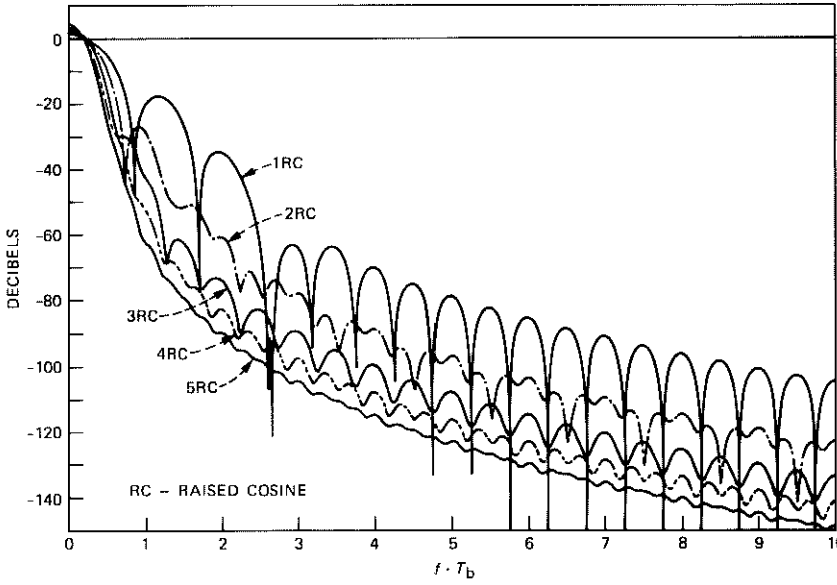



Fig. 1—Power spectra for binary RC schemes when $h = 1/2$, $L = 1$ to 5. For details, see Refs. 5, 6, and 11. $T_b$ is bit time (symbol time). (Spectra are shown one-sided normalized with total power.)

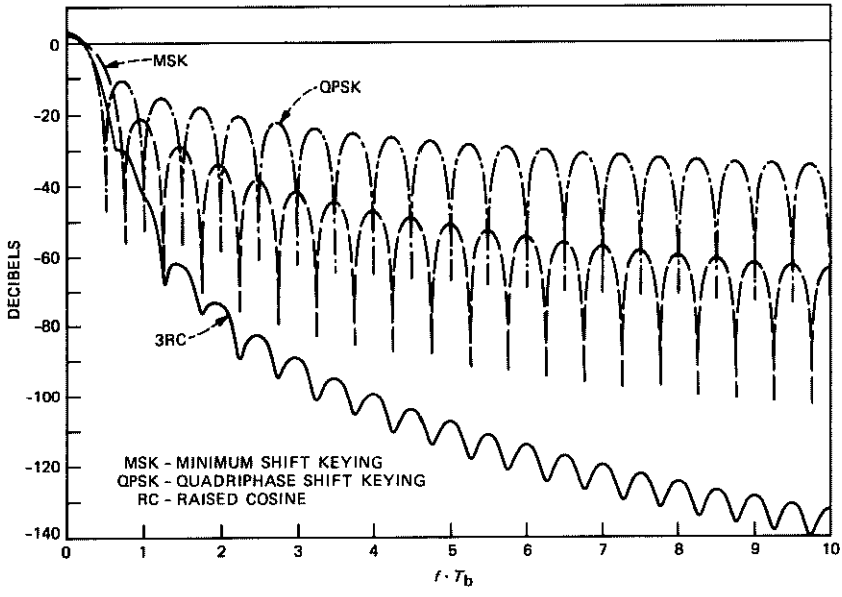



Fig. 2—Power spectra for binary 3RC with $h = 1/2$, QPSK, and MSK.

constant-amplitude modulation schemes in the raised cosine family. A scheme with a pulse shape $g(t)$ of a raised cosine shape with length $3T$ is denoted by 3RC (raised cosine). Minimum shift keying,[9] tamed frequency modulation (TFM),[8] and Gaussian MSK (GMSK)[10] are obtained from eqs. (1) and (2) by selecting proper pulses $g(t)$ (see Refs. 5 to 7 for details).

Even better spectral tail behavior is obtained with spectral raised cosine (SRC) pulse shapes instead of raised cosine pulse shapes.[5,6] These schemes are more difficult to analyze exactly, however. A 3SRC pulse $g(t)$ has a length of $3T$ between the first nulls in time around the peak value. The Fourier Transform is a raised cosine. Thus, the time function is infinitely long. In practice, these pulses are truncated.

Figures 4 and 5 show examples of error probability curves for coherent binary phase shift keying (BPSK), quadriphase shift keying (QPSK), 3RC and 4RC for slow Rayleigh fading, space diversity with $M$ branches, and ideal maximal-ratio combining. These are examples of calculations in Refs. 7 and 12. The analysis for the Gaussian channel is presented in Ref. 7, using analytical tools from Refs. 13 and 14. The formulas for the Gaussian channel are generalized to the fading channel in Ref. 12. Note that the error probability curves in Figs. 4 and 5 are shown without differential encoding/decoding. In practice, differential encoding is employed as the simplest way of resolving phase ambiguity in the coherent MSK-type receiver for $h = 1/2$

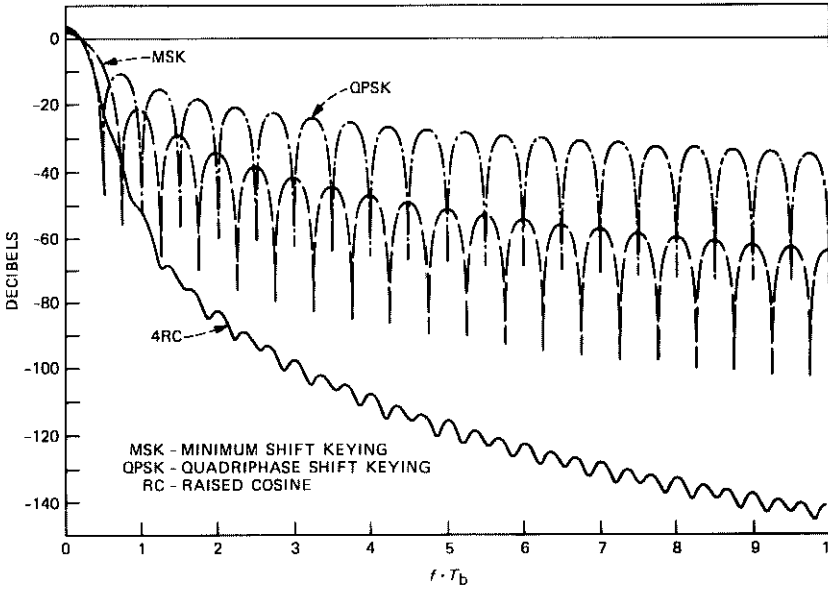



Fig. 3—Power spectra for binary 4RC with $h = 1/2$, QPSK, and MSK.

schemes.[7,8,9,15] For this case, all error probability values in Figs. 4 and 5 are multiplied by a factor of two.

The longer the pulse $g(t)$, the more narrow the power spectrum.[6] However, the detection efficiency is reduced with increasing $L$ for fixed $h = 1/2$, particularly for suboptimum receivers.[7] The optimum receiver (the Viterbi detector) is more robust.[6] Thus, as we will see, choosing the length of the pulse shape $g(t)$ plays an important part in the overall system optimization.

Power amplifiers, both in the mobile unit and at the base stations, sometimes contain nonlinearities. For such cases, it is advantageous to use a constant envelope modulation scheme. Such a scheme will not be subject to widened spectra after the nonlinearity. This is the case for nonconstant amplitude modulations. A QPSK signal passed through a bandpass filter can be quite narrowband. The signal amplitude varies, however, and after a nonlinearity, the sidelobes are restored to a level that is unacceptable from an adjacent-channel interference point of view.

### *2.2 Conventional cellular arrangement*

This paper will consider frequency reuse in cellular systems for digital mobile radio systems. Figure 6a shows an example of a cellular system with $N = 3$ hexagonal cells per cluster. Frequencies are assigned as in Fig. 6b. Note that for a fixed total number of channels, a fixed

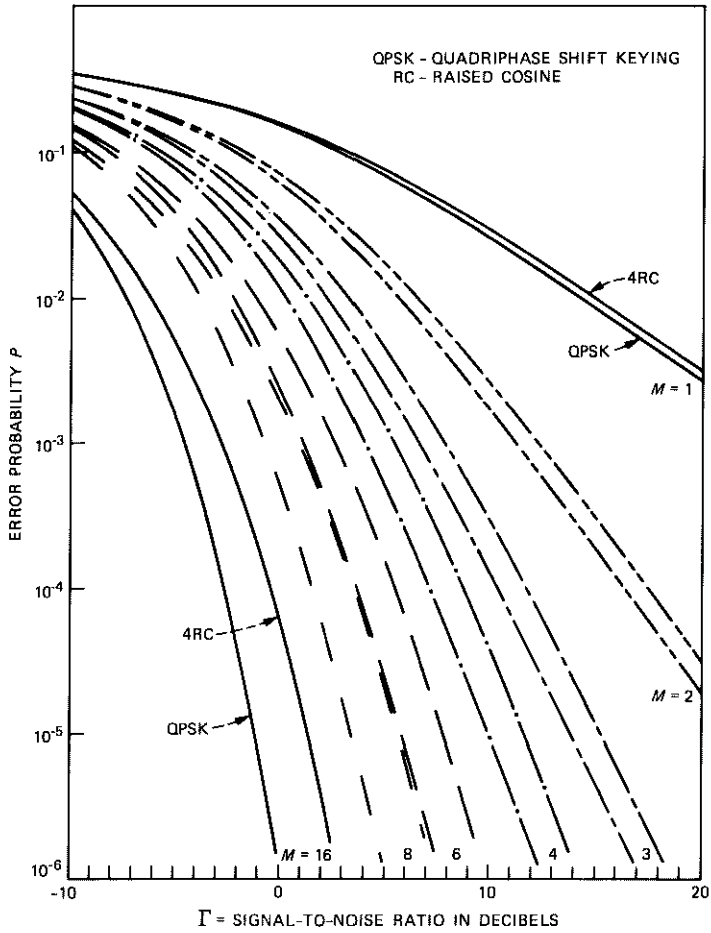



Fig. 4—Error probability, $P$, versus average per diversity branch (and per bit) signal-to-noise ratio $\Gamma$ for a coherent MSK-type receiver with Selected Pulse Amplitude Modulation (SPAM) receiver. Maximal-ratio combining and modulation scheme 4RC are used. Curves shown without differential encoding/decoding.

antenna configuration, and a fixed cell size, a large frequency reuse factor, $N$, gives good cochannel interference (the interfering cells get further away) but low system capacity, since the number of available channels in one cell is inversely proportional to $N$.[1-3] In this paper, we will basically confine the discussion to one type of cell, namely the type with three 120-degree corner antennas, as shown in Fig. 7. The maximum distance from any base station $d_0$ is equal to the cell radius $r$. For details about cellular systems based on this cell, see Refs. 1 and 3. Other cellular concepts are considered in Ref. 16.

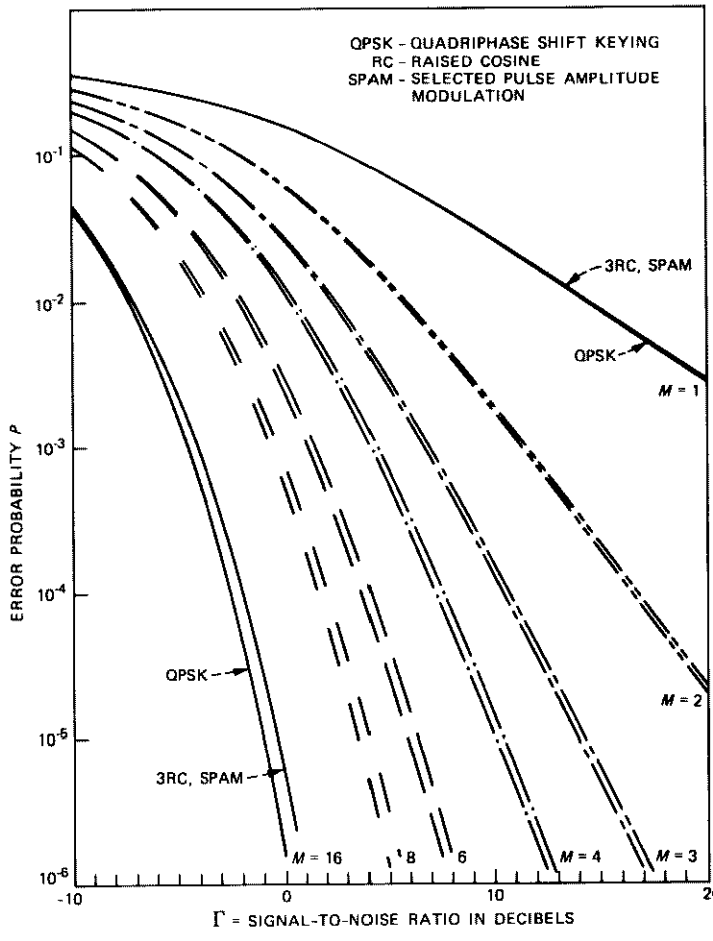



Fig. 5—Same as Fig. 4 with 3RC and SPAM-filter $M$ branch diversity with maximal-ratio combining.

### *2.3 Time-division retransmission*

The time-division retransmission (TDR) concept is described in Refs. 3 and 4. The basic ideas are as follows. The fading channel changes "slowly" (during one burst or package). Communication in both directions takes place in packages transmitted in the same frequency band: first mobile-to-base and then base-to-mobile. During mobile-to-base transmission, the channel is estimated for maximal-ratio, space-diversity combining in both directions. Several schemes[3] have been proposed for performing the required co-phasing for transmission from and reception at the base station. Because all diversity combining takes place at the base stations, the mobile equipment is

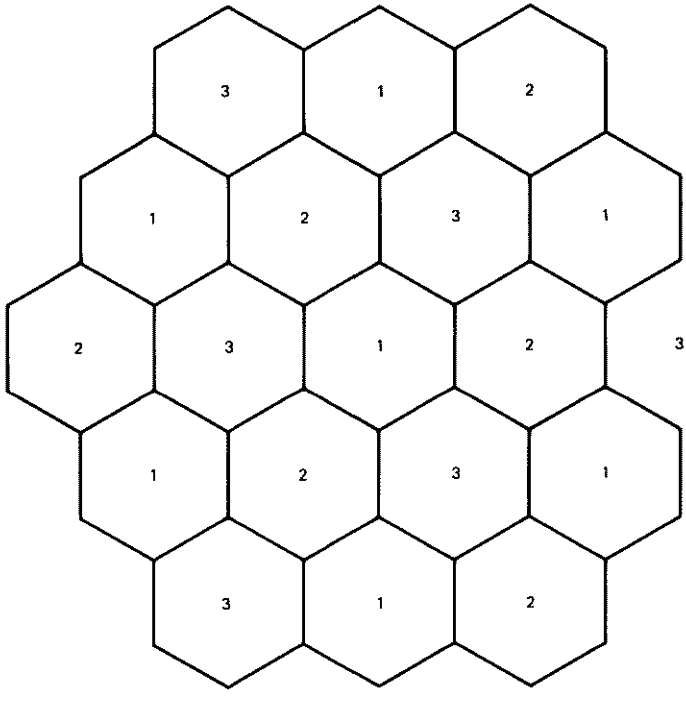



Fig. 6a—Example of cellular system. Each cell is a hexagon of equal area, and frequencies are reused. Cells marked with same digit use same frequency channel set (see Fig. 6b). In cluster of $N = 3$ cells, all channels are used.

relatively simple. As a consequence, it is feasible to use more than two branches of diversity with TDR.

### *2.4 Signal propagation and interference*

It is assumed that mobile radio reception in an urban environment is characterized by

$$P(\bar{r}) = |\bar{r}|^{-\alpha} S(\bar{r}) \cdot R^2(\bar{r}), \tag{5}$$

where $P(\bar{r})$ is the received signal power at location $\bar{r}$ (position vector relative to a transmitter).[1,3,4] The first factor, $|\bar{r}|^{-\alpha}$, is a reduction factor due to the distance between the mobile unit and the transmitter, and $\alpha$ is the propagation constant. It is normally assumed that $\alpha$ is in the range three to four in the urban environment.[1,3] In free space, $\alpha = 2$.

The second factor, $S(\bar{r})$, represents shadow fading,[1,3,4] and the third factor, $R^2(\bar{r})$, represents Rayleigh fading.[4,17] $R$ is the envelope of the received signal and is modeled as a random variable with the density function

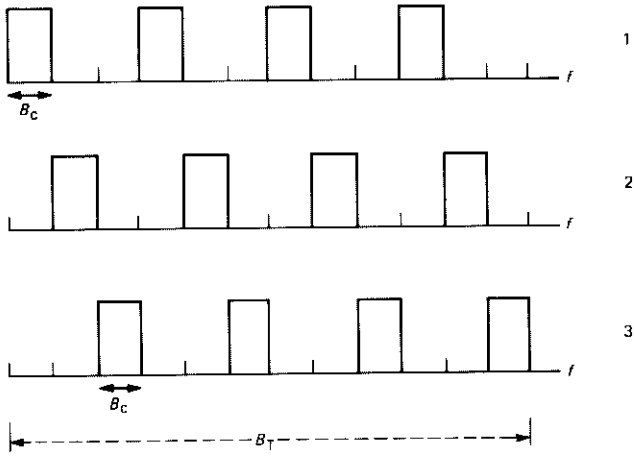



Fig. 6b—Frequency plan for cellular system with $N = 3$, shown in Fig. 6a. Channel set marked 1 used in each cell marked 1, etc. Channel bandwidth is denoted $B_c$, and total system bandwidth is $B_T$. Channel means two-way channels, including overhead for synchronization (when required).

$$p(R) = 2Re^{-R^2}, \tag{6}$$

with $E\{R^2\} = 1$ (see Refs. 4 and 17). In general, $R$ varies with vehicle location and signal frequency.

This paper also considers propagation and interference in cellular systems with frequency reuse. We will make use of the same basic assumptions as in Refs. 1 and 3.

It is assumed that the cochannel interference and adjacent-channel interference from cells other than those containing the desired mobile unit are formed by the incoherent sum of contributions from many interfering sources. This sum is assumed to be equivalent to stationary Gaussian noise.[1,3] It is also assumed that the shadow and Rayleigh fading of the total interference is negligible compared to the fading of the signal.[1,3,4]

It is also assumed that cochannel interference is the main source of additive signal degradation. Adjacent-channel interference from other cells will be considered to some extent in a few cases, where it is assumed that the adjacent-channel interference is the incoherent sum of many sources forming a stationary additive Gaussian adjacent-channel noise, which is added to the cochannel interference. The additive thermal background noise is assumed to be negligible compared to the cochannel and the adjacent-channel interference. Thus, the transmitter power and the cell sizes are assumed to be such that alien background noise from sources other than mobile units and transmitters in the cellular system is negligible compared to cochannel and adjacent-channel interference.

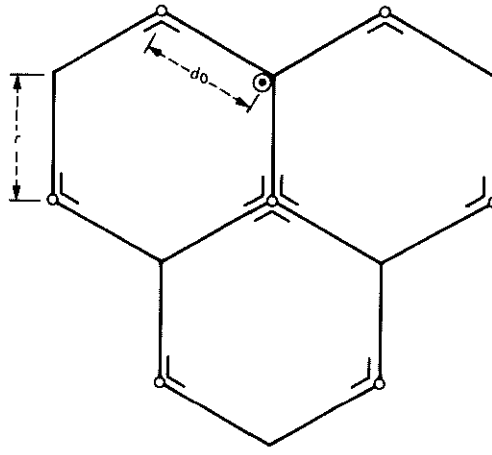



Fig. 7—Details for conventional three-corner, 120-degree antenna cell. Desired mobile shown in its worst-case locations.

Below, we will use the same technique as in Refs. 1 and 3 for calculating adjacent-channel interference and cochannel interference for continuous-phase modulation schemes. The signal-to-interference ratio is defined as the ratio of the signal power to the total noise power, based on the $|\bar{r}|^{-\alpha}$ propagation law and averaged over shadow and Rayleigh fading. It is assumed that the fading is flat over the band of each channel.

## III. A BASEBAND COMBINER FOR TIME-DIVISION RETRANSMISSION WITH CONSTANT-AMPLITUDE MODULATION

The time-division retransmission (TDR) concept has so far been applied only to binary modulation schemes with nonconstant amplitude.[3] For constant-amplitude modulation schemes, such as continuous-phase modulation, the combiner in Ref. 3 needs to be slightly generalized. That is the subject of this section. We have not yet analyzed the impact of the constant-amplitude format on synchronization. As a first rough estimate, it is assumed that the same scheme as that in Ref. 3 can be used also for constant-amplitude modulation. Further work in considering other schemes[3] for synchronization and establishing phase references is required for constant-amplitude modulation schemes.

The baseband combiner for space diversity with maximal-ratio combining used by P. S. Henry and B. Glance[3] for binary phase shift keying (BPSK) is easily generalizable to quadrature constant-amplitude modulation. The base-station signal processing equipment for one branch of diversity is shown in Fig. 8 for time-division retransmission with quadrature constant-amplitude modulation. During the

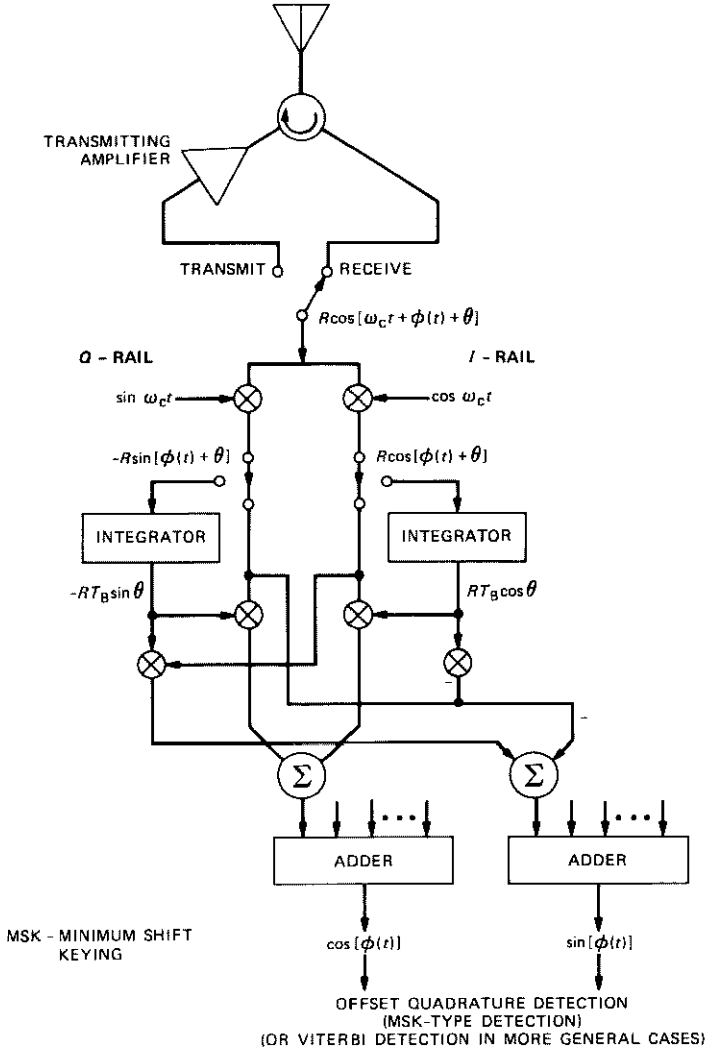



Fig. 8—Baseband combiner for constant-amplitude modulation and time-division retransmission. Compare this to Fig. 5 in Ref. 3.

reference transmission interval, the carrier burst of duration $T_B$ is received, having the form

$$R\cos(\omega_c t + \theta), \tag{7}$$

where $R$ is the Rayleigh amplitude, and $\theta$ is the unknown phase. It is assumed that the $R$ and $\theta$ are essentially constant until the next reference burst is transmitted.[3] Then the stored $R$ and $\theta$ are updated.

After down-conversion and integration, the reference coefficients $T_B R \cos\theta$ and $T_B R \sin\theta$ are produced and stored for message demodulation. During information transmission, the received signal can be written

$$R\cos[\omega_c t + \phi(t) + \theta] + I_c \cos\omega_c t + I_s \sin\omega_c t, \tag{8}$$

where $\phi(t)$ is the information-carrying phase (see the introduction). $I_c$ and $I_s$ are Gaussian interference with zero mean and variance $S^2$ (see Ref. 3). After down-conversion and multiplication with the reference coefficients and combinations following the network in Fig. 8, we have in each quadrature arm the signals

$$\begin{aligned} S_I = {} & T_B R^2 \cos[\phi(t) + \theta]\cdot\cos\theta \\ & + T_B R^2 \sin[\phi(t) + \theta]\cdot\sin\theta \\ & + I_c T_B R\cos\theta - I_s T_B R \sin\theta \end{aligned} \tag{9}$$

in the cosine branch ($I$-rail), and

$$\begin{aligned} S_Q = {} & T_B R^2 \sin[(\phi(t) + \theta]\cdot\cos\theta \\ & - T_B R^2 \cos[\theta(t) + \theta]\cdot\sin\theta \\ & - I_c T_B R \sin\theta + I_s T_B R\cos\theta \end{aligned} \tag{10}$$

in the sine branch ($Q$-rail). Using simple trigonometric formulas we have

$$S_I = T_B R^2 \cos[\phi(t)] + I_c T_B R\cos\theta - I_s T_B R\sin\theta, \tag{11}$$

$$S_Q = T_B R^2 \sin[\phi(t)] - I_c T_B R\sin\theta + I_s T_B R\cos\theta. \tag{12}$$

The special case of BPSK in Ref. 3 is obtained by letting $\phi(t)$ be

$$\phi(t) = \begin{cases} 0 & \text{for data } +1 \\ \pi & \text{for data } -1. \end{cases} \tag{13}$$

Thus $\cos\phi(t) = \pm 1$, and $\sin\{\phi(t)\} = 0$. Only one output is required in this case. Maximal-ratio combining at the output is obtained by adding the components from the $M$ branches.[3] Coherent reception is obtained and the data can be demodulated by processing $\cos\phi(t)$ and $\sin\phi(t)$. This can, for example, be done by means of the "MSK-type" receiver (offset quadrature receiver)[7–9] or, for the more general case, even a Viterbi detector.[6,13]

The combiner in Fig. 8 followed by an "MSK-type receiver" supplies the motivation for considering the family of $h = 1/2$ partial response FM or continuous-phase modulation (CPM) schemes[6,7] for the case of slow Rayleigh fading with $M$-branch space diversity with maximal-ratio combining.[4,7]

For transmission from the base station to the mobile unit, all stored phase references in Fig. 8 are changed from $\theta$ to $-\theta$. Thus, through the time-division retransmission procedure, maximal-ratio combining is approximately obtained at the mobile unit with only one antenna and one receiver.[3] All signal processing is performed at the base-station transmitter.

## IV. COCHANNEL AND ADJACENT-CHANNEL INTERFERENCE ANALYSIS FOR CPM

### *4.1 Cochannel interference analysis*

The technique for the cochannel interference analysis of CPM is basically the same as that for BPSK QPSK in Refs. 1, 3, and 16. The signal-to-cochannel interference ratios are given by the geometry of the cells and antennas and of the propagation exponent $\alpha$ [in the range three (pessimistic) to four]. From Refs. 1 and 3, we have the average signal-to-cochannel interference ratio with a three-corner cell system with three cells per cluster

$$P_S/P_I = 7.5 \text{ dB} \tag{14}$$

for mobile-to-base transmission and

$$P_S/P_I = 8.0 \text{ dB} \tag{15}$$

for base-to-mobile transmission. Figure 4 shows the theoretical number of diversity branches required in a space diversity (TDR) 4RC and QPSK scheme. Figure 5 shows the corresponding curves for 3RC.

With a larger number of cells per cluster (large channel sets), the required signal-to-cochannel interference ratio is larger, and thus fewer diversity branches are required.

It is assumed in the calculations that the background noise from other sources is negligible. The calculated signal-to-interference ratio determines which modulation method can be used (in terms of required detection efficiency) and how many branches of diversity are required.

Note that the signal-to-cochannel interference ratio is not dependent on a particular modulation scheme. It is given by the relationships of distances from the desired base station to the desired mobile unit and from the interfering base stations or mobile units. See Refs. 1, 3, and 16 for details. Thus, the cell geometry gives the available signal-to-cochannel interference ratio. This number plus the bit error probability requirement determines the number of diversity branches for a given modulation scheme by using results like those in Ref. 12 (see Figs. 4 and 5).

Reference 16 gives cochannel interference results for a generalized class of cells.

### 4.2 *Adjacent-channel interference analysis*

The effect of adjacent-channel interference in cellular mobile radio systems is briefly considered for some cases with constant-amplitude modulation schemes.

Figure 9 shows the so-called fractional out-of-band power curves for the binary-raised cosine family for modulation index $h = 1/2$.[6,7] These curves show the fraction of the spectral power relative to the total power outside the band $-f + f_0$, $f + f_0$ for the data rate $f_b = 1/T_b$. Thus, the bandwidth of the 4RC with the portion $10^{-3}$ of the total power outside the band is approximately $0.95/T_b$. Note that the figure shows half the bandwidth of one channel. Figures 10 and 11 show 3SRC and 4SRC. Similar curves for GMSK are published in Ref. 10.

Curves like those in Fig. 9 are useful for calculating the adjacent-

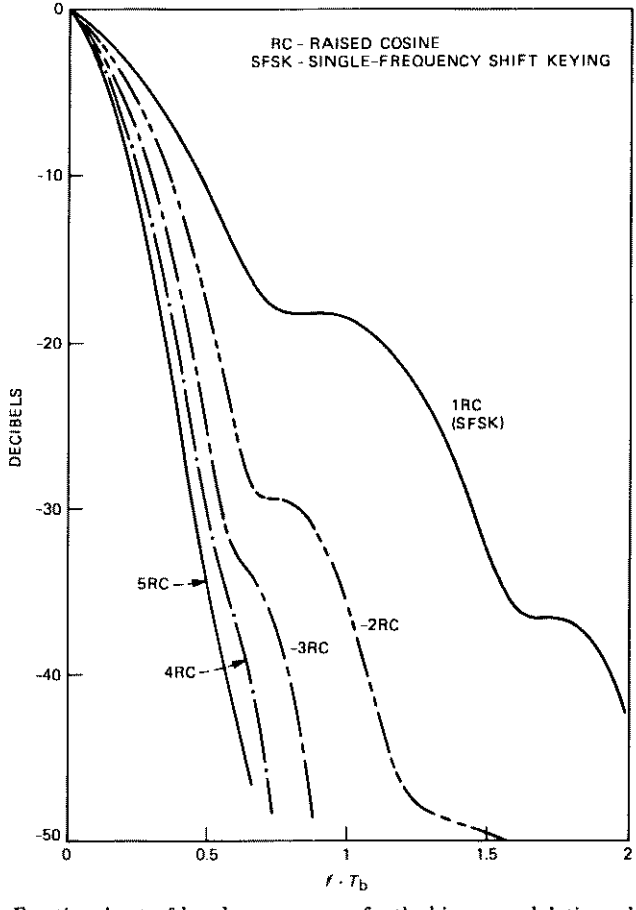



Fig. 9—Fractional out-of-band power curves for the binary modulation schemes 1RC, 2RC, 3RC, 4RC, and 5RC, $h = 1/2$.

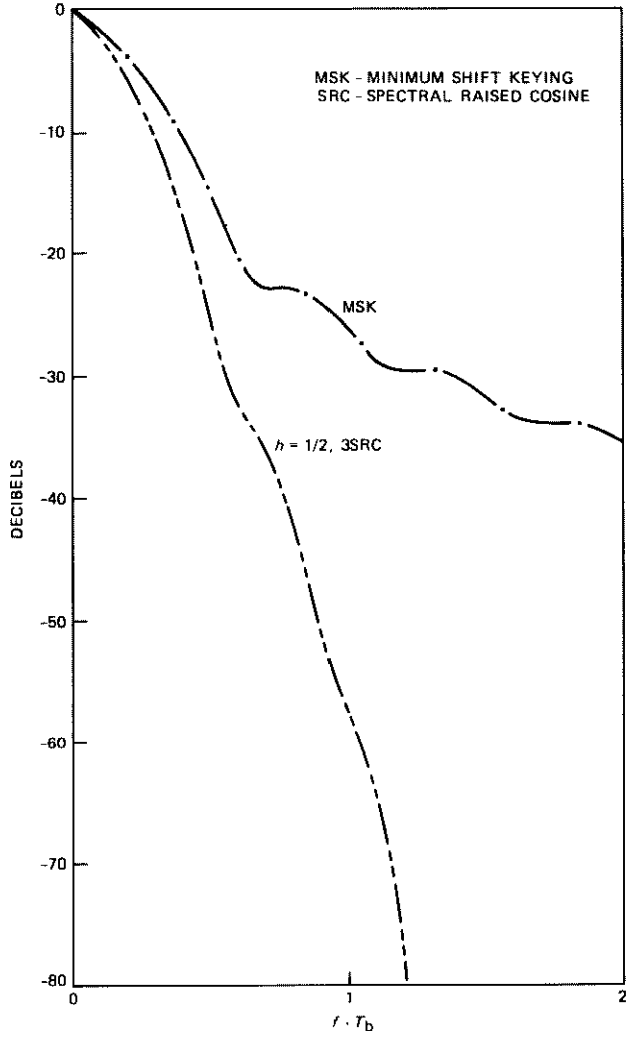



Fig. 10—Fractional out-of-band power curves for 3SRC, $h = 1/2$, and MSK.

channel interference. From the curves in Fig. 9, we obtain an upper bound on the interference power in the two adjacent channels on each side. Since the power spectra fall off rapidly with $f$ for increasing frequencies, this upper bound is a good estimate of the adjacent-channel interference power level.

From the results in the appendix and from the cochannel interference results above and in Refs. 1, 3, and 16, it is evident that with "wide" channels compared to the bit rate, average adjacent-channel interference is very small compared to cochannel interference. In the

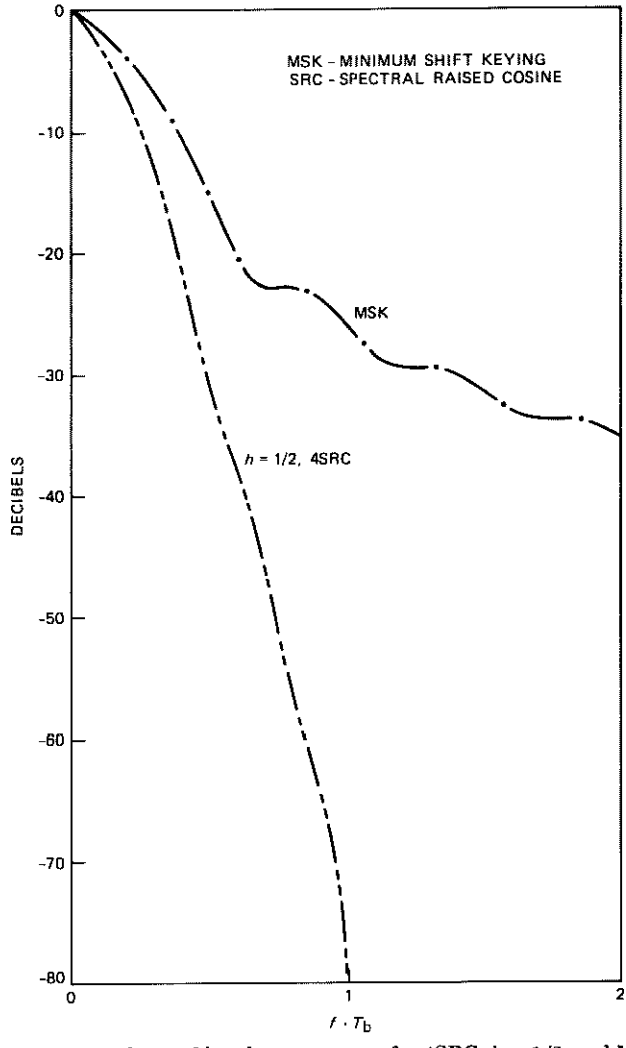



Fig. 11—Fractional out-of-band power curves for 4SRC, $h = 1/2$, and MSK.

appendix, a parameter, $A$, is introduced, which is a measure of the adjacent-channel interference suppression with a given modulation scheme and a given relationship between the bit rate and the channel bandwidth. Even with $A$ as large as $10^{-2}$, the average adjacent-channel interference is almost negligible compared to the cochannel interference. Much smaller $A$'s are quoted in the literature ($10^{-6}$–$10^{-7}$).[8,10,18] With $A$ as small as $10^{-2}$, the detection efficiency of the modulated signal will be affected by the filtering. In the system discussion below, we will choose "small" $A$'s.

For schemes with $N \geq 3$, the adjacent-channel interference with one particular cell is small with modulation schemes like 3RC-4RC (see Figs. 1, 6, and 7). The average adjacent-channel interference from adjacent cells is small compared to the cochannel interference. The same is true for $N = 1$ for the adjacent-channel interference from adjacent cells. However, for $N = 1$, the adjacent-channel interference within one cell is a more severe problem. This will restrain the value of bit rate $f_b$, relative to the channel bandwidth $B_c$. Cellular systems with $N = 1$ are considered in Refs. 1 and 16.

## V. DISCUSSION AND CONCLUSIONS

We can now put the various pieces of information together and evaluate the number of available channels for cellular schemes with different modulation schemes. Time-division retransmission is used in several cases but not all. For the cases without time-division retransmission, it is expected that the number of space diversity branches cannot be larger than $M = 2$. It is inconvenient to have more than two antennas on the mobile unit.

Let $B_T$ be the total system bandwidth, and let $B_c$ be the bandwidth of one two-way TDR channel (or two one-way channels without TDR). Let the bit rate over one channel be $f_b = 1/T_b$ bits/s. The total number of available channels is $B_T/B_c$. Let $N$ be the number of cells in a cluster. The number of channels, $C$, available in each cell in the cluster is $B_T/B_c \cdot 1/N$. The total system bandwidth will be kept constant for comparison of various systems. It is evident that $C$ can be improved by decreasing $N$ or decreasing $B_c$ (at a given $B_T$ and a given transmitted bit rate $f_b$). Decreasing $N$ has been discussed in Ref. 16. Decreasing $B_c$ can be done to a certain degree by selecting narrowband modulation schemes. However, the detection efficiency decreases for schemes with simple receivers. Furthermore, the adjacent-channel interference increases. There seems to be a preferred modulation scheme with an intermediate degree of smoothing (see Ref. 7 for details). These we have chosen below.

The total system bandwidth is assumed to be 40 MHz, and the carrier frequency $f_c = 850$ MHz. The propagation constant is $\alpha = 3$, and the error probability is $10^{-3}$. It is assumed that digitized speech is transmitted with 32 kb/s. If 16 kb/s is used (and all other system parameters are kept unchanged), the total system capacity in terms of channels is of course doubled for all schemes. When comparisons are made to other schemes,[3,15] these will be scaled to "our" system bandwidth 40 MHz and data rate $f_b = 32$ kb/s.

The calculations of number of diversity branches given below assume ideal conditions all the way through, i.e., no degradation due to

filtering, nonlinearities, timing, synchronization, combining, etc. Sometimes there is a margin, though.

Below we will calculate the number of channels per cell (antenna) for a few different cases.

1. $N = 3$, i.e., a cell with 120-degree base stations in three corners. Time-division retransmission. BPSK with 81 kb/s data rate (the overhead over 64 kb/s is due to the co-phasing procedure for the TDR scheme). Channel bandwidth $B_c = 100$ kHz (see Ref. 3). With this channel bandwidth, the transmitted BPSK signal must be filtered quite hard, resulting in a nonconstant amplitude. $M = 3$ branch diversity. This scheme gives about 133 two-way channels.

2. $N = 3$ and the same cells as in 1. TDR with 81 kb/s in a channel bandwidth of 81 kHz, 4RC (or better 4SRC[6]) modulation (see Figs. 9 to 11 for adjacent-channel interference estimates). Mild filtering of the 4RC might be necessary, causing small fluctuations of the envelope (see Fig. 4 for error probability considerations). TDR and $M = 4$ branches of diversity. The number of channels is 164.

3. $N \cong 11$, a cell with centrally located, omnidirectional antenna. $M = 2$ branches of diversity, no TDR. GMSK modulation with $B_bT \cong 0.25$ (see Refs. 10 and 15 for parameter definitions and other details). This corresponds roughly to 3RC (3SRC). Adjacent-channel interference is about $-70$ dB. 64 kb/s in 107 kHz. The results in Ref. 15 were scaled to the total bandwidth of 40 MHz and the data rate of 32 kb/s in the one-way speech channel. About 33 channels are available.

4. Same as 3 above with harder filtering. 64 kb/s in 58 kHz. Adjacent-channel interference is about $-20$ dB.[15] This leads to 62 channels.

The reasons that schemes 3 and 4 above yield such a low number of channels compared to the others are twofold. Time-division retransmission is not considered. Therefore an $M = 2$ is the highest degree of diversity considered. This leads to the high $N$ value. The other important reason is that only centrally located omnidirectional antennas are considered. Thus, even with a good modulation scheme, the number of channels remains low.

A class of constant-amplitude modulation schemes is considered for the cellular digital radio system. Such features as power spectral density, fractional out-of-band power, and detection efficiency with simple near-optimum detectors for the slow Rayleigh fading channel with space diversity and maximal-ratio combining are reported.

From the analysis and discussions above, we have seen that a large number of channels can be provided in a digital cellular system by the proper combination of antenna configuration, modulation scheme, and diversity scheme. We have seen that constant-amplitude digital mod-

ulation schemes can give high-capacity systems. It should be pointed out that higher system-capacity levels can be expected if nonconstant-amplitude modulation schemes are acceptable.

A large number of unsolved problems remain. The analysis above (as indeed that in Refs. 1, 3, and 15) is based on very simple and idealized channel and interference models. More refined analysis and simulations will be necessary. The analysis above was carried out under the idealized assumptions of flat fading and of negligible effects of filtering and nonlinearities on the modulation scheme. Uniform transmission conditions were assumed for all cells. No delay spread was considered. Perfect timing and synchronization were assumed with coherent detection and ideal maximal-ratio combining. It was furthermore assumed that perfect synchronization for the time-division retransmission scheme was established. All of the above problems and others have to be taken into account in a refined system analysis.

At least the same number of channels seems to be within reach with the space-diversity and time-division retransmission schemes as with the spread spectrum multiple access approach to digital mobile radio (see Refs. 3, 19, 20, and 21).

## VI. ACKNOWLEDGMENT

Thanks are due to Tor Aulin, who calculated the fractional out-of-band power curves in Figs. 9 to 11 and to Arne Svensson, who calculated the error probability curves in Figs. 4 and 5 (see Ref. 12).

## REFERENCES

1. Y. S. Yeh and D. O. Reudink, "Efficient Spectrum Utilization for Mobile Radio Systems Using Space Diversity," IEEE Trans. Commun., *COM-30*, No. 3 (March 1982), pp. 447–55.
2. V. H. MacDonald, "The Cellular Concept," B.S.T.J., *58*, No. 1 (January 1979), pp. 15–41.
3. P. S. Henry and B. S. Glance, "A New Approach to High Capacity Digital Mobile Radio," B.S.T.J., *60*, No. 8 (October 1981), pp. 1891–904.
4. W. C. Jakes, Jr., *Microwave Mobile Communications*, New York: Wiley, 1974.
5. T. Aulin and C.-E. Sundberg, "Continuous Phase Modulation—Part I: Full Response Signaling," IEEE Trans. Commun., *COM-29*, No. 3 (March 1981), pp. 196–209.
6. T. Aulin, N. Rydbeck, and C.-E. Sundberg, "Continuous Phase Modulation—Part II: Partial Response Signaling," IEEE Trans. Commun., *COM-29*, No. 3 (March 1981), pp. 210–25.
7. T. Aulin, C.-E. Sundberg, and A. Svensson, "MSK-Type Receivers for Partial Response Continuous Phase Modulation," Int. Conf. Commun., Philadelphia, Pennsylvania, June 1982, pp. 6F3.1–6.
8. F. deJager and C. B. Dekker, "Tamed Frequency Modulation, A Novel Method to Achieve Spectrum Economy in Digital Transmission," IEEE Trans. Commun., *COM-26*, No. 5 (May 1978), pp. 534–42.
9. R. deBuda, "Coherent Demodulation of Frequency-Shift Keying with Low Deviation Ratio," IEEE Trans. Commun., *COM-20* (June 1972), pp. 429–35.
10. K. Murota and K. Hirade, "GMSK Modulation for Digital Mobile Radio Telephony," IEEE Trans. Commun., *COM-29*, No. 7 (July 1981), pp. 1044–50.
11. T. Aulin and C.-E. Sundberg, "Numerical Calculation of Spectra for Digital FM

Signals," Nat. Telecommun. Conf., New Orleans, Louisiana, Conference Record, December 1981, pp. D.8.3.1–7.
12. C.-E. Sundberg, "Error Probability of Partial Response Continuous Phase Modulation With Coherent MSK-type Receiver Diversity and Slow Rayleigh Fading in Gaussian Noise," B.S.T.J., *61*, No. 8 (October 1982), pp. 1933–63.
13. T. Aulin, C.-E. Sundberg, and A. Svensson, "Viterbi Detectors with Reduced Complexity for Partial Response Continuous Phase Modulation," Nat. Telecommun. Conf., New Orleans, Louisiana, Conference Record, December 1981, pp. A.7.6.1–7.
14. W. Hirt and S. Pasupathy, "Suboptimum Reception of Binary CPSK Signals," Proc. IEE, Part F, *128*, No. 3 (June 1981), pp. 125–34.
15. K. Murota, K. Konoshita, and K. Hirade, "Spectrum Efficiency of GMSK Land Mobile Radio," Int. Conf. Commun., Denver, Colorado, Conference Record, June 1981, pp. 23.8.1–5.
16. C.-E. Sundberg, "Alternative Cell Configurations for Digital Mobile Radio Systems," B.S.T.J., this issue.
17. M. Schwartz, W. R. Bennett, and S. Stein, *Communication Systems and Techniques*, New York: McGraw-Hill, 1966.
18. D. Muilwijk, "Tamed Frequency Modulation—A Bandwidth-Saving Digital Modulation Method Suited for Land Mobile Radio," Philips Telecommun. Rev., *37*, No. 1 (March 1979), pp. 35–49.
19. G. R. Cooper and R. W. Nettleton, "A Spread-Spectrum Technique for High-Capacity Mobile Communications," IEEE Trans. Veh. Technol., *VT-27* (November 1978), pp. 264–75.
20. P. S. Henry, "Spectrum Efficiency of a Frequency-Hopped DPSK Spread-Spectrum Mobile Radio System," IEEE Trans. Veh. Technol., *VT-28* (November 1979), pp. 327–32.
21. D. J. Goodman, P. S. Henry, and V. K. Prabhu, "Frequency-Hopped Multilevel FSK for Mobile Radio," B.S.T.J., *59*, No. 7 (September 1980), pp. 1257–75.

## APPENDIX

### *Details of adjacent-channel interference calculations*

In this appendix, we will demonstrate how the method for analyzing cochannel interference in Refs. 1 and 3, with minor modifications, can be used for calculation of adjacent-channel interference from adjacent (and further away) cells.

The average adjacent-channel interference from adjacent cells is calculated with basically the same method as the cochannel interference. It is assumed that the total adjacent-channel interference is formed by incoherent additions of several interference sources forming a stationary additive Gaussian noise, which is added to the cochannel interference noise. We will see that with "reasonable" spectral shape of the modulation scheme, this intereference source is, in most cases, small compared to the cochannel interference.

Let $A$ be the relationship between the adjacent-channel interference power on one side of the channel to the total power in this channel. This number is upper bounded by 1/2 of the fractional out-of-band power level at $f = B_C$ (see Figs. 9 to 11 and Ref. 5). It is assumed that the receiver filter consists of an ideal bandpass filter of width $B_C$ in these idealized calculations (the same assumptions are used in Ref. 15).

The base stations and the mobile units in nearby cells will cause adjacent-channel interference because of nonideal, nonband-limited

signals. We will consider the constant-amplitude raised cosine schemes (see Section II) transmitted without bandpass filtering. The spectral tails will cause adjacent-channel interference.

First, we will consider adjacent-channel interference for the case $N = 3$, $\alpha = 3$ with centrally located omnidirectional base stations (see Fig. 12). It is immediately clear from the frequency plan (Figs. 6a and 6b) that the adjacent-channel interference from sources within the same cell is extremely small with any spectrally efficient modulation scheme because each adjacent channel within one cell is three channels apart. The significant contributions to the total adjacent-channel interference come from adjacent channels in adjacent cells.

Once the major sources of interference are identified, it is straightforward to use the same methods as in Refs. 1, 3, and 16 for the calculation of the interference. The average adjacent-channel interference is suppressed by a factor of $A$ compared to the cochannel inter-

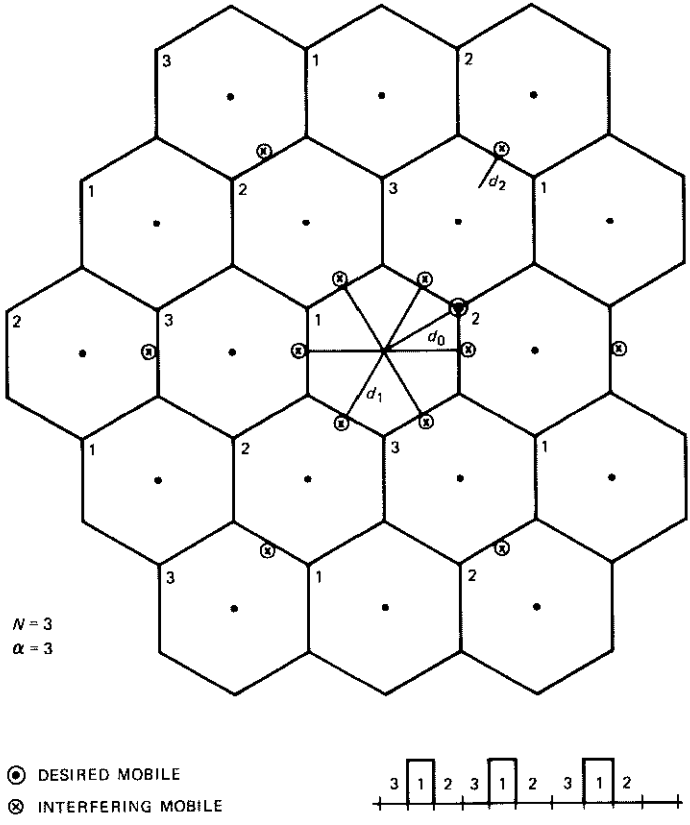



Fig. 12—Worst-case adjacent-channel interference cases for mobile-to-base transmission with centrally located antennas.

ference from the same sources. Space diversity helps against average adjacent-channel interference just as well as cochannel interference.[1,3]

Figure 12 shows the calculation of the worst-case (noncoherent addition) adjacent-channel interference for the $N = 3$, $\alpha = 3$ case with centrally located base stations and mobile-to-base transmission. Adjacent-channel interference comes from all cells marked 2 and 3. (Cochannel interference comes from cells marked 1; this is considered in Refs. 1, 3, and 16.) The distance to the desired mobile unit serviced from the omnidirectional antenna in the center of the cell is $d_0$. The distance to the nearest (worst-case) six interfering mobile units is $d_1$ (see Fig. 12). The distance to the next six interfering mobile units is $d_2$. Thus, the worst-case adjacent-channel interference to signal ratio for mobile-to-base transmission is

$$\frac{P_I}{P_S} = A\left[6\left(\frac{d_0}{d_1}\right)^3 + 6\left(\frac{d_0}{d_2}\right)^3 + \text{further terms}\right]. \tag{16}$$

For the case in Fig. 12,

$$P_I/P_S \cong 9.5 \cdot A. \tag{17}$$

Assuming that the interfering mobile units can be in any equally probable location in their cells, we found that the average adjacent-channel interference to signal ratio for mobile-to-base is

$$P_I/P_S \cong 3.6 \cdot A. \tag{18}$$

It is straightforward to obtain similar results for base-to-mobile transmission for the case in Fig. 12. We have the adjacent-channel interference to signal ratio

$$P_I/P_S \cong 2.5 \cdot A. \tag{19}$$

It is also straightforward to calculate the formulas for a cell with three 120-degree corner stations in each cell (see Fig. 7). For $N = 3$, $\alpha = 3$, we have for this case:
base-to-mobile,

$$P_I/P_S \cong 1.2A, \tag{20}$$

mobile-to-base average,

$$P_I/P_S \cong 1.5A. \tag{21}$$

In principle, it is straightforward to calculate the corresponding adjacent-channel interference formulas for the $N = 3$, $\alpha = 3$ case for other cells.[16] However, as a rough estimate one can use $A$ times the cochannel-interference to signal ratio for the corresponding $N = 1$ case.[16]

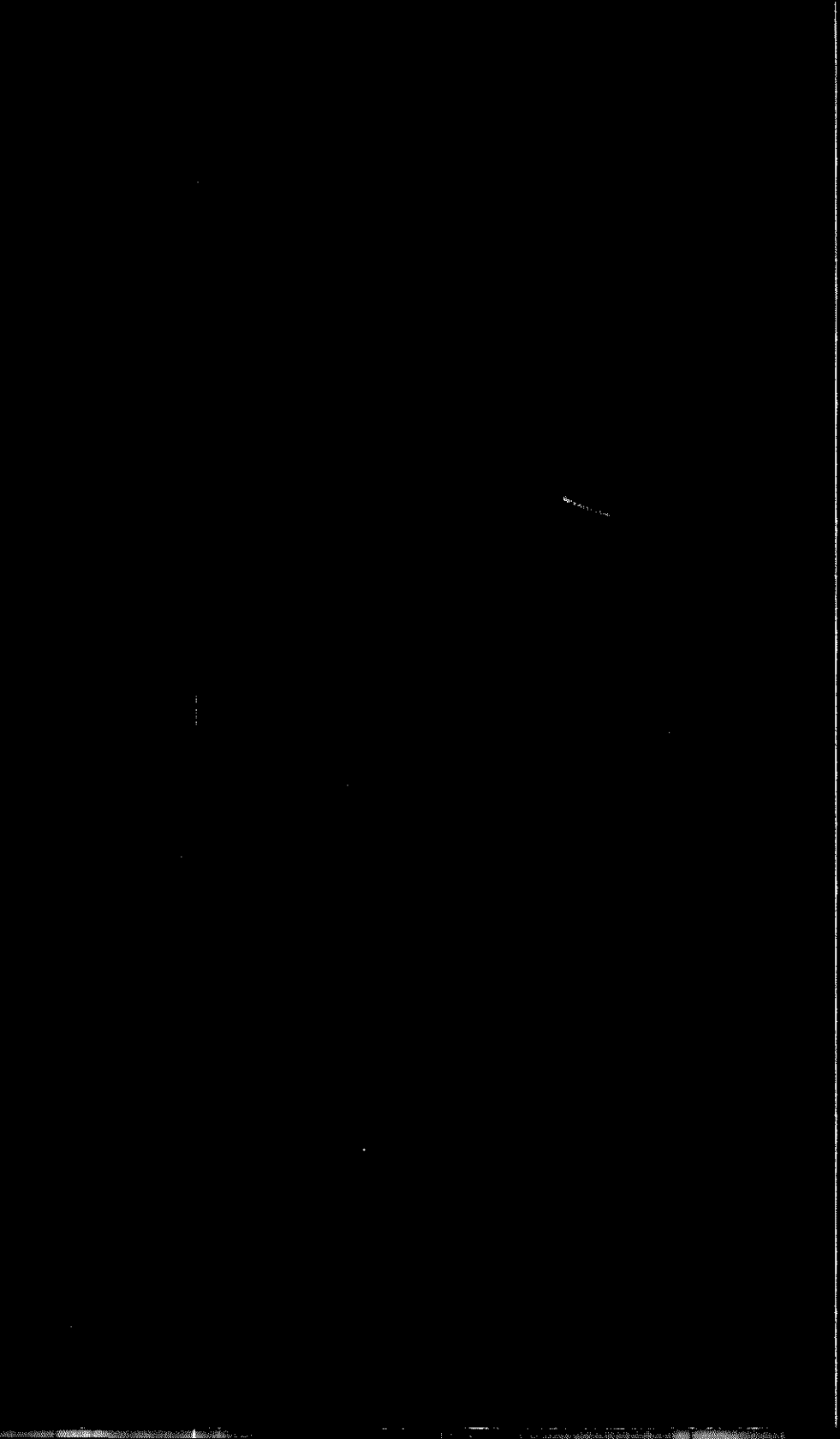
For all calculations above, it was assumed that the desired mobile unit is in the least favorable position in the cell, much the same way as in Refs. 1 and 16.

Note that in the approximate calculations above, we only considered averages. It is possible that the desired mobile unit and the interfering mobile unit in the adjacent cell occupying a frequency channel immediately adjacent to that of the desired mobile unit are almost at the same location near the cell dividing line. For such cases, the adjacent-channel interference might be such that the space diversity does not suppress this interference (the combiner in the desired mobile unit adds the interference and treats this interference just as the desired signal). For this case, a sufficiently small $A$ is required. It is, of course, also conceivable to have a frequency channel change for one of the mobile units in that case.

For the $N = 1$ case, the adjacent-channel interference problem becomes larger than for the $N = 3$ case. The background average adjacent-channel interference from sources in adjacent cells is still there as before. On top of that, there is interference from (primarily two) adjacent channels within the cell of the desired mobile unit. A small $A$ is required to suppress this interference.

## AUTHOR

**Carl-Erik W. Sundberg,** M.S.E.E and Dr. Technology, Lund Institute of Technology, University of Lund, Lund, Sweden, in 1966 and 1975, respectively. Currently, Mr. Sundberg is a Research Professor in the Department of Telecommunication Theory, University of Lund. He is also Director of the consulting company SUNCOM, Lund. During 1976 he was with the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands, as an ESA Research Fellow. He has held positions as Consulting Scientist at LM Ericsson and SAAB-SCANIA, Sweden, and at Bell Laboratories, Crawford Hill, N. J. His research interests include source coding, channel coding, especially decoding techniques, digital modulation methods, fault-tolerant systems, digital mobile radio systems, spread spectrum systems, and digital satellite communication systems. He has published a large number of papers in these areas during the last couple of years. Dr. Sundberg holds several U.S. and European patents. Senior Member, IEEE; member, SER (Sveriges Elektroingenjörers Riksförening).



THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# A Compatible High-Definition Television System

By T. S. RZESZEWSKI*

(Manuscript received March 11, 1983)

A compatible HDTV system with the potential of having significantly better picture quality than the present NTSC color TV system is proposed. It realizes an increase in horizontal and vertical resolution and has considerably less crosstalk between the composite signal components compared to the NTSC signal. The increased resolution will allow a display as large as present home projection televisions with a sharper-looking and more detailed image than is possible with the present NTSC system. Also, the elimination of crosstalk adds to picture quality. A large screen size together with improved image quality should provide the user with a feeling of realism and involvement. This system also allows the use of more detailed graphics and more text per page for new services such as teletext.

## I. INTRODUCTION

A compatible high-definition television (HDTV) system capable of producing an image quality significantly better than NTSC is proposed. Viewing the pictures produced by this system on a large screen should result in an increased sense of realism over the present NTSC television. This new system uses a split-luminance and split-chrominance (SLSC) type of transmission. The areas where the primary benefits are expected from this HDTV system over the National Television System Committee (NTSC) system are:

1. Increased horizontal resolution,
2. Increased vertical resolution, and
3. Less crosstalk between the components of the signal.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

Several different approaches to HDTV can offer excellent quality. However, there are quality-independent attributes that will aid the acceptance of this type of service. In particular, compatibility with present NTSC receivers and a bandwidth of no more than twice the present 6-MHz channel for broadcast are critical.

Many different forms of compatibility are discussed today. However, as used here, compatibility means receiver compatibility. The signal must be able to feed an HDTV and an NTSC TV simultaneously and be received on the NTSC receiver with substantially the same quality picture that those sets presently realize, while the HDTV receiver realizes all the benefits, such as increased resolution.

Spectrum usage is another problem with some HDTV approaches. The NHK (Japanese) system uses 30 MHz of baseband.[1] This bandwidth is so large that this type of service could not be broadcast in the same manner as the present broadcast service, and would leave this service with fewer delivery systems. Of course, video tape or cable are still possible delivery systems. Direct broadcast by satellite (DBS) is also possible; however, the prime allocations that are not affected by the weather are likely to be used for DBS of NTSC in the near future.

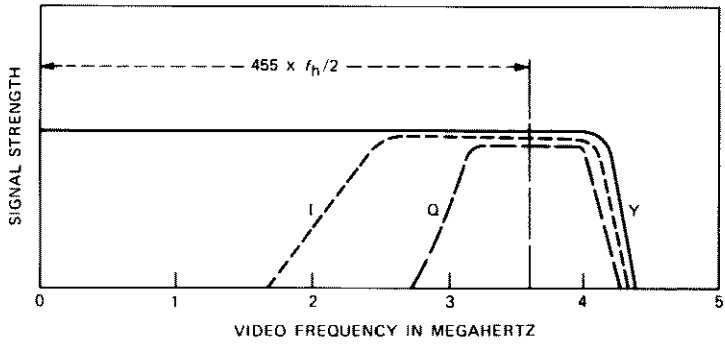
Some other proposed systems have preserved some aspect of compatibility.[2,3] They retain the scanning format, but change the encoding such that a present receiver will not operate correctly.

A format is proposed here that uses a 10-MHz baseband composite signal that can be transmitted as a vestigial sideband, amplitude-modulated signal in a bandwidth of 12 MHz. Also, an NTSC receiver will operate with the same quality as at present when receiving this signal. The NTSC receiver must be tuned to the lower 6-MHz portion of the 12-MHz spectrum. The price that is paid for this compatibility is a reduction in the number of broadcast or CATV channels to approximately one half the present NTSC allocations. However, VHF stations are normally spaced at least every second channel allocation apart in a given location for broadcast, and UHF channels are spaced even further apart. Therefore, the impact on the present broadcast service is likely to be small. Also, new systems with bandwidth requirements like the NHK system would produce at least a six-to-one reduction in the number of channels compared to an NTSC service.

## II. A COMPATIBLE APPROACH

### *2.1 System description*

This new system is built around the NTSC baseband spectrum shown in Fig. 1. The NTSC baseband signal is modulated for broadcast as a vestigial-sideband, amplitude-modulated signal as shown in Fig. 2. The composite HDTV baseband is illustrated in Fig. 3. Note that there is approximately 0.75 MHz of spectrum left over for other




Y - THE LUMINANCE SIGNAL
I - I COMPONENT OF CHROMINANCE
Q - Q COMPONENT OF CHROMINANCE

Fig. 1—NTSC luminance and chrominance bandwidth allocation.

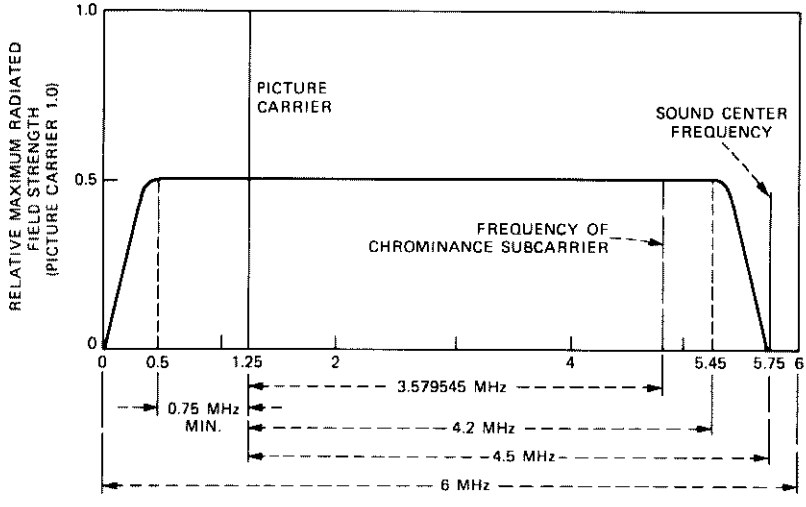



Fig. 2—Idealized picture transmission amplitude characteristic for NTSC.

services such as teletext and/or multichannel sound. The composite HDTV baseband signal could be modulated as a vestigial-sideband, amplitude-modulated signal for broadcast, as illustrated in Fig. 4. The selectivity of the NTSC receiver will reject the additional high-frequency portion of this signal at least as well if not better than an adjacent NTSC station.

The composite signal of Fig. 3 is obtained by starting with a 1050-line progressive scan source of wide-bandwidth red, green, blue (R, G, B) signals. The technique for improved vertical resolution is described

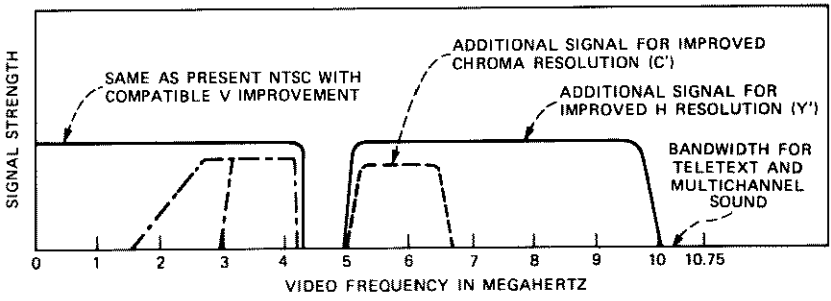



Fig. 3—SLSC HDTV baseband spectrum.

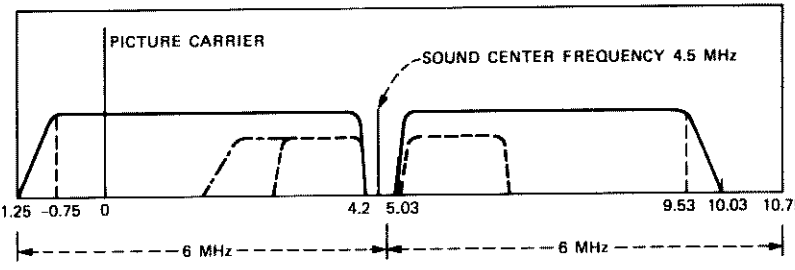



Fig. 4—Idealized broadcast picture transmission amplitude characteristic for SLSC HDTV.

in the literature.[4] That technique will be described in greater depth later. For now, it is sufficient to say that the wideband R, G, B signals are filtered and converted to a 525-line signal by a scan conversion that deletes every second line to obtain a 525-line signal suitable for transmission. Next, the wideband R, G, B signals are matrixed into a Y, I, Q format. This process is illustrated in Fig. 5. The NTSC encoder provides the appropriate selectivity and processing to create a standard NTSC signal that occupies the lower 4.2 MHz of the baseband spectrum in Fig. 3. The additional signal for improved horizontal resolution (Y′) that occupies the frequency region of approximately 5 to 10 MHz in the baseband is processed by the High Frequency Luminance Encoder. The High Frequency Chrominance Encoder creates the signal C′.* These signals are added to the signal obtained from the NTSC encoder to produce the composite HDTV baseband signal of Fig. 3. A detailed block diagram that elaborates on the encoding of these extra signals will be given later.

* A method of recreating the high-frequency components of the composite NTSC signal that were filtered out at the transmitter has recently been developed.[5] If this technique is proven to be acceptable for this application, the additional signal for extra chrominance resolution (C′) is not needed.

The system description in this paper utilizes the anti-aliasing filtering of the R, G, B signals in the encoder, as shown in Fig. 5. Consequently, the interpolation filtering in the decoder will also be performed on R, G, B as described later. An alternative approach that may have some advantages is to transform the R, G, B signals into a Y, I, Q format before performing the anti-aliasing filtering. In that case the I and Q signals will be filtered to a smaller bandwidth before the anti-aliasing filter. This approach simplifies two of the anti-aliasing and interpolation-filter designs and requires less frame-store memory. It will be considered further in the section on system alternatives.

Figure 6 contains the general block diagram of the decoder. Not shown are the tuner, IF, and video detector that are required to select and demodulate the HDTV signal to a baseband signal if the composite signal is modulated for broadcast; these functions will be described later. In the decoder, the three portions of the HDTV signal (the low-frequency signals obtained from the NTSC composite signal, the

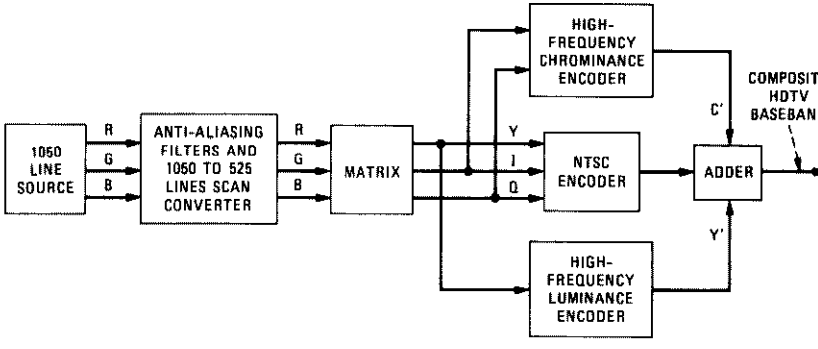



Fig. 5—SLSC HDTV encoder.

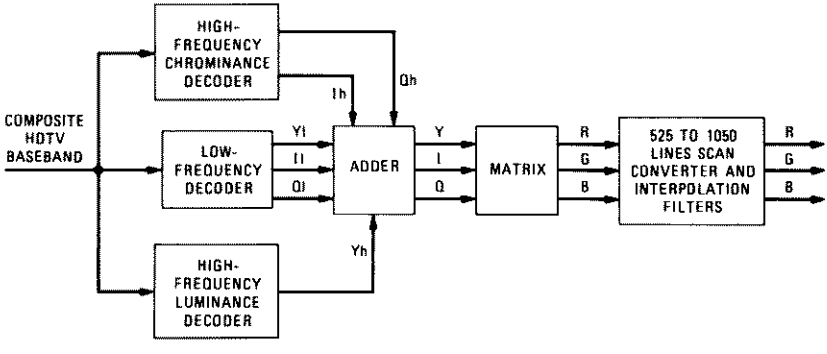



Fig. 6—SLSC HDTV decoder.

additional high-frequency chrominance signal, and the additional high-frequency luminance signal) are selected and processed in parallel. A detailed block diagram of these portions will be given later. The additional chrominance signals (Ih and Qh) are added to the chrominance produced by the low-frequency decoder to get I and Q signals that have a 2-MHz bandwidth each. These signals are then matrixed to produce 2-MHz color-difference signals. They are then added to the full 7.5-MHz-bandwidth luminance signal to produce R, G, B signals. The wideband luminance, Y, results from the addition of Yl and Yh.

However, before being fed to the display, the signals must be processed to realize the increased vertical resolution. The process consists of a scan conversion back to 1050 lines from the interlaced 525-line transmission format. The monitor may be either a 1050-line progressive scan or interlaced scan at the discretion of the manufacturer. However, progressive scan is needed to realize the full potential of this system. Interpolation filtering is performed in the scan-conversion process, and the high-definition R, G, B signals are fed to the display device.

### *2.2 Vertical resolution*

An analysis of the resolution capability of the NTSC signal is helpful to assess the improvements that HDTV will realize. Resolution is expressed in terms of vertical and horizontal equivalent TV lines. The vertical resolution tells the number of horizontal lines alternating between black and white that can be resolved in the TV image. It is tempting to equate this to the total number of scan lines minus the lines in the vertical interval that are not used for display. Unfortunately, the scanning process that changes the image into an electrical signal in the camera and then reassembles the image on the display is really a sampling process. It is well known that sampled signals must first be bandlimited or aliasing will occur. It is the aliasing and the replicated spectra that further reduce the vertical resolution. Equation 1 below expresses the actual vertical resolution in a TV picture by the addition of a Kell factor ($k$) that takes this extra loss into account.

$$Rv = (Nt - 2Nv)k. \tag{1}$$

$Rv$ is the vertical resolution; $Nt$ is the total number of scan lines (525 for NTSC); $Nv$ is the number of lines in the vertical interval (21 for NTSC); and $k$ is the Kell factor. The Kell factor normally ranges between 0.6 and 0.7 for TV systems and results in an $Rv$ range between 290 and 338 lines for NTSC TV. The method of vertical resolution improvement employed for the SLSC HDTV system allows the vertical

resolution to approach the full 483 lines of the active video; the Kell factor approaches unity.[4]

The modulation transfer function (MTF) in the camera and the display are analogous to the frequency response in linear system theory. It can be adjusted by shaping the electron beam in the camera and a CRT display. The contour of the scanning spot can be thought of as a two-dimensional impulse response commonly called the point-spread function. A narrow scanning spot in the vertical direction means a wide vertical spatial frequency spectrum and aliasing, and a wide scanning spot means overlapping of adjacent lines and low-pass filtering in the vertical direction (defocusing). In NTSC, the scanning spot is adjusted to compromise between aliasing and defocusing. Anti-aliasing (prefiltering) on a 1050-line progressive-scan source signal and interpolation (postfiltering) that eliminates replicated spectra aid the compromise. These filters can be used together with more lines at the camera in a compatible fashion to increase the vertical resolution.[4]

### *2.3 Horizontal resolution*

The horizontal resolution is also expressed in terms of lines (equivalent vertical lines) that are the same width as the horizontal lines used to determine the vertical resolution above. There are two lines per cycle of video bandwidth. In other words, in the time—and therefore the horizontal space—it takes to display a cycle of the highest-frequency signal that will pass through the system bandwidth, the system will produce two lines on the display, one white and one black. The width of the lines is the same as for the vertical resolution, and the 4-to-3 aspect ratio is taken into account. Equation 2 below can be used to determine the horizontal resolution of a television system per unit of video bandwidth ($Rh'$).

$$Rh' = 2Ta/AR. \tag{2}$$

$Ta$ is the total active time for a horizontal line, and $AR$ is the aspect ratio (these are 53.5 microseconds and 4/3 respectively for the NTSC system). This results in approximately 80 lines/MHz. Most NTSC receivers have at least 3 MHz of bandwidth, yielding a minimum of 240 lines of horizontal resolution. However, the total system luminance bandwidth can be 4.2 MHz if a comb filter is used in the NTSC receiver. This results in 336 lines.

The horizontal resolution is increased for the HDTV system proposed here by adding extra bandwidth to the baseband luminance signal. Figure 7a shows the 7.5-MHz bandwidth that is allocated to luminance in this compatible system. The same scanning format for transmission as NTSC is used; therefore, eq. (2) above indicates that the system will produce 80 lines/MHz. The 7.5-MHz bandwidth will

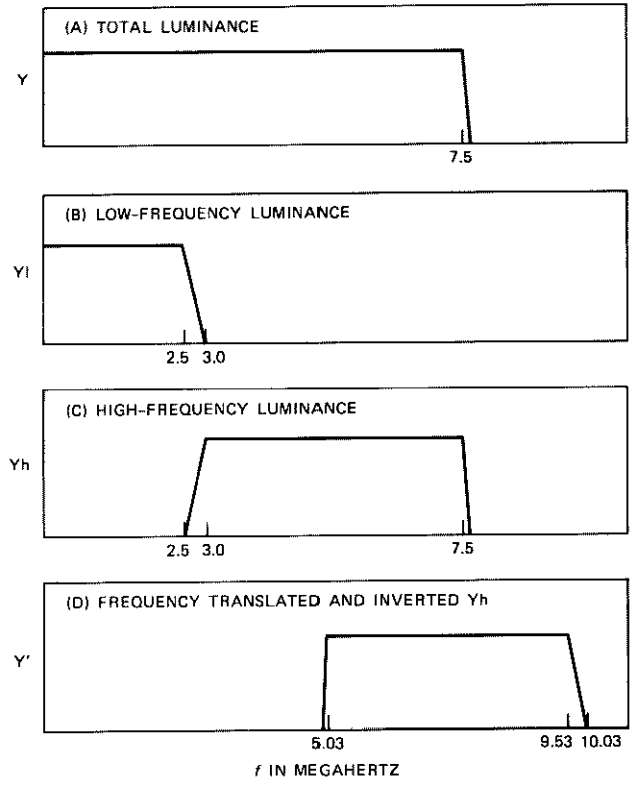



Fig. 7—Luminance (Y) processing in SLSC HDTV.

result in 600 lines of horizontal resolution.* This 7.5-MHz-bandwidth luminance spectrum feeds (1) an NTSC encoder that only uses the lower 4.2 MHz, and (2) a circuit that processes the high-frequency luminance detail (2.5 to 7.5 MHz) in a separate parallel channel.

Figures 7b through 7d illustrate the high-frequency luminance processing. Luminance (Y) is split up into a low-frequency portion (Yl) and high-frequency portion (Yh), each with a controlled roll off between 2.5 and 3 MHz (see Fig. 7c). If these two signals are added together, they will result in the original spectrum (Y) with a 7.5-MHz bandwidth. The high-frequency luminance will be processed so that it can be multiplexed with the NTSC baseband signal into a new, compatible HDTV baseband signal. This is accomplished by reversing

* This will result in 24 percent more horizontal resolution than vertical resolution for this system. There is the possibility of limiting the horizontal resolution to equal the vertical and simplifying the system. The amount of benefit produced by allowing greater horizontal resolution than vertical can be investigated when the system is implemented.

the frequency sense of Yh and translating it up in frequency to the location illustrated in Fig. 7d as Y′.

Note that the spectrum for Yh is cut off at a frequency below 2.5 MHz. Only the lower portion of the NTSC signal, which is substantially free from chrominance, is used for low-frequency luminance information (Yl).* Use of this lower cutoff frequency reduces the cross luminance problems that trouble the present NTSC system. An optimum cutoff frequency should be determined based on experiment since the higher the cutoff frequency the greater the horizontal resolution. The value indicated in Fig. 7 is a likely choice.

Figure 3 shows the complete baseband selectivity that defines the HDTV spectrum. Notice the additional signal for improved chrominance resolution (C′). Additional luminance resolution requires extra associated chrominance resolution and therefore additional chrominance bandwidth to appreciate the full improvement. The additional chrominance signal, C′, is time multiplexed between the Ih signal and the Qh signal components on alternating scan lines. However, unlike SECAM, the C′ signals are modulated as a single-sideband signal. The frequency of the carrier is selected to be a multiple of the horizontal frequency to minimize the crosstalk between this additional signal and the high-frequency luminance information, Y′. The purpose is to interleave the luminance (Y′) and chrominance (C′). The next section includes an explanation of the interleaving.

### *2.4 Detailed block diagrams*

Figure 8 is a detailed block diagram of the encoder shown in Fig. 5. The method for increasing the vertical resolution mentioned previously is used,[4] so that a 1050-line progressive-scan source (camera) feeds an anti-aliasing filter. The R, G, B signals are all processed in the same fashion, each one is passed through a scan converter that converts from 1050 lines in a progressive-scan format to 525 lines in an interlaced format for compatible transmission. These signals are then matrixed into a Y, I, Q format.

The additional high-frequency signal for improved chrominance is time multiplexed to carry the Ih and Qh signals on alternate horizontal lines. However, it is not frequency modulated as in SECAM; rather, it is single-sideband amplitude modulated. This gives a spectrum that tends to cluster at multiples of the horizontal frequency and odd

* Yl will be completely free from any cross-luminance produced by the Q component of the color signal. If the I component is allowed the present specification of 1.5-MHz bandwidth, it can cause some cross-luminance. However, there is some question of whether that bandwidth should be transmitted since there are no consumer receivers that make use of this extra bandwidth; consequently, the extra bandwidth of the I signal can only cause cross-luminance problems in present NTSC receivers.

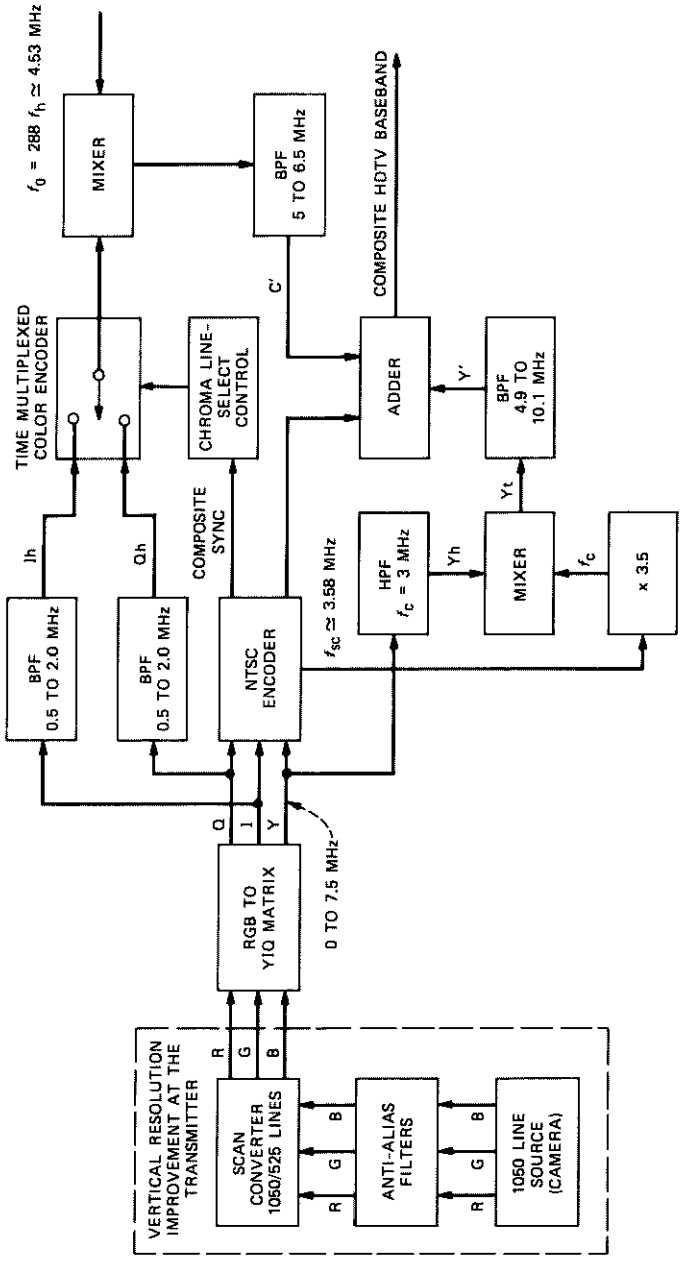



Fig. 8—Detailed block diagram of SLSC HDTV encoder.

multiples of half the horizontal frequency. The outputs of the two bandpass filters that select the Ih and Qh signals and the composite synchronization signal from the NTSC encoder feed the time-multiplexed color encoder (switch) that provides the additional signal for improved chrominance resolution, C′. The switch connects the Ih signal and then the Qh signal to the mixer on alternate scan lines. A carrier frequency, $f_0$, is the second input.*

$$f_0 = 288\, f_h \sim 4.53 \text{ MHz}. \tag{3}$$

A tone burst of frequency $f_0$ could be inserted into the vertical interval for phase reference at the receiver. The bandpass filter at the output of the modulator selects only the sum signal, C′, that is in the frequency range of 5 to 6.5 MHz.

The NTSC encoder functions as it would normally. It provides the composite synchronization signal to the line-select control for the high-frequency chrominance processing described above, and the color subcarrier for the high-frequency luminance processing.

The translated and frequency-inverted high-frequency luminance signal is formed by first high-pass filtering the 7.5-MHz luminance to produce Yh. When Yh is applied to the mixer illustrated in Fig. 8, a double-sideband suppressed carrier signal is created. The carrier input to the modulator is $f_c$.

$$f_c = 3.5\, f_{sc} = 3185(f_h/4) \sim 12.53 \text{ MHz}. \tag{4}$$

The NTSC color subcarrier, $f_{sc}$, is available from the NTSC encoder and can be used to derive $f_c$ as indicated by eq. (4). However, a tone burst of $f_c$ would be inserted into the vertical interval for phase reference at the receiver. The bandpass filter on the output only allows the frequency-inverted lower sideband, Y′, to pass. Y′ has a quarter-line frequency offset as can be seen from eq. (4). Consequently, $f_0$ is made equal to an exact multiple of $f_h$ in eq. (3) so that the time-multiplexed signal C′ will interleave with Y′. This signal, Y′, is added to the output of the NTSC encoder and to the output of the high-frequency chrominance encoder, C′, to produce the composite HDTV signal of Fig. 3. Audio information is added just prior to transmission; this is the conventional manner of adding audio information to the NTSC broadcast signal. Additional subcarriers could be added to the baseband of the composite HDTV signal for multichannel sound or teletext as indicated in Fig. 3.

A detailed block diagram of the decoder is shown in Fig. 9. The decoder input comes from the video detector(s); more will be said

* This is just one of several strategies for selecting $f_0$. $f_0$ could be made 300 $F_h \sim 4.72$ MHz; then a low-level tone of this frequency can be inserted in the SLSC baseband.

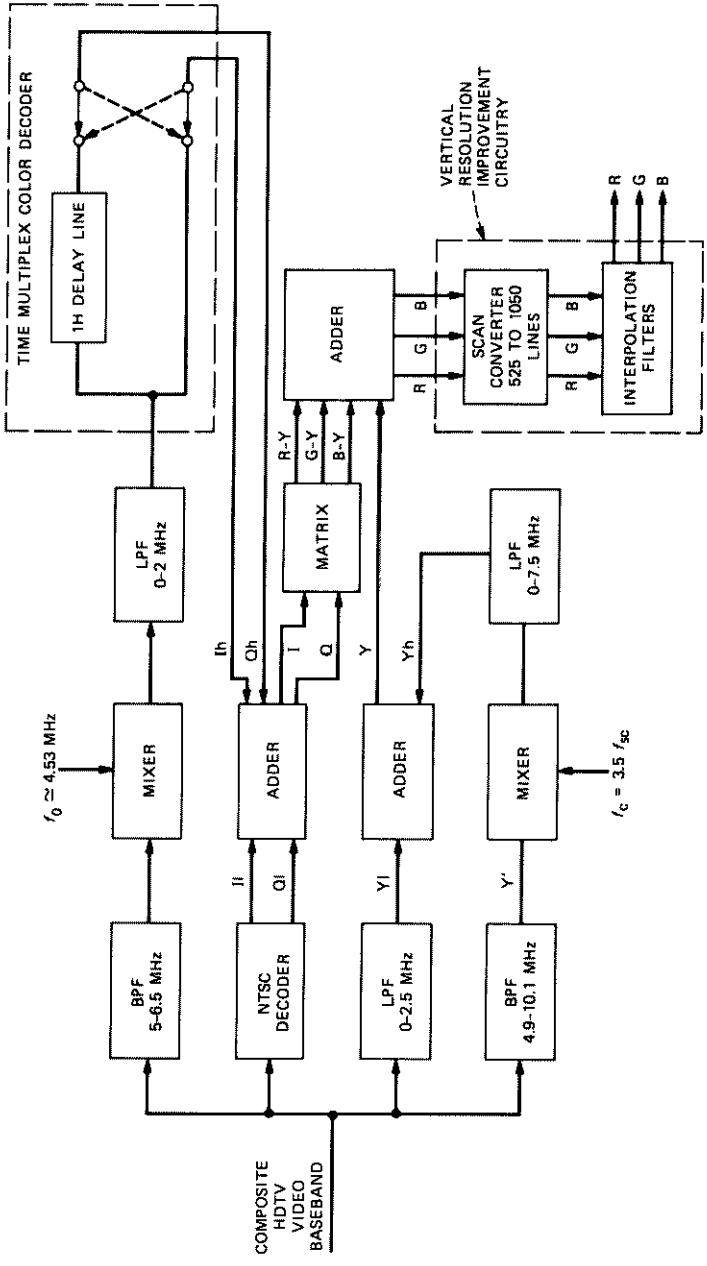



Fig. 9—Detailed block diagram of SLSC HDTV decoder.

about the receiver circuitry between the antenna and the decoder in Section 3.1. Since the composite HDTV signal was made up of three separate parts, each of these parts must be decoded. The high-frequency chrominance is decoded by first selecting it from the composite signal via the 5- to 6.5-MHz bandpass filter. This signal feeds a single-sideband demodulator that consists of a mixer and a bandpass filter that extracts the 0.5- to 2-MHz high-frequency color signal. However, this signal is still time multiplexed at the single-sideband demodulator output, so this output is fed into a time-multiplexed decoder to obtain simultaneous I and Q signals. The time-multiplexed decoder consists of a delay line that provides the storage of one horizontal line of color information, and the appropriate switches indicated in Fig. 9. The time-multiplexed decoder functions such that the present Ih (Qh) signal and the previous line Qh (Ih) signal are both present on the outputs providing the simultaneous Ih and Qh signals as outputs.

The NTSC decoder provides its normal I and Q color signals, called Il and Ql to emphasize that they convey the low-frequency portion (0 to 0.5 MHz) of the HDTV color signal. (Virtually all NTSC decoders produce I and Q signals of 0.5 MHz bandwidth in spite of the fact that the I channel should have extra bandwidth.) Il and Ql are added to Ih and Qh to form the complete 2-MHz I and Q signals. The high-frequency luminance, Y′, is passed by the 4.9- to 10.1-MHz bandpass filter and fed to the mixer. The carrier input is 3.5 $f_{sc}$, derived from the color subcarrier provided by the NTSC decoder with phase coherence obtained from a tone of $f_c$ in the vertical interval. The demodulator output is filtered by a low-pass filter with a 7.5-MHz cutoff. The resultant signal, Yh′, which occupies a 2.5- to 7.5-MHz spectrum, is added to the 0- to 2.5-MHz low-frequency luminance signal obtained by passing the composite HDTV spectrum through a low-pass filter with a 2.5-MHz cutoff. The adder outputs the luminance signal, Y. Y, I, and Q are matrixed to provide color-difference signals and then added to Y to output R, G, B signals to the circuit that provides the vertical-resolution improvement. This circuit consists of a scan converter that inserts extra lines, converting the interlaced 525-line transmission standard to a progressive or interlaced 1050-line format along with interpolation filtering for each of the R, G, B signals. The resulting R, G, B signals feed circuitry that drives the picture display.

### *2.5 System alternatives*

This system is flexible in that several variations are possible depending upon the needs and possibilities that surface during testing. The vertical resolution is limited to a maximum of 483 lines because the technique for vertical improvement can only promise a maximum resolution equal to the maximum number of active scan lines in the

transmission standard. The actual resolution could be slightly less than this because the anti-aliasing filter and the interpolation filter may reduce the vertical frequency response somewhat. The maximum horizontal resolution that corresponds to 7.5 MHz of bandwidth is 600 lines. This resolution can be achieved by using a comb filter at the receiver (covering the 5- to 6.5-MHz portion of the baseband signal shown in Fig. 3), or by not transmitting C′ and using the inferred highs processing to restore wideband chrominance signals at the receiver.[5] Both approaches result in less vertical resolution (483 lines) than horizontal resolution (600 lines).

Alternatively, a horizontal resolution equal to the vertical resolution could be chosen, thus simplifying the system. If 480 lines of horizontal resolution were chosen, only 6 MHz of luminance bandwidth is required. The high-frequency luminance of Fig. 3 need not overlap the high-frequency chrominance. Therefore, a comb filter is not needed to realize the 480 lines of horizontal resolution while using a C′ signal. An additional simplification is possible; the carrier, $f_c$, used to translate the high-frequency luminance spectrum in Fig. 8 can be three times the color subcarrier, 3 $f_{sc}$, rather than three and one half times that frequency.

Less extensive changes are possible in the interest of optimizing the system. For example, the 2.5-MHz cutoff of the low-pass filter that forms Y1 (illustrated in Figs. 6 and 9) could be reduced to 2 or even 1.5 MHz in order to completely eliminate any possible crosstalk between low-frequency chrominance and luminance. Alternatively, the cutoff could be increased to 3 MHz if it is shown that the extra resolution is a more important benefit than the penalty of a small amount of extra crosstalk. The 2.5-MHz cutoff and many other parameters given are reasonable choices that may change somewhat once the system is tested. Another possibility is to bandlimit the I channel in the NTSC encoder to 0.5 MHz since virtually no NTSC decoders use more than 0.5 MHz at this time and the extra I channel bandwidth can only cause crosstalk in NTSC consumer receivers. With equal-bandwidth (0.5 MHz) I and Q channels in the NTSC encoder, the 2.5-MHz cutoff illustrated in Fig. 7 will result in no cross luminance from Ih and Qh.

A change in the encoder and decoder (illustrated in Figs. 5, 6, 8, and 9) is possible to simplify the system by taking the output of the 1050-line source in Figs. 5 and 8 and transforming to a Y, I, Q signal format immediately. Then the I and Q signals could be bandlimited to 4 MHz (twice the 2-MHz transmission bandwidth) rather than 15 MHz for R, G, B signals (twice the 7.5-MHz Y bandwidth) for a source with a 30-Hz frame rate. Anti-aliasing and interpolation filter implementation in these channels is simplified because of the lower-frequency

operation. Framestore memory is also minimized because of the smaller bandwidth of I and Q.

## III. DELIVERY SYSTEMS

### *3.1 AM broadcast*

The front end of the HDTV set consists of a tuner, IF amplifier, and video detector. These functions can be handled in three basic ways, as illustrated in Fig. 10. Figures 10a and b represent the case where the two 6-MHz channels are adjacent to each other, as illustrated in Fig. 4.

Figure 10a illustrates a single wideband tuner and IF amplifier that is wide enough to pass the entire HDTV spectrum. The IF amplifier may be centered around the same IF frequency presently used, or a different center frequency may be chosen. Figure 10b uses a wideband tuner, but splits the spectrum up into two parts: an NTSC IF to receive that portion of the spectrum that is the same as the NTSC signal, paralleled by a second IF specially designed to receive the extra portion of the HDTV signal. After video detection in each channel, the two baseband spectra are added together to produce the baseband spectrum of Fig. 3.

A more versatile but complicated approach is illustrated in Fig. 10c; shown are two separate tuners for each portion of the spectrum, the NTSC portion and the extra portion. After processing by the respective IF amplifiers and detectors in each channel, the two portions of the baseband spectrum are added back together to obtain the spectrum of Fig. 3. If the arrangement of Fig. 10c is used, the extra 6-MHz spectrum for HDTV need not be located at the upper 6-MHz band shown in Fig. 4. It could be a totally separated channel of 6-MHz bandwidth. If the additional HDTV information is carried in the 6-MHz channel directly above the NTSC channel, 40.75 to 35.75 MHz is a likely IF frequency range for the additional IF. However, if the additional HDTV information is one or several channels away, another frequency may be chosen.

### *3.2 Cable*

Modern CATV systems usually use a tree structure that has the capability of handling up to 50 or more NTSC TV channels per cable. Since present channel allocation is 6 MHz, any additional bandwidth requirement results in a minimum of two channels per HDTV signal. This two-to-one reduction in stations would be more acceptable to the CATV industry than HDTV systems requiring more than two 6-MHz channels per signal. It is almost certain that a higher-bandwidth system would cause the CATV industry large problems, since it would

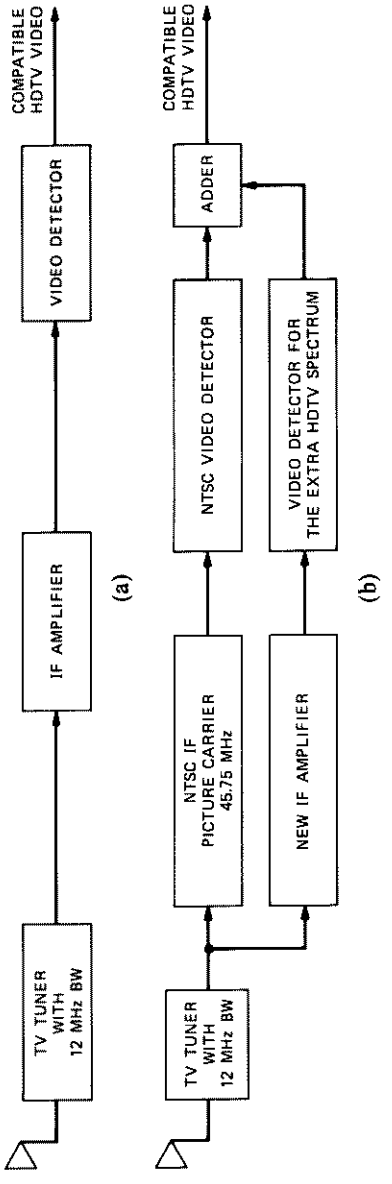



Fig. 10—Front-end alternatives for SLSC HDTV. (a) Single IF amplifier approach. (b) Single-tuner multiple IF amplifier approach. (c) Multiple-tuner approach.

severely limit the amount of programming that could be carried by a CATV system.

Switched video cable systems that have the ability to provide video on demand are just beginning to surface. It is very difficult to switch extremely wideband video; therefore, a reasonably compact HDTV spectrum is needed for this new type of cable system.

This compatible HDTV system has been carefully designed to occupy a 10-MHz baseband bandwidth. It will fully utilize two 6-MHz channels for transmission if desired. Therefore, it should have a minimum negative impact on present and future cable systems.

### *3.3 Satellite*

Direct satellite broadcast to the home is a delivery system that is just about to become important. The modulation format and the frequency allocation of the service will eliminate compatibility with the present NTSC receivers. NTSC compatibility may lose some importance for this delivery system. However, bandwidth is always at a premium for satellite systems; thus the compact baseband as shown in Fig. 3 is very important. Also, the ability to modulate the HDTV signal into two NTSC channels could be advantageous.

### *3.4 Prerecorded*

Even when the prerecorded media such as video tape and video disc are used as a delivery system, bandwidth is still important. It is much easier to record a 10-MHz baseband signal than a 30-MHz signal. Consequently, a tape recorder for the 10-MHz signal should be more economical. There may be a considerable problem with making a video disc unit that handles a 30-MHz baseband signal and still has sufficient playing time per disc.

## IV. COMPARISONS

### *4.1 Vertical resolution comparisons*

The vertical resolution, $Rv$, of a TV system is given by eq. (1). For the current NTSC system, the following is obtained:

$$Rv = (525 - 2\text{x}21)0.65 = 313 \text{ lines.} \tag{5}$$

This assumes a nominal Kell factor of 0.65. If the same Kell factor and 21-line vertical interval is applied to the NHK system, the result is 704 lines. For the Dortmund system using USA scanning standards and the proposed split-luminance, split-chrominance (SLSC) compatible system, the result should be the same as the NTSC with a Kell factor approaching unity.[4] Therefore, the vertical resolution should approach 483 lines. These comparisons together with modifications of the BBC and IBA systems for the USA scanning standards are shown

in Table I.[2,3] (Note that there are actually two Dortmund systems described in the literature.[4] The system considered here is the diagonal sampling approach that provides both increased vertical and horizontal resolution over the present standard. The other system does not provide increased horizontal resolution.)

### *4.2 Horizontal resolution comparisons*

Table I also contains a comparison of some HDTV systems with respect to horizontal resolution, crosstalk, and bandwidth requirements adapted to the NTSC environment where appropriate. The NTSC system is used as a reference. The horizontal resolution of the NTSC system ranges from 240 lines to 336 lines. The NHK system should be capable of approximately 30 lines/MHz based on eq. (2). This predicts a horizontal resolution, $Rh$, of approximately:

$$Rh = (30 \text{ lines/MHz})(20 \text{ MHz}) = 600 \text{ lines.} \quad (6)$$

The Dortmund system approach applied to the NTSC system should have the ability to reproduce the same horizontal resolution as vertical. Further, the vertical resolution should approximate the number of active scan lines. Therefore, it should result in 480 lines.[4]

### *4.3 Crosstalk*

Crosstalk is a very important factor in the image quality of a video system. There are three types of component crosstalk in present television standards. They are cross luminance (crosstalk of chrominance into luminance), cross color (luminance crosstalking into chrominance), and chrominance to chrominance crosstalk. Cross luminance, also called dot crawl, normally shows up in the NTSC picture at the edges of color areas. The high-frequency subcarrier signal appears in the luminance channel as a dot pattern crawling up the edges of color areas. Cross color is most obvious when the scene contains a detailed pattern such as a striped shirt or tweed suit. It is

Table I—HDTV comparison chart

| System | *V* Resolution ($k$ = 0.65) | *H* Resolution | Min. No. NTSC Channels When Broadcast | Compatible | Crosstalk |
|---|---|---|---|---|---|
| NTSC | 313 | 240–336 | 1 | Yes | Bad–Small |
| NHK | 704 | 600 | — | No | Small |
| Dortmund* | 483 | 480 | 1 | — | — |
| BBC* | 313† | 336 | 2 | Can Be | None |
| IBA* | 313† | 336 | 2 | No | None |
| SLSC | 483 | 600 | 2 | Yes | Small–None |

* The values are adapted to the USA scanning standard.
† This value can be made 483 by applying the vertical improvement technique.

seen as a color pattern over the area of detail that obviously does not belong, and destroys the ability to see the detail in that area. Chrominance-to-chrominance crosstalk is less obvious than the other two because it is correlated with the scene. It results in color distortions.

The first two types of crosstalk mentioned above can produce significant picture degradation in the NTSC system unless the luminance bandwidth is sacrificed or a comb filter is used. However, a comb filter tends to produce some luminance degradation of its own. It reduces vertical resolution by averaging two or more lines at a time and rejects diagonal lines in the picture. This situation can be improved for luminance by only comb filtering the high-frequency luminance where the actual crosstalk frequency components occur. Further, a comb filter will produce a dot crawl on horizontal edges of saturated colors in a picture that is not present without it.

As shown in Fig. 7, the low-frequency luminance (Yl) used by the SLSC HDTV system is the lower 2.5 MHz of the NTSC luminance. This portion of the NTSC luminance is not interleaved with any of the Q component of the NTSC color signal. There is some interleaving of the I component of the NTSC signal and the low-frequency luminance, Yl, of the HDTV signal. However, there is some question of whether the full frequency range of the I signal should be transmitted for NTSC receivers since there are no consumer receivers that make use of it, and it can only cause crosstalk in the present NTSC receivers.

Cross luminance in the high-frequency luminance (Yh) can also be minimized. The upper 1.5 MHz of the luminance—the 6-MHz to 7.5-MHz region of the original luminance signal—interleaves with the high-frequency chrominance. This upper end of the luminance ends up occupying the 5- to 6.5-MHz region of the frequency inverted and translated spectrum Y′ in Fig. 3. This region can be comb filtered—rolled off at 6 MHz (480 lines of resolution)—or simply allowed to pass with the small amount of cross luminance mentioned above.

The last alternative is possible with only a small amount of degradation because of the high-frequency nature of the cross luminance and the fact that the chrominance that is talking into the luminance will be down in level compared to the signal that is producing cross luminance in the NTSC system (it is only the 0.5- to 2-MHz region of the chrominance that will contribute to crosstalk here). Since this relatively low-level chrominance will also be producing a much smaller dot pattern in the luminance than the NTSC color subcarrier would cause in the NTSC system, the crosstalk should be much less obvious.

Cross color can be reduced in several ways. A comb filter could be used to remove that portion of the luminance that interferes with the low-frequency chrominance. Alternatively, use can be made of the fact that a portion of the luminance that represents 2.5 to 4.2 MHz of the

NTSC luminance is repeated in the high-frequency luminance portions of the composite HDTV signal. This portion of the high-frequency luminance is free of any interleaved chrominance. However, the corresponding portion of the NTSC part of the signal contains chrominance interleaved with the luminance. These two corresponding portions of luminance can be subtracted, leaving an uncontaminated chrominance signal. Also, the luminance above 3 MHz could be rolled off in the NTSC part of the baseband as another way to eliminate cross color without any degradation to the vast majority of NTSC receivers that do not use a comb filter. Cross color into C′ can be reduced with a comb filter.

### *4.4 Encoding errors*

There are a number of color deficiencies in the NTSC color standard that degrade the quality of the reproduced image.[6-8] The crosstalk problems mentioned in the previous section are a part of these deficiencies. Encoding errors are another facet of these problems. The study of encoding errors or distortions in the NTSC system is an involved topic that will only be briefly mentioned here. The problems center around the fact that part of the luminance is carried by the chrominance when color is broadcasted, and it is aggravated by the nonlinear gamma characteristics of the system.[6] The transmitter corrects for a nominal gamma of 2.2 by raising the signal to the 1/2.2 power so that the overall response is linear. The result is that saturated (vivid) colors lose details. Also, there may be substantial errors in the transients between certain colors (usually complementary colors produce the worst transients). With wideband color signals, there will be fewer errors. The luminance is still carried partly by the chrominance, but the chrominance is now wideband and it degrades the end result less.

## V. SUMMARY

A split-luminance, split-chrominance (SLSC) HDTV system has been described that is NTSC compatible and uses a 10-MHz baseband signal. This baseband signal can be modulated to produce an amplitude-modulated, vestigial-sideband signal in a 12-MHz bandwidth for broadcast. Alternatively, this compatible signal can also occupy two separate 6-MHz channels. Compatibility and bandwidth conservation are two of the most important attributes of this HDTV system. A compatible system is likely to penetrate the market place much more rapidly than a non-compatible system because of the huge investment in NTSC equipment. The two most common delivery systems are broadcast and CATV; both are very sensitive to the bandwidth requirements of a new system.

The present NTSC channels are 6 MHz. Conventional channel allocation forces additional bandwidth to come in increments of 6 MHz. A two-channel, 12-MHz requirement is the largest bandwidth that these systems can reasonably accommodate for this new HDTV service. Therefore, this new compatible system has been designed to occupy a 12-MHz bandwidth when using the present broadcast format of vestigial-sideband amplitude modulation.

The important parameters of this new compatible HDTV system are a horizontal resolution potential up to 600 lines, a vertical resolution potential up to 483 lines, and less crosstalk between the individual components of the color signal compared to the NTSC system. Further minimization of color-encoding errors may be possible by using recently developed processing techniques.[5]

## VI. ACKNOWLEDGMENTS

The author gratefully acknowledges Joseph LoCicero for many useful discussions that aided his understanding of much of the HDTV area. The author expresses his gratitude to Bruce Briley, Kristin Kocan, and Harvey Lehman for their review of this paper.

## REFERENCES

1. T. Fujio, "High-Definition Wide-Screen Television System for the Future-Present State of the Study of HD-TV Systems in Japan," IEEE Trans. Broadcasting, *BC-26* (December 1980), pp. 113–24.
2. C. Sandbank and M. Moffat, "High-Definition Television and Compatibility with Existing Standards," *Tomorrow's Television*, Scarsdale, New York: SMPTE, 1982, pp. 170–85.
3. T. Robson, "A Compatible High Fidelity TV Standard for Satellite Broadcasting," *Tomorrow's Television*, Scarsdale, New York: SMPTE, 1982, pp. 218–36.
4. B. Wendland, "High Definition Television Studies on Compatible Basis with Present Standards," *Television Technology in the 80's*, Scarsdale, New York: SMPTE, 1981, pp. 151–65.
5. D. Richman, "Color Television Receiving System Utilizing Inferred High Frequency Signal Components to Reduce Color Infidelities in Regions of Color Transitions," U.S. Patent 4,181,917, Jan. 1, 1980; "Color Television Receiving System Utilizing Multimode Inferred Highs Correction to Reduce Color Infidelities," U.S. Patent 4,183,051, Jan. 8, 1980; and "Color Television Receiving System Utilizing Inferred High Frequency Signal Components to Reduce Color Infidelities in Regions of High Color Saturation." U.S. Patent 4,245,239, Jan. 13, 1981.
6. K. McIlain and C. Dean, *Principles of Color Television*, New York: John Wiley, 1956, Chapter 11.
7. W. Gibson and A. Schroeder, "Color Television Luminance Detail Rendition," Proc. I.R.E., *43* (August 1955), pp. 918–23.
8. D. Livingston, "Reproduction of Luminance Detail by NTSC Color Television Systems," Proc. I.R.E., *42* (January 1954), pp. 228–34.

## AUTHOR

**Theodore S. Rzeszewski,** B.S.E.E., 1964, M.S.E.E., 1965, University of Illinois; E.E., 1971, D.E. (honorary), 1973, Midwest College of Engineering; Bell Laboratories, 1982—. Mr. Rzeszewski is a member of the Exploratory Wideband Telecommunications Department. His current areas of interest include high-definition television and digital television. Before joining Bell Laboratories, he was Manager of Technical Operations of Matsushita Applied Research.

## LETTER TO THE EDITOR

### Comments on "Three-Stage Multiconnection Networks Which Are Nonblocking in the Wide Sense," by F. K. Hwang*

Two theorems presented in this paper are incorrect. Theorem 2 stated by Hwang can be reformulated as follows.

*Theorem 2:* $\nu(m, n_1, r_1, n_2, r_2)$ *is nonblocking as a* $(q_1, q_2)$ *multiconnection network under Strategy 2, for* $r_1 \geq q_1 q_2 n_1$ *and* $r_2 \geq q_1 q_2 n_2$, *if and only if*

$$m \geq q_1 q_2 (n_2 - 1) + q_1 + n_1 - 1.$$

*Proof*: Sufficiency. Consider the connection of the pair $(x, Y)$. The input switch that contains $x$ can be connected already to at most $n_1 - 1$ distinct middle switches under Strategy 2. Each output switch in $Y$ can be connected already to at most $n_2 q_1 - 1$ distinct middle switches under Strategy 2. Since $|Y| \leq q_2$, we need $q_2$ sets of $n_2 q_1 - 1$ distinct middle switches, if the sets are disjoint. However, under Strategy 2 these sets are not disjoint and the number of middle switches must be replaced by

$$q_2(n_2 q_1 - 1) - (q_1 - 1)(q_2 - 1).$$

Then the total number of middle switches, including one switch that must be available to connect the pair $(x, Y)$, is

$$(n_1 - 1) + q_2(n_2 q_1 - 1) - (q_1 - 1)(q_2 - 1) + 1.$$

After rearrangement we obtain

$$m \geq q_1 q_2 (n_2 - 1) + q_1 + n_1 - 1.$$

The necessity can be proved with ease by presenting the network with

$$m < q_1 q_2 (n_2 - 1) + q_1 + n_1 - 1,$$

in which a new call is blocked.

Similarly, Theorem 3 can be reformulated as follows.

*Theorem 3:* $\nu(m, n_1, r_1, n_2, r_2)$ *is nonblocking as a* $(q_1, q_2)$ *multiconnec-*

* B.S.T.J., *58*, No. 10 (December 1979), pp. 2183–87.

*tion network under Strategy* 3, *for* $r_1 \geq q_1q_2n_1$ *and* $r_2 \geq q_1q_2n_2$, *if and only if*

$$m \geq q_1q_2(n_1 - 1) + q_2 + n_2 - 1.$$

*Proof:* Analogous to the proof of Theorem 2.

Andrzej Jajszczyk
Technical University of Poznań
Institute of Electronics
ul. Piotrowo 3a, 60-965 Poznań
Poland

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Barnes E. S., Sloane N. J. A., **New Lattice Packings of Spheres.** Can J Math 34(1):117–130, 1983.

Barnes E. S., Sloane N. J. A., **The Optimal Lattice Quantizer in Three Dimensions.** Siam J Alg 4(1):30–41, 1983.

Bazelow A. R., Raamot J., **On the Microprocessor Solution of Ordinary Differential-Equations Using Integer Arithmetic (Letter).** IEEE Comput 32(2):204–207, 1983.

Buckingham M. A., Golumbic M. C., **Partitionable Graphs, Circle Graphs, and the Berge Strong Perfect Graph Conjecture.** Discr Math 44(1):45–54, 1983.

Fleckenstein W. O., **Challenges in Software-Development.** Computer 16(3):60–64, 1983.

Garey M. R., Johnson D. S., Tarjan R. E., Yannakakis, M., **Scheduling Opposing Forests.** Siam J Alg 4(1):72–93, 1983.

Grambsch P., **Sequential Sampling Based on the Observed Fisher Information to Guarantee the Accuracy of the Maximum-Likelihood Estimator.** Ann Statist 11(1):68–77, 1983.

Hwang F. K., **Fast Solutions for Consecutive-K-OUT-OF-N-F-System.** IEEE Reliab 31(5):447–448, 1982.

Lin G. K., Menon P. R., **Totally Preset Checking Experiments for Sequential-Machines.** IEEE Comput 32(2):101–108, 1983.

Logan B. F., **An Integral-Equation Connected with the Jacobi-Polynomials.** Siam J Math 14(2):269–322, 1983.

Logan B. F., **Extremal Problems for Positive-Definite Bandlimited Functions. 1. Eventually Positive Functions with Zero Integral.** Siam J Math 14(2):249–252, 1983.

Logan B. F., **Extremal Problems for Positive-Definite Bandlimited Functions. 2. Eventually Negative Functions.** Siam J Math 14(2):253–257, 1983.

Logan B. F., **Hilbert Transform of a Function Having a Bounded Integral and a Bounded Derivative.** Siam J Math 14(2):247–248, 1983.

Mallows C. L., **Minimax Aspects of Bounded-Influence Regression Comment.** J Am Stat A 78(381):77, 1983.

Stroustrup B., **Adding Classes to the C-Language—An Exercise in Language Evolution.** Software 13(2):139–161, 1983.

## ENGINEERING

Alley G. D., **High-Voltage Two-Dimensional Simulations of Permeable Base Transistors.** IEEE Device 30(1):52–60, 1983.

Azbel M. Y., **Resonance Tunneling and Localization Spectroscopy.** Sol St Comm 45(7):527–530, 1983.

Capasso F., **Avalanche Photo-Diodes with Enhanced Ionization Rates Ratio—Towards a Solid-State Photomultiplier.** IEEE Nucl S 30(1):424–428, 1983.

Chemla D. S., Maruani A., **Non-Linear Optical Effects Associated with Excitonic-Molecules in Large Gap Semiconductors (Review or Bibliog.).** Prog Q Elec 8(1):1–77, 1982.

Joy D. C., **Practical Quantitation for Energy-Loss Spectra.** Scan Elec M1982(P2):505–515, 1982.

Levinson M., Temkin H., Bonner W. A., **Electron-Bombardment Induced Defect States in P-InP.** J Elec Mat 12(2):423–432, 1983.

Meyer W. G., Fair R. B., **Dynamic Behavior of the Buildup of Fixed Charge and Interface States During Hot-Carrier Injection in Encapsulated MOSFETs.** IEEE Device 30(2):96–103, 1983.

Salem A. A. M., Cox D. C., **Improving the Power-Added Efficiency of FET Amplifiers Operating with Varying-Envelope Signals.** IEEE Micr T 31(1):51–56, 1983.

Shang H. T., Stone J., Burrus C. A., **Low-OH MCVD Fibers Without a Barrier Layer Using OH→OD-Exchanged Substrate Tubes.** Electr Lett 19(3):95–96, 1983.

## MANAGEMENT/ECONOMICS

Gordon R. H., **Interest-Rates, Inflation, and Corporate Financial Policy.** Brookings P1982(2):461–488, 1982.

## PHYSICAL SCIENCES

Abrahams S. C., et al., **Detwinning Cell for High-Transition-Temperature Atmosphere-Sensitive Ferroelastic Crystals.** J Appl Crys 16(FEB):96–98, 1983.

Alavi K., Temkin H., Wagner W. R., Cho A. Y., **Optically Pumped 1.55-$\mu$m Double Heterostructure $Ga_xAl_yIn_{1-x-y}As/Al_uIn_{1-u}As$ Lasers Grown by Molecular-Beam Epitaxy.** Appl Phys L 42(3):254–256, 983.

Alpern M., Kitahara K., Krantz D. H., **Perception of Color in Unilateral Tritanopia.** J Physl Lon 335(FEB):683–697, 1983.

Bally J., Predmore R., **Ionized-Gas in Active Molecular Cloud Cores.** Astrophys J 265(2):778–790, 1983.

Bally J., Lada C. J., **The High-Velocity Molecular Flows Near Young Stellar Objects.** Astrophys J 265(2):824–847, 1983.

Baraff G. A., Schluter M., Allan G., **Calculation of Si and Al Interstitials in Silicon Using the Cluster-Extended Greens-Function Technique.** Physica B&C 116(1–3):76–78, 1983.

Baraff G. A., Schluter M., Allan G., **Theory of Enhanced Migration of Interstitial Aluminum in Silicon.** Phys Rev L 50(10):739–742, 1983.

Bayman A., Hansma P. K., Kaska W. C., Dubois L. H., **Inelastic Electron-Tunneling Spectroscopic Study of Acetylene Chemisorbed on Alumina Supported Palladium Particles.** Appl Surf S 14(2):194–208, 1983.

Bean J. C., Mcafee S. R., **Silicon Molecular-Beam Epitaxy—A Comprehensive Bibliography 1962–82.** J Physique 43(NC-5):153–172, 1982.

Beck S. M., Brus L. E., **Transient Spontaneous Raman Study of Photo-Ionization Kinetics at the Hydrocarbon Water Interface in Micellar Solutions.** J Am Chem S 105(5):1106–1111, 1983.

Benton J. L., Kimerling L. C., Stavola M., **The Oxygen Related Donor Effect in Silicon.** Physica B&C 116(1–3):271–275, 1983.

Besomi P., Wilson R. B., Wagner W. R., Nelson R. J., **Enhanced Indium-Phosphide Substrate Protection for Liquid-Phase Epitaxy Growth of Indium-Gallium-Arsenide-Phosphide Double Heterostructure Lasers.** J Appl Phys 54(2):535–539, 1983.

Bridges T. J., Patel C. K. N., Strnad A. R., Wood O. R., Brewer E. S., Karlin D. B., **Syneresis of Vitreous by Carbon-Dioxide Laser-Radiation.** Science 219(4589:1217–1219, 1983.

Cooper J. A., Nelson D. F., **High-Field Drift Velocity of Electrons at the $SiSiO_2$ Interface as Determined by a Time-of-Flight Technique.** J Appl Phys 54(3):1445–1456, 1983.

Crane P., Tyson J. A., Saslaw W. C., **Optical-Emission in the Radio Lobes of Radio Galaxies. 2. New Observations of 21 Radio Lobes.** Astrophys J 265(2):681+, 1983.

Cross M. C., Daniels P. G., Hohenber P. C., Siggia E. D., **Phase-Winding Solutions in a Finite Container Above the Convective Threshold.** J Fluid Mec 127(Feb):155–183, 1983.

Cummins S. E., Green S., Thaddeus P., Linke R. A., **The Kinetic Temperature and Density of the Sagittarius-B2 Molecular Cloud From Observations of Methyl Cyanide.** Astrophys J 266(1):331–338, 1983.

Dutta N. K., **Calculation of Auger Rates in a Quantum Well Structure and its Application to InGaAsP Quantum Well Lasers.** J Appl Phys 54(3):1236–1245, 1983.

Edgerton M. E., Handler F. A., Wu C. W., **Methods for the Analysis of the Kinetics of Cooperative Binding of Ligands to a Two-Site Lattice.** Biopolymers 22(2):787–805, 1983.

Eibschutz M., Salomon D., Disalvo F. J., **Mossbauer Studies of the 6.2 keV Gamma-Rays of $^{181}$Ta in Ta-Dichalcogenides.** Phys Lett A 93(5):259–261, 1983.

Eilenberger D. J., Isaacs E. D., Aumiller G. D., **Near-Infrared, Tunable, Oxazine-750 Perchlorate, Synchronously-Pumped Picosecond Ring Dye-Laser.** Opt Commun 44(5):350–352, 1983.

Furukawa T., Date M., Johnson G. E., **Polarization Reversal Associated with Rotation of Chain Molecules in Beta-Phase Polyvinylidene Fluoride.** J Appl Phys 54(3):1540–1546, 1983.

Gallagher P. K., Sinclair W. R., **Oxidation of Polycrystalline Niobium Nitride.** ISR J Chem 22(3):222–226, 1982.

Gurvitch M., Washington M. A., Huggins H. A., **High-Quality Refractory Josephson Tunnel-Junctions Utilizing Thin Aluminum Layers.** Appl Phys L 42(5):472–474, 1983.

Hasegawa A., Wakatani M., **Plasma Edge Turbulence.** Phys Rev L 50(9):682–686, 1983.

Heaven M., Dimauro L., Miller T. A., **Laser-Induced Fluorescence-Spectra of Free-Jet Cooled Organic Free-Radicals—Vinoxy, Cyclopentadienyl, and Benzyl.** Chem p Lett 95(4–5):347–351, 1983.

Helfand E., **On Inversion of the Williams-Watts Function for Large Relaxation-Times.** J Chem Phys 78(4):1931–1934, 1983.

Hertel G., Bishop D. J., Spencer E. G., Rowell J. M., Dynes R. C., **Tunneling and Transport Measurements at the Metal-Insulator-Transition of Amorphous Nb-Si.** Phys Rev L 50(10):743–746, 1983.

Hutson A. R., **Charge Transport by Dislocations in II-VI Compounds.** Physica B&C 116(1–3):650–656, 1983.

Julian M. M., Stillinger F. H., Festa R. R., **The 3rd Law of Thermodynamics and the Residual Entropy of Ice: "Stillwater" or $\Delta S^{H_2O}_{f,T=0} = 0$.** J Chem Educ 60(1):65–66, 1983.

Khanarian G., Schilling F. C., Cais R. E., **Kerr Effect and Dielectric Study of Polyvinyl-Chloride and its Oligomers.** Macromolec 16(2):287–291, 1983.

Kim S., et al., **Resonance Raman-Spectra and Electronic-Structure of the $\mu$-Disulfur-Bridged Diruthenium Complex $[(H_3N)_5RuSSRu(NH_3)_5]^{4+}$ and Related Systems.** J Am Chem S 105(3):336–339, 1983.

Kimerling L. C., **Defects in Semiconductors—1982 (Editorial).** Physica B&C 116(1–3):1–3, 1983.

Kimerling L. C., Benton J. L., **Electronically Controlled Reactions of Interstitial Iron in Silicon.** Physica B&C 116(1–3):297–300, 1983.

Lamola A. A., et al., **Spectral Study of the Photochemistry of Dipyrrole Models for Bilirubin Bound to Human-Serum Albumin.** Photochem P 37(3):263–270, 1983.

Le H., Rosenbluh M., Lax B., Miller T. A., **Observation of $^1$P-$^1$D, $^1$P-$^{1,3}$F, and $^1$P-$^{1,3}$H Motional Stark-Effect-Induced Anticrossings in $^4$He: Determination of Zero-Field Level Separations.** Phys Rev A 27(2):895–905, 1983.

Lin W., Hill D. W., **Oxygen Segregation in Czochralski Silicon Growth.** J Appl Phys 54(2):1082–1085, 1983.

Lines M. E., Eibschutz M., **Correlation-Effects in the Mossbauer-Spectra of Amorphous Metallic Magnetic-Materials.** Sol St Comm 45(5):435–439, 1983.

Macrander A. T., Johnston W. D., **Deep Levels in InGaAsP/InP p+n Diodes Grown by Vapor-Phase Epitaxy.** J Appl Phys 54(2):806–813, 1983.

Manzione L. T., Osinski J. S., **Predicting Reactive Fluid-Flow in RIM and RTM Systems.** Mod Plast 60(2):56–57, 1983.

Mills A. P., **Measurement of the Decay-Rate of the Positronium Negative-Ion.** Phys Rev L 50(9):671–674, 1983.

Muhlhausen C. W., Serri J. A., Tully J. C., Becker G. E., Cardillo M. J., **Scattering of N-2 From Ag(001).** ISR J Chem 22(4):315–320, 1982.

Pearsall T. P., Carles R., Portal J. C., **Single Longitudinal-Mode Optical Phonon-Scattering in $Ga_{0.47}In_{0.53}As$.** Appl Phys L 42(5):436–438, 1983.

Pearsall T. P., Eaves L., Portal J. C., **Photo-Luminescence and Impurity Concentration in $Ga_xIn_{1-x}As_yP_{1-y}$ Alloys Lattice-Matched to InP.** J Appl Phys 54(2):1037–1047, 1983.

Pelleg J., Murarka S. P., **Phosphorus Distribution in $TaSi_2$ Films by Diffusion From a Polycrystalline Silicon Layer.** J Appl Phys 54(3):1337–1345, 1983.

Shank C. V., Greene B. I., **Femtosecond Spectroscopy and Chemistry.** J Phys Chem 87(5):732–734, 1983.

Silfvast W. T., Wood O. R., **Gain Scaling of Short-Wavelength Plasma-Recombination Lasers.** Optics Lett 8(3):169–171, 1983.

Sternheim M., Vangelder W., Hartman A. W., **A Laser Interferometer System to Monitor Dry Etching of Patterned Silicon.** J Elchem So 130(3):655–658, 1983.

Stolen R. H., Mollenauer L. F., Tomlinson W. J., **Observation of Pulse Restoration at the Soliton Period in Optical Fibers.** Optics Lett 8(3):186–188, 1983.

Sturge M. D., Cohen E., Logan R. A., **Dynamics of Intrinsic and Nitrogen-Induced Exciton Emission in Indirect-Gap $Ga_{1-x}Al_xAs$.** Phys Rev B 27(4):2362–2373, 1983.

Tewksbury S. K., **Formation of High-Density Electron-Hole Plasma in Silicon Metal-Oxide-Semiconductor Transistors Below 25-Degrees-K.** J Appl Phys 54(2):868–875, 1983.

Thompson M. O., Mayer J. W., Cullis A. G., Webber H. C., Chew N. G., Poate J. M., Jacobson D. C., **Silicon Melt, Regrowth, and Amorphization Velocities During Pulsed Laser Irradiation.** Phys Rev L 50(12):896–899, 1983.

Wagner W. R., Schumaker N. E., Zilku J. L., Anthony P. J., Swaminathan V., **Lattice-Parameter Changes in $Al_{0.39}Ga_{0.61}As$ Due to O, Ge, Si, and S Doping.** J Elchem So 130(3):670–675, 1983.

Watson E. A., Gibbs H. M., Mattar F. P., Cormier M., Claude Y., Mccall S. L., Feld M. S., **Quantum Fluctuations and Transverse Effects in Superfluorescence.** Phys Rev A 27(3):1427–1434, 1983.

Wood T. H., White J. C., Thacker B. A., **Ultraviolet Photo-Decomposition for Metal-Deposition—Gas Versus Surface Phase Processes.** Appl Phys L 42(5):408–410, 1983.

Yan M. F., Cannon R. M., Bowen H. K., **Space-Charge Distributions Near Interfaces During Kinetic Processes.** J Appl Phys 54(2):779–791, 1983.

Yan M. F., Cannon R. M., Bowen H. K., **Space-Charge, Elastic Field, and Dipole Contributions to Equilibrium Solute Segregation at Interfaces.** J Appl Phys 54(2):764–778, 1983.

## SOCIAL AND LIFE SCIENCES

Breitmeyer B. G., Kropfl W., Julesz B., **The Existence and Role of Retinotopic and Spatiotopic Forms of Visual Persistence.** Act Psychol 52(3):175–196, 1982.

Eisinger J., Flores J., **Cytosol-Membrane Interface of Human-Erythrocytes—A Resonance Energy-Transfer Study.** Biophys J 41(3):367–379, 1983.

Graedel T. E., Mcrae J. E., **Total Organic-Component Data—A Study of Urban Atmospheric Patterns and Trends—Reply (Discussion).** Atmos Envir 17(3):670–671, 1983.

Nelson W. L., **Physical Principles for Economies of Skilled Movements.** Biol Cybern 46(2):135–147, 1983.

Reed A. V., **Parallel Models of Associative Memory—Hinton, G. E., Anderson, J. A. (Book Review).** Cont Psycho 28(2):137–139, 1983.

Smith E.R., Miller F. D., **Mediation Among Attributional Inferences and Comprehension Processes—Initial Findings and a General-Method.** J Pers Soc 44(3):492–505, 1983.

## SPEECH/ACOUSTICS

Sondhi M. M., Resnick J. R., **The Inverse Problem for the Vocal-Tract Numerical-Methods, Acoustical Experiments, and Speech Synthesis.** J Acoust So 73(3):985–1002, 1983.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Introduction

By L. SCHENKER*

(Manuscript received April 18, 1983)

Since the earliest days of telephony telephone traffic measurements have been needed to determine the proper quantities of circuits, operators, and switching systems. In this context "proper" is defined as the efficient and effective utilization of operating factors that provide good service to customers at the lowest possible cost.

The Total Network Data System (TNDS), comprising thirteen different operations systems, is a very large, complex, coordinated set of computer systems developed by Bell Laboratories and now used throughout the Bell Operating Companies. The systems that comprise the Total Network Data System collect, validate, process, archive, and retrieve the traffic data required to fill the entire spectrum of user needs, from near-real-time network management and performance monitoring, through weekly/monthly provisioning and administration, to long-range engineering and planning. The TNDS maintains interfaces with the many telecommunications systems, as well as with operations systems such as the Central Office Equipment Engineering System (COEES), which depend on traffic data.

As the size and complexity of the telephone network expanded, a need evolved for larger quantities of more accurate and timely traffic

* Bell Laboratories.

Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

data. Early measurements of busy equipment were made manually (switch counts, peg counts, etc.) and processed with the aid of desk calculators. The mechanical registers of the 1940s and 1950s evolved into camera register recorders and traffic usage recorders, which partially automated the process. In the mid-1960s, data were being keypunched and then summarized and processed on the large billing computers already in use by comptrollers' organizations. In the early 1970s the advent of the minicomputer facilitated on-line data collection and real-time processing to support such functions as network management. By the mid-1970s, a variety of computer systems collected and processed traffic data to support trunk and switching engineering, administration, and network management. These systems have evolved into a tightly coupled family of operations systems, TNDS, which have been implemented and updated in all the Bell Operating Companies. The systems operate on dedicated minicomputers and large general-purpose computers. The TNDS collects, processes, and utilizes traffic data to provide excellent, efficient, economical telephone service.

The TNDS was developed, used, and enhanced during the thirteen years from 1969 to the present. It has become an indispensable element of Bell operating activity. Without the extensive mechanization provided by the TNDS, it is inconceivable that we could have accommodated the explosive network growth, reconfiguration, and churning. Without the TNDS, operating and investment costs would have risen and service would have been degraded.

This special issue of the *Journal* addresses the TNDS from many points of view. The first two articles describe the TNDS environment and objectives and outline the system plan. The third article describes the conceptual framework and theory upon which the TNDS is built. The eight succeeding, more detailed, articles describe the functions performed by the TNDS and the component operations systems that have been developed to support these functions. The final article, prepared by an employee of Southern Bell, describes the TNDS from an operating telephone company perspective.

As we contemplate the future, we envision continued evolution in two areas: Modifications will be required to keep pace with new services and new technology, to provide interfaces with new telecommunications equipment, and to facilitate new network requirements (such as Dynamic Non-Hierarchical Network Routing); and enhancements will be needed to improve efficiency and make the systems user-friendly. These enhancements will include simplified architecture and new computing facilities, enhanced data communications among the TNDS elements, interactive update features, consolidation of record bases, and on-line user documentation.

This special issue of *The Bell System Technical Journal* appears at an appropriate time—the end of an era. The individual articles reflect the way in which business was conducted prior to 1982. As a result of divestiture, responsibility for much of the TNDS will be transferred to the Central Services Organization, which will be owned and operated by the seven regional Bell Operating Companies. However, the TNDS will continue to provide the independent regional companies with primary traffic data in the future, as it has to the Bell System in the past, without, we hope, missing a beat during the transition.

## AUTHOR

**Leo Schenker,** B.S. (Civil Engineering), 1942, University of London; M.S., 1950, University of Toronto; Ph.D., 1954, University of Michigan; Bell Laboratories, 1954—. After joining Bell Laboratories, Mr. Schenker was involved in the development of telephone station equipment, including *TOUCH-TONE*® dialing and the *TRIMLINE*® phone. In 1968, he was promoted to Director of the Military Electronic Technology Laboratory in North Carolina and, in 1971, became Director of the Loop Maintenance Systems Laboratory, which was closely involved with the development of new software and hardware systems aimed at reducing the expense of maintaining the customer's telephone service. In 1978, he became Director of the Loop Systems Engineering and Methods Laboratory and, in 1979, he was made Director of the Customer Network Operations Systems Engineering Center. In 1980 Mr. Schenker was appointed Executive Director of the Central Office Operations Division. Currently, he is Executive Director, Network Systems Planning Division. Mr. Schenker has been awarded seven patents in connection with *TOUCH-TONE* dialing, *PICTUREPHONE*® meeting service, and *TOUCH-A-MATIC*® repertory dialer. Fellow, IEEE; member, Sigma Xi, Phi Kappa Phi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Environment and Objectives

By A. L. BARRESE,* D. E. PARKER,† T. E. ROBBINS,* and L. M. STEELE*

(Manuscript received January 19, 1983)

Planning for computer-based tools to help in the measurement and analysis of network traffic data began in the late 1960s. During this same period, the rapid introduction of computer technology was changing the character of the network. The network was becoming more efficient and economical but, at the same time, more sophisticated and complex. This intensified the need to provide timely and complete information to those responsible for the management, administration, engineering, and planning of the network. The computer-based systems that were developed to meet this need are collectively called the Total Network Data System (TNDS). This paper discusses the operating environments of the network and telephone company, and provides a framework for the remainder of the papers in this issue.

## I. INTRODUCTION

Managing, engineering, and planning the telecommunications network are essential tasks for the future health and vitality of the network. These complex tasks cannot be performed effectively without detailed data about network traffic. The Bell System has mechanized the collection and processing of such data with a family of computer-based systems collectively called the Total Network Data System

* Bell Laboratories. † AT&T.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

(TNDS). This paper provides an overview of the telecommunications network environment, introduces the need for network traffic data in the operating environment of the telephone company, and briefly discusses the primary mechanization objectives for collecting and processing the network traffic data through the TNDS.

## II. THE TELECOMMUNICATIONS ENVIRONMENT

To appreciate the need for network traffic data, it is necessary to have a general understanding of the telecommunications network.

### *2.1 Network description*

In the most general sense, a network can be defined as a set of nodes interconnected with links. We can further define the telecommunications network (hereafter referred to as "the network") both on the basis of its function and its physical characteristics. From a functional standpoint, the network carries a variety of telecommunications traffic (e.g., voice, data) between a number of stations that can be connected on demand, or that are permanently connected. From the physical standpoint, the network consists of switching systems (nodes), transmission facilities (links), and stations (sources and receptors of traffic) that are connected together in an organized and controlled manner. These two views are complementary and together provide a framework for understanding aspects of telephone engineering and operations.[1]

Let us examine the physical elements of the network more closely. To construct a telecommunications network that would allow complete, direct interconnection of all stations would be both impractical and cost prohibitive. Therefore, every large communications network is based on the principle of shared facilities, whereby facilities that can be shared among different elements of traffic are utilized extensively when designing and building the network. Central offices are entities built to provide switchable interconnection of customer stations. A central office allows each of its customers, with a single pair of wires connected to the office (i.e., the customer's loop facilities), to talk to any other customer served by that office. Such a central office, having customer station equipment directly connected to it, is said to serve "local" traffic from those stations.

In general, however, customer stations being connected are served by different central offices. Therefore, central offices are connected to one another by transmission paths called trunks, so that customers in one office can reach customers in another. The network of trunks interconnecting central offices is referred to as the Interoffice Facility Network. To enable all stations in the network to be interconnectable while making efficient use of network equipment and facilities, switching and trunking arrangements employ a hierarchical network config-

uration and the principle of automatic alternate routing.[1] [A new traffic routing technique called Dynamic Non-Hierarchical Routing (DNHR), planned for deployment in the intercity network beginning in 1984, promises significant cost savings in the network.]

The hierarchical network configuration provides for the collection and distribution of traffic and permits switching systems to be completely interconnectable. The hierarchical aspect prevents "loop arounds" that might otherwise occur when calls are automatically alternate routed through the network. Each switching system is given one of five classifications based on the highest switching function performed within the hierarchy, its interrelationship with other switching systems, and its transmission requirements.

With the automatic alternate routing principle, a call that encounters an "all trunks busy" condition in the interoffice facility network on the first route tested is automatically offered sequentially to one or more alternate routes for completion, if such alternate routes are possible for that item of traffic. Arrangements, defined by the Network Routing Plan, that dictate the final routing of traffic are called "homing arrangements." Central offices whose only function is to route calls through the hierarchy are called toll offices. These offices do not directly serve customers (i.e., do not have any connecting loop facilities). Regional Centers, the switching systems at the top of the switching hierarchy, are examples of toll offices. Many central offices perform a tandem switching function (i.e., route traffic by simply interconnecting interoffice trunk groups). These offices may perform a pure tandem switching function or they may also function as local or toll switching systems.

### *2.2 The concept of service*

Because only a small percentage of telephone customers originate calls at any time, the amount of equipment needed to carry the actual simultaneous traffic is only a fraction of that needed to carry simultaneous traffic from all customers. If we install a very limited amount of equipment, there is no assurance that we can meet service demands satisfactorily at all times. Therefore, we can anticipate that a percentage of calls will not be completed (i.e., will be "blocked") during peak traffic periods [e.g., during the busiest hour(s) of the day, during the busy season (busiest three months) of the year]. When calls are "blocked" too frequently, calling customers perceive service as unsatisfactory. Conversely, if we have too much equipment in the network, too much of it will remain unused even during peak traffic periods, which is not cost effective.

The proportion of calls blocked during the busy hour of a switching system is an index of the grade of service rendered. We define the

grade of service as the probability that a certain percentage of calls originated during the busy hour will be blocked from utilizing the installed equipment and network facilities. There are strong justifications for providing a high grade of service (i.e., low probability of calls being blocked). For example, when a customer must dial a number more than once because of failure to properly connect, it generates additional traffic, and during peak periods this further aggravates poor service conditions. Acceptable service levels are determined by analyzing both network traffic data and customer reactions.[2] Objective grades of service, or service objectives, are established to balance customer service and network cost. It is the effort to strike a cost-effective balance between providing service and utilizing network equipment that imposes the need for network traffic data.

### *2.3 Requirements for network data*

There are four key network functions that require network traffic data: network management, network administration, network design, and long-range network planning. The function of network management is to control network traffic overloads by distributing loads among circuits and equipment to meet customer service demands in a way that is best from a total network viewpoint. To do this, network management requires a near-real-time view of the traffic in the network. This is made possible through analysis of network traffic data.

The function of network administration is to control the assignment of lines and trunks to take maximum advantage of the installed equipment in the central office for serving the offered call traffic. This involves implementing a plan to spread the office load efficiently over all equipment, as well as to monitor the current load, service levels, and office capacities. A major task within network administration is to use traffic data to monitor the flow of traffic through the central office and to detect changes in office performance (e.g., service degradation) or in offered load. Network administration includes the task of providing sufficient, accurate network traffic data for all functions.

The function of network design is to estimate where, when, and how much equipment will be required within a five-year period so that the necessary additional equipment (i.e., relief equipment) can be ordered and installed in time to satisfy the service objectives.[4] This activity requires data that reflect traffic volumes being carried by existing equipment, as well as knowledge of equipment capacities.

The function of long-range network planning is to determine the most economic growth and replacement strategies for the network to meet estimates of future demand. This future demand is estimated using current traffic load trends and marketing information. The basic output of this function is a broad view of network topology 20 years

into the future. Section 3.3 further discusses these key network functions relative to telephone company operations.

In addition to the network functions discussed above, the Bell Operating Company (BOC) marketing organizations request network traffic data to aid in the administration and provisioning of each individual customer's telecommunications service. Due to such factors as increased availability of Stored Program Control (SPC) features and vertical services, increased data sophistication of subscribers and increasing competitive pressures, the number and frequency of these traffic data requests have been increasing consistently. Regulatory needs for these data have also appeared. For example, to support requests for rate increases before regulatory bodies, the BOCs conduct very detailed traffic studies. Regulatory study data requests tend to be complex and difficult to anticipate. These marketing and regulatory needs are only examples of a growing family of special applications that require network traffic data. Though they don't constitute a basic network function, they do represent a significant environmental element that must be considered when planning for the collection and processing of traffic data.

Another function requiring traffic data is operator services force administration. This function involves forecasting the load that will be offered in each half-hour and determining the operator force necessary to carry that load at an objective grade of service. This requires data on traffic volumes, service, and force levels. Because this function has limited interaction with the above-defined key network functions, its data tend to follow a separate but somewhat parallel flow that will not be discussed in this article.

### *2.4 Network data*

The fundamental types of traffic measurements include:

1. Event counts—These measurements simply represent the number of occurrences of an event that occurred in a specific time interval (e.g., hour ending 11:00). "Offered" calls are termed *peg* counts, while "blocked" calls are termed *overflow* counts.

2. Usage—These measurements represent the estimated amount of time an equipment component was busy during a specific time interval, generally an hour. In electromechanical (EM) switching systems, usage is typically obtained through a separate, special measuring device, the Traffic Usage Recorder (TUR). In SPC switching systems, usage measurements (as well as the others) are generated internally by the switch.

3. Delay—This type of measurement usually indicates, on the average, how long a particular type of event would have been delayed, say by an all-servers-busy condition, if it had chosen to wait. Some

different measures of delay include: average delay, average delay of events delayed, probability of delay, and probability of delay exceeding some specified time interval.

4. Status—This measurement, used for network management and administration, indicates the presence or absence of a condition (e.g., equipment outage, severe overload) and is usually in the form of a lamp indication.

With few exceptions, measurements are made at the location of the switching system. All switches make some traffic measurements internally. The interfaces for data collection from EM systems are traffic register leads for peg count, overflow, grouped usage and delay, as well as status-indicating leads. In the case of EM systems, it is usually necessary to provide special traffic measurement equipment (e.g., the TUR) at the switch location. This equipment, in conjunction with special software in the data collection machine, makes possible the gathering of more detailed individual circuit usage data rather than circuit group data. This concept, used for selected EM offices to help increase data accuracy, is called Individual Circuit Usage Recording (ICUR). For SPC switching systems, traffic measurements and status indications are collected by the basic internal programs. The information, equivalent to that stored in EM registers, is retained in memory until the accumulated results are read out on schedules established by the users.[4]

In the Bell System, the volume of network traffic data processed is overwhelming. It was estimated that in the average BOC, over 50 million pieces of traffic data per week were collected and processed in 1980 to support the management, administration, design, and long-range planning of the network.[3] Because of the size, complexity, and importance of the network data job, the management of the data has become a major internal function of the BOCs.

Next we discuss another fundamental element of the network data system environment—the BOC operational environment.

## 3. THE OPERATIONAL ENVIRONMENT

### *3.1 Operations planning*

Planning for modern computer-based tools to help in the measurement and analysis of network data began in the late 1960s. During this same period, the rapid introduction of computer technology was changing the character of the network. The network was becoming more efficient and economical but, at the same time, more sophisticated and complex. The people responsible for the network's day-to-day operation (i.e., management, administration, engineering and planning) needed more timely and complete information. Bell System management recognized that the largely manual methods in use during

the 1960s could not keep pace with the emerging operational needs of the 1970s and 1980s.

Three properties of network data made the planning of computer-based support tools particularly challenging.

1. The data are almost useless until they are processed.
2. A much larger volume of data must be gathered than will be used ultimately because of the way that data are generated and output by the SPC switch and the external measurement devices of the EM switch.
3. The same basic data must be summarized in varying ways to meet the needs of a diverse family of users (see Section 3.3).

Measurement, data processing, and user applications must, therefore, be considered together in planning for the mechanization of an "end-to-end" flow of data.

The formulation of such an end-to-end view of the data flow requires an understanding of the functions performed by various work groups and how these groups relate to one another. This allows the formulation of requirements that describe how the work groups can best be supported by computer-based tools. Finally, the tools must be related to one another and to the elements of the telecommunications network. This concept of a process view, encompassing functions to be performed by people and computers, has been one of the principles used in the design of the individual components, termed *Operations Systems (OSs)*, that comprise the Total Network Data System (TNDS).[4]

This view of the overall data flow not only facilitated the allocation of functions to the specific TNDS-related OSs, but also resulted in the identification of functions that must be performed by people. Consequently, work groups, termed *operations centers*, were established to oversee the operation of the elements of TNDS and ensure the integrity of the process. The allocation of functions among people and computers, together with a specification of how they interact to accomplish an overall process, was among the initial efforts in a general discipline known as *operations planning*.

Since then, the concept of operations planning has been expanded to encompass virtually all areas of telephone company operations. It is an organized, disciplined procedure to provide an integrated operations structure. This, in turn, will permit the Bell System to meet customer and market needs while minimizing the operational cost through applied computer technology. These efforts have produced a family of operations plans that guide AT&T, Bell Laboratories, and the telephone companies in the planning and deployment of the over 100 currently available OSs.[5]

Planning for the evolution of the network data collection process is now one element of an overall plan known as the Total Network

Operations Plan (TNOP). The planning process is not static; efforts are now under way to define the directions in which operations centers, systems, and processes should evolve to meet the future needs of the telecommunication network and the people who operate it.

### *3.2 Modeling the environment*

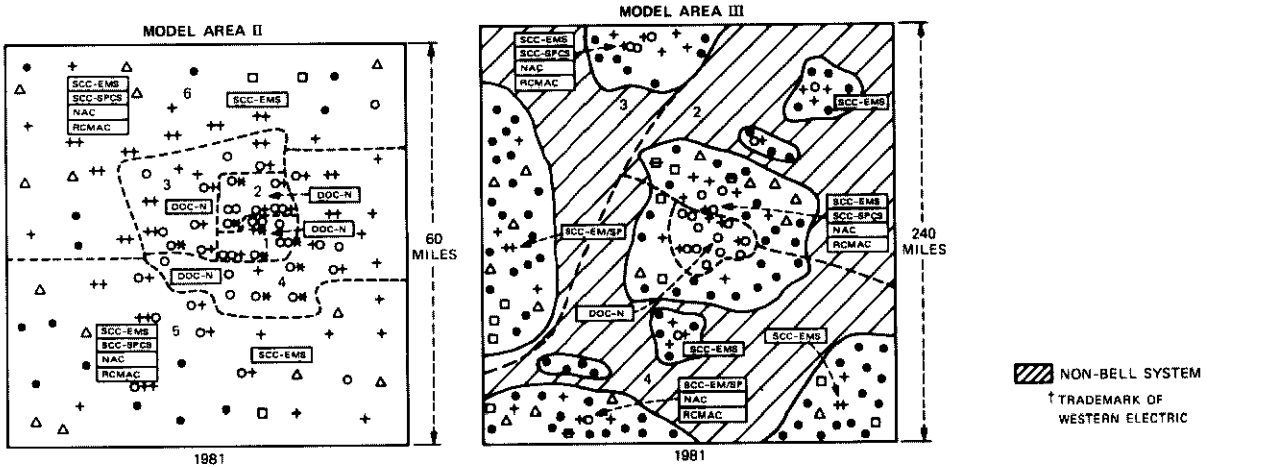
A key factor that influences the plan for mechanized support of the network traffic data acquisition and analysis process is the telephone company operating environment. We can gain insight into the requirements imposed by this environment by analyzing models that represent typical situations and by conducting detailed field studies and experiments. These types of activities ensure that both the operations strategy and the operations system developments meet the needs imposed by both the evolving telecommunications and operations environments.

Many existing operating areas in the Bell System have similar characteristics and face similar network operations requirements. Thus, four basic types of model areas can be constructed to represent the various geographies, populations, and network equipment configurations that presently exist in the Bell System.

One, termed Model Area II, represents a large, densely populated metropolitan region, such as Detroit or Cleveland, with about two million main stations. Another, Model Area III, characterizes a state or portion of a large state containing over one million main stations and having an urban center with over two hundred thousand main stations. Indiana and Georgia resemble this model. Figure 1 shows maps of the geographic distribution of local switching equipment and key line operations centers in these two model environments. Table I shows a summary of physical plant statistics in each model environment. Of the two remaining models, Model Area IV characterizes large, primarily rural states, while Model Area I describes New York City and its immediate vicinity.

While line operations functions can be reasonably studied on an area basis, the span of other operations functions, such as trunking network design, often covers more than one operating area. Hence, the model areas were combined to form model BOCs. Figure 2 illustrates the typical deployment of selected operation functions for one such model company. It consists of one Model Area II and one Model Area III and resembles a single-state company like the Illinois Bell Telephone Company. Similarly, model companies can be combined to form a model territory to study toll operations.

These models provide a basis for quantitatively evaluating the effect(s) of the environment on alternative mechanization strategies. Among the key elements that influence the design and evolution of network data mechanization are:




**LOCAL SWITCHING PROFILE**

| DISTRICT | 1 | 2 | 3 | 4 | 5 | 6 | TOTAL |
|---|---|---|---|---|---|---|---|
| SWITCH TYPE | | | | | | | |
| * 1XB | 3 | 2 | 1 | 3 | | | 9 |
| + 5XB | 2 | 3 | 12 | 12 | 17 | 19 | 65 |
| o 1 *ESS*† | 8 | 8 | 8 | 10 | 7 | 1 | 42 |
| △ 2 *ESS* | | | | | 7 | 6 | 13 |
| □ 3 *ESS* | | | | | 1 | 2 | 3 |
| • SXS | | | | | 8 | 6 | 14 |

**CHARACTERISTICS**

3600 MI$^2$ AREA
URBAN AND SUBURBAN AREAS
ONE NPA
NO INDEPENDENTS

**LOCAL SWITCHING PROFILE**

| DISTRICT | 1 | 2 | 3 | 4 | TOTAL |
|---|---|---|---|---|---|
| SWITCH TYPE | | | | | |
| + 5XB | 4 | 10 | 11 | 12 | 37 |
| o 1 *ESS* | 11 | 5 | 3 | 2 | 21 |
| △ 2 *ESS* | | 5 | 5 | 5 | 15 |
| □ 3 *ESS* | | 1 | 3 | 5 | 9 |
| ⊖ RSS | | 2 | | 1 | 3 |
| • SXS | | 18 | 26 | 45 | 89 |

**CHARACTERISTICS**

58,000 MI$^2$ AREA
TYPICALLY A STATE
TWO NPAs
MEDIUM-SIZE URBAN AREAS
LARGE AREAS SERVED BY INDEPENDENTS

**SWITCH AND OPERATIONS CENTERS**

DOC-N – DISTRICT OPERATIONS CENTER-NETWORK (LOCATED SCCs, NAC, RCMAC)
EMS – ELECTROMECHANICAL SYSTEMS
EM/SP – COMBINED FUNCTIONS
NAC – NETWORK ADMINISTRATION CENTER
NPA – NUMBERING PLAN AREA
RCMAC – RECENT CHANGE MEMORY ADMINISTRATION CENTER
RSS – REMOTE SWITCHING SYSTEMS
SCC – SWITCHING CONTROL CENTER (MAINTENANCE)
SPCS – STORED PROGRAM CONTROL SYSTEM
SXS – STEP BY STEP
XB – CROSSBAR

Fig. 1—Two representative model operating areas.

Table I—Physical plant statistics in two model areas

| | Model Area II (1.93M Main Stations) | | Model Area III (1.39M Main Stations) | |
|---|---|---|---|---|
| | No. of Entities | Main Stations (1000's) | No. of Entities | Main Stations (1000's) |
| Local switching | | | | |
| No. 1 Crossbar | 9 | 189 | 0 | 0 |
| No. 5 Crossbar | 65 | 738 | 37 | 450 |
| Step-by-Step | 14 | 28 | 89 | 281 |
| 1 *ESS** | 42 | 904 | 21 | 542 |
| 2 *ESS* | 13 | 65 | 15 | 97 |
| 3 *ESS* | 3 | 6 | 9 | 18 |
| RSS[1] | 0 | 0 | 3 | 2 |
| DCO[2] | 0 | 0 | 0 | 0 |
| DRSS[3] | 0 | 0 | 0 | 0 |
| Total local entities: | 146 | | 174 | |
| Wire centers: | 98 | | 159 | |
| Toll and local tandem switching | | | | |
| 4 *ESS* | 1 | | 1 | |
| 1 *ESS*[4] | — | | 6 | |
| No. 4A Crossbar | 3 | | 2 | |
| Crossbar Tandem | 5 | | 1 | |
| No. 5 Crossbar with tandem features[4] | 3 | | 6 | |
| SXS with tandem[4] features | — | | 7 | |
| Trunks (1000's)[5,6] | 162 | | 118 | |
| Special Service circuits (1000's)[6] | 114 | | 80 | |
| T-Carrier channels (1000's)[6] | 165 | | 101 | |
| N-Carrier channels (1000's)[6] | — | | 16 | |
| High-Capacity Carrier channels (1000's)[6,7] | 21 | | 26 | |

1. Remote Switching System.
2. Digital Central Office.
3. Digital Remote Switching System.
4. Local switches with toll or tandem features are included in local entity count.
5. Interoffice circuits. Intraoffice circuits are not included.
6. Count includes intra-area plus one-half interarea circuits.
7. Includes low-capacity carrier systems multiplexed directly to broadband carrier.

* Trademark of Western Electric.

1. Number, type, and geographic distribution of data sources (i.e., switching systems)

2. The relative scope, number, and geographic distribution of operations centers that require mechanized support.

These factors, together with the nature of the functions performed by the various operations centers (see Section 3.3) and insights gained through detailed field studies, form the foundation for a network traffic data mechanization plan for the 1980s. Section 3.3 discusses the key network traffic functions in the BOCs.

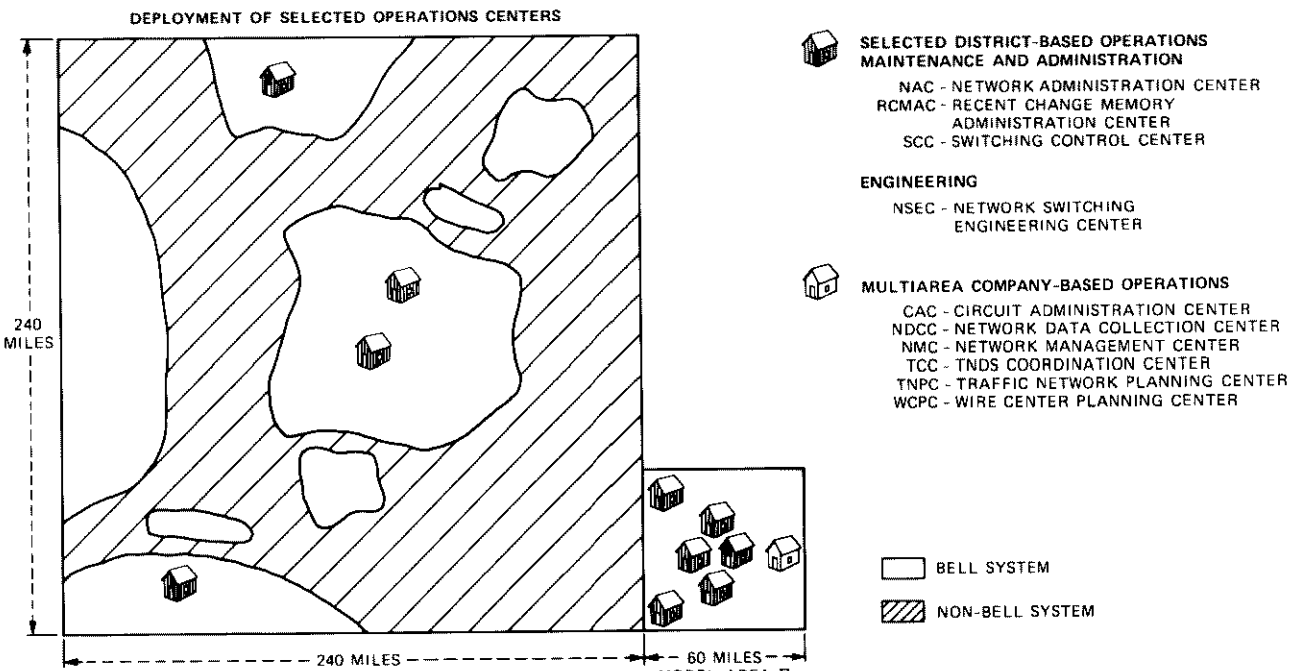



Fig. 2—A model operating company.

### *3.3 Managing, engineering, and planning the network*

The BOC operations centers are responsible for network management, network administration, network design, and long-range planning. This section describes how the centers interact and how the information flow among centers is managed.

#### *3.3.1 BOC Operations Centers*

***3.3.1.1 Network management.*** The Network Management Center (NMC) keeps the network operating efficiently when unusual traffic patterns or equipment failures would otherwise result in congestion. The NMC analyzes network performance and prepares contingency plans for situations such as peak days, telethons, and major switching system failures. The NMC also routinely monitors near-real-time network performance data to identify abnormal situations. Once an abnormal situation is detected, appropriate network management controls are implemented. When the critical situation has passed, the network management controls are removed. For further details, see the papers entitled "Network Management" and "National Network Management" later in this issue of the *Journal.*

***3.3.1.2 Network administration.*** The Circuit Administration Center (CAC) and the Network Administration Center (NAC) perform network administration. The NAC is responsible for optimum loading, balancing, and utilization of installed central office equipment. It performs daily surveillance of central office and connecting trunk groups to ensure that service objectives are being met. In addition, the NAC reviews profiles of office load relative to profiles of anticipated capacity growth. It initiates corrective actions to deal with current as well as potential service problems by working with the Network Switching Engineering Center (NSEC) to initiate work orders to increase equipment in service (Section 3.3.1.3).

The CAC ensures that in-service trunks meet current as well as anticipated customer demands at acceptable levels of service. There are two activities, planned and demand servicing, that the CAC performs to discharge this responsibility. In planned servicing, the CAC compares current traffic loads with the forecasted loads for the upcoming busy season. If the loads are consistent, the CAC issues the orders to provide the forecasted trunks. When inconsistencies are found, the CAC examines the variation, develops a modified forecast for the busy season, and issues orders (if appropriate) based on the new forecast. In demand servicing, the CAC reviews weekly traffic data to identify trunk groups that imminently need augmenting beyond what the forecast states, and issues the appropriate trunk orders.

***3.3.1.3 Network design.*** The network design function is performed by the CAC and NSEC work centers. The CAC projects current traffic

loads for a one- to five-year period based on estimates of main station and call rate growth trends, and also develops corresponding forecasts of network trunk requirements. The NSEC is responsible for developing an analogous forecast of loads for traffic-sensitive switching equipment, setting office capacities, and determining relief size and timing.

The forecasts developed specify the amount of equipment (switching and trunking) that will be needed for the busy seasons of each of the next few years. Based on these forecasts, construction budget dollars are committed. The CAC and NSEC ensure that these forecasts are consistent with long-range planning.

***3.3.1.4 Long-range planning.*** Long-range network planning is performed by work centers such as the Traffic Network Planning Center (TNPC) and the Wire Center Planning Center (WCPC).

The TNPC conducts studies to determine the most economic growth and replacement strategies for the network to meet its estimates of future demand. The estimates of future demand are based on current traffic load trends and marketing information. The plans developed start with the present environment and provide corporate guidance for future network configurations over a 20-year period. This planning process includes the tandem switching systems, operator services networks, trunks interconnecting all switching systems, and switching terminations to accommodate the trunks.

The WCPC conducts similar studies for the local wire center areas. The WCPC planning process includes the local switch and its interaction with other network elements (such as the subscriber network and interoffice facility network).

The basic output of the long-range planning function is a broad view of the future network topology. This includes the numbers, types, and locations of switching systems and the homing arrangements. The resulting long-range plan embodies various routing rules, such as whether local and toll traffic will flow through the same tandems in a metropolitan network and what sequences of alternate routes will be used. Such planning does not involve commitments to spend money but rather to ensure that the long-term consequences of current decisions are foreseen and that the evolution of the network proceeds smoothly and economically.

### *3.3.2 Work center interrelationships*

Figure 3 illustrates how these work centers interact to perform network management, network administration, network design, and long-range planning.[6]

The TNPC and WCPC provide the CAC and NSEC with the fundamental office and network evolution plans (20-year horizon).

The CAC and NSEC review current trends in traffic offered to in-service network and switching equipment. They develop forecasts of specific equipment needs for the next one to five years, consistent with the fundamental plans. The forecasts result in construction budget allocations and equipment orders.

The CAC and NAC review the current level of traffic offered to in-service equipment. The CAC and NAC identify current (within a year) equipment rearrangement and growth needs. Accordingly, equipment is rearranged, ordered, and installed.

The NMC, with the help of the CAC and NAC, prepares contingency plans for situations such as peak days, telethons, and major switching system outages. The NMC routinely monitors the load, in near-real time, to identify unusual traffic patterns and equipment failures that have resulted or will result in congestion. The NMC will implement the network management plans, as required, to deal with the problems.

As an example of the interrelationships of these centers, Table II outlines the role of the NMC, NAC, and CAC relative to the surveillance of network service. The primary differences that can be noted are: the time frame of interest and action, the geographic domain (network), and the network components of interest.

Table II—Summary of relative service surveillance roles in BOC Operations Centers

| Center | Surveillance Objective | Network Traffic Data Used |
|---|---|---|
| Network Management Center | Monitors traffic congestion in a portion of the network (e.g., a numbering plan area) and initiates controls to maximize call completion during times of overload or equipment failure | 5- to 20-minute traffic data and 30-second network status data for selected central office equipment and trunk groups in selected central offices |
| Network Administration Center | Monitors load and service status for each switching system and its trunk groups in a central office district, determines if service objectives are being met, detects or is informed of potential or actual service-affecting problems, initiates corrective action when necessary, and verifies that problems are being resolved | Hourly and weekly central office equipment and trunk group traffic data and selected status indicators |
| Circuit Administration Center | Monitors traffic loads on the message trunk network in an operating area or company, determines the need for near-term trunking rearrangements and additions to resolve conditions that are network service-affecting and that are expected to persist | Weekly and longer-term summaries of trunk group traffic data |

### *3.3.3 Management of information flow*

Coordinating the flow of information to and among the work centers is a complex task for which work centers have been established. The Network Data Collection Center (NDCC) coordinates schedules and ensures that the requested data are collected and distributed appropriately (i.e., in the most timely and cost-effective manner). Operations systems have been developed for internal operations and information exchange. The NDCC coordinates the execution of these systems on a day-to-day basis.

Up to this point, we have discussed the network and the BOC operational environment. Now let us turn our attention to the functional objectives of a data system to mechanize the network data process.

## IV. MECHANIZATION OBJECTIVES

Until the early 1970s, the collection and processing of network traffic data by the BOCs was a combination of manual and semi-mechanized processes. A number of environmental changes made this type of network data environment inadequate to meet the needs of the business. These influences included:

1. Growing sophistication and complexity in an increasingly SPC network
2. Need for immediate information for network management and for more complete and timely information for all network-related functions
3. Increasing regulatory and competitive pressures.

Accordingly, the trend was toward mechanizing the network data process. The overall objective of this effort has been to automate the process of collecting and summarizing network traffic data so that BOC decision makers have adequate quantities of timely and accurate information to administer and engineer the network. To be effective, the system must be robust to varying BOC environments (e.g., organizational responsibilities, mechanization deployment) and must be expandable to meet the growing processing needs of the BOCs. The system must offer a manageable and cost-effective implementation for the BOCs in the context of their operations.

Before describing the implementation of the Bell System's computer-based network data system, let us look at the basic data function objectives that such a system must meet to satisfy the needs of users performing key BOC traffic functions.

### *4.1 Data collection*

After the traffic measurements have been taken by the switching

system or by special measurement equipment located near the switch, it is desirable to collect the measurements at a centralized location where large-scale data processing can be brought to bear. The system must be capable of collecting network traffic data from all types of switching entities—EM and SPC systems, local, tandem, and toll switches.

Because the equipment collecting the data is centralized and has access to the switching systems it serves, it is practical to delegate measurement control (e.g., turning on a TUR to begin taking measurements on the basis of a collection schedule) and network management control (e.g., signaling to a switching system to cancel alternate routing) to the same collection equipment.[4]

### *4.2 Data administration*

The system must also support a number of "data administration" activities. Measurements to be collected from switching systems must be scheduled so they are not collected during periods when they are not required (e.g., during low traffic periods). In addition, the system must allow users to control the data processing schedules (e.g., hours of the day for which the data collected are to be summarized into the various reports).

The measurements must also be linked to records that associate each measurement with the equipment being measured. In addition, records of the individual switching system characteristics (e.g., inventories of installed central office equipment) must be maintained. No less important than the network traffic data, complete and accurate records are absolutely required by the system to transform raw traffic measurement data into useful information. The generation and maintenance of these records are major tasks that must be supported by the system.

At times, more data are collected than are actually needed due to the way measurements are accumulated and "packaged" by the switch or by the special measurement equipment. The system should identify and remove these unneeded data as early as possible to avoid wasteful processing.

It is necessary for the system to validate data (i.e., inspect them to determine if they truly reflect network traffic conditions that they are intended to measure) soon after collection to avoid processing incorrect data and allowing them to contaminate the associated good data.

As a final data administration task, the system must ensure that measurement data are distributed as appropriate between the different processing elements of the system as well as to any other Operations Systems that require those data.

### 4.3 Report generation

To present BOC users with useful information, the system must perform a number of processing and reporting tasks. As Section 3.3 showed, there are four key traffic-related functions that depend on network traffic data. A primary factor in designing a successful network data system is satisfying the needs of the BOC people performing these functions—both in the type of information reported and the timeliness with which it is made available.[4]

Reported information should be in formats that allow users to quickly identify and interpret the information. The system should provide a set of predefined, standard reports as well as capabilities for selective, demand query of information to allow users to obtain information in formats customized to their special needs. In addition, the system must allow the users to control the routing of processed outputs and must be able to deliver processed results to users in the form to be used in their local operations (i.e., paper reports, microfiche reports, hard copy or cathode ray tube terminals, status board).

Automatic, more detailed, validation should be performed to augment that carried out shortly after data collection. Validation reports should be generated to help users detect suspect data. In addition, the system must allow users to highlight and exclude information that is judged suspect so that it is appropriately treated in affected reports.

The system should provide access to network information in time frames consistent with the spectrum of user functions (see Fig. 3). For instance, the system must provide five-minute network load information in an immediate time frame as the basis for real-time management of the trunk network. However, the system must summarize and maintain information for a year or more to support central office and facility engineering and long-range planning. In general, the system should store data and results for periods of time based on anticipated information usage patterns and economic considerations.

The system must collect and report information on its performance (e.g., data collection availability) to allow the BOCs to promptly identify and resolve system performance problems and to otherwise manage the system.

### 4.4 Robustness

A mechanized network data system must continue to evolve as new switching machines are introduced, as new network services and features demand the processing of new measurements, as new provisioning or network design algorithms are formulated, and as new functions arise for the network data system. In addition to simply evolving to meet new processing needs, the network data system must do so at or near the time of introduction of new technology to help minimize the impact on Bell System operations.

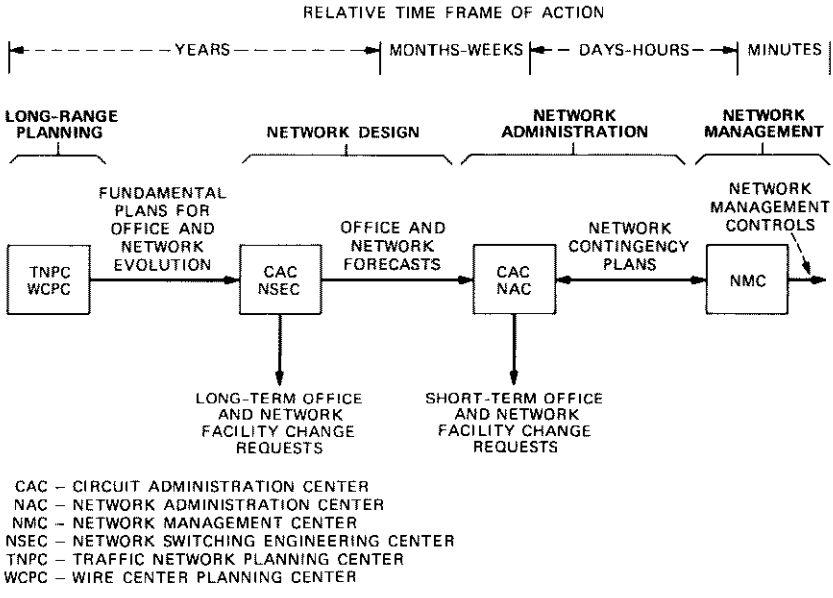



Fig. 3—Work center interrelationship.

## V. CONCLUSION

The Bell System has implemented a mechanized process known as the Total Network Data System to meet the needs we have outlined. This TNDS consists of a set of subsystems, each performing part of the overall process. The formulation of and adherence to a system plan has helped assure that all parts of TNDS have been available when required to support the network management, administration, design, and long-range planning needs of the BOCs. The paper that follows discusses this "TNDS System Plan" and the component TNDS subsystems that have evolved.

## REFERENCES

1. *Engineering and Operations in the Bell System,* Murray Hill, N.J.: Bell Laboratories, Indiana Publication Center, 1977.
2. David Talley, *Basic Telephone Switching Systems,* Rochelle Park, N.J.: Hayden Book Company, 1969.
3. M. M. Irvine, "An Electronic Watchdog for the Network," Bell Lab. Rec., *58,* No. 8 (September 1980), pp. 267–73.
4. M. M. Buchner and W. S. Hayward, "Total Network Data System," Proceedings of the Eighth International Teletraffic Congress, November 1976, Melbourne, Australia.
5. M. M. Buchner, Jr., "Planning for the Evolving Family of Operations Systems," Bell Laboratories Record, *57,* No. 5 (May 1979), pp. 118–24.
6. R. F. Grantges, G. W. Riesz, and B. E. Snyder, "Evolution of the Total Network Data System," Proceedings of the Ninth International Teletraffic Congress, October 1979, Porremolinof, Spain.

## AUTHORS

**Anthony L. Barrese,** B.E., 1970, M.E.E., 1971, Ph.D., 1978, Stevens Institute of Technology; Bell Laboratories, 1969—. Mr. Barrese was first involved in the development and testing of the Safeguard antiballistic missile system. He then worked as a systems engineer, developer, and tester on the Total Network Data System project. He supervised a group with methods development responsibilities for the Circuit Administration Center. He is currently Supervisor, System Architecture Group, with long-range planning responsibilities for the Operations Systems Network. Member, IEEE, Tau Beta Pi.

**Donald E. Parker,** Southern Bell Telephone and Telegraph Company, 1967–1968; South Central Bell Telephone Company, 1968–1976; Bell Laboratories, 1976–1979; American Telephone and Telegraph Company, 1979—. Mr. Parker's involvement with TNDS began in 1974 when he was assigned responsibility for training and implementation of Total Network Data System/Equipment (TNDS/EQ) in the Alabama Area of South Central Bell. It has continued through a rotational assignment in the Personnel Subsystems Group at Bell Laboratories, where he was responsible for user documentation and consultation for Traffic Data Administration System (TDAS), Common Update (CU), and Individual Circuit Analysis Program (ICAN). He is currently assigned to the Network Services Department of AT&T as Project Manager for TNDS/EQ and Stored Program Control Systems Central Office Equipment Reports (SPCS COER).

**Thomas E. Robbins,** B.S., 1968 (Mathematics), Westminster College; M.S., 1971 (Operations Research), New York University; Bell Laboratories, 1968—. Mr. Robbins has been involved in various aspects of the Total Network Data System since 1972, including both development (SPCS COER) and systems engineering (SPCS COER, TNDS/EQ) assignments. He is presently Supervisor of the TNDS Planning Group responsible for planning the feature evolution of the TNDS elements to support new network switches and services. Member, IEEE.

**Larry M. Steele,** B.S.E.E., 1969, University of Kentucky; M.S.E., 1971, Princeton University; AT&T, 1975–1977; Bell Laboratories, 1969–1975, 1977—. Mr. Steele joined Bell Laboratories in 1969 and employed modeling and simulation techniques to evaluate the accuracy and vulnerability of proposed military missile detection systems. In 1971 he developed the requirements and the software for a system to analyze the performance of No. 1 Crossbar switching entities. He was subsequently given responsibility for the determination of downstream analysis requirements and the formulation of an administrative plan for the individual circuit usage recording feature of a standard Bell System traffic data acquisition system. In 1975 Mr. Steele went on a rotational assignment with the General Departments of the American Telephone and Telegraph Company, where his responsibilities included project management and methods development for a family of traffic data acquisition and analysis systems. Upon his return from AT&T in 1977, he conducted a comparative analysis of system architectures for providing a standard methodology for gathering detailed data relating to point-to-point traffic loads. Between 1978 and early 1981, Mr. Steele supervised a group responsible for studying and recommending improved methodologies and mechanized capabilities in the area of telephone network administration. The group was also

responsible for synthesizing administrative plans for incorporation into the Total Network Operations Plan (TNOP). Currently, Mr. Steele is the Head of the Network Operations Planning Department, which is responsible for planning for the integration and use of computer-based systems to support network-related operations functions.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

# *Total Network Data System:*

# System Plan

By M. S. HALL, JR.,* J. A. KOHUT,* G. W. RIESZ,* and J. W. STEIFLE†

(Manuscript received July 23, 1982)

The Total Network Data System (TNDS) is a coordinated family of operations systems that work together to mechanize telephone traffic data gathering and reporting. This paper describes the component parts of the TNDS, their functions, their interactions, and their inputs and outputs. The paper also presents the managerial view of TNDS: its product, its size, the number of people involved, how the overall project is organized and coordinated, and how the system evolves to meet the needs of a continually changing environment.

## I. INTRODUCTION

The planning effort for traffic data collection and processing for what was to become the Total Network Data System (TNDS) began in the mid-1960s. This plan gradually evolved as the initial steps were taken to mechanize portions of the traffic data collection and data processing functions for the Bell Operating Companies and AT&T Long Lines.

This paper is divided into two parts. The first part, Sections II and III, presents an architectural view of the Total Network Data System.

* Bell Laboratories. † American Bell.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

The second part of the paper (Section IV) is written from a managerial point of view.

The first part begins (in Section II) with an overview of the current configuration of the TNDS, and the decisions that led to the current architecture; it concludes with brief descriptions of the functions of the individual subsystems and summaries of their interactions. (Subsequent papers in this volume describe the subsystems and their internal operations in detail.) Section III describes TNDS operations from a different point of view, i.e., by following the flow of a representative data item through the system and describing the database update process.

The second part, which begins with Section IV, describes the product delivered to the end users, including software, documentation, and user support, as well as the organization of the development and maintenance effort, and the planning, requirements, development, test, and delivery cycle. Section IV also examines the need to enhance and update TNDS subsystems to meet changing needs, using the integration of the 5 *ESS** switching system as an example. This article ends with a brief look at possible future directions.

## II. CURRENT CONFIGURATION

This section introduces the architecture of the Total Network Data System and briefly describes the individual component systems and their interactions. Subsequent articles in this issue of the *Journal* describe the elements of TNDS in greater detail.

TNDS is now widely deployed throughout the Bell System. All or a portion of TNDS is now in use in every Bell operating company. Of the almost 10,000 switching entities, TNDS collects and processes data for more than 7000. Since the switching entities not now connected are primarily the small electromechanical Community Dial Offices (CDOs), TNDS actually handles more than 90 percent of all Bell System traffic information.

TNDS is a coordinated family of operations systems, which work together to mechanize the data gathering and reporting process. It consists of manual procedures and computer systems that enable operating company managers, network administrators, and network designers to analyze traffic data in a variety of ways.

The development of TNDS has been under way since the mid-1960s. It was the Bell System's first set of integrated operations systems, and has required the planning, development, testing, modification, and introduction of a number of new features and generic programs. This

* Trademark of Western Electric.

systems planning served as the forerunner of the main operations planning activities that have since characterized several plans, including the Total Network Operations Plan (TNOP).

TNDS has now evolved into a system that encompasses 13 major component systems. They are:

- EADAS—Engineering and Administrative Data Acquisition System
- EADAS/NM—Engineering and Administrative Data Acquisition System/Network Management
- TDAS—Traffic Data Administration System
- LBS—Load Balance System
- 5XB COER—No. 5 Crossbar Central Office Equipment Reports
- SPCS COER—Stored Program Control Systems Central Office Equipment Reports
- ICAN—Individual Circuit Analysis
- SONDS—Small Office Network Data System
- CU/EQ—Common Update/Equipment
- TSS—Trunk Servicing System
- TFS—Trunk Forecasting System
- CU/TK—Common Update/Trunking
- CSAR—Centralized Systems for Analysis and Reporting.

Figure 1 shows the overall TNDS architecture.

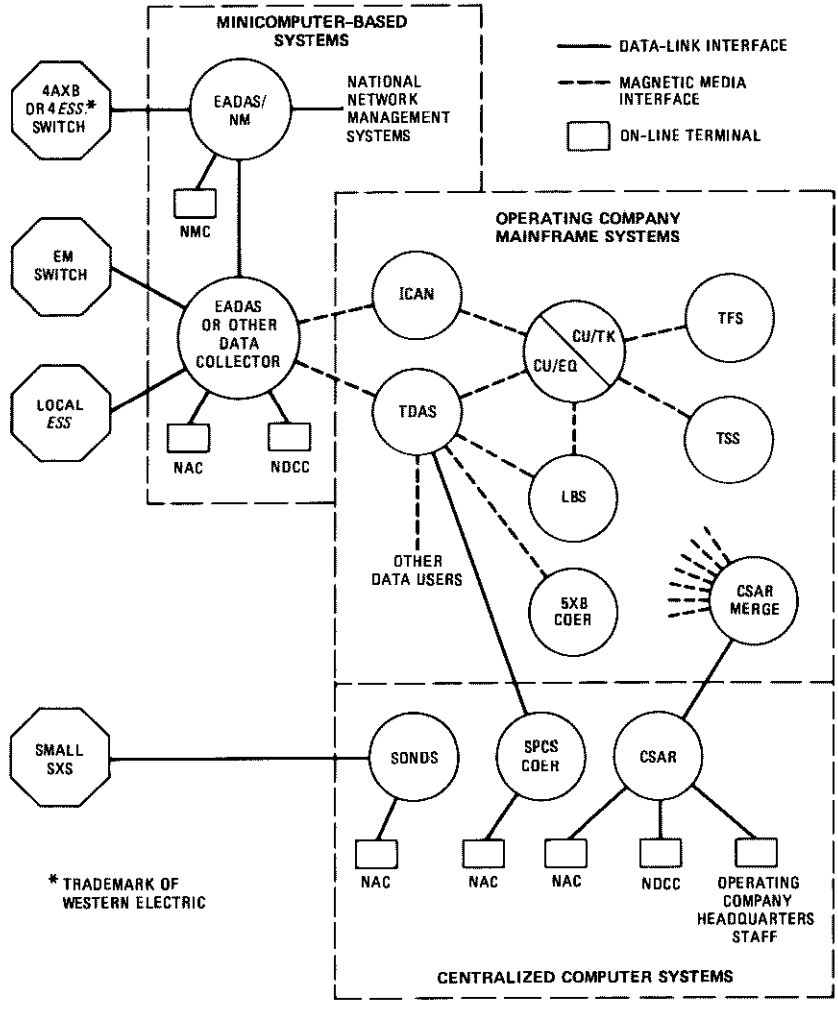
The elements of TNDS are operated in distributed locations. The switching systems, each serving thousands of customers, are generally based in a community's central office. The data collection systems typically employ dedicated minicomputers and gather traffic information from anywhere between 20 and 80 switching systems. Most of the remaining engineering and administrative reporting systems run on either general-purpose operating company computers, or at the AT&T computer center.

Once collected and stored, a number of different systems of the TNDS may process the same items of traffic data to produce a variety of reports.

Despite these distributed operations, the TNDS for any operating company can be thought of as a single, integrated entity designed to provide managers with comprehensive, timely, and accurate information about the network.

### *2.1 Architectural overview*

The Total Network System did not, like Athena from Zeus, spring full blown from the head of a grand planner. Instead it evolved, like most things, from a combination of several individual system developments. R. L. Martin has observed[1] that the architecture of a system




5XB COER – NO. 5 CROSSBAR CENTRAL OFFICE EQUIPMENT REPORTS
CSAR – CENTRALIZED SYSTEMS FOR ANALYSIS AND REPORTING
CU/EQ – COMMON UPDATE/EQUIPMENT
CU/TK – COMMON UPDATE/TRUNKING
EADAS/NM – ENGINEERING AND ADMINISTRATIVE DATA AQUISITION SYSTEM/NETWORK MANAGEMENT
ICAN – INDIVIDUAL CIRCUIT ANALYSIS
LBS – LOAD BALANCE SYSTEM
NAC – NETWORK ADMINISTRATION CENTER
NDCC – NETWORK DATA COLLECTION CENTER
NMC – NETWORK MANAGEMENT CENTER
SONDS – SMALL OFFICE NETWORK DATA SYSTEM
SPCS COER – STORED PROGRAM CONTROL SYSTEMS CENTRAL OFFICE EQUIPMENT REPORTS
SXS – STEP BY STEP
TDAS – TRAFFIC DATA ADMINISTRATION SYSTEM
TFS – TRUNK FORECASTING SYSTEM
TSS – TRUNK SERVICING SYSTEM

Fig. 1—TNDS architecture.

is the product of the history of the organization that builds it, the present and near-present technology, and the intended application. This observation is certainly true for TNDS. Its roots reach back into the 1960s. Its components originated in a number of organizations at

Bell Laboratories and the operating companies. The uses of its output within the operating companies are widespread and diverse. It is, therefore, not surprising that the TNDS architecture (shown in Fig. 1) is not uniform or highly integrated. It is perhaps amazing that it fits together as well as it does.

### *2.1.1 The primal components*

One of the earliest threads leading to TNDS began in 1966. A group of planning department representatives from eight associated companies, along with Long Lines and AT&T, met in New York City and recommended a Bell System effort to centralize the development of a computerized trunk facilities system. Among the components recommended for development were a Traffic Trunk Estimating Subsystem and a Traffic Trunk Servicing System. Development of these two components began shortly thereafter in the Business Information Systems (BIS) area of Bell Laboratories as the Trunk Forecasting System (TFS) and the Trunk Servicing System (TSS). A third program, designed to acquire data from a variety of manual and mechanized sources and to distribute it to client programs, was also included in the planning. This program, called Identify and Edit, later became TDAS.

Identify and Edit served as a data "warehouse." It labeled traffic measurements, stored them, and distributed batches of data to downstream users as they were needed. It and the downstream systems it fed were designed to serve entire operating companies, and in some cases the entire Bell System. These downstream systems were placed on large mainframe computers to take maximum advantage of economies of scale in processing, and to reduce the administrative problems of operating multiple systems. Another factor that led to these systems being company-wide in scope was the need for TFS to design company-wide trunk networks. This capability required a centralized database for trunk traffic measurements. It led to the centralization of the downstream systems with that database. The database, to which was added data for central office equipment, later became the CU/EQ and CU/TK components.

During this early period, the place later taken by EADAS (see Fig. 1) was occupied in planning by the Traffic Data Recording System (TDRS). TDRS was a special-purpose system that collected traffic data from electromechanical switching offices. Also during this period, AT&T planners expected to meet central office equipment data needs by standardizing programs developed by various telephone companies for individual switch types. The programs would receive data from Identify and Edit. Chief among these "Central Office Equipment"

programs was one developed by New Jersey Bell for No. 5 Crossbar offices (it was running in 1968). Later standardized by AT&T and still later revised by Bell Laboratories, this program became the present-day 5XB COER component of TNDS.

The basic allocation of processing functions among these early systems has prevailed until now. TDRS was responsible for the real-time collection of traffic measurements for about 40 electromechanical switching offices. This number of offices was selected to balance the cost of switch-to-TDRS data links with the economics of scale available with centralized processing functions. This basic configuration and size have continued with little change even though the role of TDRS has been taken over by EADAS.

The early downstream systems (TSS, TFS, Identify and Edit, and 5XB COER) were designed for batch processing on mainframe computers. This approach was the prevalent technology for large software systems in that era. It provided an economical way to meet user needs with minimal technical risks. Quick turnaround was not a requirement on the reports produced by these systems. These early systems have evolved and been enhanced over the last fifteen years. LBS and ICAN have been added. However, the fundamental architecture established in the late 1960s for this set of operating company, mainframe-based systems has changed little. Only now are on-line, real-time features being added.

The arrangement of Central Office Equipment Reports (COER) systems within the TNDS architecture was driven by different factors than the other downstream systems. Each type of switching equipment has unique traffic data processing requirements. The switch architectures and service features strongly influence these requirements. This characteristic led to separate COER developments for each switch type. Only recently has the technology needed for a generic COER been pursued.[2] Several approaches were taken by the various organizations that developed the COER modules. As discussed above, 5XB COER is implemented on a mainframe system, and SPCS COER (which is a collection of COER modules) is implemented on a centralized, time-shared system (as is SONDS). The selection of time sharing for these systems is discussed below.

#### *2.1.2 Evolution and integration*

In the early 1970s economical new minicomputer systems with greater flexibility and versatility replaced the specialized TDRS hardware being used for data collection. Some locally developed and vendor-supplied systems were installed for this purpose. The predominant system in use, however, is EADAS, which is a product designed by Bell Laboratories. Being real-time systems, these minicomputer sys-

tems could easily provide the short-term traffic reporting functions that were also required.

Another major evolutionary step was the addition of network management functions in 1975. Network management requires near-real-time data processing and requires an overview of a larger number of switching offices than can be served by one EADAS. However, the EADAS system provided a convenient and economic concentration point for the real-time, local office, traffic measurements needed for network management. Therefore, a decision was made to create a hierarchical architecture with up to six EADAS data collection minicomputers concentrated on an EADAS/NM minicomputer. For 4A Crossbar and 4 *ESS* toll switching offices, however, a direct switch-to-EADAS/NM connection was chosen because the number of these toll offices is relatively small and the data volumes interchanged with them are relatively high. It also provided a higher level of reliability for the communications with these critical switching entities.

Three systems—SPCS COER, SONDS, and CSAR—were implemented on Bell System national-based time sharing systems. All three systems began as experiments conducted at Bell Laboratories to investigate improved traffic data management methods. For SPCS COER and CSAR, centralized time sharing was a means to deploy required on-line processing functions without the need for operating companies to buy additional hardware, or to support software in a number of geographically dispersed sites. These are attractive attributes for an experimental prototype.

SPCS COER and CSAR were standardized on centralized time sharing because it was easier and more economical to evolve the prototype software than it would have been to rewrite it for another computer system and perhaps purchase new hardware. The decision to deploy SONDS on centralized time sharing was more difficult, however.

The SONDS prototype was implemented on a minicomputer system at Bell Laboratories. It was determined that a single computer installation could economically support all the Bell System's small step-by-step offices expected to utilize SONDS. These electromechanical switching offices were expected to be gradually replaced by electronic offices (served by EADAS) over a fifteen- to twenty-year period. The decision therefore was between: (1) standardizing the minicomputer prototype and supporting a centralized, dedicated minicomputer for 15 to 20 years of declining switch population; or (2) rewriting all the prototype software for a general-purpose, time-sharing system. It was decided that users could get better support if the system was on a general-purpose computer. So SONDS joined CSAR and SPCS COER on the AT&T time-shared computer facility.

Integration of TNDS occurred as a series of steps. An early step was linking data collection to downstream processing. Minicomputer-based data collection systems like EADAS wrote traffic measurements on magnetic tapes that could be used as direct input to TDAS. The tape formats used for this link have been changed to improve processing efficiency, but the basic arrangement is still used. Integration of SPCS COER was another major step. Initially SPCS COER received data from punched paper tape via dial-up connections. In 1975, EADAS was upgraded to collect 1 *ESS* traffic data and pass it to TDAS via magnetic tapes. Data transmission from TDAS to SPCS COER was then used to complete the link. Other key integration steps were the automatic input to TFS of trunk group base load data from TSS, and linking the standard CSAR system by means of the downstream CSAR merge program. As part of their initial development, EADAS/NM, ICAN, and LBS were all integrated. TNDS was officially recognized as an integrated system for planning purposes in 1974. This was accomplished by an internal Bell Laboratories memorandum which, in fact, declared it an integrated system and documented its architecture.

The slow pace of evolution from the early architectural choices attests to their workability, but also is evidence of how difficult it is to change established architectures. As discussed at the end of this article, more changes are occurring. They are driven by the needs of users for new processing functions and technological advances that can be effectively incorporated into TNDS. Recently, developers have explored: (1) the use of on-line functions in mainframe systems for database management, report distribution and documentation delivery; and (2) the migration of some functions on centralized time sharing to on-line mainframe systems.

The remainder of this section describes, in more detail, the existing component systems and their current roles within the TNDS architecture.

### *2.2 Component system functions*

To understand the interrelationship of TNDS systems and how they collect and use traffic data, let's look at the four primary functions of TNDS in their general order of occurrence—data acquisition and management, central office equipment reporting, trunk network reporting, and system performance measurement—and at the systems responsible for each function.

#### *2.2.1 Data acquisition collection and management*

Managers need data on network performance and traffic loads carried by trunk groups and switching systems to assess the quality of service and to plan for network growth.

This network information begins as bits of data collected in switching offices. For example, in electromechanical offices a specialized data collection device, called a Traffic Usage Recorder, scans the trunks and other components periodically (typically every 100 seconds) and counts how many are busy. Other traffic data such as peg count and overflow are also collected. In *ESS* offices, a similar process is used, but there is no need for specialized equipment, since traffic data are collected by the switching system's central processor.

***2.2.1.1 EADAS.*** Traffic data are transmitted to the first of the 13 systems of TNDS—the Engineering and Administrative Data Acquisition System (EADAS). EADAS runs on dedicated minicomputers located at an operating company's Network Data Collection Centers. As its name suggests, it is the major data-collecting TNDS system. However, a few operating companies use locally developed or non-Bell vendor-supplied systems instead of or in addition to EADAS for this first step in the TNDS process. In addition, the large toll machines, 4 *ESS* and No. 4 Crossbar [with its associated computer, the Peripheral Bus Computer (PBC)] collect their own data and do not connect to EADAS.

Each EADAS serves up to 80 switching offices and, upon receiving traffic data, performs three basic functions.

1. It processes some data in near-real time (shortly after they are received) to provide hourly and half-hourly reports and a short-term database for network administrators.
2. It collects and summarizes data for processing by remaining "downstream" TNDS systems.
3. It performs on-line surveillance and reporting functions.

EADAS also links other TNDS systems by forwarding traffic data to them via a data link or magnetic tape. Three systems receive these data directly from EADAS.

***2.2.1.2 EADAS/NM.*** Two of three direct recipients of traffic data from EADAS are the Individual Circuit Analysis (ICAN) Program and the Engineering and Administrative Data Acquisition System/Network Management (EADAS/NM). They are the only systems that use data collected and processed by EADAS without first having it formatted by TDAS. ICAN is not a data acquisition system, but rather one of the central office engineering and administrative reporting systems, and is described in that section. TDAS is the third direct recipient.

EADAS/NM is located at operating company Network Management Centers, and uses data received from EADAS or directly from some types of switching systems (e.g., 4A and 4E). It watches switching systems and trunk groups designated by network managers, and reports existing or anticipated congestion on a display board at the center. EADAS/NM also provides information to AT&T Long Lines'

Network Operations Center at Bedminster, N.J. This center is supported by an operations system, the Network Operations Center System (NOCS), which performs functions similar to EADAS/NM, but on a national scale.

*2.2.1.3 TDAS.* The Traffic Data Administration System (TDAS) formats traffic data for use, in turn, by a number of the other "downstream" systems.

But unlike EADAS and EADAS/NM—both of which employ dedicated minicomputers—TDAS runs in an operating company computer center supporting the Network Data Collection Centers. If TNDS were to be pictured as a series of distinct steps, then TDAS can be thought of as the second data acquisition step. Each week it accepts millions of pieces of data from EADAS (or the equivalent of locally developed or vendor-supplied systems) or directly from the large toll machines. In response to requests from downstream systems users, TDAS sorts, labels, stores, and, at the appropriate time, provides the data in the proper format to the engineering and administrative reporting systems. In effect, TDAS acts primarily as a warehouse and distribution facility for traffic data.

TDAS treats its data acquisition job as a basic order/inventory problem. Orders, or Traffic Measurement Requests, for data are manually prepared and sent to TDAS by operating company personnel. These orders are stored in a master database.

As traffic data are received by TDAS, they are matched against the orders held in Common Update. When the data necessary to fill an order have been received, a weekly data summary—either printed or on magnetic tape—is sent to the system that requests it to use in preparing an engineering or administrative report.

Once TDAS has received, formatted, and sent data to the appropriate downstream reporting system, the data acquisition function is complete. At this point, traffic data have moved from the switching system to EADAS and then directly to TDAS and the requesting engineering and administrative reporting system. The next step is to look at how managers employ the TNDS systems to analyze the data that have been gathered.

*2.2.1.4 CU/EQ.* In addition to the above systems, a common record base is used. TDAS and several of the downstream engineering and administrative systems need much of the same record-base reference information. Those data are maintained in a common system, called Common Update/Equipment (CU/EQ), rather than duplicated in each system. The information for each central office includes the configuration of switching equipment as well as specifications of traffic registers. This database is updated as changes occur in the physical arrangements of central offices.

### *2.2.2 Central office equipment reporting*

There are five downstream TNDS engineering and administrative systems that send reports about central office switching equipment to operating company personnel. These systems, which run on either operating company computers or at the AT&T computer center, are the:

1. Load Balance System (LBS)
2. No. 5 Crossbar Central Office Equipment Reports (5XB COER)
3. Stored Program Control Systems Central Office Equipment Reports (SPCS COER)
4. Individual Circuit Analysis (ICAN) Program
5. Small Office Network Data System (SONDS).

The first three receive traffic data from TDAS. ICAN receives data directly from EADAS, but also uses Common Update for some of its reference information. SONDS collects its own data directly from small step-by-step offices.

***2.2.2.1 LBS.*** The Load Balance System, LBS, which is run on the operating company computer, helps assure that the customer traffic load is uniformly distributed over each switching system. Customer lines are connected to the switching system "concentrators," which allow customers to share switching equipment. LBS analyzes traffic data coming to it from TDAS to establish the traffic load on each line group of each switching system. Then, personnel in the Network Administration Center use LBS reports to determine "lightly loaded" line groups to which new subscriber lines can be assigned. This minimizes congestion on a given concentrator. In addition, LBS calculates "load balance indices" for the entire operating company, indicating how effectively each central office has avoided congestion by efficiently distributing traffic.

***2.2.2.2 5XB COER, SPCS COER.*** Two other central office equipment reporting systems—5XB COER for No. 5 Crossbar offices, and SPCS COER for 1, 2, 3, and 5 *ESS* offices—also use traffic data collected by EADAS and formatted by TDAS to support the Network Administration Center. While 5XB COER runs on operating company computers and SPCS COER at the AT&T computer center, the two systems perform similar functions. Both analyze traffic data to indicate the overall load carried and the service provided by the switching systems, and to determine how much of the switching system's capacity is being used. This information helps planners decide when new equipment is needed and answers many other important administrative questions.

***2.2.2.3 ICAN.*** The two remaining central office equipment reporting systems, ICAN and SONDS, do not use TDAS to format their data. ICAN, which receives data directly from EADAS for certain electro-

mechanical offices equipped with an EADAS option called Individual Circuit Usage Recording (ICUR), detects switching system equipment faults by identifying abnormal load patterns on individual circuits. ICAN then produces a series of reports used at the Network Administration Center to analyze individual circuit usage and verify circuit grouping.

*2.2.2.4 SONDS.* SONDS—the fifth downstream central office equipment reporting system—is the only TNDS system that performs a full range of data manipulation functions. It economically provides a number of TNDS features for the smaller electromechanical step-by-step offices.

To do that, SONDS—which runs at the AT&T computer center—collects traffic data directly from offices measured, processes them, and automatically distributes weekly, monthly, exception, and on-demand reports to managers at the Network Administration Center via dial-up terminals.

### *2.2.3 Trunk network reporting*

Three TNDS systems—all run on operating company computers—support trunk servicing and forecasting at the Circuit Administration Center:

1. Trunk Servicing System (TSS)
2. Trunk Forecasting System (TFS)
3. Common Update/Trunking (CU/TK).

Together these three systems are sometimes referred to as Total Network Data System/Trunking (TNDS/TK). TSS helps trunk administrators develop short-term plans to relieve unacceptable blocking on final trunk groups. It processes traffic data supplied by TDAS, and computes the "offered load" for each trunk group. TSS calculates the number of trunks theoretically required to handle that traffic load at a designated grade of service—an objective expressed in terms of percent of blocking. TSS produces weekly reports showing which trunk groups are underutilized and which trunk groups are performing below the grade-of-service objective.

The traffic load data calculated by TSS also help support the trunk forecasting function performed by TFS. Those data, in conjunction with information on network configuration and forecasting parameters stored in Common Update, are used for long-term construction planning. The Trunk Forecasting System uses that information to forecast message trunk requirements for the next five years. These forecasts are a fundamental input to the planning process, which leads to the provisioning of additional facilities.

CU/TK provides a common record base for TSS/TFS. It contains information describing the configuration of the trunk network. The traffic data are provided from TDAS.

### *2.2.4 CSAR system performance measurement*

Centralized System for Analysis and Reporting (CSAR) is the newest TNDS system. It was designed to monitor and measure how well data are being processed through TNDS from beginning to end. CSAR quantitatively measures the accuracy, timeliness, and completeness of the data flow for operating company personnel at the Network Data Collection Center, Network Administration Center, and the Circuit Administration Center. It also supplies sufficient information to locate trouble so corrective action can be taken. It does not presently analyze data from EADAS/NM, SONDS, or TFS.

CSAR is an on-line, interactive system that puts TNDS performance information at the user's fingertips. At the conclusion of each run of a system in TNDS, data required by CSAR are placed in a special file. Then, at an appropriate time, they are transferred to the AT&T computer that contains the CSAR program. CSAR then performs the proper association and analysis of data.

Operating company managers can access the report information from terminals at their own work locations. The reports can be arranged in a number of formats, and can provide details on overall TNDS performance or individual system effectiveness. Reports can be broken down by traffic unit (trunk group or switching system), district, division, or area to help identify and resolve problems at various operating company organizational levels.

## III. HOW IT WORKS

This section will describe the end-to-end data flow of TNDS using two examples. One example will explain the steps necessary to pass *Touch-Tone** dialing originating register data from a No. 5XB office to the 5XB COER system, and the other will focus on the delivery of trunking data from a 1 *ESS* office to TSS. In both cases considerable preparation is necessary in the switching office and in the databases of the affected TNDS subsystems before data flow can begin. This preparatory work will be described first.

### *3.1 Addition of* Touch-Tone *originating registers in a No. 5XB*

Generally, the marketing department would decide that the level of customer demand was sufficient to justify the cost of purchasing and establishing a *Touch-Tone* originating register (OR) group in a No. 5XB office. Following this decision, the Traffic Engineer would engineer the addition using forecasts of usage and would write a traffic order for the new equipment. A copy of the order would be received

* Registered service mark of AT&T.

by Network Administration personnel whose job it is to assign the *Touch-Tone* equipment to measurement devices. After the assignments are made, central office personnel will connect the leads of the Electronic Traffic Data Collection (ETDC) unit to the *Touch-Tone* equipment.

Traffic counts are maintained on a data collection device (DCD). Before data collection was computerized, traffic measurements were accumulated in mechanical registers. The modern equivalent to these registers are the DCDs, which are the software storage locations of the accumulated counts. For our example these storage locations are in the EADAS memory, although many of the subsystems of TNDS use DCD locations to refer to traffic measurements. The DCD is the common thread that links the traffic measurements from the switching office through EADAS and on to other systems.

A typical No. 5XB may require 1500 DCDs. The important measurements collected in DCDs on the *Touch-Tone* group are identified and defined in Table I below. As illustrated in Fig. 2, the next step in the sequence of preparing individual systems is loading the EADAS database (by defining the new DCDs) to accept the new data and to print exception and hourly reports.

The mapping between the ETDC at the switching office and the EADAS is accomplished via the DCDs. Likewise, the DCD is the link between EADAS and TDAS. The user prepares TDAS to receive this new data from EADAS by modifying the CU/EQ database using input transactions. Each TNDS equipment (TNDS/EQ) system except 5XB COER shares some portion of the CU/EQ database; hence, each system is assigned a series of numerically coded transaction commands to allow the database to be changed. As examples, a 750 transaction is used to associate a DCD with a particular measurement type, such as *Touch-Tone* OR peg count, and the days of the week and hours of the day that measurements need to be processed by TDAS are specified on the Traffic Measurement Request (790) transaction.

The CU/EQ database can be modified to enable 5XB COER to begin processing *Touch-Tone* OR data and printing reports. Once this is accomplished, all pieces are in place to permit traffic data to flow

Table I—DCD measurements on the *Touch-Tone* group

| Measurement Type | Definition of Measurement |
|---|---|
| Usage | An estimate of the total amount of time the *Touch-Tone* ORs were busy for any reason. |
| Peg Count | The total number of attempts made to access the *Touch-Tone* ORs. |
| All *Touch-Tone* Registers Busy | The number of attempts to seize *Touch-Tone* ORs when the entire group was busy. |
| Maintenance Usage | The total amount of time the *Touch-Tone* ORs were busy for maintenance reasons. |

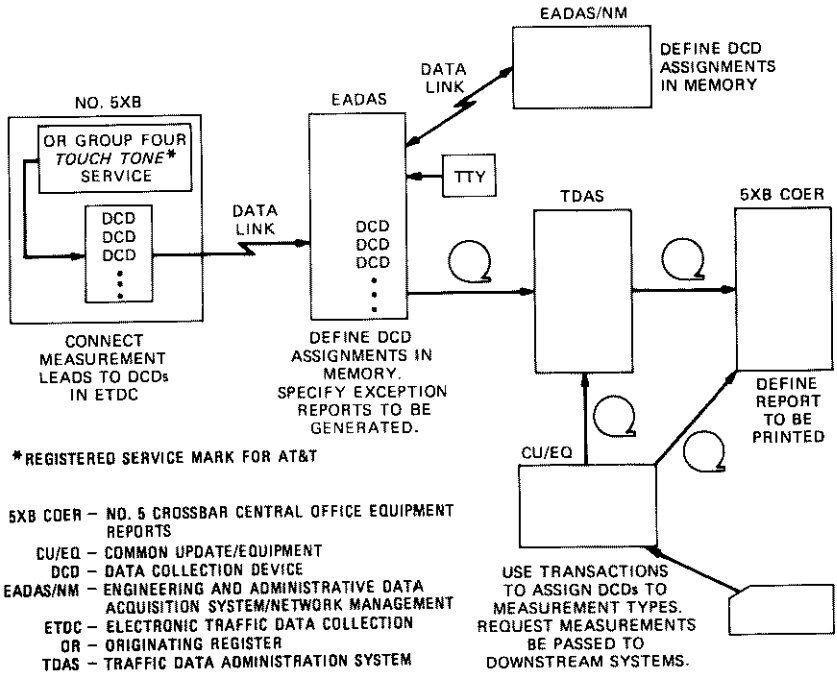



Fig. 2—Database synchronization for *Touch-Tone* originating register data.

from the physical equipment to the Network Administrator and Traffic Engineer. However, two potential users of this data have not yet been included. Those users receive data by an indirect path rather than the main-line path described thus far. Personnel at the network management center receive traffic measurements on critical switching office components in near-real time. To prime EADAS/NM to receive certain of the *Touch-Tone* OR measurements directly from EADAS, the EADAS/NM database is modified by identifying the new measurements (DCDs) that will be available at the EADAS.

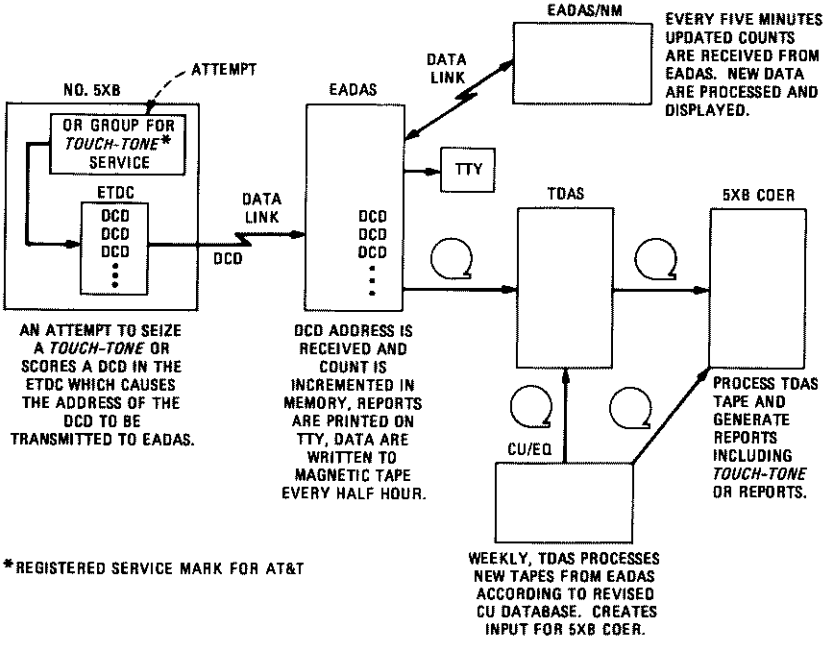
Selected data from each of the 27 EADAS/NM systems are sent to the NOCS computer every five minutes. These data describe the status of toll switching offices and trunk groups, but not the status of the local network. Thus, the new No. 5XB data would not be forwarded to NOCS. Data from toll switching offices are sent to NOCS by identifying it in the EADAS/NM database as being of NOCS interest. An EADAS/NM program automatically notifies NOCS that new data will be sent to it by forwarding a database map to NOCS. This program synchronizes the NOCS database to the EADAS/NM database.

The final system to receive these new measurements is CSAR. However, since CSAR measures overall performance rather than specific items, no database changes are required.

Now that all record base components are ready, the flow of new data can be monitored through each system. Figure 3 graphically illustrates this flow.

Data collection starts with the customers; their *Touch-Tone* calls cause attempts (peg counts), usage, and, during high load periods, an occasional all-busy condition on the *Touch-Tone* ORs. (Maintenance usage is generated by the central office work force when they remove an OR from service to maintain or repair it.) The ETDC is signaled each time an attempt is made to seize a *Touch-Tone* OR. The ETDC, in turn, codes a message, which is sent to EADAS when a seizure is attempted. This coded message is simply the DCD for that measurement. The ETDC, located at the switching office, is connected to the EADAS, which generally is physically remote from the ETDC.

For each switching office served by an ETDC, a block of DCDs in the memory of the EADAS records all the traffic measurements being




5XB COER – NO. 5 CROSSBAR CENTRAL OFFICE EQUIPMENT REPORTS
CU/EQ – COMMON UPDATE/EQUIPMENT
DCD – DATA COLLECTION DEVICE
EADAS/NM – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM/NETWORK MANAGEMENT
ETDC – ELECTRONIC TRAFFIC DATA COLLECTION
OR – ORIGINATING REGISTER
TDAS – TRAFFIC DATA ADMINISTRATION SYSTEM

Fig. 3—Flow of *Touch-Tone* originating register data through TNDS.

collected. These banks of DCDs are forwarded to EADAS/NM every five minutes when a command is sent from EADAS/NM over a data link. EADAS transmits data from all of the offices requested by EADAS/NM, including the No. 5XB with the new *Touch-Tone* ORs. After receiving data, EADAS/NM performs calculations on that data and displays the results nearly instantaneously to the network manager. If data from the No. 5XB were of national interest, every five minutes they would be forwarded to the NOCS where calculations would be performed and data displayed to national network managers.

Network management requires 5-minute data, whereas traffic engineering requires hourly data. Starting either on the hour or at half past the hour, EADAS writes the current DCD counts to magnetic tape and resets these registers to zero.

One of the major benefits of EADAS is its exception-reporting capability. The network administrator can define an exception condition such as some number of occurrences of all ORs busy in a 30-minute period. If this frequency occurs or is exceeded, a report would be printed at the EADAS teletypewriter (TTY) and the network administrator would take the appropriate action.

Periodically, the magnetic tape will be forwarded to a data processing center. These tapes will be run through TDAS, which will perform some validations and "warehouse" the data. TDAS will then create new tapes, one for each TNDS downstream system. One tape produced by TDAS would be processed by 5XB COER, which would then perform validation tests and calculations on the data and generate output reports to be used by the network administrator and traffic engineer. These reports would be used to determine if the new equipment was being utilized effectively while providing an acceptable grade of service to the customer. In most companies, the 5XB COER program is run weekly with a TDAS tape containing each day of the week used as input. The key report generated by 5XB COER is the Machine Load Service Summary (MLSS) report, which displays the ten high-day loads and the average busy season load on most switching office components, including the new *Touch-Tone* ORs. The MLSS report is also used by traffic engineers to determine equipment quantities for the next busy season.

At key checkpoints, traffic measurements are monitored by CSAR to ensure that they are received, processed, and are reasonably accurate. If, for example, the DCD associated with *Touch-Tone* OR usage were improperly wired, the occupancy of this equipment could erroneously appear to be greater than 100 percent. CSAR would be notified of these invalid measurements and would include this problem in the calculation of the index. Besides producing the index, CSAR identifies problem areas, thus facilitating their correction.

### *3.2 Addition of a new trunk group to TSS*

The previous example gave an overview of how TNDS collects and processes equipment data from an electromechanical office. In contrast, the next example describes how trunking data is collected from a 1 *ESS* office and processed by TSS. As before, we will start with a description of the record base process needed to support this data flow. In this case, we will describe the effort needed to add a new trunk group between two 1 *ESS* offices. For this example, measurements will be collected at only one end of the group, hence, data will be needed from only one 1 *ESS* office.

Generally, the recommendation to add a new trunk group would come from TFS as part of the forecasting process. At the time the group need is forecasted, it is added to the CU/TK database. The trunk servicer would write an order to have this new group installed and specify the schedule for data collection. The order would pass through organizations responsible for assigning and installing equipment to make the trunk group operational. One step of this operation would be assigning the group a trunk group serial number (TGSN). The TGSN is similar to the Common Language Location Identifier (CLLI); it is the unique identification for a trunk group in the electronic switching system office. These offices do not require ETDCs; they store counts in their memory and later forward them to the data collection computer.

Figure 4 illustrates the steps necessary to prepare each subsystem to collect and process data from this new trunk group. These steps are similar to those in Fig. 2; the major difference occurs at the switching offices. Traffic measurements are specified in software of the 1 *ESS* switching equipment. This includes the EADAS/NM requirements for five-minute data on the new trunk group and requirements for half-hourly data on the TSS and the other downstream systems. The new trunk group is added to the EADAS and EADAS/NM databases with the TGSN being used to identify the group and the DCDs used to identify individual measurements. If the trunk group is of national interest, it will be marked in the EADAS/NM database to be sent to NOCS.

Record base information for this new group is entered for TDAS and TSS via CU/EQ and CU/TK. As before, transactions are used to define the new measurements, and the time period during which these measurements will be processed, and to specify the new trunk group. Once the record bases have been modified, all subsystems are prepared to process this new group. The next section will trace the data flow from the office to the end users.

Figure 5 shows each step of this data flow. The major difference between this example and the previous one occurs at the front end—

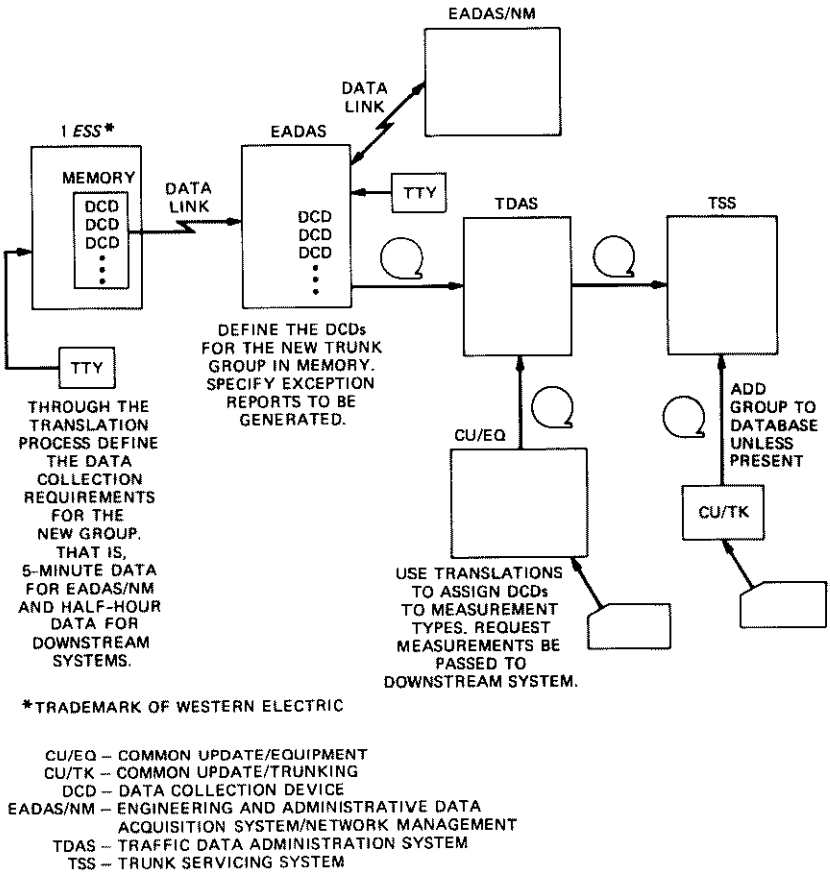



Fig. 4—Database synchronization for trunk group data.

between the switching office and data collection computer. When an attempt is made to seize the new group, a peg count is stored in a memory location at the 1 *ESS* office. The system passes these data forward upon request from the EADAS. Every five minutes EADAS/NM requests data from the EADAS. In the previous example, data were already in the data collection computer. However, for this example the EADAS must request data from the *ESS* offices that it serves. Raw counts are forwarded to the EADAS, which then subtracts the previous readings and forwards them to EADAS/NM. As before, EADAS/NM processes these data and displays them to the network manager. If this 1*ESS* is a key part of the toll network, data from the new group could be forwarded to NOCS for display.

To satisfy the needs of engineering, much more data are passed from the ESS office to the EADAS on the hour and half hour. These

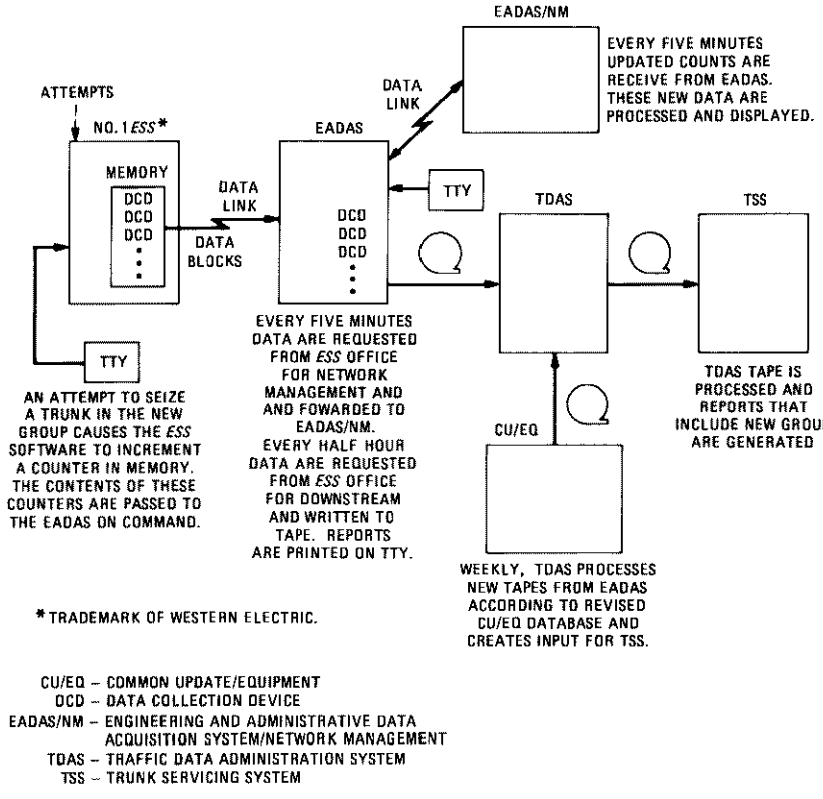



Fig. 5—Flow of trunk group data through TNDS.

data are written to tape and handled identically to the previous example. TDAS generates a separate tape to be processed by TSS and, as frequently as weekly, the TSS subsystem is run to process the data and generate reports. In the case of the new group, the servicer would closely examine its performance to determine if the number of circuits in the group was a good match to the load being offered to the group. If a mismatch occurred, an adjustment in the circuit quantity would be made after a pattern of performance for that group was established by running the TSS program several times with new traffic data.

As was the case for equipment data, CSAR monitors the flow of trunking data, calculates an index on this portion of TNDS, and produces reports that highlight problem areas.

## IV. MANAGING SYSTEMS ENGINEERING AND CONTINUING DEVELOPMENT

From the descriptions above, it is clear that from the user viewpoint, TNDS is large and complex. It is even more complex from a software

engineering viewpoint. This section describes how the software engineering for this large and complex system is managed.

### *4.1 End products and services*

#### *4.1.1 What is provided*

The TNDS organization provides various end products and services to the users. Although the specific details vary among the systems in TNDS, the end products and services can be categorized as follows:

1. Software
2. Documentation
   a. End user
   b. System Operation
3. Training
4. User Support.

The software is in the form of load modules for the specific machine on which the system is run. The load modules for systems run in each operating company are sent by tape or high-speed data transmission and then loaded. System operations personnel install the load modules using the installation guide provided. For centralized systems (e.g., SPCS COER, CSAR, and SONDS), development personnel install the new load modules.

The documentation provided falls into two categories: end user and system operation. End user documentation provides network personnel with information on how to interface with the system. This documentation is usually in paper form, but in the case of SPCS COER and CSAR, it is on-line at the user terminal with the option of printing. Each operating company also receives documentation for system operation from an Electronic Data Processing (EDP) viewpoint for the systems run at that site. This documentation includes information on backup and recovery procedures, resource requirements, etc.

The TNDS/EQ-TNDS/TK training philosophy is to train the trainers—TNDS coordinators and trainers in each operating company—by giving them the information they need to hold similar training courses in their companies. Both user and EDP training is provided. In addition, special release-oriented training usually supplements the basic system training.

User support for each operating company is usually in the form of an assist line (or "hot line") number, which the user can call when questions or problems arise.

#### *4.1.2 Size of the product*

Table II shows the relative size of each end product and service for each system in the TNDS. The lines of source code are in various high-level languages (PL/I, COBOL, FORTRAN, and C).

Table II—TNDS software and documentation

| | Lines of Code (Thousands) | Pages of Documentation | |
|---|---|---|---|
| | | User | EDP |
| EADAS/NM | 900,000 | 700 | 800 |
| EADAS | 200,000 | 3500 | 500 |
| TDAS | 75,000 | 1725 | 1334 |
| CU (CU/EQ and CU/TK) | 119,000 | 586 | 2527 |
| ICAN | 43,000 | 1445 | 318 |
| LBS | 42,000 | 1764 | 560 |
| 5XB COER | 128,000 | 2215 | 340 |
| SPCS COER | 450,000 | 1000* | 20 |
| CSAR | 125,000 | 200* | 48 |
| TSS | 166,000 | 1805 | 641 |
| TFS | 80,000 | 2086 | 2788 |
| SONDS | 178,000 | 380* | 30† |
| Total (Rounded) | 2,328,000 | 17,000 | 10,000 |

* On-line lessons.
† Binders.

#### *4.1.3 TNDS deployment*

Another important factor that affects managing the TNDS is its extensive deployment. Table III illustrates the extent of deployment of each component system in TNDS. Looked at another way, some or all of these systems are deployed in all of the operating companies and several are deployed in Bell of Canada.

### *4.2 Managing change*

Although TNDS is a mature system that has been widely deployed and operating in Bell operating companies for some time, the systems are still undergoing extensive development to keep up with the changing operating environment and to furnish improved capabilities. A goal of this continuing development process is to furnish users periodically (usually yearly) with new versions of the systems that incorporate needed changes in a timely manner. To achieve this goal, several factors must be considered.

1. The total number of possible improvements is generally much greater than the resources available for a particular release.
2. The calendar time required to develop a given release is much longer than one year.
3. A change frequently has impact in more than one of the systems of the TNDS.

These factors result in a need for careful planning to choose among the capabilities to be incorporated into a release, to begin planning well before the targeted release date, and to carefully coordinate planned changes among the systems. A formal approach called the TNDS Release Cycle is used to satisfy these needs.

Table III—TNDS subsystem development and utilization

| Function | Subsystem | Parameter | Percent Coverage (End 1982) |
|---|---|---|---|
| Data Acquisition | EADAS (or equivalent) | Main and equivalent main telephones | 93 |
| | TDAS/CU | Number of companies | 100 |
| Network Management | EADAS/NM | Class 1,2,3,4 and sector tandem offices | 84 |
| | | End offices with over 5000 telephones | 63 |
| Central Office Equipment | No. 5 Crossbar COER | Marker groups | 97 |
| | SPCS COER | Control groups | 100 |
| | SONDS | Step offices over 2000 lines | 37 |
| | ICAN | Eligible traffic units | 60 |
| | LBS | Eligible traffic units | 98 |
| Trunking | TFS | Message and auxiliary trunk groups | 84 |
| | TSS | Message and auxiliary trunk groups | 93 |
| TNDS Performance | CSAR | TNDS systems monitored | 100 |

#### 4.2.1 The nature of change

TNDS software and hardware are modified and enhanced for many reasons. First of all, designers must run to stay even with the rapidly evolving telephone network equipment, services, and operating procedures. Next additional features are needed by TNDS users and operators. And to prevent the system from becoming technically obsolete, it is necessary to make changes that improve the efficiency, maintainability, and expandability of software and hardware components. Examples of changes to the telephone network that have been accommodated in TNDS include the addition of new switching system equipment such as 5 *ESS* (the management of this change is discussed further below) and the use of new extreme value load engineering procedures. Recent additions include mechanization of *ESS* busy-hour determination studies, and new reports for No. 5 Crossbar administrators. There have been technical improvements to TNDS. Some examples are run time and disk utilization improvements, the complete rewrite of the No. 5XB COER module to improve its maintainability, and the conversion of EADAS to the *UNIX** operating system, which simplified the addition of new features.

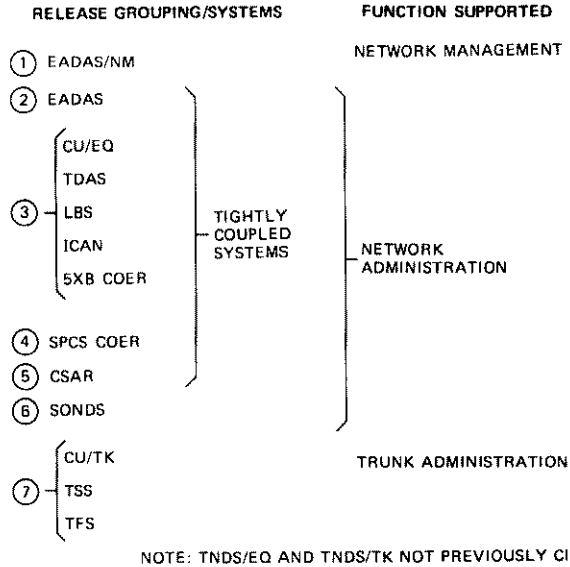
New releases of software modules usually contain a mixture of enhancement types. A recent TNDS/EQ release is typical. It contains features that support 5 *ESS* data collection, a new 2B *ESS* traffic data collection interface, and improvements to allow automatic transfer of *ESS* capacity statistics from EADAS to SPCS COER. This release also eliminates software that supported now obsolete data collection equipment. Several software functions were simplified, including the interface with one non-TNDS system, the distribution of traffic register definition listings, and company parameter tables. Also, thirteen separate changes to improve operational efficiency were made.

#### 4.2.2 The release cycle

The TNDS release cycle covers the overall planning, development, and project control processes to be employed to produce the major releases of systems in the TNDS. For release purposes, these systems are grouped into seven sets as outlined in Fig. 6.

Each of these groupings has a separate release cycle, all of which are coupled to varying degrees. In particular, the cycles of EADAS, TNDS/EQ, SPCS COER, and CSAR are tightly coupled because of feature interactions. These systems all support the Network Administration function. On the other hand, release of EADAS/NM and TNDS/TK (supporting network management and trunk administration, respectively) are less dependent on the other systems and on

* Trademark of Bell Laboratories.




5XB COER – NO. 5 CROSSBAR CENTRAL OFFICE EQUIPMENT REPORTS
CSAR – CENTRALIZED SYSTEMS FOR ANALYSIS AND REPORTING
CU/EQ – COMMON UPDATE/EQUIPMENT
CU/TK – COMMON UPDATE/TRUNKING
EADAS/NM – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM/NETWORK MANAGEMENT
ICAN – INDIVIDUAL CIRCUIT ANALYSIS
LBS – LOAD BALANCE SYSTEM
SONDS – SMALL OFFICE NETWORK DATA SYSTEM
SPCS COER – STORED PROGRAM CONTROL SYSTEMS CENTRAL OFFICE EQUIPMENT REPORTS
TDAS – TRAFFIC DATA ADMINISTRATION SYSTEM
TFS – TRUNK FORECASTING SYSTEM
TSS – TRUNK SERVICING SYSTEM

Fig. 6—TNDS release coupling.

each other. Finally, SONDS is a complete mini-TNDS for a specific class of offices (SXS CDOs), and consequently is more independent of intersystem interactions than other systems in the TNDS.

The following discussion applies in general to all the TNDS systems, but in detailed examples, the release cycle for TNDS/EQ will be used.

#### *4.2.3 Project phases*

The project phases in the release cycle define a continuing development process. The phases are divided into major categories and subcategories as shown in Fig. 7 and as listed below:

1. Planning
2. Implementation
   a. Requirements
   b. Design, code, and test
   c. Soak

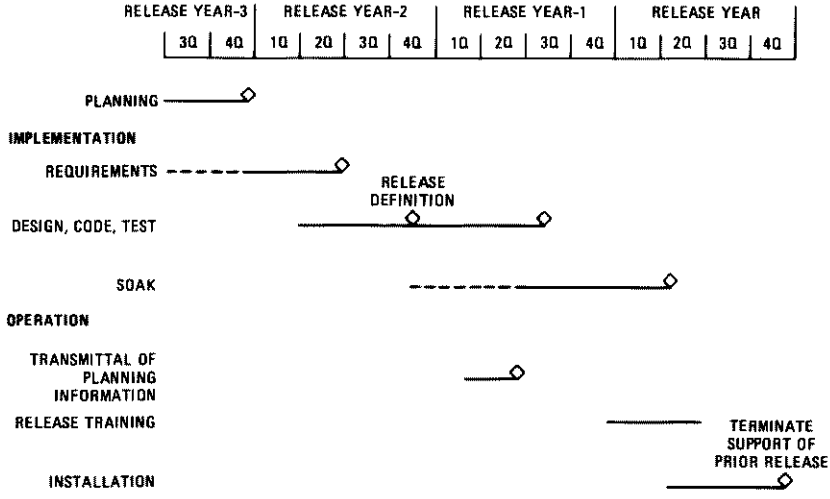



Fig. 7—TNDS release cycle.

3. Operation
   a. Transmittal of planning information
   b. Release training
   c. Installation.

***4.2.3.1 Planning.*** For the continuing development process, inputs come from many sources—operating companies, AT&T, and Bell Laboratories. Operating companies submit change requests to the AT&T project managers. These managers as well as Bell Laboratories designers and systems engineers also initiate proposals. These inputs are diverse. They range from requests to fix bugs or design errors to major development programs. The AT&T project managers and Bell Laboratories systems engineers, often with the advice of telephone company user representatives, perform an initial screening and classification of the inputs. In this step the inputs are classified into maintenance items, desirable improvements, and rejected suggestions. Maintenance items are transferred to the maintenance organization for the component system involved. Rejects are returned to the originator with an explanation of the reason for rejection. Desirable improvements are included in one or more "feature" packages.

Feature packages are specific development items that can be assigned individual priorities. They usually involve only one component system, but may be dependent on feature packages for other component systems for implementation. The assignment of individual items to feature packages has several advantages. Related items can be developed more efficiently as a group rather than separately. Items with similar objectives can be combined or modified to create improve-

ments with more global and more universally useful objectives. And since improvements often are suggested in terms of implementation, the same objectives sometimes can be accomplished as part of an already scheduled or more easily accomplished feature.

Certain time-critical features may be worked into ongoing development of forthcoming releases. These "expedited enhancements" generally must be small and very important because they displace other planned work. The level of management approval required for expedited enhancements depends upon the nature of the work being displaced. The remaining features are subjected to a much longer obstacle course. They must compete for the limited resources available for TNDS ongoing development.

Feature packages are carefully evaluated by Bell Laboratories systems engineering and development, by AT&T project managers, and by telephone company user representatives to establish their priority. Metrics for comparative evaluation include economic benefits, estimated development cost, and estimated length of time to develop. Other intangible factors that are considered in setting priorities are potential quality-of-service improvements, needs to adhere to AT&T corporate policies or operating practices, and the need to respond to FCC or state regulatory commission requests for data.

Feature lists in priority order are maintained on a permanent basis and are updated periodically. Separate lists are maintained for TNDS/TK, TNDS/EQ, SPCS COER, EADAS, and EADAS/NM. Any dependencies on other TNDS or non-TNDS features are noted. These priority lists are used as input for planning specific new TNDS software releases. Priorities may be time-dependent, that is, a certain feature, such as support of a new switching entity may not be needed until some future date. After that date, the importance of the support increases with the expected deployment of the switch.

Release planning is the responsibility of Bell Laboratories systems engineers working with each of the TNDS component systems. They consider the engineering resources available, the priority of proposed features, and length of time for development to formulate the content of proposed releases. These proposals are reviewed by AT&T project managers, telephone company user representatives and, when appropriate, Western Electric product line managers. Changes, if any, are made, and the priority of each of the features in the release package is established. This release priority is used to eliminate work when the inevitable resource constraints and expedited enhancements arrive. While approvals for the proposed release packages are being obtained, work will usually start on requirements for the features in the proposed package. Sign work, however, starts after final approval.

***4.2.3.2 Implementation.*** The first step toward implementing a

release is to complete detailed requirements for each feature. These are reviewed and approved, and they are one of the key project control documents.

The design step then translates the requirements into the code changes to be made. Design is divided into two parts: (1) system design, which identifies the modules, files, and interfaces (e.g., transaction and report layouts, intersystem interface definitions) to be changed; and (2) detail design, which identifies the coding changes to be made in each module. After system design is complete, the development organizations formally commit themselves to the contents and schedule for the release. After detail design, the changes are then coded and tested.

Finally, a soak test of the product is performed in one operating company site. The soak process determines if the end product performs in accordance with the requirements, is capable of satisfying user needs, and is worthy of systemwide release. The soak site is selected based on environmental requirements needed to exercise the features in the release. Soak periods vary by system but usually last from three to six months. After a soak is successfully completed, the release is made available for installation in other sites.

***4.2.3.3 Operation.*** The first step an operating company takes in installing a release is receipt of planning information in the form of a Product Release Description, which is usually made available approximately four months prior to the general availability date.

Also prior to general availability, release training information is made available and training classes are held. Documentation for the release is also made available in this time frame.

When the release is available it is installed in other sites. Each site is expected to install the release within a reasonable period (usually six months), at which time continued support of the previous release is terminated.

General availability dates are usually chosen to allow installation during a time period that is not in the traffic busy season when data collection is critical.

### *4.2.4 Release cycle timing*

The phases of the release process are integrated into an overall release cycle as shown in Fig. 7. This figure depicts the major activities within the cycle and the time intervals typically required to conduct each of these activities. Roughly three years are required to complete all of the phases for a particular release. With releases planned for annual delivery, this implies that at any point in time, development activities will be in progress on three different releases. The overlapping of releases is shown in Fig. 8.

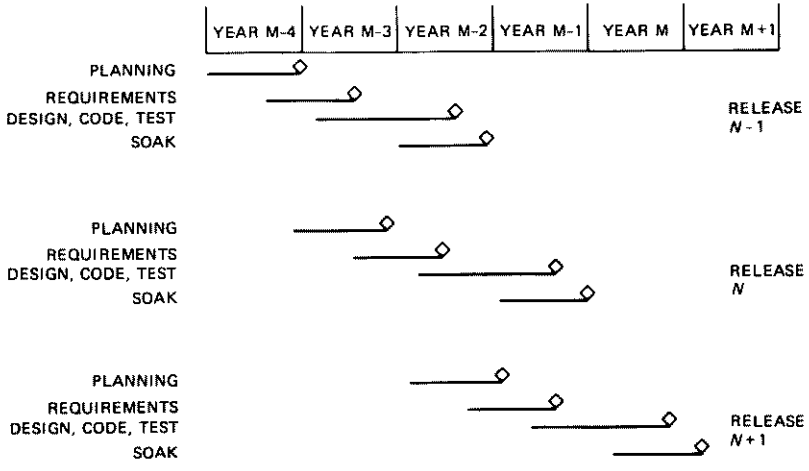



Fig. 8—Parallelism between releases.

#### *4.2.5 Interproject coordination*

For the subset of TNDS features that involve development in multiple component systems or in non-TNDS systems, an interproject coordination activity must be overlaid on the internal TNDS control process. Features in this category are few but often large, important development efforts. An example of such a development is supporting a new switching system like the 5 *ESS* system. Coordination involves establishing overall requirements for TNDS support, identifying specific "features" to be developed in each component system, and negotiating a coordinated development program for these releases. The development program may involve work in both TNDS and non-TNDS systems. Often, project committees are formed to coordinate requirements, design, testing, and soak of both the hardware and software components. Always there is an exchange of documentation at each step in the process. Interfaces are defined and documented as early as possible.

To make the continuing development process work, it is essential that continual communication among all the people involved be maintained. A permanent AT&T, Bell Laboratories, Western Electric management committee has the responsibility to see that communications are maintained and that problems in the continuing development process are resolved. The process has enabled TNDS to evolve with changes in the telephone business, become more efficient, grow in services to its users, and introduce new technologies, where appropriate.

### *4.3 The organization structure*

TNDS software engineering takes a large number of people working effectively together with well-defined individual responsibilities. Because of the size of TNDS, the organization is structured functionally with each organizational group having responsibility for one or more of the project phases defined above and for one or more of the systems in the TNDS.

#### *4.3.1 Functional structure*

Three types of organizations divide up the work in the release cycle—Systems Engineering, Development, and Western Electric Product Engineering Control Center (PECC).

***4.3.1.1 Systems engineering.*** Systems Engineering organizations have responsibility for the planning and requirements phases, and they participate in the soak evaluation.

***4.3.1.2 Development.*** The development organizations are responsible for design, coding, and test, and they also participate in the soak evaluation.

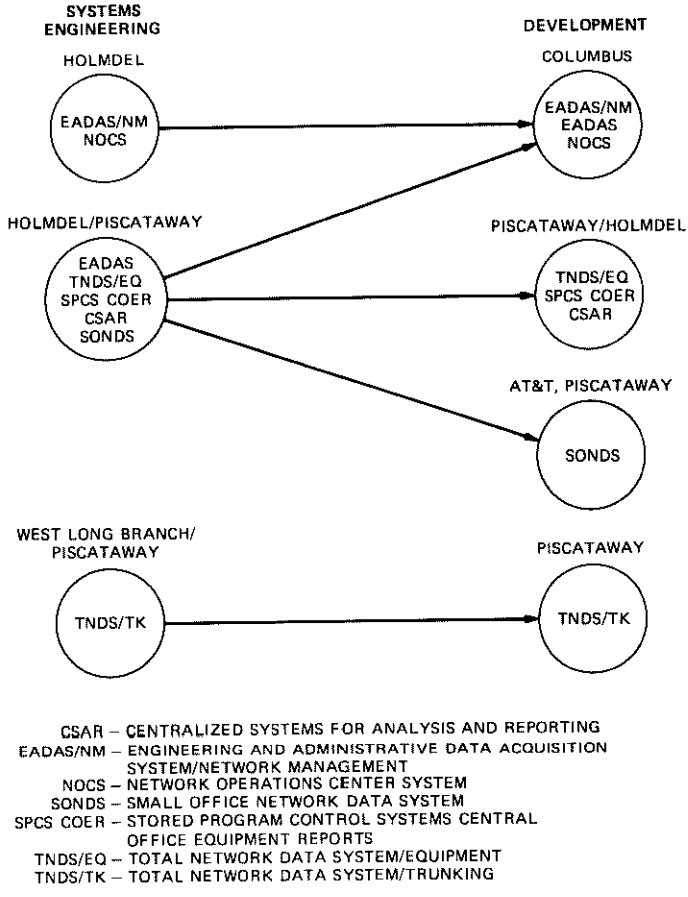
***4.3.1.3 Role of Western Electric.*** Western Electric PECC organizations are responsible for (1) developing documentation, (2) developing training and giving classes, (3) coordinating and performing soak evaluation, (4) maintaining installation support, and (5) providing customer consultation.

#### *4.3.2 Project structures*

Figure 9 shows the structure of the TNDS organization in terms of responsibility for systems engineering and development. Each circle represents a separate department in Bell Laboratories. Those organizations on the left are responsible for systems engineering, and those on the right for development. The systems for which the organization is responsible are indicated in the circle. Also indicated above the circle is the physical location of the organization (Holmdel, Columbus, Piscataway, West Long Branch, and AT&T Data Systems in Piscataway). Note that many Western Electric organizations involved in TNDS are not shown on the chart.

### *4.4 A continuing development example—the marriage of TNDS and 5 ESS*

The 5 *ESS* switching system is a new, Bell Laboratories-designed system that recently went into service with full TNDS support in place. Planning and implementation of this support has occurred in parallel with the development of the 5 *ESS* switching equipment. To illustrate the TNDS continuing development process, we will use the history of TNDS support for 5 *ESS* switching equipment.




CSAR – CENTRALIZED SYSTEMS FOR ANALYSIS AND REPORTING
EADAS/NM – ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM/NETWORK MANAGEMENT
NOCS – NETWORK OPERATIONS CENTER SYSTEM
SONDS – SMALL OFFICE NETWORK DATA SYSTEM
SPCS COER – STORED PROGRAM CONTROL SYSTEMS CENTRAL OFFICE EQUIPMENT REPORTS
TNDS/EQ – TOTAL NETWORK DATA SYSTEM/EQUIPMENT
TNDS/TK – TOTAL NETWORK DATA SYSTEM/TRUNKING

Fig. 9—TNDS organizational interactions.

Almost four years before the scheduled cutover of the first 5 *ESS* switching equipment, its development organization began to consider alternatives for handling the traffic data, and the needs for specific types of measurements. A joint AT&T, Bell Laboratories task group was formed to consider 5 *ESS* traffic data-handling needs. Experts on 5 *ESS* development, traffic engineering methods, TNDS, and telephone company data needs were assigned to the group. This task force recommended an overall strategy for handling 5 *ESS* traffic data and identified a number of specific work efforts needed to implement the strategy. The recommendations of this group established the objectives for the several work activities that followed. Key recommendations of this group included:

1. Allocating all traffic data-handling functions except basic measurement and measurement distribution to TNDS.

2. Using extreme value engineering techniques for timing and sizing of 5 *ESS* additions.

3. Collecting traffic data for trunk and switching equipment engineering, division of revenue, service provisioning, switching administration, and network management through EADAS.

4. Selecting peak values in EADAS for extreme value engineering.

Work activities included:

1. Developing extreme value engineering methods and measurement requirements for 5 *ESS* switching equipment.

2. Formulating a TNDS plan to support 5 *ESS* switching equipment.

3. Formulating requirements for 5 *ESS* features needed to deliver required measurements to EADAS.

The task group's recommendations were accepted by AT&T and Bell Laboratories managers, and work started on the above tasks. The work proceeded in parallel.

The first step in the TNDS planning effort was identifying TNDS outputs required for 5 *ESS* switching equipment and the TNDS developments needed to provide these outputs. Descriptions were developed, by individual component system, of the feature packages needed to support 5 *ESS* switching equipment. Table IV identifies the changes needed to TNDS. Following identification of the necessary features a tentative schedule was established for requirements formulation and component system design. This schedule was coordinated with the estimated schedules for the engineering methods work,

Table IV—TNDS developments for initial 5 ESS service

| System | Development Required |
|---|---|
| EADAS | X.25 interface with 5 *ESS* switching equipment<br>Peak measurement selection<br>Complete Network Operation Report Generation (NORGEN) report package |
| EADAS/NM | No development for first service. (Full EADAS/NM support will be provided in stages with development of advanced versions of 5 *ESS* switching equipment.) |
| TDAS<br>CU/EQ | 5 *ESS* measurement identification<br>Peak measurement handling<br>5 *ESS* statistics for CSAR |
| SPCS COER | Complete package of engineering and administrative reports for processing extreme value statistics<br>5 *ESS* statistics for CSAR |
| LBS | No development needed for first service. (Development to support a new 5 *ESS* load balance index will be required later.) |
| TSS, TFS, CSAR | No development required as current software is sufficiently general to handle 5 *ESS* switching equipment. |
| 5XB COER<br>ICAN<br>SONDS | These systems support electromechanical switching entities. No changes for 5 *ESS* switching equipment required. |

and the 5 *ESS* development work. The resulting TNDS "Integration Plan" for 5 *ESS* switching equipment was reviewed by Bell Laboratories and AT&T management, and then was distributed to all organizations having responsibility for the TNDS component systems. The TNDS integration plan was completed two years before the scheduled first 5 *ESS* service.

At this point the features needed to support 5 *ESS* switching equipment were added to the feature priority lists for each of the TNDS component systems. These 5 *ESS* features thus became part of the overall continuing development process for TNDS. Because of the expected widespread deployment of 5 *ESS* switching equipment in the operating companies and the large economic penalties that would occur without TNDS support for 5 *ESS*, the 5 *ESS* packages generally received the highest priority for development. Approval for development and for requirements formulation was obtained for each of the component systems in time to meet the initial 5 *ESS* service date. Work then started on requirements and design under the normal TNDS release process.

Three interface definitions (5 *ESS* switching equipment and EADAS, EADAS and TNDS/EQ, TNDS/EQ and SPCS COER) were formulated. These were required because of the decisions to utilize an X.25 protocol between EADAS and 5 *ESS* switching equipment, and the need to specify new schedules for extreme value data being selected by EADAS. Other requirements for new EADAS, SPCS COER, and TNDS/EQ reporting and processing were written. Actual design and coding of the various TNDS features began one to two years before the scheduled first 5 *ESS* service.

A committee of development representatives was formed to coordinate development activities for first service. Experts from 5 *ESS* switching equipment, EADAS, and TNDS/EQ served on it. This group worked out problems in design details and coordinated laboratory-to-laboratory testing activity, which generally proceeded in stages from basic hardware communications to application process communications. In addition, load simulators were used in the development organizations to do further interface testing and debugging. A large number of measurement definition changes occurred during the 5 *ESS* development. These changes had to be accommodated in the TNDS during development.

The first 5 *ESS* is owned by the Illinois Bell Telephone Company. To test the TNDS features designed to support the 5 *ESS* switching equipment, it was necessary to arrange soak testing of EADAS, TNDS/EQ, and SPCS COER releases in Illinois Bell. Planning for this soak testing began about one year before the scheduled first office service.

Final end-to-end laboratory tests were conducted three months

before first service. Initial soak testing with the first 5 *ESS* began about six weeks prior to service. A simulated call load on the 5 *ESS* switching equipment was used for these tests. When the 5 *ESS* switching equipment started handling live traffic, TNDS was in place to measure the traffic. Coordinated testing of TNDS components under live traffic conditions continued for several months after first service. Then the new release packages were made available to other TNDS users.

TNDS was ready to serve the first 5 *ESS* switching equipment because the need for TNDS support and planning was recognized by the 5 *ESS* developers early in their development cycle. Planning, requirements formulations, design, code, test, and soak proceeded in an orderly manner with careful coordination at each step. Problems in scheduling, interface details, and detailed measurement definitions were all solved through close coordination of the development efforts.

### *4.5 Future directions*

TNDS will continue to evolve gradually. Setting the direction of this continuing evolution is the purpose of TNDS long-range planning. Long-range planning for the TNDS is a continuous process whose output is a series of TNDS feature packages designed to avoid obsolescence and to take advantage of new technology. The results of recent long-range planning studies have indicated that further major improvements are desirable and appear to offer significant economic benefits. Several improvements relating to additional support for network administration functions are being studied. These studies are aimed at reducing the volume of paper reports produced by the TNDS, and providing users more direct control of TNDS operations, and flexible user-defined output reports. These objectives are planned to be achieved through on-line updating capabilities, expanded user programmability features, and provision of a simple standardized human interface to TNDS component systems. The next steps are to expand the scope of the planning work to include support for other telephone company organizations and to define the specific features that will be implemented. Through this process of continuous renewal guided by a periodically revised long-range plan, it is expected that TNDS will continue to serve the traffic-data handling needs of operating companies into the next decade and beyond.

## REFERENCES


1. R. L. Martin, "Automated Repair Service Bureau: System Architecture," B.S.T.J., *61*, No. 6, Part 2 (July–August 1982), pp. 1115–29.
2. R. F. Grantges, et al, "Central Office Equipment Reports for Stored Program Control Systems", B.S.T.J., this issue.

## AUTHORS

**George W. Riesz,** B.S. (Mechanical Engineering), 1950, M.S., (Electrical Engineering), 1952, Princeton University; Bell Laboratories, 1952—. Mr. Riesz joined Bell Laboratories in 1952 as a member of the Analog Computer Design Group. During 1955 and 1956 he served in the U.S. Army at Redstone Arsenal, where he participated in the initial ballistic missile defense studies. In 1956, he joined Nike-Zeus Systems Engineering, and from 1958 to 1960, was responsible for the design of Nike-Zeus battery control equipment. In 1962, he became Head of the Advanced Design Department. He then worked as Head of the Discrimination Measurements Department in Kwajalein from 1963 to 1965, when he became Head of the Nike-X Systems Analysis Department. In 1968, he was named Head of the CAMAR Data Systems Department, working on data systems design for advanced measurements radar. In 1970, he became Head of the Resource Allocation Studies Department, and in 1971, Head of the Traffic Network Studies Department. In 1974, he was named Head of the Network Administration Department. He assumed his present position as Head of the Operations Systems Implementation Planning Department in 1979. Mr. Riesz holds one patent and is the author of several technical articles. Member, Phi Beta Kappa, Sigma Xi.

**J. Wayne Steifle,** B.S. (Electrical Engineering), 1968, Clemson University; M.S. (Electrical Engineering), 1969, Stanford University; Bell Laboratories, 1968–1983; American Bell, 1983—. After joining Bell Laboratories, Mr. Steifle worked on radar and ABM system studies, and then spent three years in the Air Force working at NASA on guidance systems for manned space programs. After returning from military leave, he worked on guidance design for Safeguard, and later on systems engineering and development of various Business Information System projects. From 1977 to 1980 Mr. Steifle was Supervisor of the TNDS/EQ Engineering Group, where he was responsible for systems engineering for the TNDS/EQ systems. He is currently Supervisor of the Interconnect Software Group in the AIS/NET 1000 Node Hardware Department at American Bell. Member, IEEE.

**John A. Kohut,** B.A. (Mathematics), 1964, Gannon College; M.S. (Systems Engineering), 1967, Stevens Institute of Technology; Bell Laboratories, 1964–1977, 1980—; AT&T, 1977–1980. Early in his career at Bell Laboratories, Mr. Kohut evaluated the performance of various antiballistic missile systems using computer simulation techniques. From 1970 to 1977, he analyzed the behavior of the message telephone networks and various control systems during failure and overload conditions. From 1977 to 1980, he worked at AT&T, where he was the project manager of an operational support system that aids in the real-time management of the network. Presently, Mr. Kohut is Supervisor of the Trunk Servicing Engineering Group, which plans new capabilities for the Trunk Servicing System (TSS).

**Milton S. Hall, Jr.,** B.E.E. 1959, Georgia Institute of Technology; M.E.E. 1962, New York University; Bell Laboratories, 1959–1968, 1970—; AT&T, 1968–1970. Initially Mr. Hall worked in the data systems engineering organization. His responsibilities there included computer performance analysis, requirements for store-and-forward switching, and plans for wideband data services. He also spent six months on active duty for training in the U.S. Army Signal Corps. In 1968, he transferred to AT&T, where he was involved in long-haul transmission facilities planning. Then in 1970, he returned to

Bell Laboratories to supervise traffic engineering studies. In addition, his group designed a large, on-line system for processing telephone traffic data. From 1977 to October 1981, when he assumed his current assignment, Mr. Hall supervised a group responsible for network administration systems planning. In this work, his group investigated on-line access to information and records as a means to eliminate paper reports and to provide direct user updating of a distributed datebase. Since October 1981, he has supervised a group responsible for operations systems evolution planning. Member, IEEE, ACM, Tau Beta Pi, Phi Kappa Phi, Phi Eta Sigma.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Theoretical and Engineering Foundations

By W. S. HAYWARD* and J. P. MORELAND*

(Manuscript received July 14, 1982)

The Total Network Data System (TNDS) is a coordinated family of computer-based systems that collect and process network measurements to aid the engineers, administrators, and managers of the Bell System network in efficiently meeting service objectives. This paper describes these service objectives, the nature of telephone traffic and traffic measurements, and the theories and engineering assumptions underlying the use of these measurements in the design and administration of the trunk network and switching systems.

## I. INTRODUCTION

The Total Network Data System (TNDS) is a coordinated family of computer-based systems that collect and process network measurements to aid the engineers, administrators, and managers of the Bell System network in efficiently meeting service objectives. In this paper we describe these service objectives, the traffic measurements used to monitor and design the network, and the theories underlying the use of these measurements in the various TNDS systems.

It is important to realize that, as a result of the continuing changes in the switching systems and methods of routing traffic in the Bell

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

System network, new mathematical models and new service criteria are likely to be used as time goes by. Therefore what we are reporting here is a snapshot of what is going on today. We fully expect that there will be significant changes in the years to come.

The TNDS is described in the other articles of this issue. Our presentation of the theoretical foundation of the activities that the TNDS supports is divided into two parts, trunking and central office. We start with trunking because its simplest model is widely known and serves as an introduction.

## II. TRUNKING

### *2.1 Service objectives*

Consider the simple two-node network shown in Fig. 1. The trunk group joining end offices A and B provides the only route for calls between A and B and, as such, is called an only-route trunk group. Call attempts which arrive when all trunks are busy are said to be blocked and the customer is requested (by a reorder tone) to hang up and try the call at a later time.

The grade-of-service for an only-route group is defined to be the unweighted average blocking observed in the time-consistent busy hour of the busy season (defined below). For a given hour, the blocking is defined to be the ratio of the total number of blocked calls (overflows) to the total number of call attempts (peg count). When the hourly blockings are averaged over the time-consistent busy hour of the busy season, the resulting number is the observed grade-of-service for the group. The service objective is 0.01 average blocking.

The measure of telephone traffic used to define the time-consistent busy hour of the busy season is called the trunk-group offered load. Offered load is measured in units called erlangs, and is equal to the average number of busy trunks for a situation in which there is no blocking. Of course, in practice, offered load cannot be measured directly since calls are blocked. It can, however, be estimated from measurements of the carried load (average number of busy trunks) and blocking, as explained in Section 2.3.

When the hourly offered loads are averaged over the same hour for 20 consecutive business days, the maximum of these averages defines the time-consistent busy-hour load for the 20-day period.* The busy season is then defined to be the 20-day period for which the time-consistent busy-hour load is a maximum.

Formally, if $PC_j$ and $O_j$ denote, respectively, the number of call

* A different definition of busy hour is used for trunking than for central office equipment. The difference has grown up over the years because of fundamental differences between trunks and central offices, particularly with regard to forecasting capability, available measurements, and ability to respond to unforeseen shifts of traffic.

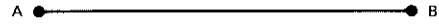



Fig. 1—Only-route trunk group.

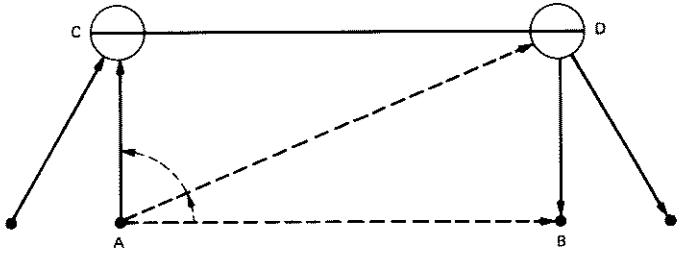



Fig. 2—Alternate-routing network.

attempts and blocked attempts during the $j$th time-consistent busy hour of the busy season, the observed blocking, $b_j$, during hour $j$, is defined to be

$$b_j = \begin{cases} O_j/PC_j, & \text{if } PC_j > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

and the observed grade-of-service is defined to be

$$b = \frac{1}{n} \sum_{j=1}^{n} b_j, \tag{2}$$

where $n$ is typically 20 consecutive business days. As explained in Section 6.3, the statistic $b$ is used to detect service problems (blocking significantly greater than the 0.01 objective) and, if necessary, trigger corrective actions (e.g., trunk augments).

Figure 2 shows a more complex traffic network consisting of high-usage trunk groups (dashed lines) and final trunk groups (solid lines). Calls originating at end ofice A and destined for end office B are first offered to the primary high-usage group AB. If, however, all AB trunks are busy, the call is alternate routed to the intermediate high-usage group between A and the tandem switch D or, if all AD trunks are busy, to the final group AC. If the call arrives at D it is offered to the final group DB; if the call arrives at C, it is offered to the final group CD. Whenever a call is offered to a final group with all trunks busy, the customer receives a reorder tone or recorded announcement.

The grade-of-service for final trunk groups is defined in the same way as described above for only-route groups; the service objective is again 0.01 average blocking.

Service objectives are not, however, specified for high-usage groups. As explained in Section 2.4.1, they are sized so as to minimize the cost of the trunk network.

### 2.2 Traffic models

Modern trunking theory shows that three traffic parameters are required to predict busy-hour busy-season average blocking[1]: (1) average offered load during the time-consistent busy hour of the busy season, (2) peakedness (the ratio of the variance to the mean of the number of busy trunks for a trunk group so large that there is no blocking), and (3) the day-to-day variation of the busy-hour loads (variance of the daily offered loads in the time-consistent busy hour of the busy season).

#### 2.2.1 Poisson/Erlang formulas

Prior to 1970, the Poisson formula,

$$P(c, a) = e^{-a} \sum_{n=c}^{\infty} \frac{a^n}{n!}, \tag{3}$$

where $c$ is the number of trunks and $a$ is the offered load in erlangs, was the standard formula for sizing trunk groups in the Bell System. It gives the blocking probability when Poisson traffic (i.e., traffic with random call arrivals) is offered to a group of $c$ trunks and blocked calls are held. That is, calls are assumed to remain in the system—either waiting or holding a trunk if one is available—for intervals that are independent of whether they were initially blocked.

The Erlang loss formula,

$$B(c, a) = \frac{a^c}{c!} \Big/ \sum_{n=0}^{c} \frac{a^n}{n!}, \tag{4}$$

published in 1917 by A. K. Erlang of the Copenhagen Telephone Company, also assumes Poisson arrivals but assumes that blocked calls are cleared; i.e., blocked calls are assumed to leave immediately and to have no further effect on the telephone system. Note that both the Poisson and Erlang formulas are independent of the distribution of holding times.

While the Erlang formula was derived from an apparently more realistic assumption of blocked call behavior, it, in practice, underestimated the measured blocking corresponding to an average busy season load $a$. Accordingly, after considerable discussion during the 1920s, AT&T decided to use the Poisson formula, which predicts a higher probability of blocking than the Erlang formula. It was generally thought that blocked call behavior, while not the same as assumed for the Poisson formula, was the prime reason for the Poisson formula's superiority. Much later it was found that day-to-day load variation was the major underlying cause.

### 2.2.2 Wilkinson's Equivalent Random method

The assumption of Poisson (random) call arrivals accurately models most first-offered traffic, but does not adequately describe overflow traffic from high-usage trunk groups. That is, overflow traffic is more variable than Poisson traffic because it arrives in bunches; consequently, more trunks are required than the Poisson formula predicts when final groups receive overflow traffic. Accordingly, in 1956, shortly after the introduction of automatic alternate routing of customer-dialed calls, R. I. Wilkinson developed the Equivalent Random method to size final trunk groups in an alternate routing network.[2]

The basis of Wilkinson's method, as illustrated in Fig. 3, is the assumption that the superposition of the individual overflows offered to a final trunk group with $c$ trunks can be represented by a single overflow from a (fictitious) group of $s$ trunks with Poisson offered load $a^*$. The parameters $a^*$ and $s^*$ are chosen so that the resulting overflow has the same mean, $\alpha$, and variance, $\upsilon$, as the actual traffic offered to the final group. With this approximation, the Erlang loss formula, with $c + s^*$ trunks and offered load $a^*$, can be used to estimate the blocking on the final group. That is, the Equivalent Random approximation to blocking probability, which simulation studies have shown to be remarkably accurate, is given by

$$B(c, \alpha, z) = \frac{a^*}{\alpha} B(c + s^*, a^*), \tag{5}$$

where $z = \upsilon/\alpha$ is called the peakedness of the traffic offered to the final. Formulas for computing $z$, $a^*$, and $s^*$ are given in Ref. 1.

### 2.2.3 Day-to-day variation

Wilkinson showed that day-to-day load variation causes trunk group average blocking to be significantly higher than that predicted under the assumption of a constant offered load.[3] Furthermore, he showed that the distribution of measured daily offered loads is well approximated by a gamma distribution, $\Gamma(\alpha \mid \bar{a}, \upsilon_d)$, with mean $\bar{a}$ and variance $\upsilon$ related to the mean $\bar{a}$ by

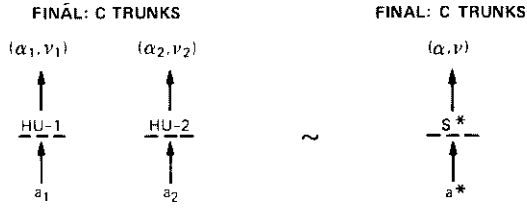



Fig. 3—Wilkinson's Equivalent Random method.

$$v_d = 0.13\bar{a}^{\phi}, \tag{6}$$

where $\phi$ is a parameter that describes the level of day-to-day variation. Wilkinson's studies—which showed that $v_d$ is relatively larger for overflow traffic than for Poisson traffic—led to the use of three values of $\phi$ for engineering applications: $\phi = 1.5$, $\phi = 1.7$, and $\phi = 1.84$, which are referred to respectively as low, medium, and high day-to-day variation.

Using this model for day-to-day variation, Wilkinson's formula for predicting trunk group average blocking is given by

$$\bar{B}(c, \bar{a}, z) = \int_0^{\infty} B(c, \alpha, z) d\Gamma(\alpha \mid \bar{a}, v_d). \tag{7}$$

Since it was not practicable to apply Wilkinson's method without the use of a computer, his now famous $\bar{B}$ capacity tables were not available in the Bell System until about 1970.

#### *2.2.4 Neal-Wilkinson tables*

In 1976, Hill and Neal refined Wilkinson's $\bar{B}$ tables by developing mathematical models that account for the effects of the finite (one-hour) measurement interval.[1] That is, the service objective is defined in terms of the expected single-hour blocking, $E(O/PC)$. Thus, since eq. (7) provides an estimate of the probability of blocking, $E(O)/E(PC)$, it must be modified to remove the assumption that the measurement interval is infinite. Furthermore, Hill and Neal modified the formula for day-to-day load variation to account for the fact that part of the measured variation in daily offered loads is due to the finite measurement interval and must, therefore, be subtracted from the observed variation. Their work led to the new Neal-Wilkinson trunk group capacity tables, which are now the Bell System standard for sizing final and only-route trunk groups.

### *2.3 Traffic measurements and data transformations*

This section describes the traffic measurements used to monitor network service, and the conversion of these measurements into estimates of the traffic parameters used to design the network.

#### *2.3.1 Measurements*

The Trunk Servicing System (TSS) derives estimates of average blocking on final groups. These estimates are used to detect service problems that trigger the demand servicing action described in Section 2.5. In addition, TSS derives estimates of trunk-group offered load, peakedness, and day-to-day variation. These estimates are used in demand servicing to correct existing service problems and in planned

servicing to forecast future trunk requirements. These estimates are derived from three traffic measurements: peg count, overflow, and usage (i.e., carried load).

In electromechanical offices, usage is measured in units called CCS (hundred call seconds) by a traffic usage recorder (TUR) which scans the trunk group every 100 seconds and increments a register for each busy trunk. Thus, the usage or carried load in erlangs (average number of busy trunks) is obtained by dividing hourly register count by 36; i.e., one erlang equals 36 CCS. In electronic switching system offices, a similar process is used to estimate usage, but there is no need for specialized equipment since traffic data are collected by the switching system's central processor.

Because of the discrete scanning (i.e., once every 100 seconds) there is, of course, some statistical sampling error associated with the usage estimate. However, our studies have shown that this error is negligible compared with the unavoidable statistical errors associated with the finite measurement interval.

#### *2.3.2 Data transformations*

The process used to derive estimates of blocking, offered load, peakedness, and day-to-day variation consists of three basic steps: data validation, computation of traffic parameters, and data substitution. Data validation is a mechanized process used to determine whether there are unusual traffic conditions (indicated by unusually high blocking due, for example, to a snowstorm) or problems in the data collection process (indicated by inconsistent measurements such as overflow exceeding peg count or usage exceeding 36 CCS per trunk). Such data must be detected and deleted; otherwise they would propagate through the trunk provisioning process and possibly cause unnecessary trunk augments to be made.

Under normal conditions, when peg count ($PC_i$), overflow ($O_i$), and usage ($U_i$) are available for each day in the study period (normally twenty consecutive business days), the required traffic parameters are estimated as follows:

1. The study period average blocking, $b$, is estimated as given by eqs. (1) and (2).
2. The study period offered load $a$ (in erlangs) is estimated by

$$a = \frac{1}{n} \sum_{i=1}^{n} a_i, \tag{8}$$

where

$$a_i = \frac{u_i/36}{i - b_i} \tag{9}$$

is an estimate of the daily offered load.

3. The observed variance, $v$, of the daily offered loads is estimated by

$$v = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - a)^2. \tag{10}$$

Peakedness is not directly measured. Instead, to reduce data collection costs, a value of $z$ is inferred from the relation

$$b = B(c, a, z), \tag{11}$$

where $c$ is the number of trunks in the group and $B(c, a, z)$ is the equivalent random blocking formula.

Since there are many cases when complete UPCO (usage, peg count, and overflow) data are not recorded or are invalidated, TSS includes procedures to estimate traffic parameters with less than complete UPCO data. For example, if PCO is available but U is not, and if $b_i \neq 0$, an estimate of the daily offered load $a_i$ is obtained by solving the equation

$$b_i = B(c, a_i, z), \tag{12}$$

where a typical value for $z$ is assumed. Alternatively, if U is available, but PCO is not, the daily blocking is estimated by solving the equation

$$u_i/36 = a_i[1 - B(c, a_i, z)] \tag{13}$$

for $a_i$; then

$$b_i = 1 - \frac{u_i/36}{a_i}. \tag{14}$$

To minimize the impact of sampling error, TSS requires a minimum of three days per week of both peg count and overflow or three days per week of usage measurements. If less than this amount of data is available, the traffic parameters are estimated by using the most recent historical values; this process is called data substitution.

### *2.4 Trunk forecasting*

The Trunk Forecasting System (TFS) is used by most Bell operating telephone companies to provide forecasts of interoffice message-trunk requirements for each of the next five years. The process consists of two basic steps: the estimation of future busy-season traffic loads and the design of a traffic network that minimizes the cost of the trunks required to satisfy these anticipated demands.

Since the network design process determines the traffic loads that must be forecast, we first discuss the concepts underlying the design

of minimum cost traffic networks and then describe the load forecasting process.

### *2.4.1 Network design*

The objective of the network design process is to determine the number of trunks in each high-usage and final trunk group that minimizes the cost of the trunks provided subject to the constraint that the average blocking on final groups does not exceed 0.01 in any hour.

To illustrate the basic concepts, we first consider the simple case of engineering the network shown in Fig. 4 for a single hour. The problem is to determine the value of $n$, the number of trunks in the high-usage group, which minimizes the cost function

$$\text{COST} = nC + N_a C_a, \tag{15}$$

where $C$ is the cost per trunk on the high-usage route, $C_a$ is the cost per circuit on the alternate route (which, in reality, consists of at least two trunk groups and a tandem switch), and $N_a$ is the number of alternate route circuits required to meet the service objective. The load offered to the high-usage group is $a$; the traffic offered to the alternate route, A, consists of overflow traffic from the high-usage group under consideration plus "background traffic" consisting, in general, of overflow from other high-usage groups and first-routed traffic.

Qualitatively, the trade-off is between the less expensive high-usage trunks (i.e., the direct route has no switching cost and is usually physically shorter) and the more efficient alternate route circuits (i.e., the total number of circuits, $n + Na$, is minimized by providing all circuits in a common group where they can be shared by all traffic).

Since $N_a$ depends upon both the mean $A$ and the peakedness $Z$ of the traffic offered to the alternate route, we should account for the change in both $A$ and $Z$ as $n$ is varied. However, the procedure is considerably simplified without significant loss in accuracy by ignoring the variation of $Z$ with $n$ and by assuming that the rate of change of $A$ with respect to $N_a$ is a constant, $\gamma$. Thus, with these approximations, the condition for minimum cost may be written as

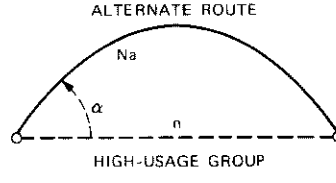



Fig. 4—ECCS engineering.

$$-\frac{dA}{dn} = \frac{\gamma}{C_R},$$

where $C_R = Ca/C$ is called the cost ratio.

Since only the overflow portion of $A$ varies with $n$, and since the overflow load is $aB(n, a)$, where $B(n, a)$ is the Erlang-$B$ blocking formula, the value of $n$ that minimizes the COST is determined by

$$-\frac{d}{dn}[aB(n, a)] = \gamma/C_R. \tag{16}$$

The quantity of the left is approximately $a[B(n-1, a) - B(n, a)]$ and is called the load carried on the last trunk. The quantity on the right is called the economic load on the last trunk or, when the load is expressed in CCS, the ECCS; in fact, a more careful derivation, which recognizes that $n$ is restricted to integer values, shows that the optimum value of $n$ is the largest value for which the load on the last trunk is greater than or equal to the ECCS.

If the time-consistent busy hour for each group in the network were the same, that hour would be the only one needed in the sizing process. However, since such complete coincidence seldom occurs and since the existing algorithms are designed to use only a single-hour load, the question arises as to which hour to use to size each group.

At first glance, the solution may appear simple: since the service objective must be satisfied in all hours, final-trunk groups must be sized for their busy-hour loads, and high-usage groups should be sized for their offered loads in the alternate route's busy hour. However, there are two problems here. First, the various legs of the alternate route may have different busy hours and second, these busy-hours may depend upon the size of the subtending high-usage groups.[4] Thus, in theory, it is not possible to preselect a single hour to use in designing a minimum-cost network that satisfies the service objectives in all hours.

In practice, however, a heuristic, called the significant-hour method, has been found to produce networks that do not differ significantly from those obtained by using all hourly data. To illustrate this method, consider the two-level network shown in Fig. 2. For group AB, for example, two significant hours are considered: the A-office originating cluster busy hour (the hour for which the total load offered to the set of high-usage groups and the final originating at A is maximum) and the B-office terminating cluster busy hour (the hour for which the total load terminating at B is a maximum). Of these two hours, which are preselected in TSS from current traffic measurements, the one for which the AB load is larger is called the control hour and the group is sized using its control-hour load. A more complete discussion of this method is given in Ref. 4.

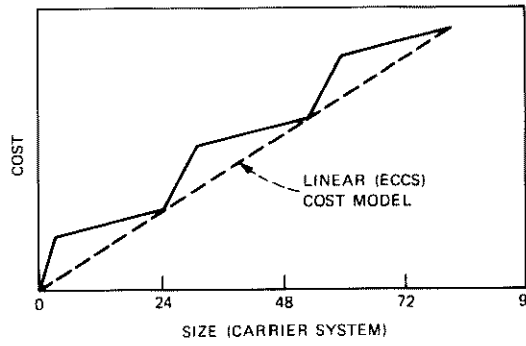



Fig. 5—Modular engineering.

The trunk quantities provided by the ECCS method must be modified to account for several factors. First, the administrative cost for a high-usage group is not included in the ECCS model. Since this cost is independent of group size, its inclusion leads to a prove-in or minimum group size; for larger groups it plays no role in the determination of the optimum size. While the exact prove-in size depends upon the specific costs, in practice we use a three-trunk minimum for local networks and a six-trunk minimum for toll networks.

Second, the ECCS model assumes a linear relationship between the cost and size of a high-usage group. However, for carrier systems, such as the T1 system, which provides the capacity for 24 time-division multiplexed channels on two pairs of wires, the cost is the step-like function illustrated in Fig. 5. Consequently, the ECCS solution is rounded to the nearest module of 24 for two-way groups or 12 for one-way groups. This rounding procedure is called modular engineering.

Finally, the ECCS method does not address the dynamic aspects of the trunk forecasting process, specifically, the time-varying nature of demands (which may call for the removal of trunks in one year only to have them added back in a future year) and forecast uncertainty. Currently, these factors are accounted for by manual adjustments to smooth the trunk requirements and avoid uneconomical disconnects and to introduce some reserve capacity to limit the amount of activity required in demand servicing.

#### *2.4.2 Load forecasting*

The objective of the load forecasting process is to estimate future busy season (control hour) first-route loads for each trunk group. First-route load, i.e., total offered load minus overflows from subtending high-usage groups, is projected since future offered load depends, of course, on the future sizes of subtending high-usage groups.

The standard load-forecasting algorithms currently in use in the

Bell System obtain estimates of future busy-season first-route loads by multiplying the most recent measurement of busy-season first-route load by an aggregate growth factor, for example, the average of the growth factors obtained by trending the total office loads at each end of the trunk group. Descriptions and comparisons of the various algorithms currently in use are given in Ref. 5.

As we explained in Ref. 5, the existing algorithms have two significant sources of error: First, on account of the finite amount of data, measured loads can have large statistical errors; standard deviations fall in the range of 5 to 10 percent for trunk-group data. Second, individual trunk-group growth factors can differ from the aggregate growth factor. These errors are significant since they lead to an increase in the reserve capacity required to satisfy anticipated customer demands.[6]

To reduce forecast error, and hence reserve trunking capacity, a new algorithm, called the Sequential Projection Algorithm (SPA), has been developed to forecast busy-season traffic loads within the Bell System.[5,7] SPA is based on a linear Kalman filter model, which establishes a linear trend for individual trunk group loads, together with logic for detecting and responding to outlier measurements. A complete description of SPA is given in Ref. 7.

### *2.5 Trunk demand servicing*

To compensate for the effects of forecast error, demand servicing determines which trunk groups should be augmented to satisfy current busy season demands. The process uses current traffic measurements to detect service problems on finals (blocking significantly greater than objective) and to determine which high-usage and/or final groups should be augmented to correct these problems in a cost-effective manner. In theory, the demand-servicing process could also direct the removal of trunks in groups providing significantly better than objective service. In practice, however, the decision to remove trunks is, generally, part of the trunk-forecasting process described in Section 2.4.

The first step in demand servicing is to determine when action should be taken. This decision is based upon the observed average blocking $b$ defined by eq. (2). Specifically, when $b$ exceeds a threshold $b_u$, the final group is declared to be overloaded and corrective action is taken; otherwise the measured blocking is assumed to be acceptable.

Because of the statistical nature of the demands, the finite number of days in the study period, and the finite (one-hour) measurement interval each day, the measured blocking can be smaller or larger than the underlying true mean blocking.[8] For example, Fig. 6 shows a

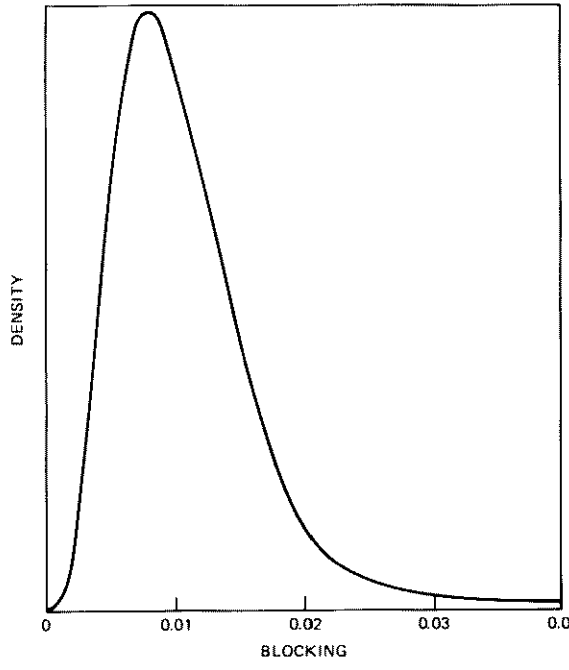



Fig. 6—Distribution (density) of measured blocking.

typical distribution of observed blocking for a correctly sized final group.

Accordingly, the threshold $b$ has been selected to achieve a reasonable balance between two types of servicing errors: false alarms, or Type I errors, occur when $b$ exceeds the threshold but the group is not overloaded; misses, or Type II errors, occur when a service problem is not detected; i.e., when the measured blocking falls below the threshold but the group is underprovided. The threshold $u$ has been selected to give a false alarm probability of less than 2.5 percent; in most cases this threshold corresponds to a miss probability of less than 10 percent when the group is overloaded to at least a 0.05 blocking level.[9] For the standard 20-day study period, the threshold $b_u$ falls in range of 0.07 (small groups) to 0.02 (large groups).

If a service problem has been detected, the next step is to determine which groups to augment. At present, this decision is based on the servicer's judgment, but a new trunk-demand servicing policy, described below, has been developed for possible inclusion in TSS.[9]

Although the simplest procedure would be to augment the final group, this procedure would tend to drive the network away from the optimal balance between high-usage and final trunks. Consequently, the demand servicing policy should attempt to resolve a service prob-

lem at its source. That is, the problem should be corrected by augmenting those groups—starting at the lowest level in the hierarchy—that are significantly underprovided (based upon the ECCS design) and are contributing to the service problem on the final group.

Since the determination of when a high-usage group is underprovided is subject to the same sources of statistical error as blocking estimates, thresholds that complement those for final groups have been developed for high-usage groups.[9] Thus, when the difference between the required number of trunks and the number of trunks in service exceeds the threshold, the group is a potential candidate for servicing. However, since only those groups contributing to the blocking problem are considered as candidates for servicing, an overloaded primary high-usage group that subtends an acceptably loaded intermediate high-usage group is not selected as a candidate.

Thus, as explained in Ref. 9, the proposed demand-servicing process is an iterative one that starts at the top of the hierarchy (the final) to identify those undertrunked groups that subtend undertrunked groups at the next higher level; such groups are selected as candidates for servicing. The lowest-level candidate, whose servicing will contribute the maximum reduction in load offered to the final, is then augmented. The blocking on the final is then recomputed. If a problem still exists, the trunk requirements and candidacy of all groups is then reevaluated. The algorithm is terminated when the blocking on the final has been reduced to an acceptable level.

## III. SWITCHING SYSTEMS

### *3.1 Service objectives*

Switching systems typically divide the process of switching calls into functional parts. Each part may require specialized equipment, such as customer receivers to receive dial pulses, or dedicated areas of memory in which to store information relative to the function, such as the number dialed. Equipment and memory of this kind are generally provided only as required to handle the traffic of each installation—a traffic engineering job requiring data from TNDS. The central processor which time-shares the control of most functions has a finite capacity, so it too must be traffic engineered. Because of the different kinds of traffic and the different function of a switching system in serving customers, the service criteria and hours of measurement differ, not only from those applied to trunks but even among switching systems according to their place in the network. This and the next section discuss service objectives and measurement timing; the remaining sections discuss application.

In setting service criteria to balance customer service against the cost of providing it, a comprehensive consideration has to be given to

what blocking or delaying calls means to a customer. In toll switching and in trunking, any particular subscriber is likely to place only a small proportion of calls through a particular trunk group or switch. If average blocking is achieved here, the customer is not likely to build up a feeling of frustration when a call is blocked; the next call is very likely to go through. In a local office, however, an overloaded switching system can affect all calls placed by the customer, regardless of destination, and blocking may remain high for a considerable time. The feeling of frustration can be much higher therefore, and service criteria must be chosen for closer control of service than can be achieved by considering averages alone. Day-to-day variation is, obviously, most important. Another important factor is balance; during the high-traffic hours, a switching system can give excellent *average* service while giving consistently poor service to a subset of customers.

A further difference applies to administration. As point-to-point loads vary with time, the components from which trunks are constructed can be reconfigured to meet changing demands. Very little of this kind of administration of equipment can be done with a switching system. Additions can normally be made only at the end of an engineering period, which is usually two years.

In addition to criteria of call blocking, which are expressed in terms of fraction of calls not completed because of congestion, delay criteria are used for engineering and administering switching systems. Delay criteria, which were not discussed in the section on trunking, are expressed in terms of the probability of exceeding an objective delay time. They require measurement methods different from those of load or loss. Ideally, delays would be measured separately for each call; however, a very busy period is a poor one for the switching system to divert its activity from call processing to service measurement. Instead, the usual delay measurement is made by placing test calls, at fixed intervals either within the system or "externally" by an attached test-call originator. In this way, a fixed amount of time is devoted to the measurement which is independent of the load on the system. The total number of test calls and the number of test calls that are delayed over an objective time are reported to the data acquisition system of TNDS.

### *3.2 Measurement timing*

As with trunking, most traffic measurements of switching systems are based on a concept of a busy hour, that is, a single clock hour of the day such as 9:00 to 10:00 am, which is identified as the hour of the day in which, on the average, more traffic is carried than in any other. It has long been recognized that the busiest hour of a particular day may not coincide with the "busy hours" as defined above. A newly

developed engineering concept to take this into account is described at the end of this section.

As we mentioned previously, definitions involved in traffic measurement differ from those for trunks. Various "engineering hours" used in switching are described in the following sections.

#### *3.2.1 Average busy-season busy hour*

Through special studies, often made at the start of the busy season (which is itself usually well known through previous data) by means of special programs available in TNDS, the hour of the day that exhibits the highest average load, when averaged over the selected days of the study period, is identified as the busy hour. The load may be number of calls, usage, or both, depending on the type of switching system component being measured. Different components may have different busy hours. Measurements are taken in the busy hours on all days throughout the year. The traffic measurements of those three months that have the highest average busy-hour traffic are averaged to get the average busy-season busy-hour traffic. (Weekends and holidays are usually excluded from this calculation.) Provision is made within TNDS to measure loads in more than one candidate busy hour, to handle cases where different switch components have different busy hours, or when two hours are so close together in load that a clear decision between them cannot be made ahead of time.

#### *3.2.2 Average 10-high-day busy hour*

As the name implies this measurement period still uses the busy hour; however, the average usage or call volumes of the ten highest busy hours of all the months of the year are used rather than just those of the busiest three months.

#### *3.2.3 High-day busy hour*

This is the highest of the 10-high-day busy hours for a given year. Provision is made for deleting measurements made on days of a very unusual type that are not expected to recur. Service impairment on days with such unusual traffic extremes is generally accepted by the public because the cause is obvious: blizzard, flood, tornado, local or national disaster. Provision of sufficient equipment to handle such extremes without service impairment is uneconomical; network management serves under such circumstances to help the network complete as many calls as possible.

To assist the traffic engineer in evaluating the 10-high-day and high-day busy-hour values, a comparison is made by the TNDS Central Office Equipment Report (COER) system based on a theoretical model of traffic variation. Studies have shown that the probability distribu-

tion of the busy hour traffic carried by central office components can often be approximated by a gamma distribution. The gamma distribution is determined by its first two moments, so that a fitting curve can be generated from the first two moments of the observed data. COER makes such a calculation for the current year of data being collected and prints out in parallel columns the values of the observed highest 15 measurements and the values computed for a gamma distribution with the same mean and standard deviation as all of the busy-hour measurements taken to date. A quick visual check will give the engineer some idea of whether this traffic is typical. In particular, an impression can be given of whether there is more or less volatility in the traffic under study. Also, if high-day engineering is required, the difference between actual high-day load and the gamma projection will help the administrator judge whether the high-day load may have been a data outlier.

#### *3.2.4 Extreme-value measurement period*

This type of measurement period differs markedly from the three previous ones. As the costs of collecting data have changed, it has been possible to do a limited amount of processing at the collection point so that long-term storage of many measurements can be eliminated. The hour with the highest measured value for the day can be retained and all others discarded. The resulting measurement can be used to obtain reliable estimates of the extremes of traffic and service to be encountered. A different criterion of service is possible now; instead of average or high-day service, the objective can be set in terms of the expected number of hours or days in which service fails to meet a given criterion. For example, one objective now used is that on the average no more than one day a month will contain an hour in which dial tone delay exceeds 8 percent over 3 seconds. Because the criterion applies to only one day of the month, the probability of delay can be made significantly higher than the average 1.5 percent over 3 seconds used for average busy-season busy hour. In this example the criterion was picked so that in an average office there would be no noticeable change in service on changing from busy-hour engineering to extreme-value engineering.

Extreme-value methods are still being incorporated into the TNDS. Studies of the statistics of extremes indicate that extreme-value methods result in much more accurate projection of peak period loads than do current methods.[10,11] This leads to more accurate forecasting and engineering.

### *3.3 Types of equipment*

Measurement periods and the accompanying criteria of service vary

among the three general types of central office components that are described next. Also in this section, we discuss both load (traffic intensity) and call volume. For example, the common control elements of a system are affected primarily by the number of calls processed, while the switching network is affected primarily by the average number of simultaneous conversations in progress.

#### *3.3.1 Switching network*

The capacity of the switching network portion of a switching system is almost always a function of the conversation load. Load capacity is specified in units of erlangs or CCS. (See Section 2.3.1.)

Electromechanical switching networks in general react with rather slowly rising blocking to increasing load, so that a criterion of loss averaged over the busy hours is used similar to the trunking criterion. On the other hand, time-division networks generally display very low blocking until high load is reached where blocking rises quickly.[12] In the latter case a high-day criterion is used.

The theoretical reasons for this can be seen by considering the effect of the number of parallel paths on the load-loss characteristics. Blocking in a typical three-stage network tends to follow the simple formula:[13]

$$B = [1 - (1 - a)^3]^n, \tag{17}$$

where:

$B$ = blocking (loss)
$a$ = average load in erlangs carried by a link connecting two stages
$n$ = number of parallel paths.

This formula is plotted in Fig. 7 for 10, 30, and 128 parallel paths to show the effects that different blocking characteristics may have upon the selection of network service criteria. For illustration purposes, the ratios of high-day to 10-high-day to average busy season, busy hour are selected at 1.15:1.10:1.00. If the objectives are $B = 0.02$ for the average busy-season busy hour, $B = 0.05$ for the 10-high day average, and $B = 0.10$ for the high-day, three different networks might require three different engineering criteria.

The curves show the result of having selected the appropriate criterion. For the network with ten parallel paths the *ABS* criterion is appropriate; the 10-high-day and high-day service will be better than required. For a network with 30 parallel paths, however, the 10-high-day criterion is picked; the high-day and *ABS* service will be better than required. Finally, for the network with 128 parallel paths only the high-day is of concern; essentially no calls will be blocked on any other day. In fact, for this network it is likely that a service criterion will not be used directly, but that a maximum occupancy for the high-

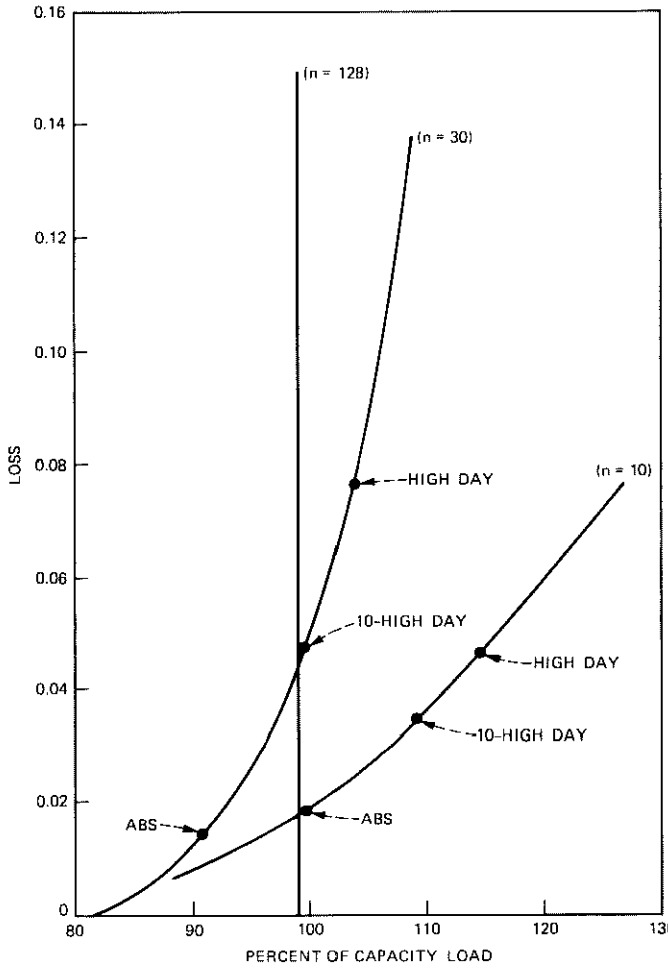



Fig. 7—Effect of parallel paths on service criterion—based on Loss = $[1 - (1 - a)^3]^n$.

day load will be adopted in order to leave room for error in predicting the high-day load.

### *3.3.2 Service circuits*

Service circuits are the circuits that are provided in switching systems to give tones or to record customer dialing, to accept in-band signals from other offices, or to transmit such signals to other offices. Depending on the size of the office, these are provided usually in small groups ranging from five to ten, but they may occur in large groups of well over 100. Because of the way in which service circuits are used by the switching systems, service circuits delay rather than block cus-

tomers' calls during periods of congestion. The delay criterion of service used here is expressed in percent of calls delayed over a specified number of seconds. Depending on the group's size and holding time, the average, 10-high-day or high-day criterion may be used. The choice of criteria depends to a large extent on the judgment of the traffic engineer. Customer reaction, cost of equipment, stage in the progress of a call, traffic volatility, and interaction with other parts of the switching system all play a part.

The capacities of many service circuit groups are determined from delay calculations using the erlang delay formula when service times are highly variable or using the Crommelin-Pollaczek delay curves when holding times are nearly constant.

#### *3.3.3 Central processors and controls*

The control components of switching systems that are mainly used in setting up a call generally have very short holding times and are provided in small quantities, often only one per system. The probability of delay is very high, but, because of the very low holding time per call, delays are usually unnoticeable. Criteria are based on delays on the highest day, because when delays start to become noticeable the processor is usually close to saturation, where no further increase in load can be handled. In such cases if additional load is offered, customer calls will be blocked. Customers try again when blocked, so the number of attempts will rise faster than the carried load. Eventually, so much processing time is used for uncompleted attempts that the number of good calls completed by the switching system is reduced.

### *3.4 Traffic engineering models*

The engineering of most central office equipment is based on assumptions of random call originations with constant source loads during the busy hour but with day-to-day variation. In general, peakedness is assumed at unity, although it is known that there are phenomena, such as customer retrials, that can produce some of the effects of peakedness. Also, toll offices high in the hierarchy will handle a large quantity of overflow traffic, and thus peaked traffic. The effect of this peakedness may seriously affect capacity in the periods of overload. In spite of the differences from the model underlying the Poisson formula, that is, constant traffic intensity and random call arrivals, the formula has been the work horse for much of central office engineering because it was the best choice in predicting actual system performance.

The Poisson formula is not applicable to internal switching networks. Formulae of the general form of eq. (17) give remarkably close approximation to the performance of this type of network. Observa-

tions show that a particular network will display a load-loss relationship that parallels such a theoretical curve. Differences may be attributed not only to the simplified assumptions of the formula but also to imbalance effects, day-to-day variation, systematic load variation within the hour, or minor variations in the network structure from that assumed in the model. The TNDS supplies data from which comparisons of actual with theoretical load-loss relationships can readily be made.

Common control equipment, including call processors, present a different traffic modeling problem from what has been considered so far. Loss or delays are insignificant most of the time. The event of call processor overload is so infrequent that measurements are usually too few for making useful load-loss curves to compare with theory.

With electromechanical switching systems, the proportion of time that a processor, such as a marker, is busy can be measured with little complication. A plot of occupancy against calls gives a clear picture of the number of calls that will fully occupy the processor. Capacity is usually picked at a lower number, say 95 percent of full occupancy, and applied to high-day busy-hour traffic.

A stored program controlled processor, however, is busy all of the time; when it is not processing calls, it is performing maintenance or audit functions or looking for work. The control program is designed to reduce the time spent in non-call processing activities as the call processing load increases. In many systems, this is done by organizing the processing into a fixed order of tasks. Processing jobs that must be done with little delay are given priority over those with less stringent time requirements. So that some maintenance and audit work is done even in periods of heavy load, the search for work is made in a cyclical fashion over the various priorities with many repeat searches for high-priority work. Thus the time taken between visits to the lowest priority level becomes longer with increased processing load. The capacity of the processor, in terms of high-day busy-hour calls, is then found by analyzing observations of offered originating-plus-incoming calls with the corresponding observations of basic processor cycles, i.e., visits to the lowest priority level. A first-order least-squares fit to these points is made and a 90-percent confidence line is computed and drawn below the fitting line (90 percent of the observations are expected to fall above this line). Next a horizontal line is drawn at the minimum number of visits that allow acceptable service as determined by simulations and field studies of many systems. The intercept of the 90-percent line and the minimum-number-of-visits line gives the estimated call capacity for the high-day busy hour. This is illustrated in Fig. 8.

The choice of 90-percent confidence is, of course, a judgment choice.

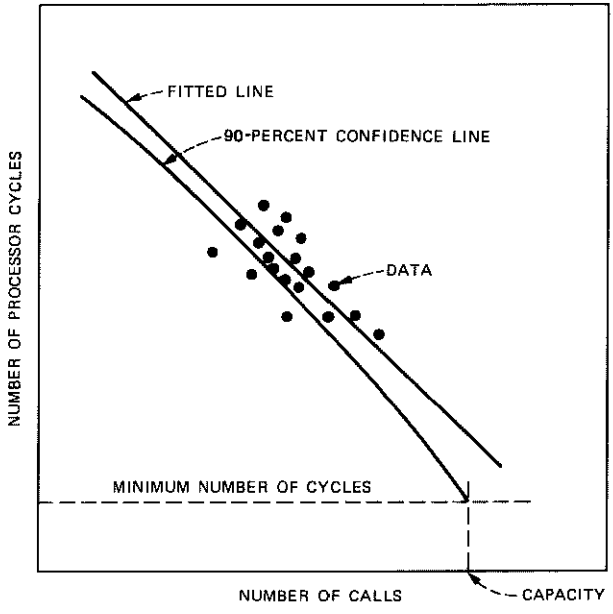



Fig. 8—Use of measurements to determine central processor capacity.

It is based on experience gained from many switching systems. A load greater than capacity will often cause no problem; on the other hand, a load less than capacity may rarely overload the processor.

### *3.5 Load balance*

The switching networks of most switching systems require balancing of offered traffic over the network switches and frames. Unbalance arises from the nature of traffic sources—some lines or trunks generate more traffic than others. Correction of unbalance is particularly needed in the concentrating stages of the network, where the number of input lines exceeds the number of output links. While the first objective of load balancing is to equalize service among customers, it also achieves a second objective of improving the load-carrying capacity of the network. The latter effect of load balancing arises from the nonlinear load-loss curve that exists for most networks. As Fig. 9 illustrates, the higher loss in a section of network loaded above average is not completely compensated for by the lower loss in a section loaded, by the same amount, below average. Therefore, reducing the spread of the distributions of loads around the mean will reduce average blocking at a given load or enable increased loading at a given service objective.

It is an objective of line-load balancing to control the assignment of

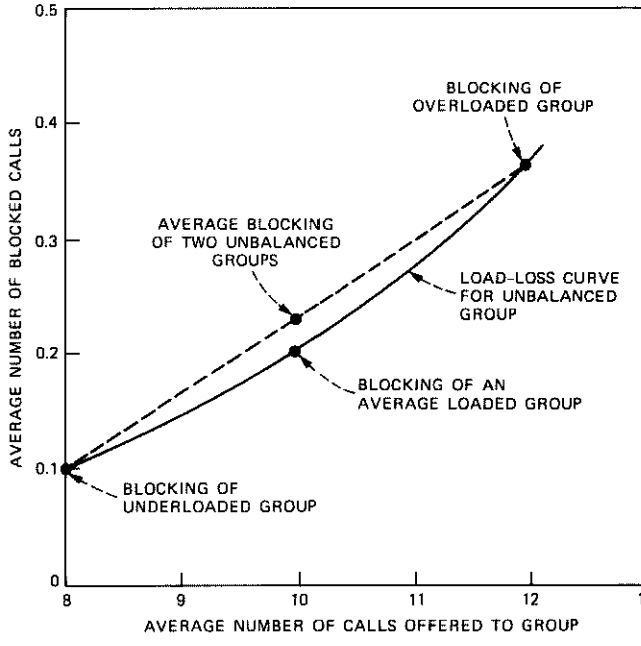



Fig. 9—Effect of unbalance on blocking.

new lines in such a fashion that all units stay in traffic balance. In a 55,000-line 1A *ESS** switching equipment for example, this implies keeping track of a number of separately tracked "loading modules" that may approach a thousand. Even in smaller offices the problem is far from trivial. The measurement and processing systems of TNDS provide the means of controlling load balance—without excessive manual effort—by accumulating line-group usage for all of the designated hours of the week. Also, measurements of individual line usage are taken for guidance in selecting lines when extreme unbalance requires shifting lines from one particular group to another.

The first step in balancing concentrator loads is to assign lines in such a fashion that every concentrator has the same number of lines of each identifiable traffic or class of service. Unfortunately, this is not enough to ensure balance.

The first systematic application of traffic measurements to load balance was reported in a study by D. H. Barnes in 1958.[14] Barnes proposed a method of statistical control that would separate effects of chance load variations from those of systematic load balances. It was implemented on the computers of that day as well as in a manual

* Trademark of Western Electric.

procedure (the "score method") that would permit the calculations to be made with simple mathematical operations.

The type of solution proposed by Barnes has been refined over the years as more computing power has become available and more measurement refinement has become possible. The operational problem to be solved remains the same however: to separate the random variations from the systematic ones out of the large variations that occur in the hourly measured traffic on a concentrated group of lines. This is the same problem faced in the design of a statistical quality control procedure. It involves estimating load variance and measurement error, and displaying the results in a form that will enable managers to take appropriate action without moving lines needlessly or letting service deteriorate. The TNDS Load Balance system takes over the computation of balance indices and processing of the balance data.

## IV. NETWORK MANAGEMENT

The background of network management is covered in a companion article to this one and in Ref. 15. The main difference in the requirements placed by network management on TNDS from those of other traffic data applications is the very fast response required between traffic events and the reporting of them. Because of the long-established need to keep network managers informed within seconds of a major control loss or within minutes of serious call completion problems, the traffic-gathering parts of TNDS must have—and do have—the capability of passing data quickly and accurately to the network management computer and relaying control orders quickly back to the measured switching systems.

## V. CONCLUSION

Throughout the evolution of TNDS there has been a continuous increase in the accuracy and availability of traffic data. This in turn has made possible more accurate traffic models and has spurred activity for better traffic theory and better network operating practices. What has been reported here is the state of the art at this time. The coming years may be expected to bring more changes and improvements.

## REFERENCES


1. D. W. Hill and S. R. Neal, "Traffic Capacity of a Probability-Engineered Trunk Group," B.S.T.J., *55*, No. 7 (September 1976), pp. 831–2.
2. R. I. Wilkinson, "Theories for Toll Traffic Engineering in the U.S.A.," B.S.T.J., *35*, No. 2 (March 1956), pp. 421–514.
3. R. I. Wilkinson, "Some Comparisons of Load and Loss Data with Current Teletraffic Theory," B.S.T.J., *50*, No. 9 (October 1971), pp. 2807–34.

4. M. Eisenberg, "Engineering Traffic Networks for More Than One Busy Hour," B.S.T.J., *56*, No. 1 (January 1977), pp. 1–15.
5. A. J. David and C. D. Pack, "The Sequential Projection Algorithm: A New and Improved Traffic Forecasting Procedure," 9th International Teletraffic Congress, Torremolinos, Spain, 1979.
6. R. L. Franks et al., "A Model Relating Measurement and Forecast Errors to the Provisioning of Direct Final Trunk Groups," B.S.T.J., *58*, No. 2 (February 1979), pp. 351–78.
7. J. P. Moreland, private communication.
8. S. R. Neal, "Blocking Distributions for Trunk Network Administration," B.S.T.J., *59*, No. 6 (July 1980), pp. 829–44.
9. C. R. Szelag, "Trunk Demand Servicing in the Presence of Measurement Uncertainty," B.S.T.J., *59*, No. 6 (July 1980), pp. 845–60.
10. D. H. Barnes, "Extreme Value Engineering of Small Switching Offices," 8th International Teletraffic Congress, Melbourne, Australia, 1976.
11. K. A. Friedman, "Extreme Value Analysis Techniques," 9th International Teletraffic Congress, Torremolinos, Spain, 1979.
12. J. F. Huttenhoff et al., "No. 4 ESS: Peripheral System," B.S.T.J., *56*, No. 7 (September 1977), pp. 1029–55.
13. C. Y. Lee, "Analysis of Switching Networks," B.S.T.J., *34*, No. 6 (November 1955), pp. 1287–1315.
14. D. H. Barnes, "Statistical Methods for the Administration of Dial Offices," 2nd International Teletraffic Congress, The Hague, 1958.
15. D. G. Haenschke, D. A. Kettler, and E. Oberer, "Network Management and Congestion in the U. S. Telecommunications Network," IEEE Trans. Commun., *COM-29*, No. 4 (April 1981), pp. 376–85.

## AUTHORS

**Walter S. Hayward, Jr.,** A.B., 1943, S.M. (Electrical Engineering), 1947, Harvard University; Bell Laboratories, 1947—. Mr. Hayward has worked in the field of telephone traffic and switching systems engineering. In 1961, he was appointed Head, Electronic Switching Studies Department. In 1964, he was appointed Director of the Traffic Studies Center. He is now Consultant, SPC Studies Center. Member, IEEE, ORSA, and ACM.

**James P. Moreland,** B.S.E.E., 1964; M.S.E.E., 1964; Ph.D. (E.E.), 1967, Ohio State University; Research Associate, Electroscience Laboratory, 1964–1968, Instructor, Electrical Engineering, 1967–1968, Ohio State University; Bell Laboratories, 1968—. At Ohio State, Mr. Moreland worked on studies of scattering theory and optical heterodyne detection. At Bell Laboratories, he has been concerned with clock-synchronization schemes for digital communications networks, optical-fiber transmission studies, and traffic and facility network planning. He is presently a Supervisor in the Trunk Traffic Engineering Department. Member, IEEE, Eta Kappa Nu, Tau Beta Pi, Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

*Total Network Data System:*

# Data Acquisition and Near-Real-Time Surveillance by EADAS

By C. J. BYRNE,* D. J. GAGNE,* J. A. GRANDLE, Jr.,* and G. H. WEDEMEYER†

(Manuscript received May 3, 1982)

The Total Network Data System (TNDS) requires a facility to collect the traffic data generated by electromechanical and electronic offices. The Engineering and Administrative Data Acquisition System (EADAS) fulfills this function. EADAS systems that serve electromechanical offices employ unique self-testing circuitry to interface to central office signals. A novel buffering scheme also improves system efficiency. For electronic offices, a specialized file system has been developed, and the input data are specified in a high-level language. All of these features permit EADAS to be a cost-effective, flexible, and reliable data collection system. EADAS also provides real-time surveillance and reports for the switching offices it serves. A set of reports is predefined; however, the user may also specify reports using the Network Operations Report Generator feature (NORGEN). System capacity is specified in simple formulas which are derived and presented. With these features, EADAS provides a comprehensive facility for presenting real-time traffic data to the telephone companies.

## I. INTRODUCTION

High-quality telephone service requires facilities that collect and process data on the traffic being handled by a telephone system. In

* Bell Laboratories. † AT&T.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

particular, it needs data on the number of calls being handled by the system, the duration of each call, and the number of times a call encounters difficulties as it proceeds through the system to its destination.

Such data are needed for surveillance of the quality of service being provided. This requires, for example, measuring in real time the interval required to obtain dial tone, and informing responsible personnel when such intervals exceed acceptable limits.

Traffic data are also used to determine the proper quality and distribution of central office equipment and trunk facilities. Other uses are to provide telephone customers with special reports on the performance of their own particular facilities and grade of service, and to assist telephone company personnel in proper maintenance of equipment.

In the Bell System, traffic data are collected and processed by several different systems. Among these are the Telesciences, Inc. "Automatic Traffic Recording and Analysis Complex" (AUTRAX) System, the Conrac Corp. "Alston Traffic, Engineering and Management Information System" (ATEMIS), and the Western Electric "Engineering and Administrative Data Acquisition System" (EADAS). The operation of EADAS will be described in more detail in succeeding paragraphs to illustrate how traffic data are typically collected and processed.

## II. SYSTEM CONCEPT

Figure 1 shows the overall system concept of EADAS. A centrally located minicomputer collects data from the central offices. Within certain limitations described later, these offices can be electromechanical or electronic and can be large or small, with and without toll features.

Traffic data are collected via a data-link facility between each central office served and the EADAS central unit (CU). As mentioned above, EADAS provides information to telephone company network administration personnel for near-real-time surveillance on the quality of telephone service being provided. EADAS also makes available information to assist maintenance personnel in remedying equipment problems before they can affect service. This information is in the form of reports produced by the Network Operations Report Generator (NORGEN) feature of EADAS. These reports can be in printed form or displayed on a Cathode Ray Tube (CRT). NORGEN produces a wide variety of such reports, which are made available on demand, on a previously established schedule, or automatically when an abnormal service condition (exception) develops.

NORGEN was introduced in 1977 to replace the demand for re-

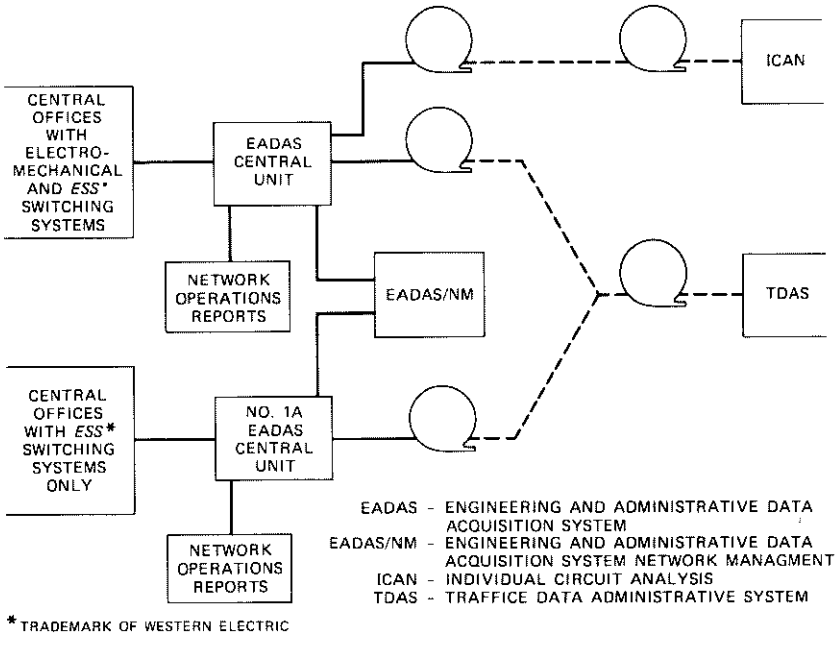



Fig. 1—The Engineering and Administrative Data Acquisition System concept.

porting features that had exceeded the calculation capacity and flexibility of the original reporting features of EADAS. NORGEN provides a calculation capacity six times larger and with greatly increased flexibility, as well as a set of standard applications programs designed to meet those near-real-time report needs required on a systemwide basis. A programmability feature allows the user to modify the standard reporting features and to provide new ones to meet local conditions.

Figure 1 shows that traffic data collected by EADAS can also be recorded at regular intervals on magnetic tapes. These tapes are physically transported to the Traffic Data Administration System (TDAS) and to the Individual Circuit Analysis (ICAN) program described in the article on TNDS equipment systems[1] in this issue. TDAS formats the data for use by other downstream software systems. These systems report on longer-term engineering functions, such as determining how much of a switching system's capacity is being used and forecasting future trunking requirements. ICAN analyzes subsets of the data collected by the Individual Circuit Usage Recording (ICUR) feature of EADAS to detect record base errors, equipment faults on individual circuits in electromechanical offices, and malfunctions in the data collection and processing functions performed by the central office and EADAS facilities.

Figure 1 also shows the data link between EADAS and the EADAS/

Network Management (EADAS/NM) System. Over this link, traffic data are sent at five-minute intervals to alert the local Network Management Center or Regional Operations Center to network problems. Signals from EADAS/NM, which activate various network controls such as trunk reroute, and signals to EADAS/NM, which indicate the status of these controls, are also transmitted over this link.

## III. FIELD OF APPLICATION

There are two versions of EADAS. The original system, No. 1 EADAS,* uses a 16-bit minicomputer with 128K words (word equals two bytes) of memory. This system was designed to serve No. 1 and No. 5 Crossbar, Crossbar Tandem, and Step-by-Step type electromechanical offices, as well as 1/1A and 2/2B *ESS*† switching equipment, and Remote Switching System (RSS) offices. The No. 1 EADAS was first installed in Kansas City, Missouri, in 1973. In 1974, the ICUR feature was first installed in Miami, Florida.

A later version—the No. 1A EADAS—was designed to serve *ESS* switching equipment offices only. The No. 1A EADAS employs a faster minicomputer (three times the processor speed) with expanded memory capability (larger address range and use of cache memory). The No. 1A EADAS has approximately twice the capacity for serving *ESS* switching equipment offices, with a lower installed cost than No. 1 EADAS and the flexibility to grow with the continued expansion of Stored Program Control electronic switching in the Bell System. All No. 1A EADAS installations have the NORGEN feature. The No. 1A EADAS was first installed in New York City in 1977. The No. 1A EADAS is presently arranged to serve 1/1A, 2/2B, 3, and 5 *ESS* switching equipment offices, as well as RSS offices. In addition, coverage is provided for Northern Telecom's DMS-10‡ offices.

The No. 1 EADAS met the Network Administration needs of electromechanical offices. However, the Bell System trend toward adoption of Stored Program Control (SPC) for new and replacement offices is reducing the number of electromechanical offices in service. This, together with the advantages of No. 1A EADAS for SPC entities, has caused a leveling off in the number of No. 1 EADAS installations. On the other hand, the number of No. 1A EADAS continues to increase in proportion to the installation of Stored Program Control offices.

## IV. DATA COLLECTION

The data collection process differs between the electromechanical offices and the electronic offices. Electromechanical data are collected

* The offical designation of the original system is simply "EADAS."
† Trademark of Western Electric.
‡ Trademark of Northern Telecom.

from several terminal types. The predominant terminal type provides real-time data, while data are also accumulated and held by other types of terminals in end offices. The accumulated data sources operate over either a dialed-up or dedicated link, resulting in three different interfaces for the electromechanical offices. On the other hand, electronic offices send only accumulated data to the EADAS, and each electronic system has a unique output format. The data collection process in electromechanical offices will be discussed first.

A number of requirements were considered in developing the electromechanical design:

1. It was advantageous to use the data collection equipment of the existing Traffic Data Recording System (TDRS) already installed in a number of central offices. The main concern here was the high cost of installing new traffic measurement equipment in electromechanical offices. This is caused by the large number of points to be monitored, which are scattered throughout the office and require many individual wiring runs.
2. Data were to be collected in real time to provide information for exception reports and the network management feature (see other articles on Network Management).
3. The system was to provide new features such as Individual Circuit Usage Recording (ICUR), described later, and office control functions for the network management feature mentioned above.

The traffic data equipment used by the then existing TDRSs, known as the Traffic Data Converter, operated as follows. The unit, located in the central office, is connected to the data collection center via a dedicated link. The occurrence of an event in the central office (i.e., generation of a call or detection of a busy on a usage scan) causes the converter to generate a data word representing the input address or connection point for the event. Thus, the function of the data collection center is to sort and sum the occurrences of these address words by address for the desired measurement period. While this function is simple in concept, it can quickly become complex to provide a capacity of 100 data channels, each capable of generating 1000 input addresses. Since all of the real-time data are available at the collection center at any instant, centralized accumulation is ideal for real-time data collection. Thus, the network management and real-time reporting functions can be readily accomplished.

### *4.1 Input network*

The desire to provide new features and a capacity greater than that of the TDRS led to the design of a new converter for collecting data that could be used in those offices not already containing TDRS equipment. In addition to collecting peg counts, this design provided

remote office control, fast scan usage data, individual input scaling (to reduce the data channel load for rapidly occurring events), concentration, and ICUR. This converter, called the EADAS Traffic Data Converter or ETDC, had a number of interesting features. Perhaps the most interesting is the input circuitry used to interface the system to the central office. This input circuit is shown in Fig. 2. The circuit has to provide a high impedance input which could monitor ground closures provided to relays (without changing operating characteristics), while absorbing inductive spikes and contact bounce associated with the relays. In addition, a periodic self-test should be provided. The input circuit provides these features as well as an input verification feature.

Meeting the high impedance requirement was a problem. Transistor-transistor logic (TTL) of the early 1970s employed 2K pull-up resistors, and a quick analysis will show that to achieve a low on the input, the maximum resistance to −48V could be no greater than 20K. This was not anywhere near the 100K to 300K desired. By power switching the TTL gate, the high input impedance can be obtained since the TTL gate is effectively removed until the input is scanned. A capacitor (C) added to the circuit to remove noise pulses serves as a charge storage point and holds the input state during the short interval when the power is switched onto the TTL gate. With this arrangement, the levels fed to the TTL gate are determined by the external voltage divider $R_P$ and $R_I$.

This network is also testable. If the power switch is turned on permanently, all of the input capacitors will charge through the TTL pull-up resistor, and all of the input points should read a high state if

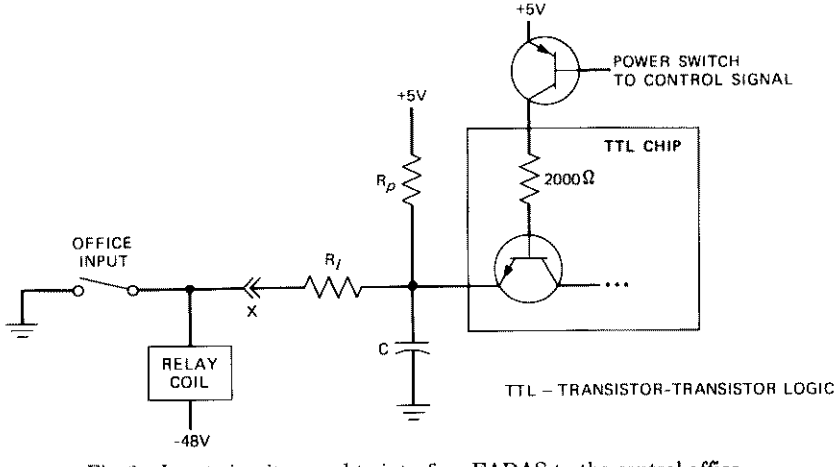



Fig. 2—Input circuitry used to interface EADAS to the central office.

a failure has not occurred. To test the low state, the +5V supply is removed from the resistor $R_P$. This results in all inputs being low since the voltage feeding the capacitor is either negative or zero.

Another test feature is available. Note that if a positive voltage is applied to the input at point X in Fig. 2, it is possible to introduce a high input on the TTL gate. Thus, if one takes a positive supply and pulses an input, it should register a count. This feature has two uses. First, input connections can be verified in a central office by going to the input source and pulsing it with positive voltage. Even though other inputs are active, only the positive input will count. Second, using this same technique, any crosses to other inputs will appear as additional counts or generation of additional addresses.

### *4.2 Electromechanical CU operation*

Having described some features of the central office circuitry, let us turn to the central data collection operation. As stated earlier, this requires the ability to collect data from up to 100 links with 1000 addresses each. The simplest approach would be to use 100,000 words of memory and use the incoming address along with its link number as the address of a location to be incremented by one. Needless to say, this operation is memory intensive, and considering that the capacity of the minicomputer is 128K words, not very efficient.

A more appropriate implementation is to store the data on a bulk storage device, such as a fixed head disk, and buffer the incoming data until a sufficient amount is collected to update the disk. Once again complications arise. Disks can only be written one sector or one track at a time. For the disk selected, 2048 words represented a track. This was used to hold two incoming channels with 48 words for header. Since the rotation speed of the disk is 1800 rpm, and two rotations are required to update two channels (one read and one write), the maximum rate for updating 100 channels is

$$1800 \frac{\text{rotations}}{\text{minute}} \times 1 \frac{\text{channel}}{\text{rotation}} \times \frac{1 \text{ update}}{100 \text{ channels}} = 18 \frac{\text{updates}}{\text{minute}},$$

if no other accesses are allowed. However, in actual operation the data must be read and used for processing so that only half of all disk accesses are allocated to input processing. Therefore, nine channel updates/minute are provided. Thus, the incoming data must be buffered for one ninth of a minute or 6.67 seconds. Since the arrival rate at maximum channel capacity is one address every 12.5 ms (an address is 15 bits long including start, stop, and parity and is transmitted at 1200 baud), then a buffer of size 534 is required for each channel. For 100 channels, the buffer contains 53,400 words of memory.

If one looks at actual data storage needs, one would find that at any

given time nearly half of the cells in these buffers are empty. Thus, in reality only 26,700 words are needed if an algorithm can be developed to access the space. This is done using a buffering scheme that employs the triangular geometry discussed below.

First, assume that every time data are gathered into the buffer for all channels, only one channel can be updated from the buffer to the disk. Second, assume a system with five channels. Then, the buffering structure will be as shown in Figs. 3 and 4. The process is as follows. For the last channel updated (initially use channel 1), place the scanned incoming data sequentially in the vacated locations, starting with the first entry in the row assigned to that channel (channel 1 has row 1; channel 2, row 2; etc.). When the right row edge is reached, the remaining data points are placed in the column under the right edge. This operation is shown for five scans in Fig. 4, where once five scans are completed the appropriate cells for the data from the next scan are empty and the process can be repeated. Generation of addresses for using this algorithm on a digital computer is quite simple. The cells are numbered as shown in Fig. 5. Then, given:

$S$ = Scan or channel being processed,

$X$ = Start address of buffer − 1,

$N$ = Number of input sources,

a. To find the first location $C_1$,

$$C_1 = X + S$$

b. To find the remaining locations 2 through $N$,

$$C_A = C_{A-1} + N - (A - 1) \quad \text{for} \quad 2 \leq A \leq S,$$

$$C_A = C_{A-1} + 1 \quad \text{for all other } A, \quad S < A \leq N.$$

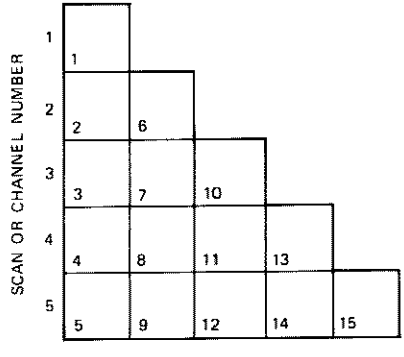



Fig. 3—Buffering scheme used to develop the algorithm that determines actual data storage needs.

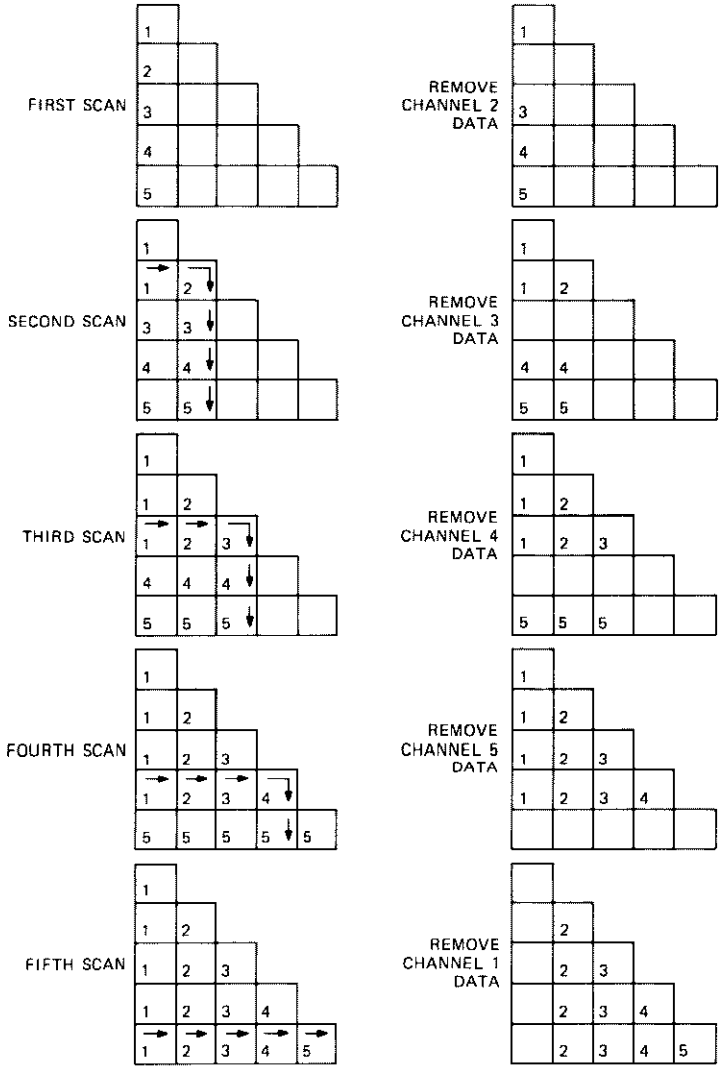



Fig. 4—Remaining data points in column under right edge of buffer structure, for system with five channels.

The above example assumed one update process for each data scan. In EADAS, scanning must occur once every 12.5 ms, while the update process takes four disk rotations for every two channels or two rotations per channel, which is equal to 66.6 ms (i.e., 66.6/6, which is the implementable rate closest to 12.5 ms). If scanning occurs every 11.1 ms, then six buffers of edge size 100 can be employed. This requires 6 $(100^2 + 100)/2$ memory locations, or 30,300. This number is slightly

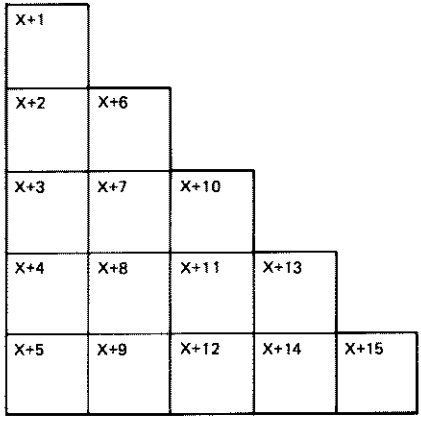



Fig. 5—Scheme for numbering buffer so that it is a contiguous string of cells.

larger than required by the 11.1-ms actual scan rate. Thus, with this algorithm, memory requirements for data collection were reduced, resulting in additional space for other functions.

In addition to regular group usage (data which are collected over all trunks in a group), the EADAS Traffic Data Converter (ETDC) includes circuitry to provide Individual Circuit Usage Recording (ICUR). These data require individual transmission of each of 3600 inputs on up to four 4A Traffic Usage Recorders (TUR), or a maximum of 14,400 inputs, as well as the 1000 inputs already available to the ETDC. Since the ETDC design had a 15-bit word (two start bits, an ICUR bit, ten data bits, a parity bit, and one stop bit) with only ten data bits, a multiword format was necessary. Two words were used to indicate the state of six inputs. The ICUR bit is set in each word, and the remaining 20 data bits are employed as follows. Two bits in one word serve as a first or second word indicator, while the remaining 18 bits indicate the six input states, and the address of those six inputs [two bits for the TUR number (0–3), four bits for the horizontal (0–9), and six bits for the vertical position (0–59)]. These data are intercepted by the scanning process, assembled and buffered, and then used to update individual circuit data, as well as form grouped totals which are merged with the accumulated peg count information. The updating process involves reading the usage and the peg count data from the disk, merging, and then writing the merged peg count data to the disk. The reads and writes are performed using Direct Memory Access (DMA) techniques. Dual-port memory arrangements are needed to accommodate the cumulative load of this processing and the data collection process. The system uses the second port on the ICUR disk as a transfer port, loading data into the added memory port. The

individual circuit data are passed downstream and analyzed by the ICAN program described in the article on TNDS equipment systems in this issue.[1]

As memory costs decreased, devices with internal data storage became available. These sources could accumulate data in the central office terminal and forward it to the central unit on command. This method of operation had the potential of a low initial cost for a small number of offices by reducing the initial complexity of the central data collection facility. In addition, there was the opportunity to handle small offices via dial-up data links where dedicated links were not cost effective. For small offices only a few inputs are required, network management data are not needed, and only busy hour information is collected. To serve small offices, the Pollable Data Terminal was designed, and an interface specification published to permit traffic-data-gathering equipment to be connected in this mode. Operation of the Pollable Data Terminal is over a dial-up link, and depending upon busy hours, one EADAS interface port can serve many offices.

### *4.3 Electronic CU operation*

The No. 1A EADAS is designed to collect data from electronic offices only, since plans to replace electromechanical offices were well under way and sufficient No. 1 EADAS Systems would soon be deployed to accommodate these offices until replaced. Initial software was developed by deleting the electromechanical features from the No. 1 EADAS. The initial No. 1A EADAS software was written in assembly language and ran under a very specialized operating system. The No. 1A EADAS software has since been rewritten in C language and now runs under the *UNIX** operating system.

Conversion to the *UNIX* operating system led to several interesting software operations. First is the reception of the accumulated data from the *ESS* switching equipment offices. Each switching machine type has its own unique output format (i.e., 1 *ESS* electronic switching is different from 2 *ESS* switching equipment, from 3 *ESS* switching equipment, etc.). Thus, a unique program was written to receive the data from each type, convert it, and store it in a common form. This task is handled by using LEX, a lexical scanner generation program, and YACC (Yet Another Compiler Compiler) to produce a scanner and parser to process the incoming data stream. The incoming data from the office are treated as a grammar. The description of this grammar is fed to LEX, producing a scanner which identifies the

* Trademark of Bell Laboratories.

tokens. A second description is fed to YACC to generate the parser. The token output from the scanner is passed on to this parser, which then prepares the data for storage by converting the data to binary and removing the header text. This approach significantly reduces the effort necessary to add new office types to the data collection portion of No. 1A EADAS. In addition, if errors are detected in data specification (either the format has changed or the specification was improperly interpreted), modifications are easily made by changing the descriptions of the grammar.

### *4.4 Overview of the No. 1A EADAS File System*

The function of the No. 1A EADAS File System is to provide storage and retrieval capabilities for near-real-time switching network data collected from *ESS* switching equipment offices. These data are needed to produce status and exception reports for network administrators, and for transmission to EADAS/Network Management Systems, as well as downstream data analysis systems. To provide these functions, No. 1A EADAS must store large amounts of data collected from the *ESS* switching equipment in a database. This task is complicated by the high volume of data, the data storage and retrieval speed required of the file system, the large number of offices from which data are collected, and the necessity for storing data on disk over time for use by the report generation programs. The impact these requirements had on the development of the file system for No. 1A EADAS is described in this section.

#### *4.4.1 File system requirements*

The No 1A EADAS File System provides the following capabilities:

1. It allows efficient input of near-real-time register data collected from *ESS* switching equipment offices. EADAS is able to collect data simultaneously from at most 48 entities and store it in the database.

2. Many different types of data collected from *ESS* switching equipment offices are stored in the database: discretes, five minute, hourly or half hourly, daily, weekly, etc. The type of data actually stored is a function of the particular office involved, and the number of registers varies by office and type of data. For example, the 1 *ESS* switching equipment may send about 7000 registers every half hour for report generation, but only about 1400 registers for its daily transmissions.

Although register data are by far the largest in volume and require the most rapid input and retrieval, there are several other types of data that are stored in the EADAS database. These include per-office reference data, thresholds for data exception generation, and time schedules for distributing reports to various network terminals connected to the system. The file system accommodates these files as well

as the register files. Currently, 36 types of data are stored in the EADAS File System.

3. Most EADAS data are collected on a periodic basis (e.g., every five minutes or every half hour). There are several reasons why successive intervals of data must be stored in the EADAS database. In some cases, calculations are performed on data received during successive collection intervals. In other cases, there is a need to have access to "old" data so that reports may be generated from special studies or for investigating office problems. Finally, a buffering scheme is needed so that reports can be generated simultaneously with data collection.

The number of collection intervals typically varies by type of data. For example, half-hour data are stored for 128 intervals (two and one-half days), whereas daily data are stored for seven days. The file system is able to store different amounts of data for each office and type of data. The total number of files necessary for EADAS to store in its database can be as high as 15,000 to 20,000.

The volume of registers collected varies not only by office and type of data, but also by collection interval. Thus, office A may normally send 2000 registers for data type 1, but at some point may decide to increase the transmission to 2500. Although this event does not occur very often, EADAS is able to accommodate the increase so that no data are lost and does so without increasing file overhead for any other office, data type, or interval. No backup of transient (register) data is necessary. The cost of a rare loss does not justify the cost of regular backups.

#### *4.4.2 The No. 1A EADAS File System*

The EADAS File System was implemented using raw (i.e., unbuffered) input/output (I/O) to meet the above requirements. It is a higher-level interface to the Logical File System (LFS), which is used in several other projects, as well as in EADAS. The overall function of the LFS is to subdivide an area of disk into contiguous files and perform all the necessary file management. The following is a list of the most important features of the LFS, as well as some information about how EADAS uses it:

1. Unlike the standard *UNIX** operating system, which provides a hierarchical file system using ASCII path names, LFS provides a "flat" file system. The entire LFS disk area is treated as a single *UNIX* system file that when opened allows access to all logical files. Each file corresponds to a unique number, called a logical file number (lfn),

* Trademark of Bell Laboratories.

that is the sole means of referencing the file. Up to 65,535 lfn's per file system are permitted.

2. The LFS performs all space management and low-level I/O, which is transparent to the user. Supported functions are creating, deleting, reading, writing, and switching logical files. LFS disk areas may be mounted and unmounted much like *UNIX* file systems.

3. One of the major functions of the LFS is to perform the mapping from logical file number to physical file location. The LFS requires no opening of individual files since they are "opened" by simply referencing them. Although efficient, access by file numbers is very cumbersome since they reveal little about file contents. Thus, the EADAS File System uses a specialized "open" function to translate retrieval keys (office, type of data, and time) into lfn's. Compared with the *UNIX* file system, the EADAS File System (using the LFS) allows many more open files and avoids costly directory searches in the opening process.

4. All logical files consist of contiguous disk storage. By computing offsets into contiguous files, the LFS avoids the overhead inherent in indirect blocks. Contiguous file storage also allows the LFS to read or write many blocks with one I/O request. This allows the physical block size to be determined by the access requirements for data in a *particular* file rather than be determined arbitrarily.

5. Since the LFS is implemented using the *UNIX* system's raw I/O, data may be transferred directly to/from disk from/to user memory.

In the present No. 1A EADAS configuration, the No. 1A EADAS File System uses three LFSs, with a maximum of 48,000 logical files. Typical EADAS database sizes range from 45M to 70M bytes, depending on the mix of switches from which data are being collected. Thus, the LFS results in efficient storage to permit the No. 1A EADAS to effectively store data from the electronic offices.

Once data are collected, the remainder of the system programs are executed to provide the reports and outputs necessary to administer the network.

## V. NORGEN REPORTS

The intent of the Network Operations Report Generator feature is to provide a time-shared programming environment for the development of reporting functions, and a set of reports designed to meet universal reporting needs including near-real-time exception reports. The reports (referred to as Network Operations Reports) are designed to meet the basic reporting needs for each switching entity type served by No. 1 and No. 1A EADAS. These programs provide users with a

package that may be modified or supplemented with locally developed features for the support of local reporting needs.

### *5.1 Basic objectives*

Most of the processes that use traffic data operate on data collected over a long period of time, to assure statistical accuracy. Examples of such long-range functions are equipment and facility provisioning, load balancing, and optimized utilization of present equipment and facilities. The reporting functions associated with the above are performed by the "downstream systems" of TNDS. The report results are generally made available to the user within one to several weeks after the end of the study period.

However, there are a number of processes that must be completed in short time ranges. Network performance levels must be monitored for service-affecting problems on a daily basis. Analysis of the traffic data provides valuable insights to the causes for many types of service-affecting problems. This problem-related information is most useful when provided on a near-real-time basis. Furthermore, validation of the data should be done in near-real time to minimize loss of data when something goes wrong. The EADAS Network Operations Reports are designed to meet the near-real-time information needs, while the long-range data collection needs, as indicated earlier, are handled by the TNDS downstream processing systems.

### *5.2 Report types*

The following report types are provided in the generic program:

1. Service Exception Reports are generated when high-priority service faults occur which may require prompt attention or corrective action.
2. Component Exception Reports provide an early warning indication of component conditions which, if allowed to persist, could affect service.
3. Trunk Exception Reports are printed when the percent overflow on final group(s) exceeds an assigned threshold. This report also provides supporting data that indicate where the pressure on the final group(s) is coming from.
4. Summary Reports provide a temporary record of busy-hour central office and trunking conditions. These reports are retained by network administrators until similar data become available from downstream processing.
5. Special Reports provide information which is useful to customers who use centrex, tie line, and multiline hunt-group facilities. The information in these reports is used by Marketing to advise customers

about the performance and needed changes in their communcations services.

6. Daily Reports are generated once a day during the early morning hours. The reports use data that was accumulated by EADAS or added up over the day by the switching system and transmitted daily to EADAS.

7. Network Switching Performance Measurement Plan reports are used to measure central office efficiency. They are organized for easy interpretation, and data are clearly labeled to simplify communications among network administrators, maintenance personnel, and traffic engineers.

8. Division of Revenue Reports provide data that help separate the cost of a switching entity into toll service, local service, and other rate base categories.

9. Measurement Apparatus Exception Reports are used to indicate failure of the data collection devices and associated data links. The report is also used to track corrective action, especially when support from maintenance organizations is required.

### *5.3 Report users*

The principal user of the Network Operations Reports is the Network Administration Center (NAC). The Switching Control Center (SCC) and the Network Data Collection Center (NDCC) also use the reports, but to a lesser degree.

The NAC is responsible for the quality of the local dial service provided by the local central offices and the connecting trunk groups. A major NAC objective is to detect problems as quickly as possible before they can affect service. It is the NAC's responsibility to take the necessary action or make the necessary recommendations to the appropriate organization for correction. These problems may be the result of underprovisioning of equipment or trunks, traffic imbalances, equipment outages, etc.

The NAC is also involved in network transition and cutover activities associated with growth. Real-time analysis of traffic data is a way to ensure that additions and rearrangements work properly.

The SCC has primary responsibility for the equipment maintenance-related aspects of local dial service. Maintenance data such as that contained in the Network Switching Performance Measurement Plan, which is generated by EADAS on a daily basis, are used as an aid in detecting and correcting equipment faults, and in setting maintenance priorities.

The NDCC is responsible for the administration of network and special studies data collection, supervising the operation and maintenance of EADAS, the data links, and data collection apparatus, such

as the EADAS Traffic Data Converter, Traffic Usage Recorder, and Pollable Data Terminal. The center is also responsible for coordinating the operation of the TNDS Central Office Equipment (TNDS/EQ) downstream systems. This center is responsible for reviewing measurement Apparatus Exception Reports (AER) and acting to correct problems which cause loss of data.

### *5.4 Users of special reports*

The near-real-time needs of various users continue to evolve. For example, special traffic studies on facilities associated with large business customers have increased substantially in the past few years and all indications are that they will increase even more in the near future. Organizations such as Business Services and Marketing are currently increasing the number of requests for special studies data. These requests are based on increased regulatory and competitive pressures as well as the increased availability of new features made possible by the widespread installation of stored program control systems and a higher level of sophistication of telephone customers. These factors have also combined to increase the frequency of these requests. The study results are often required in almost near-real time, making EADAS the logical system of TNDS to generate them. Studies not required in near-real time, or involving large amounts of data, such as those used for long-range planning, may be more efficiently produced downstream by other TNDS systems, such as TDAS.

The Network Operations Reports deliver the requested information and provide the necessary surveillance for all studies, both near-real time and long range.

### *5.5 Typical report use*

Dial tone speed tests results are part of the Network Operations Reports and are used in the real-time service analysis process. An example of a 1 *ESS* dial tone speed service exception report is shown in Fig. 6. The intent of this report is to alert the network administrator that an abnormal dial tone delay condition has occurred so that corrective action may be taken. This report is divided into four sections: dial tone speed results, customer digit receiver (CDR) data, remote switching system, and supporting data. The first section contains the measured percent dial tone speed results by customer digit receiver type, i.e., *Touch-Tone** dialing and Dial Pulse (DP) with the associated test and delay counts. This report is only generated when the percent *Touch-Tone* dialing or DP dial tone delay exceeds a user-

* Registered service mark of AT&T.

```
        * * * 1A ESS SWITCHING EQUIPMENT SERVICE EXCEPTION * * *

ENTITY:  XXXXXXXXXXX   DATE:   8/11/82   TIME:  12:30   INTERVAL:  30 MIN

*  DTS SERVICE EXCEPTION  *

                 % DTD              TSTS              DLYS
  * TT            0.3               363                 1
  * DP            0.0                87                 0

  CDR GROUPS          % OFL    OFL      PC    % OCC    # MB     NCI      HT
  * COM               0.0        0       0      0.0     0.0      16
  *  TT               0.0        0   13916     55.9     8.0     115     7.3
  *  DP               0.0        0   10646     47.1    11.0     114    11.1

  ROB Q PC  =          ROB OCC  =                ROB QTY  =     0

  RSSID  =     XXXXXXXXXXX
                 % DTD            TSTS              DLYS
  *  TT/DP         0.0             275                 0

               ORIG PC  =    224              LN-LN USAGE  =   467

                  % OFL     OFL       PC    % OCC    # MB      NCI       HT
    CHANNEL         0.0       0      428     24.0     0.0      108    109.1


  E-E CYCLES   =    8504    LN SCAN COMP  =    8504
  ORIG PC      =   24401    INC PC        =    3594
  A LK CCS     =   27899
  BLK DT Q PC  =       0
  POB %OCC     =    20.2    POB OFL       =
```

*REGISTERED SERVICE MARK OF AT&T

TT – *TOUCH-TONE*® SERVICE

Fig. 6—Example of 1 *ESS* switching system dial tone speed service exception report alerting network administrator of abnormal dial tone delay.

specified threshold. The remaining three portions of the reports provide supporting data for customer digit receivers by type.

When an excessive dial tone delay condition exists, the dial tone delay statistics are printed for both *Touch-Tone* dialing and DP. An asterisk is printed on the line(s) for which the threshold failure(s) have occurred. A threshold test of the CDR group(s) is then performed. If a common CDR group is provided, a threshold test is made for excessive overflow. If the test fails, the results are printed for the common group with an asterisk followed by two lines of information for the DP and *Touch-Tone* CDR groups. Otherwise, the common group results are printed without the asterisk. If a common CDR group is not provided, threshold tests on the DP and *Touch-Tone* dialing CDR groups are made and the results are printed with an asterisk, where required.

After the CDR results are printed, the next section will be provided only if the *ESS* switching equipment is an RSS host office; otherwise this section of the report is omitted.

The last section of the report is always provided and contains overall call processing load data.

Dial tone speed data are analyzed by the Network Administrator following reception of a dial tone speed service exception report. This report is generated when the percent dial tone delay threshold is exceeded for DP and/or *Touch-Tone* dialing. If a common CDR group is provided and it exceeds the percent overflow threshold, then the following analysis would be performed:*

1. The CDR percent occupancy is observed to determine if it exceeds the engineered capacity. If it is too high, then the CDR maintenance busy item (number of circuits made busy) is examined.
2. If the maintenance busy value is too high, then the network administrator requests restoral of the made busy CDRs.
3. If the maintenance busy value is not too high or if there would be a problem even after they are restored, then an analysis of the DP and *Touch-Tone* dialing CDRs is performed. If both the DP and *Touch-Tone* dialing CDR overflows are high and their holding times are unusually long, then permanent signal conditions are analyzed.
4. If holding times are normal and no extraordinary demand exists (such as a severe storm produces), then this condition would be referred to the traffic engineer if it persists.

If *Touch-Tone* dialing percent overflow is high and the DP percent overflow is not, then the DP *Touch-Tone* dialing CDRs should be rebalanced or *Touch-Tone* dialing CDRs added.

If DP and *Touch-Tone* dialing CDR percent overflow is not high, then a check of all related database, circuit quantities, and register assignments should be performed.

The Network Operations dial tone speed service exception report provides the network administrator with the necessary data to support this problem analysis.

In rare cases traffic peaking can be a factor in high CDR overflow. Examination of the 15-minute traffic count report which is generated by the 1 *ESS* switching equipment would provide the necessary 15-minute data required for this analysis to supplement the 30-minute data available to No. 1A EADAS. There are three other areas of investigation related to the dial tone speed analysis. They are Processor Overload, Network Blockage, and Translations. The first two areas would make use of a number of Network Operations Reports in the analysis procedure. The translation analysis would use the dial tone speed service exception report.

## VI. NORGEN IMPLEMENTATION

The development of the NORGEN feature was an ambitious under-

* Familiarity with Bell System traffic terms is assumed for this section.

taking. Data coming from 48 or more entities are interpreted and analyzed in real time. Formatted reports are prepared and distributed. The internal architectures of a large number of switches determine the report structure and algorithms in fine detail. The program can be distributed to all operating companies and run without local programmer support. Yet the system is tolerant of user programming modifications.

### *6.1 The challenge*

A major challenge was to implement NORGEN in No. 1 EADAS, with a total Random Access Memory (RAM) capacity of 256K bytes (only 64K bytes available to NORGEN), without compromising NORGEN's ultimate capability in the No. 1A EADAS, which has essentially unlimited RAM capacity (the current memory size is 1M byte).

The solution was to use two entirely different software architectures. Yet the feature appears very similar to the users of the two systems.

### *6.2 NORGEN in No. 1 EADAS*

In the No. 1 EADAS, programs are stored in interpretive language to conserve RAM. Program is brought into core in 512-byte blocks and executed block by block. A very linear coding style is used, with a minimum of subroutine calls to hold down the need for calling in extra blocks of code.

The interpretive language (object) is compiled from source code written in K language, which was designed for the project. K language is similar in style to languages with the same level of generality, but has fewer elaborate commands. Thus, K language is easier to learn than C, although C has more "power" in some applications (that is, certain features of C allow some functions to be performed in fewer lines of source code). A text editor and source-to-interpretive language compiler, both written in assembly language, are supplied with the generic.

In K language, all variables reside logically on disk; there is no distinction between local and global variables. The system keeps a cache of disk blocks in memory to improve the speed of operation. This structure was chosen because of the small amount of memory available, but has been found to ease coding problems because the programmer need not consider whether a variable is of one type or another.

The disk file access has three levels. A directory contains all named files and file offsets for particular data items. The directory is accessed using a backup function. When an item is needed, a fast access directory is checked first, to see if the file is already in RAM; then the backup function is used to access the disk directory; and then the file

itself is called from disk. Most requests are found in RAM; nearly all others are filled by two disk accesses (one for the directory and one for the actual data requested).

### *6.3 NORGEN in No. 1A EADAS*

In the No. 1A EADAS, programs are stored in assembly language, compiled from C language source code by the *UNIX* compiler. This permits more subroutine calls and a more conventional programming style. The *UNIX* text editor and C language compiler are provided with the generic.

The *UNIX* file system is used for program files and parameters, but the specialized two-level Logical File System as previously described is provided for accessing the traffic data. Otherwise, multiple file accesses would be required for the multilevel directory structure of *UNIX*.

### *6.4 Parameter administration*

In this article, the term "parameters" refers to all the user-controlled tables which the generic program reads to relate to local conditions. NORGEN parameters include scheduled times for reports to be run, exception thresholds, register definitions, etc. Any parameter administration system must have the functions add, delete, change, display, and backup.

In both the No. 1 EADAS and No. 1A EADAS, these functions are served by using text files, in controlled format, to hold all the data. The text editors, which are quite similar, are used for all of the above functions except backup; this is accomplished by dumping all the parameter files onto tape (approximately weekly) and reading them from tape in an emergency.

Programs are provided to convert each of the text files, by type, into binary object files for efficient access. When any text file has been changed (by the editor), the file is recompiled.

The text editor enables the user to work in a relatively English-like environment, minimizing training and making it easy to spot errors by browsing displays of the files.

### *6.5 Raw register access*

The No. 1 and No. 1A EADAS use very different register access mechanisms. Both use key words to provide mnemonic access to individual or summed groups of registers. However, the No. 1 EADAS systematically translates the packed register data from data collection into rigidly structured tables, with each key word assigned a fixed location (for each switch type). On the other hand, the No. 1A EADAS uses mapping algorithms and parameter tables to extract each key

word from the raw registers as it is called for from the program. This newer technique provides a great savings in disk space, allowing a fourfold increase in the number of intervals of data that can be stored, since most entities have only a minority of the total possible key words actually in use. The real time used by the two methods appears to be similar. The older technique is more efficient when key words are used repeatedly, but very few key words are used more than once or twice.

### *6.6 Scheduler*

Again, there are strong differences between the two schedulers. In the No. 1 EADAS, the scheduled reports are run one at a time, to minimize the swapping of files into and out of the severely limited RAM space. In the No. 1A EADAS, two modules are run at one time (level 2 multiprocessing). This achieves an efficiency gain due to interleaving processor and disk I/O functions.

Both executives are time-shared, allowing user access to demand reports, dump key-word values, etc. The No. 1 EADAS executive allows new users high priority. As recent run time increases for a given user, that user's priority (probability of being run at the next opportunity) decreases. This automatically allows users high priority for short, easy tasks like editing or key-word dumps while lowering the priority for compilations and running long reports.

The No. 1A EADAS uses the *UNIX* priority feature, giving manual users priority over scheduled work, independent of their tasks.

### *6.7 Reports*

The report modules are written in C language, using straightforward, linear style. This facilitates user programming as well as Bell Laboratories and Western Electric modifications of the design. Subroutines are used sparingly, so that calculation details are grouped in one place in the source program.

Special functions are defined for pervasive features. In particular, a threshold check function automatically seeks the threshold parameter and applies the appropriate check (less than, equal to, range). A special print function prints a blank in a field when appropriate. (For example, data pertaining to items which are unequipped in a given entity are carried in memory as negative numbers, which are suppressed by the print function.)

Standard page headers and section header routines are called to provide uniform structure of the reports. Automatic pagination is provided (through the special print function).

The layout of the reports emphasizes the grouping of related data for easy reading. White space is used freely to set off the groups.

Mnemonic labels are provided; abbreviations are chosen to be consistent with related Bell System documentation.

### *6.8 Report distribution*

After the ASCII image of a report is delivered to the spooling system, it is distributed to appropriate user terminals. Each report has a message class number. A parameter table lists the destination terminals for each entity and each class. In No. 1 EADAS, this table was centrally administered; in No. 1A EADAS, the user at each terminal designates the entities and message classes it is to receive. A particular report thus may be sent to many terminals.

## VII. TAPE WRITING

In the No. 1 EADAS, tape writing is done by an assembly-language-coded program that runs under the data collection executive. Data are written each half hour, under control of schedule parameters. The data are recorded in ASCII characters; each register is represented by a six-digit number.

In the No. 1A EADAS, tape data are prepared in two steps. The first step is to process and format the data; the Tape Data Formatter (TDF) module runs under the NORGEN executive. This provides opportunity for much more powerful processing of the data. In particular, the TDF combines two successive half-hour data messages into a single one-hour message. In most cases, registers from each half hour are added together; but certain registers, such as those containing averages, peaks, or trunk group numbers, are combined by specialized algorithms. Improved validity checking is done to avoid passing bad data downstream.

In the second step, the TDF writes the formatted data in a spooling area, where they are held for up to several hours. This spooling capability improves operational flexibility; for example, it is possible to hold data while a tape drive is being repaired.

A new tape format is provided in No. 1A EADAS. Data are usually written as 16-bit binary words, thus fitting a register into two bytes, where the No. 1 EADAS requires six. The effective reduction is approximately 2 to 1 since there is less compression in headers, and no compression of record gaps.

Industry standard tape labels in the new format facilitate data-link transmission of tapes and inventory control.

Two tape drives may be provided in the No. 1A EADAS; when one tape is filled, the other begins to write. This feature, together with the binary format, combination of half-hour data into one-hour data, and spooling, allows data collection to continue unattended in most installations over long holiday weekends.

## VIII. DATA FLOW

Following the course of data through the No. 1A EADAS shows how major program modules relate to one another. Let us follow the data of and *H* schedule* from a polled 1A *ESS* switching equipment office—the schedule for the interval from 10:30 a.m. to 11:00 a.m., in particular.

The 1A *ESS* switching equipment office will have collected the data and made it ready for transmission shortly after 11:00 a.m. on its clock. The No. 1A EADAS will wait several seconds after 11:00 a.m. on its clock to allow for system time differences.† Then it will send a poll asking for the first block of 256 registers of *H* schedule data. When the block has successfully been received, it will ask for the next block, and so on, until the proper number of registers (a channel parameter) has been validly received.

As the blocks of data are received, they are stored in the logical file system. A new entry has been made in the LFS directory, so that the data may be accessed by entity name, end time of data interval, and type of data.

When the schedule is complete in the LFS, a message is sent via a pipe to the NORGEN Data Analyzer (NDA) executive. The executive checks schedule parameters to see which program modules are to run on this particular *H* schedule.

Suppose that the busy hour for this entity is 10:00 a.m. to 11:00 a.m. Then the Load-Service Summary module will be scheduled among others. The executive will make an entry in its job queue, listing entity, module, and end time of data. When this entry works its way to the head of the first-in-first-out queue and a new process can be started, the executive will run it. This particular module (the Load-Service Summary) works on a full hour's worth of data. Therefore, it retrieves two successive intervals of data from the LFS and adds their register values together to form an equivalent one-hour schedule.

Next, the module accesses data to prepare its report. For example, it reads the peg count of incoming calls by use of key word INCPC (incoming calls peg count) as a variable in a program. The compiler has recognized this key word as a register reference and has compiled a function call to the key-word access routine. The key-word access routine references the Data Collection Device (DCD) parameter table to find the register number and return the corresponding value to the Load-Service Summary module.

The module prepares the ASCII text of the report, calling special

* An "*H* schedule" is a set of registers that represent traffic through the common parts of the switch (as opposed to the trunk groups).

† The No. 1A EADAS periodically requests time-of-day from each 1A *ESS* switching equipment office; if the time difference is large, a message is printed.

subroutines to form the header, subsection header, and end of report headers. Pagination is provided automatically; to make this possible, a PRINTN function is defined, which performs special NORGEN report functions before calling the PRINT functions.

The ASCII text is stored in a spooling area, but the spooler directory entry is not completed until all modules for that entity and interval are completed. This allows the NDA executive to mark the sequence of reports in a fixed order, so that they will always appear in the sequence expected by the user. This facilitates search and filing of specific reports.

The report spooling program distributes each report in order. Each report type is assigned a message class. The terminal tables are searched to determine which terminals have designated the entity name and message class for the Load-Service Summary Report. The report is sent to these terminals through the multiplexor driver program.

## IX. CAPACITY

In the No. 1 EADAS, NORGEN is usually the limiting capacity consideration. Consequently, the capacity characteristics of the reports have been analyzed carefully.

The capacity criterion of the system was to complete the report generation load of an "equivalent half hour" in one half hour, to prevent excessive delays in the availability of reports. The concept of an "equivalent half hour" was introduced because the report load is irregular. For example, for a polled interface, some reports run continuously (every half hour), some run every hour, and some only after busy hours. After discussions with users, it was decided that a backlog can build up during a busy hour, as long as the system recovers within one and one-half hours.

The load on the system for an equivalent half hour was taken to be:

continuous reports: fully weighted
hourly reports: one-half weighted
busy hour reports: one-third weighted.

The capacity analysis was thus reduced to estimating the run time of each report module. The weighted run times are then added to determine whether the system can keep up with its load.

Report run times are composed of two major components; processor time and disk access time. Each of these resources is subject to priority seizure by data collection activities, which extends the report run times significantly. Extensive modeling and measurement of the No. 1 EADAS and No. 1A EADAS Systems have been performed to determine the quantitative effects of data collection and relate those

to the mix of sizes and types of remote data sources (switching entities).

### *9.1 Modeling*

The application of analytic modeling to combination real time and time shared systems has been challenging. It has been successful because a limited accuracy goal of plus or minus five percent was set. As a result, effects which sum to only a few percent were neglected, or assigned a fixed allocation which was neither best nor worst case, but somewhere in between.

Attention was thus directed at the first-order effects. Measurements were carried out in the laboratory, using a field configuration system under approximately 50-percent load. The load was applied by a separate minicomputer acting as a load test generator. Traffic data used was based on data collected from "typical" offices in the field. These model offices were chosen to be reasonably complex in terms of the number and types of registers, but worst-case offices were avoided since they do not dominate a fully loaded system.

By taking measurements on a system loaded in the 50-percent range, the effects of approximations in the model are minimized.

#### *9.1.1 No. 1 EADAS capacity model*

In the No. 1 EADAS, the scheduled report modules are run sequentially, rather than multiprocessed, because of the limited availability of RAM. In this system, data collection takes a large amount of processor time at interrupt level (up to 60 percent in a heavily loaded system), especially for the ICUR feature. The effect of this data collection load on a given report module depends on how much processor time it needs; the data collection load has no first-order effect on disk access time. The effect, then, is to extend the processor time component of the report run time:

$$T_p = \frac{T_{po}}{1 - F},$$

where:

$T_p$ = processor component of the report run time in the presence of data collection load,
$T_{po}$ = processor component of the report run time in the absence of data collection load,
$F$ = fraction of processor used by data collection.

In the above model, it is important to note that the data collection work is not time shared with the report modules, but is performed at priority interrupt.

A second-order effect, which is typically large enough to be significant (of the order of ten percent), is caused by base-level data collection load. This load has no effect on processor time because it always takes place during times when report modules are waiting for disk accesses. However, when a disk access is complete, base-level work is usually in progress and neither a further disk access nor a report processor segment can be started until it completes. Effectively then, a fraction (taken to be 1/2) of the base-level data collection time (stretched out by interrupt data collection time) is added to disk latency.

$$L = L_o + \frac{1}{2}\frac{a}{1-F},$$

where:

$L$ = disk latency in the presence of data collection load,
$L_o$ = disk latency in the absence of data collection load,
$a$ = average base-level time for data collection,
$F$ = fraction of processor used by data collection at interrupt.

The effect on the disk portion of the report run time is given by:

$$T_d = T_{do}\frac{L}{L_o},$$

where:

$T_d$ = disk component of the report run time in the presence of data collection load,
$T_{do}$ = the disk time measured in the absence of data collection load.

The rule for capacity estimation is to add the processor and disk component times for the relevant report modules (for the given mix of office type and report schedules, according to local operation practices) weighted by the appropriate factor according to the report frequency, to apply the model equations to account for the estimated data collection loads, and to compare the resultant time to the 1800 seconds of each half hour. Ten percent of the report capacity is allocated to allow for manually generated load running in time share with the reports.

While the above process may seem complex, it has been reduced to a straightforward application of tables and worksheets and has been well accepted by the operating companies who must estimate their capacity.

#### *9.1.2 No. 1A EADAS capacity*

The No. 1A EADAS is quite similar in its modeling except that only one effect of data collection load is significant—priority access to the disk. Since scheduled reports are run in time share, two at a time, and

since the reports need about 60-percent of disk I/O time and only 40-percent of processor time, the reports are strongly disk limited. Measurements show that about 85 percent of the disk is utilized, which is quite constant even when 50 percent of the disk is taken by data collection. A change in data collection architecture has resulted in heavy disk loads for collection of ASCII data in the No. 1A EADAS.

The system is found to be so disk limited, that the distinction between processor and disk components can be neglected, and the *effective* run time (which is measured total run time in multiprocessing, divided by two, since two reports are run at the same time) can be considered to be all disk time.

Then the effect of data collection disk load is given by:

$$T = \frac{T_0}{1 - F},$$

where:

$T$ = effective report run time in the presence of data collection load,
$T_0$ = effective report run time in the absence of data collection load,
$F$ = fraction of the disk used by data collection load.

Estimation of capacity is then carried out in essentially the same manner as for the No. 1 EADAS.

## X. SUMMARY

EADAS has met its original operational objectives by providing for the collection of network data for engineering, network managment, and real-time data analysis. The result has been an overall improvement in the monitoring of network operations, thus meeting the increasing need for more efficient network utilization to optimize both capital investment and customer service. Eliminating the manual effort formerly associated with the collection and processing of network data has resulted in substantial economic savings. EADAS will continue to be enhanced to provide interfaces with new switching machines and operations systems.

## XI. ACKNOWLEDGMENTS

The No. 1 and No. 1A EADAS represent the work of many people over a full decade. The features of the system are reported here without specific accreditation, which would be difficult at this time. However, the authors wish to acknowledge the special help of J. R. McSkimin and R. L. Hardin in the preparation of this paper.

## REFERENCES

1. N. D. Fulton, J. J. Galiardi, E. J. Pasternak, S. A. Schulman, and H. E. Voigt, "Total Network Data System: Equipment Systems" B.S.T.J., this issue.

## AUTHORS

**J. A. Grandle, Jr.,** B.E.E., 1967, University of Virginia; M.S.E.E., 1968, Purdue University; Bell Laboratories, 1967—. Mr. Grandle has helped design Traffic Measurement Circuits; hardware and software for No. 1 Engineering and Administrative Data Acquisition System (EADAS), No. 1A EADAS, EADAS Individual Circuit Recording, and EADAS Network Operations Report Generators; CDA; No. 1 Automatic Message Accounting Recording Center central hardware; PDT-2A hardware and firmware; and processor terminals, including the PLURAL multiprocessor terminal. He also has three patents and has published several articles. ("A New Test Line for Coin Phones," and "A Versatile Data Gathering Tool") published in the *Bell Laboratories Record.*

**Donald J. Gagne,** B.E.E., 1943, Pratt Institute; Bell Laboratories, 1943—. Mr. Gagne has participated in the development of order wire and alarm facilities, mobile and overseas control terminals, and No. 1 and No. 5 Crossbar circuits. He has also conducted systems engineering studies for new recorded announcement systems, new electromechanical and electronic PBXs, *PICTUREPHONE*® meeting service and Automatic Number Identification for PBXs, and new Service Observing and Traffic Measurement facilities. He supervised early Systems Engineering for EADAS and at present is supervising a group responsible for EADAS Systems Engineering.

**George H. Wedemeyer,** Certificate in Aircraft Design and Technology, 1959, Academy of Aeronautics; Bell Laboratories, 1962–1981; AT&T, 1981—. At Bell Laboratories Mr. Wedemeyer participated in the hardware and software design of data collection devices. In 1981, he became a Member of Technical Staff while working for the Network Administration Systems Department in Holmdel, New Jersey. His responsibilities included formulating requirements for traffic-data collection and reporting systems, economic analysis, and system evaluations. Currently, Mr. Wedemeyer is a staff manager at AT&T responsible for the project management of EADAS.

**Charles J. Byrne,** B.S.E.E., 1957, Renesselear Polytechnic Institute; M.S.E.E., 1958, California Institute of Technology; Bell Laboratories, 1958–1962, 1968.— Before rejoining Bell Laboratories, Mr. Byrne was employed at Bellcomm, Incorporated, where he worked on systems engineering for the Apollo Project. At Bell Laboratories he has worked in communications systems research and operations systems development. He is currently Supervisor of the International Sales Planning Group of the Switching Operations Systems Laboratory. Member, Sigma Xi, Tau Beta Pi, Eta Kappa Nu.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Network Management

By G. C. EBNER* and D. G. HAENSCHKE*

(Manuscript received June 9, 1982)

This paper describes steps that have been taken to respond to the need for better network management in the evolving telecommunications network. In near-real time, network management functions recognize the onset of an overload and respond with control actions that change normal call routing through expansive or restrictive traffic controls. Emphasis is being placed on improved automatic controls that are built into the network and its switching systems. Advances in manual controls and real-time network performance monitoring have been accomplished through the introduction of computer-based systems that provide network managers with preprocessed network performance data and with the ability to intervene in problems that require human judgment.

## I. INTRODUCTION

Network management consists of real-time network performance monitoring and control of the telephone network. It is a technique designed to optimize the call-carrying capacity of the network when the network is under stress due to traffic overload or failures. In recent years significant steps have been taken to respond to the need for better network management for the evolving Message Telecommunications Service (MTS) network. Advances have been made in manual

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

network management controls and real-time network performance monitoring capabilities. These advances were accomplished primarily by the introduction of computer-based systems that support the operation of centralized network management centers. Network management centers employ automatic and manual capabilities that recognize the onset of overloads and respond with traffic control actions within a time span ranging from seconds for automatic controls to minutes for manual actions. Each center is supported by an operations system that collects the network data, monitors the status of the switching and signaling systems, and permits the implementation of appropriate control measures during overload and failure conditions. Advances have also been made in improved automatic network management controls that are built into Stored Program Controlled (SPC) switching and signaling systems. The current approach to network management is to provide an economical balance between automatic and manual network management capabilities, with greater emphasis on improvements in automatic controls.

This paper presents the status of network management and its application to the evolving MTS network. The network performance under overload and the motivations for network management are described in Section II. Section III discusses the centralized network management operations systems and the automatic network management SPC switching system controls. Section IV gives a description of the operations system Engineering and Administrative Data Acquisition System/Network Management (EADAS/NM). Section V discusses future network management needs for the evolving MTS network.

## II. MOTIVATION FOR NETWORK MANAGEMENT

### *2.1 Network performance under overload*

The North American Message Telecommunications Service (MTS) network functions as a single, integrated entity to which customers have shared access for voice telephone calls, data calls, and other uses such as facsimile transmission. The network is currently being enhanced by the rapid introduction of Stored Program Controlled (SPC) switching systems interconnected via the Common Channel Interoffice Signaling (CCIS) system.[1] In effect, the CCIS network is a separate high-speed, packet-switched data network that allows SPC switching systems in the message circuit-switched network to communicate with each other. The interconnection of SPC switching systems by a reliable, high-speed, high-capacity signaling system permits more efficient use and control of the network.

Under overload, modern telecommunications networks that employ

common control and automatic alternate routing can be forced into a congested, inefficient state, as marked by a significant decline in the number of calls that can be completed. The loss of network capacity under overload was described by P. J. Burke.[2] Figure 1 presents a key study result obtained through simulation of a 16-node hierarchical network. This figure indicates that there is a decline in the amount of traffic that can be completed when the offered load substantially exceeds the design limits of the network (about 1600 erlangs). There are two basic reasons for the loss of capacity of a network in congestion: excessive alternate routing and regenerative switching delays that are compounded by customer reattempts. Excessive alternate routing causes an increase in the average number of links a call will use before it is completed, thereby increasing call blocking and reducing the efficiency of network utilization. This increase in blocking, with increased alternate routing for a network in congestion, can be mitigated by trunk reservation schemes and by constraining the number of links in alternate routes.[3]

Switching delays are the dominant cause for the loss of the call-carrying capacity of the network under overloads. If left uncontrolled, switching delays feed on themselves and can quickly spread throughout the network. They cause time-out conditions during call setup and occur when switching systems become severely overloaded. Switching congestion time-outs result in short-holding-time attempts on trunk groups, replacing normal-holding-time messages. Today, it is recognized that the telecommunications network's response to overload and the magnitude of the decline in call-carrying capacity of an overloaded

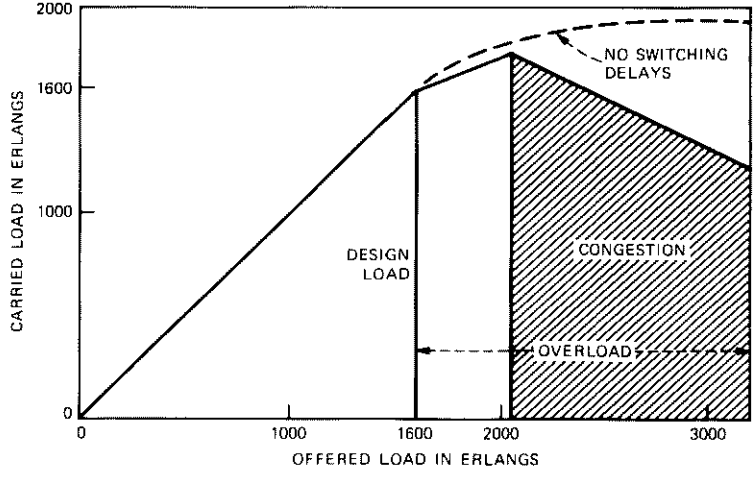



Fig. 1—Network performance under overload.

network depend on network architecture, call setup and routing procedures, signaling techniques, and switching system structure.[2-7]

### *2.2 Network management benefits*

The primary motivations for network management in the Bell System are to maintain the integrity of the network and to ensure optimal performance of the network during overloads and failures. Telecommunications networks play a vital role during emergencies and natural disasters. The benefits of investments in modern network management capabilities are realized during such emergencies.[4] Network management, coupled with redundancy in critical switching components, assures network integrity in the Bell System at less cost than alternatives that do not involve network management. Maximizing call completions during overloads by network management techniques usually produces more revenue than network management capabilities cost.

## III. NETWORK MANAGEMENT SYSTEMS

### *3.1 Automatic control systems*

Network managers at the Network Management Centers (NMCs) discussed in Section 3.2 plan the employment of automatic network management controls in the MTS network. The NMC is responsible for monitoring the performance of the automatic controls through data furnished to the network management support system EADAS/NM by the switching systems. Occasionally, the NMC must fine tune the control response actions of the automatic controls that are built into SPC switching systems.

#### *3.1.1 Automatic protective controls*

Based on results of network overload characteristic studies, automatic network management controls, such as Dynamic Overload Control (DOC) and Directional Trunk Reservation (DRE), were introduced in the mid-1960s and proved to be effective during peak-day overloads such as Mother's Day and Christmas. However, these controls are not very effective during congestion caused by focused overloads, which are characterized by a surge of traffic originating in many parts of the network to a single destination, because they cannot selectively discriminate and control traffic based on destination codes. Focused overloads, stimulated by mass-media advertising, preplanned call-ins, or natural disasters, are occurring with increasing frequency in telecommunications networks.

As the network evolved to *ESS** switching system and CCIS,

* Trademark of Western Electric.

network management control capabilities also evolved. A major advancement was the introduction of improved automatic controls with the introduction of the 4 *ESS* switching system in 1976.[8] The automatic controls that were introduced and are now in operation are summarized in Fig. 2. They include two automatic protective controls, Selective Dynamic Overload Control (SDOC) and Selective Trunk Reservation (STR), as well as a new automatic expansive control called Automatic-Out-of-Chain routing, discussed in Section 3.1.2.

SDOC responds to switching congestion by dynamically controlling the amount and type of traffic offered to an overloaded or failed switching system. SDOC response actions are taken based on control request messages from the overloaded switching system, typically transmitted via the CCIS data network. STR, on the other hand, does not require the transmission of a control message; it responds to trunk congestion in the outgoing trunking field and is triggered on a particular trunk group when less than a certain number or circuits are idle in that group. The novel feature that makes these two protective controls selective is that they control traffic to hard-to-reach (HTR) points more severely than other traffic. An HTR code is a three-digit or six-digit destination code to which calls have a low probability of completing.

Based on real-time analysis of three-digit and six-digit destination code completion statistics, HTR traffic is automatically detected by the 4 *ESS* switching system. Every five minutes the number of Ineffective Machine Attempts (IMA), defined as attempts which fail within the 4 *ESS* switching system, and the number of Ineffective Network Attempts (INA), defined as the number of calls failing in the network between the node of interest and the call's final destination, are determined by each 4 *ESS* switching system in the network. The INA count is based on the number of calls that abandon after outpulsing without obtaining answer supervision from the called customer line.

| OVERLOAD OR FAILURE IN | CONTROL | RESPONSE |
|---|---|---|
| SWITCHING SYSTEM | SELECTIVE DYNAMIC OVERLOAD CONTROL | PROTECTIVE |
| TRUNK GROUP | SELECTIVE TRUNK RESERVATION | PROTECTIVE |
| HIERARCHICAL ROUTES | AUTOMATIC OUT-OF-CHAIN ROUTING | EXPANSIVE |

Fig. 2—4 *ESS* automatic network management controls.

The current 4 *ESS* switching system code detection and control system[8] uses the 4 *ESS* switching system for identification of HTR traffic destination codes and, in the near future, will use the CCIS network to distribute this information to other nodes in the network for control purposes.[3,9] Thresholds established at switching systems by the NMC identify HTR traffic destination codes. Traffic for such codes will be selectively controlled at the switching systems by SDOC, STR, and during congestion in the CCIS network.

Concurrent with the introduction of 4 *ESS* switching system was the initial deployment of the CCIS network. Failures are rare because of the high degree of redundancy built into the CCIS network. Nevertheless, new protective automatic controls had to be provided that sense overloads and failures in the CCIS network and cause SPC switching systems to respond with control actions that can affect the flow of traffic in the telecommunications network. For instance, a focused overload can overload the terminal buffers of CCIS data links. To avoid the loss of CCIS messages and corresponding calls for the circuit-switched telecommunications network, SPC switching systems apply automatic protective controls in response to CCIS terminal buffer overloads and CCIS processor congestion. The controls temporarily restrict the amount of new messages that are offered to trunk groups that signal via the overloaded CCIS link terminals. CCIS link overloads are usually caused by unsuccessful reattempts to HTR points in the network and can, therefore, be controlled most effectively by code-selective controls. Recent studies utilizing single server queueing models and mean value analysis indicate potential benefit in increased call completion (throughput) due to selective control on CCIS messages.

Figure 3 shows the predicted effectiveness of these selective controls for a simulated focused overload in a 24-node toll network. The nonselective automatic controls are contrasted with SDOC and STR, assuming full deployment in all toll switching systems. Performance is measured in Fig. 3 by the average number of completed calls in progress versus time. While the improvements due to the selective nature of these controls are greatest during focused overloads, simulation results and experience indicate improvement also for peak-day overloads. Other automatic processor controls are incorporated in the network, such as line load control which, when enabled, denies customer access to the switching system in congestion.

#### *3.1.2 Automatic expansive controls*

In contrast to protective controls, which restrict access to overloaded network resources, expansive controls take advantage of idle capacity on out-of-chain routes, i.e., on routes that are not within the design of

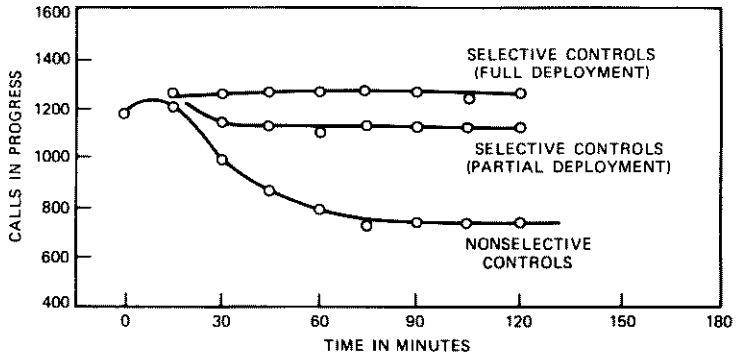



Fig. 3—Automatic control of focused overload.

the hierarchical routing structure. Automatic Out-of-Chain (AOOC) routing is a control that expands the route selection beyond the hierarchical routing constraints by offering calls overflowing the normal final-choice in-chain routes to up to seven out-of-chain routes. AOOC routing is the first step towards improved network utilization by taking advantage of network capacity that is often available through traffic noncoincidence. AOOC routing is made possible by the capabilities of the CCIS network, which permits AOOC routed calls to be classmarked, counted, and restricted to out-of-chain routes with idle capacity.

### *3.1.3 Controls for 800 Service with CCIS*

The evolving SPC network with its inherent CCIS interconnectivity is also the key to new, innovative, communication services. These new services are made possible by computer-based Network Control Points (NCPs) to which switching systems are given direct switching access via the CCIS network. NCPs are specialized network nodes that provide the logic and routing information to handle enriched network services.

An example of new network management opportunities made possible by the SPC-CCIS network is the handling and control of 800 Service, formerly called Inward Wide Area Telephone Service (INWATS). Today, more than ten percent of the total toll calls are 800 Service calls; 800 Service mass calling is a frequent cause of overloads in the telecommunications network. In the SPC-enriched network, such calls first access a centralized NCP via the CCIS network. The NCP translates the 800 number to a new destination number and returns the number to the switching system, and the call is routed as a normal telephone call.

Network management controls are provided that protect the NCP,

the CCIS network, and the circuit-switched telecommunications network from overloads. Code-selective controls are initiated automatically (or manually from the Network Operations Center described in Refs. 10 and 11) at the NCP. Call attempt thresholds are established at the NCP to detect excessive calling volumes. The Action Point (ACP) in the network (the switching system at which access to the NCP via the CCIS network is initiated) will recognize the control signals and limit attempts and NCP inquiries for high call volume destinations. The NMC will be informed via EADAS/NM when ACP controls are initiated.

### *3.2 Hierarchy of network management operations systems*

Network managers in NMCs intervene in problems for which automatic system solutions would be excessively expensive and in problems requiring human judgment. In the past, NMCs were associated with major toll switching systems on a one-to-one basis. Each NMC displayed information that pertains to only one switching system and was able to affect call completions through manual controls in only that switching system. Later, real-time network performance monitoring capabilities were expanded by telemetering data to the NMC from a few distant lower-ranking switching systems. The main drawbacks of these older NMCs were decentralization and inadequate real-time network performance monitoring capabilities.

To overcome these drawbacks, the Bell System has developed a minicomputer-based system called EADAS/NM, which supports the operation of a centralized NMC.[9,12,13] Figure 4 shows this system collecting data in near-real time from toll and local switching systems in a large geographical area, such as an entire metropolitan area or a state. This allows such an area, called a "cluster," to be managed as a whole, and breaks away from the older concept of managing individual switching systems. Local and small toll switching systems are connected to the EADAS/NM system through an intermediate traffic data collection system. This intermediate data collection system consists of the Bell System Engineering Administrative Data Acquisition System (No. 1A EADAS) or a data acquisition system supplied by the General Trade. Large toll switching systems and CCIS Signaling Transfer Points (STPs) which have their own traffic data collection systems, interface with EADAS/NM via direct data links. As explained in more detail in Section 4.1.1, the EADAS/NM processor performs exception calculations on five-minute traffic measurements and displays the results in the NMC on a wall panel, printers, and Cathode Ray Tube (CRT) display "pages."

Figure 5 shows manual network management centers organized in

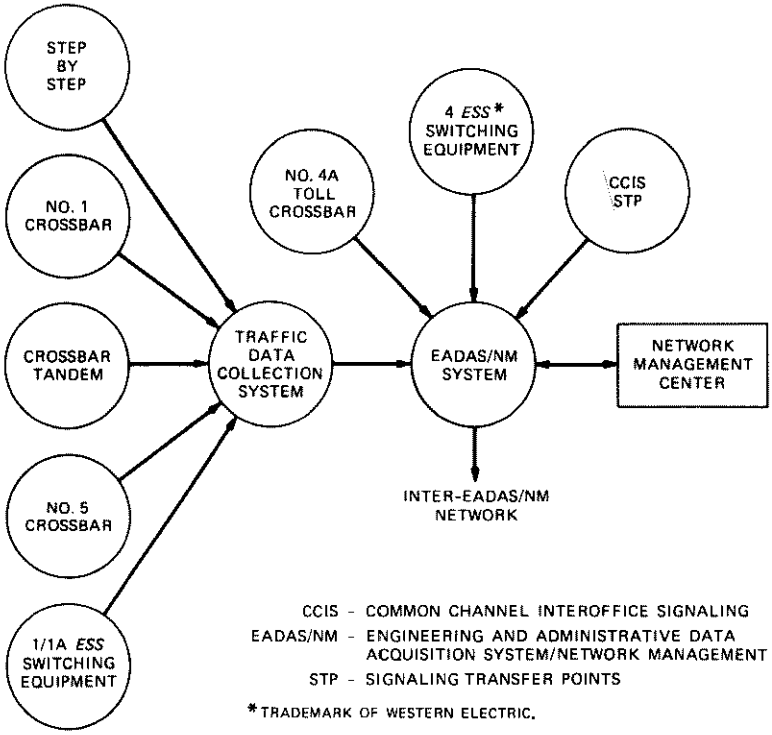
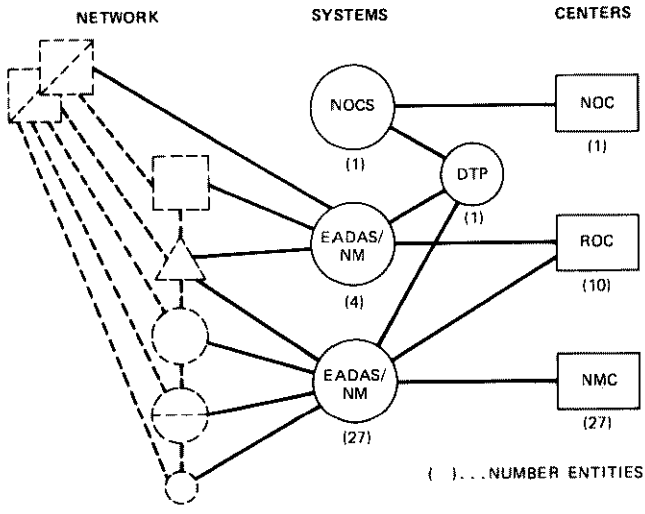



Fig. 4—EADAS/NM interfaces.

a three-level hierarchy. The first or basic level is occupied by the EADAS/NM-supported Network Management Center. Twenty-seven such centers are now in service, spanning the entire United States.[11] The switching systems and CCIS STPs of the telecommunications network are shown as dotted lines. Each NMC is jointly staffed with Bell Operating Company, Long Lines, and in some cases, Independent Company personnel.

The Regional Operations Center (ROC) occupies the next level in this hierarchy. Each ROC is supported by an EADAS/NM computer system and provides a higher level of network performance monitoring and control than the basic Network Management Center. Whereas the basic NMC has the direct responsibility for the control and real-time performance monitoring of the toll and local networks in its cluster, the ROC has the responsibility for the coordination of activities between the two or three NMCs in its region. There are ten switching regions in the United States and therefore ten Regional Operations Centers. The ROCs are also focal points for real-time




DTP - DATA TRANSFER POINT
EADAS/NM - ENGINEERING AND ADMINISTRATIVE DATA ACQUISITION SYSTEM/NETWORK MANAGEMENT
NMC - NETWORK MANAGEMENT CENTER
NOC - NETWORK OPERATIONS CENTER
ROC - REGIONAL OPERATIONS CENTER

Fig. 5—Network management structure.

performance monitoring of portions of the CCIS network. As shown in Fig. 5, four ROCs are supported by dedicated EADAS/NM computer systems. Each of the remaining six ROCs is supported by an EADAS/NM system which also supports an NMC. There are 31 EADAS/NM network management operations support systems now in use.

A single center called the Network Operations Center (NOC)[10,11] makes up the top level of this hierarchy. As described in more detail in Ref. 11, this center is supported by its own computer system and it is responsible for coordinating interregional network management problems between the 10 ROCs, the 27 NMCs, and 2 regional network management centers in Canada. It also has the main responsibility for international network management and real-time performance monitoring of the entire CCIS network. A large amount of data are transmitted between the EADAS/NMs and the NOC system via a data network. The hub of this network is a data switch called the Data Transfer Point (DTP), which furnishes the NOC with a constant flow of high-speed performance monitoring data for all key toll switching systems, CCIS STPs, and international switching systems in the United States.

### *3.3 Centralized network monitoring and manual control*

Even a highly sophisticated automatic system cannot respond optimally to all network disturbances. Thus, manual controls are required to provide additional adjustments to traffic flows in the network.

Figure 6 shows an example of an EADAS/NM-supported NMC in Boston, with the wall display panel in the background and interactive CRT systems in the foreground. The display system, which is described further in Section 4.1.2, is organized so that the activation of an indicator on the wall display panel directs the manager to CRT display pages for problem investigation and control activation. From one EADAS/NM-supported NMC, network managers can monitor and control several hundred switching systems.

Manual controls are activated from an NMC in many switching systems simultaneously through interactive CRT control pages. Control commands are communicated back to the switching systems by EADAS/NM over the same interface links over which data are collected from these systems. As shown in Fig. 7, manual network management controls are either protective or expansive in nature. Protective trunk group controls include trunk group controls which deny certain traffic access to a trunk group (cancel and skip controls) as well as trunk group directionalization. Protective controls are typically employed to limit traffic and thereby prevent the spread of traffic congestion. Expansive controls include controls that reroute traffic away from overloaded or failed facilities to facilities with idle capacity. Some computer-based tools that permit network managers to identify and solve traffic problems between regional switching systems are available at the NOC. These problems are solved through reroutes of which hundreds are implemented on peak days. Code block controls, which stop all or most of the traffic destined for overloaded points in the network, are employed mostly during focused overload congestion situations.

Because of the high number of reattempts, code block controls are not very useful in a mass-calling situation. With the deployment of CCIS, call-setup times are significantly reduced. Thus, the number of ineffective attempts per trunk can exceed 10,000 calls per hour, thereby defeating any mechanism to "choke" traffic flow based on the provision of a few, temporary, segregated trunk groups for traffic destined to a particular code. As shown in Fig. 8, a code block control based on a percentage restriction will still result in a high traffic volume under extreme conditions. Thus, a new control, "call gap," has been introduced. It consists of an adjustable timer that blocks all calls to a specified destination code for the set interval of time. The next call that arrives after the expiration of the time interval is allowed, after which the time gap begins again. This rate is strictly limited and

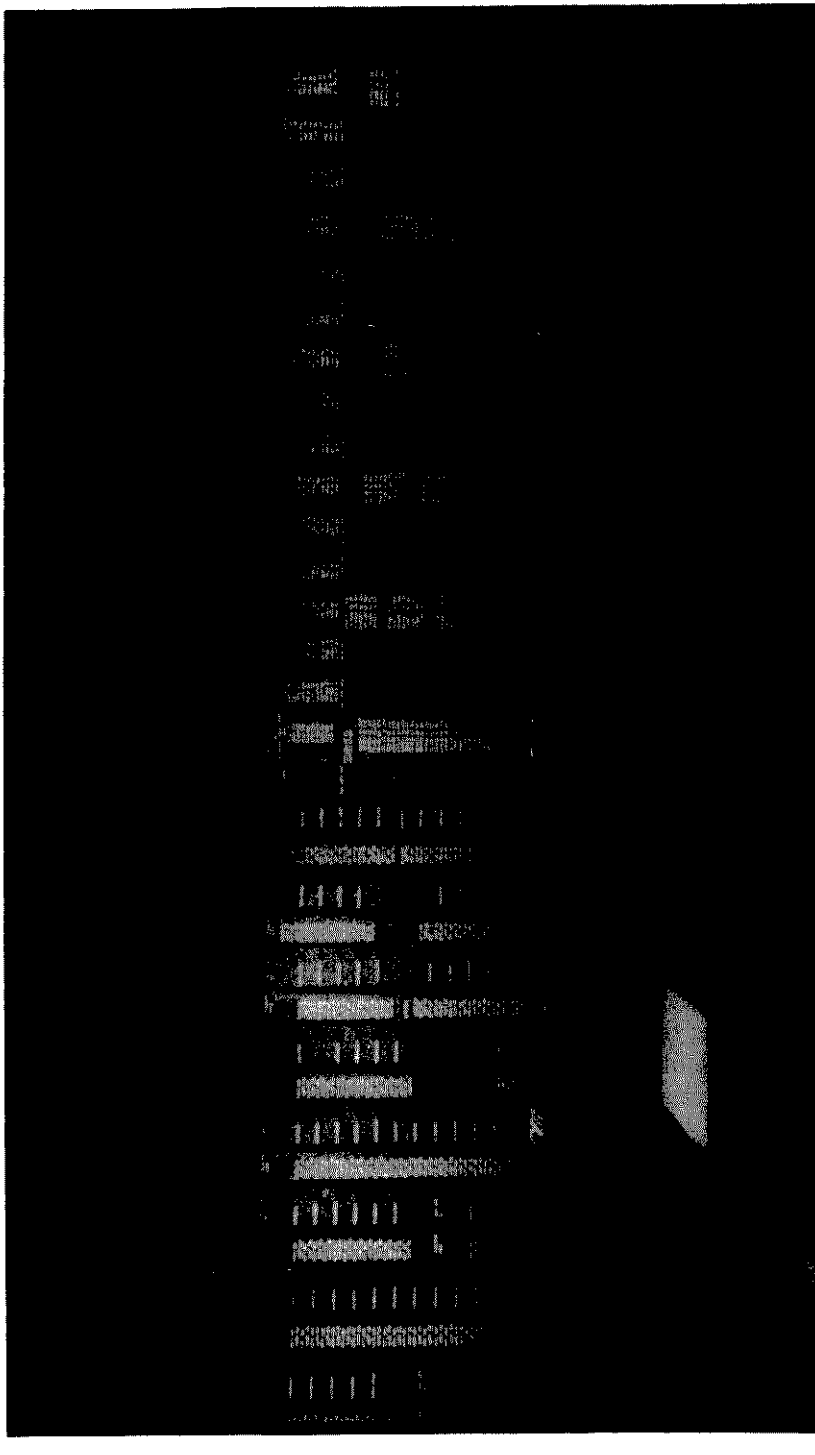


Fig. 6—Network management center.

| PROBLEM | CONTROL | RESPONSE |
|---|---|---|
| NETWORK OVERLOADS OR FAILURES | CANCEL-TO/FROM<br>SKIP<br>TRUNK DIRECTIONALIZATION | PROTECTIVE |
| | REROUTE | EXPANSIVE |
| FOCUS OVERLOADS OR MASS CALLING | CODE BLOCK<br>CALL GAP | PROTECTIVE |

Fig. 7—Manual network management controls.

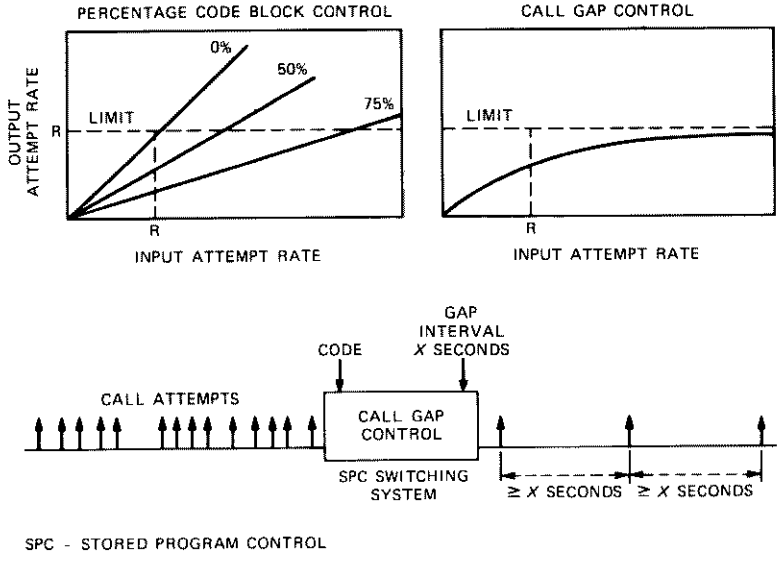



Fig. 8—Comparison of code controls.

approaches the inverse of the gap interval. This control will be useful in situations that require a set rate of traffic flow, particularly in mass calling and focused overloads.

## IV. EADAS/NM—THE COMPUTER SYSTEM

### *4.1 Basic EADAS/NM functions*

EADAS/NM is the minicomputer-based operations support system for NMCs and ROCs.[14] Its major functions are: (1) collection of network traffic and performance occurrences on an event basis every 30 seconds and associated traffic counts every five minutes; (2) analysis of traffic data to ascertain exception conditions and output of results to a wall display, printers, and interactive CRT terminals; (3) imple-

mentation of network management controls by transmission of control messages to appropriate switching systems; (4) record base maintenance by auditing selected information from SPC switching systems and by manual input of record items; and (5) transmission of selected information to the NOCS for national network performance monitoring.

#### *4.1.1 Data collection and exception calculations*

Data collection is done on a five-minute basis for traffic counts and on a 30-second basis for discrete (event) information. At the start of a five-minute interval (or a 30-second interval for discretes), EADAS/NM dispatches poll messages over direct data links to 4 *ESS* switching systems and PBCs (Peripheral Bus Computers), and through EADASs or equivalent General Trade systems to various local and small toll switching systems. The PBCs are administrative adjuncts to No. 4A Toll Crossbar Electronic Translator Switching Systems (No. 4A ETSs) and to STPs. Upon the completion of data acquisition for a given switching system, exception calculations are performed by comparing traffic data to preset thresholds. Once all polling and exception calculations are complete, the wallboard is updated to alert network managers of any changes in network conditions that may require further attention. Critical discrete changes update the wallboard on a 30-second basis. Exceptions of interest to the NOC (usually those involving switching systems and trunk groups in the upper three levels of the MTS network hierarchy) are routinely sent over a dedicated data link through the DTP to the NOCS.

The operation of an ROC EADAS/NM system is similar to the operation of an NMC EADAS/NM system except that the ROC EADAS/NM system obtains data both by direct polling and by passive monitoring. Typically, an ROC EADAS/NM system directly polls the regional switching system and the PBCs associated with STPs. It also receives data from passive connections to selected 4 *ESS* switching systems and PBCs. Figure 9 illustrates this arrangement.

#### *4.1.2 Data display*

The data display systems for EADAS/NM include the wallboard and exception printer as alerting devices, a monitor printer, and a repertoire of over sixty CRT display pages. The exception printer prints out a record of exceptions as they occur. The monitor printer allows the network manager to designate particular trunk groups (often those which are being used in a network management reroute control) for continuous observation.

The EADAS/NM CRT subsystem is comprised of a number of fixed-format displays which provide demand access to traffic data for

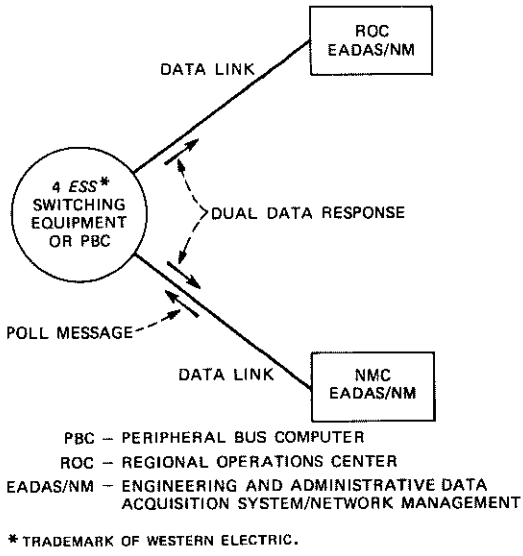



Fig. 9—ROC passive data collection feature.

the last 20 minutes and to current control and discrete status. Displays also may be used by network management personnel to implement or remove network controls, to modify certain trunk group-related reference database information, and to initiate requests for continuous observation of network components through the monitor system.

The CRT displays are divided into the following categories: *trunk group*, for traffic data on trunk groups at a selected switching system or switching systems within a geographical area; *machine*, for equipment and control status for a selected switching system; *control*, to display and implement network controls at one or more switching systems; *exception*, for resolution of data triggering exception indicators on the wallboard; *input and monitor*, for certain database and monitor system requests; and *analysis*, for analyzing traffic data to ascertain, for example, where there is idle capacity in the MTS network for network management reroutes.

The human interface to the CRT subsystem employs standard CLLI (Common Language Location Identification) codes to designate switching systems and trunk groups. In this way, a consistent identification of network components is achieved throughout the NMC/ROC/NOC hierarchy.

#### *4.1.3 Controls*

Network Management controls are activated and deactivated from selected CRT pages, after which appropriate messages are transmitted

to the switching systems where the controls are actually instituted. A control log is maintained in EADAS/NM so that network managers have a record of what actions were taken. Control changes affecting traffic in the upper levels of the MTS network are sent to the NOCS. If, for some reason, a network management control is applied locally at a switching system, an alerting discrete is picked up by EADAS/NM on the next 30-second discrete poll. EADAS/NM then does an audit to determine and record the new control status for that switching system.

#### *4.1.4 Record base definition and maintenance*

EADAS/NM requires a very large record base which describes the "cluster" or portion of the network being managed. A full set of database management tools are provided for network management personnel (system users) to input record information and to validate it. Manual and automatic audit capabilities are provided to keep the record base current. For example, if more trunks are added to a trunk group, EADAS/NM receives an alerting discrete that triggers an automatic audit to update its database.

Reference information is also exchanged between EADAS/NM and the NOCS. Specifically, the NOCS transmits to EADAS/NM a list of switching systems that are of interest to the NOCS, along with a set of switching system and trunk group exception threshold levels. In this way the NOC can control what network information it receives. The NOCS can also audit EADAS/NM for the status of controls in switching systems that are of NOC interest.

### *4.2 System hardware configuration*

Figure 10 shows a block diagram of the EADAS/NM minicomputer system and its relation to the data collecting and display devices. A brief description of the components follows:

EADAS/NM Minicomputer—The data processor for the system; it interfaces to all of the data collection, storage, and display peripherals, and executes the main system programs.

CRT Display Devices—Up to five DATASPEED Model 40 cathode ray tube terminal station arrangements in a KDP (Keyboard, Display, Printer) configuration.

Wall Display Board—Consists of an array of fluorescent, reflective, electromagnetically controlled indicators. A maximum of 4095 indicators can be used with a single wallboard driver. Up to two wallboard drivers may be equipped on an EADAS/NM system.

Disk Drives—Mass storage system using magnetic disk packs. The disk packs have a capacity of 80 million 16-bit words.

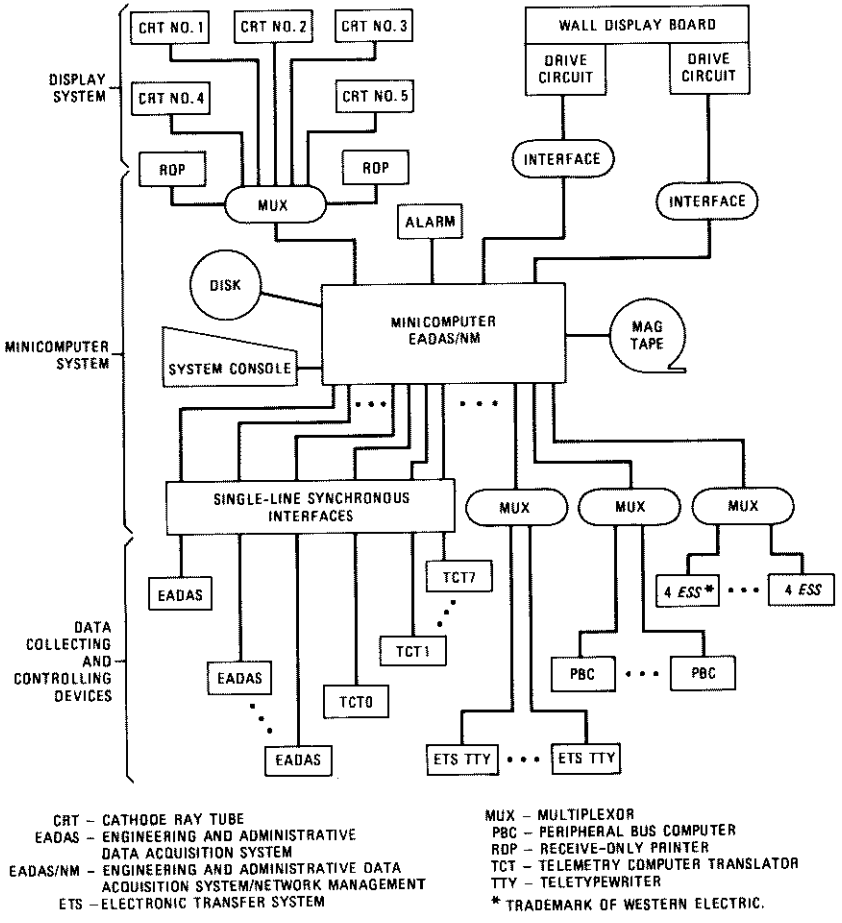



Fig. 10—Hardware configuration.

Command Console—Data terminal which may be used for any of the system commands.

Magnetic Tape System—Standard nine-track tapes at either 800 or 1600 bits per inch (bpi) used to initially load the programs, to back up the database, and to record real-time data for subsequent playback. The playback feature helps network managers review and critique specific events of network management interest using the wall display board and CRT displays. Such events can be played back in slow, real, or fast motion.

Telemetry Computer Translator (TCT) Interface—The E2A telemetry computer translators, each of which interfaces with up to four remote terminals, are used to receive discretes and send controls to

No. 4A ETS and Crossbar Tandem switching systems via the E2A telemetry system.

General Purpose Interface—Interface for both the wallboard and the E2A telemetry system.

Single Line Synchronous Interface—Interface capable of transfers at up to 9600 characters per second. These are used to transfer data from an EADAS to EADAS/NM and from EADAS/NM to NOCS.

Multiplexor (MUX)—Connects the processor with up to 16 asynchronous serial communication lines. One multiplexor is required for the EADAS/NM CRT terminals and receive-only printers (ROPs), and one or more additional multiplexors are required for No. 4A ETS, PBC, and 4 *ESS* switching system data links.

ETS TTY—100-baud teletypewriter channels used to transmit certain network controls to No. 4A ETS switching systems.

#### *4.3 Software architecture*

The EADAS/NM minicomputer software consists of the *UNIX** operating system and a collection of application programs that perform the EADAS/NM tasks. The application programs can be divided into real-time tasks, which perform data gathering, processing, and display functions; and certain non-real-time tasks. All applications programs are written in the C programming language. There are 12 major subsystems comprised of over 70 independent programs and over 1100 C-language modules. The overall software structure is depicted in Fig. 11 and will be discussed in more detail in this section.

##### *4.3.1 The* UNIX *operating system*

The *UNIX* operating system is general purpose, multiuser, and interactive. Utilized by EADAS/NM, its features include the following:

1. A hierarchical file system incorporating demountable files.
2. Compatible file, device, and interprocess Input/Output (I/O).
3. The ability to initiate asynchronous processes.
4. A system command language that can be selected on a per-user basis.

It should also be noted that the *UNIX* operating system is used for program development of the EADAS/NM project, providing a complete environment for the entry, compilation, loading, testing, maintenance, and documentation of the software product.

##### *4.3.2 The EADAS/NM database*

The EADAS/NM reference database consists of a number of user-

* Trademark of Bell Laboratories.

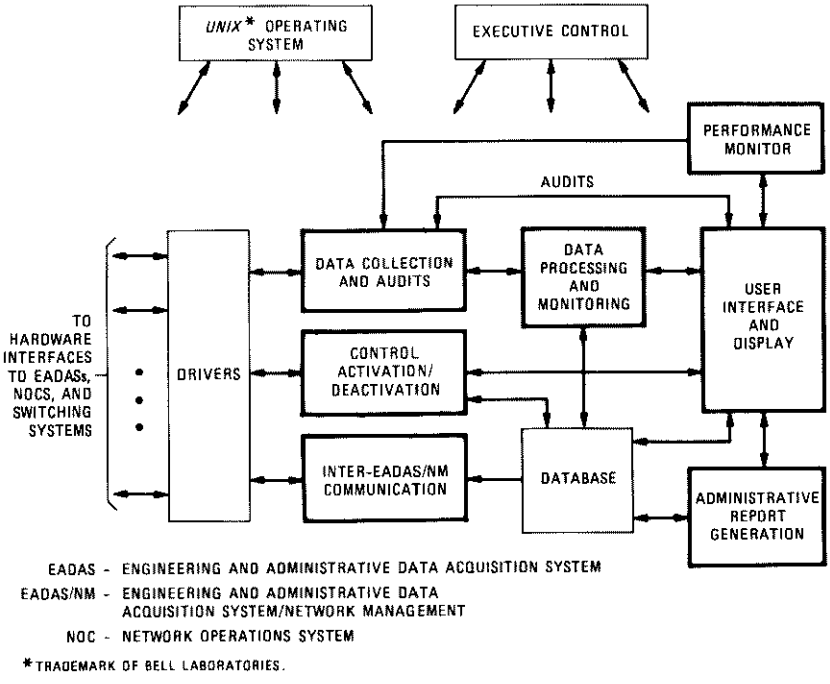



Fig. 11—EADAS/NM software structure.

built text files. These files are entered into the system using the *UNIX* operating system text editor.

In general, the information contained in these files falls into two categories. First is the information that pertains to the NMC itself, which includes the configuration of the center, the wallboard layout, and the threshold tables. The second category of information is concerned with the network to be monitored. This includes details about the switching systems and trunk groups in the NMC's cluster.

Once the database files are prepared, an operational database can be built using the database commands.

#### *4.3.3 Executive control program*

The executive control program is responsible for system initialization and for controlling the various real-time application processes. It accepts initial system parameters, starts the real-time application processes, monitors those processes for abnormal termination, and in general establishes and maintains the heartbeat of the system.

#### *4.3.4 Interface drivers*

A collection of driver programs interface the application software

with the interface hardware. These programs provide the basic communications protocols for transmitting and receiving information on the data links that connect to EADAS systems, the NOCS, and the various switching systems.

#### *4.3.5 Application programs*

Those blocks that are heavily outlined in Fig. 11 represent the actual EADAS/NM application programs. The major functions of these programs include data collection and audits, control activation and removal, inter-EADAS/NM communication, data processing, and user interface and display. In addition, a performance monitor maintains performance statistics on major system events such as data collection, exception calculations, and delivery of information to the NOCS. Also, a set of administrative report programs automatically generate needed switching system and trunk group summary reports.

The data collection and audit subsystem includes programs for polling both five-minute traffic data and 30-second discrete information. For example, the 4 *ESS* switching equipment traffic data polling programs poll 4 *ESS* switching equipment for switching system, trunk group, and HTR code data and then store these data on disk for subsequent retrieval by the data processing programs. Discrete polling programs perform similar functions and also update the wallboard based on the discrete information. Audits are essentially demand polls that may either be requested manually by the network manager or automatically whenever an alerting discrete (see Sections 4.1.3 and 4.1.4) is received. The audit program collects the new information and updates the system reference or control data accordingly.

Control activation and removal programs receive control requests from the user interface and display subsystem, format the control messages in the appropriate switching system command protocol, and send them to the switching system via the interface driver programs. The control programs perform checks to verify that controls are properly activated and deactivated. They also maintain up-to-date control status records in the database and the control log.

The inter-EADAS/NM subsystem handles all communication with the NOCS. It includes programs that receive appropriate information from the NOCS and then establish in the database a record of what switching systems and trunk groups are of NOC interest. Other programs arbitrate queues that contain exceptions, control status changes, and HTR code data for transmission to the NOCS. Still other programs format the messages and pass them to the NOCS interface driver for transmission. The inter-EADAS/NM subsystem can also handle messages that come from or go to other EADAS/NMs via the DTP.

The set of data processing and data monitoring programs perform calculations on five-minute traffic data to determine whether exception thresholds have been exceeded and to support trunk group data monitoring requests. These programs also update the wallboard, based upon the results of the exception calculations, and feed the inter-EADAS/NM subsystem with NOC-interest exceptions. Discrete monitor programs monitor user-specified discretes for user-specified time intervals and produce a discrete monitor list which is passed to the user interface and display subsystem for display.

The user interface and display subsystem includes all of the various CRT page programs (see Section 4.1.2), the programs that control the printing of information on the exception and monitor printer, and the programs that receive and interpret user commands.

## V. FUTURE NETWORK MANAGEMENT NEEDS

Future network management studies are aimed at providing improved methods for controlling the evolving switched network under overload and failure conditions. These efforts are expected to increase the reliance on automatic controls and improve centralized operations procedures and supporting system capabilities.

Simulation and analytical studies are under way to characterize the overload dynamics of a network dominated by electronic switching systems and CCIS. Specific studies focus on control strategies for new NCP-based services such as enhanced 800 Service, and the proposed evolution from the present hierarchical network to nonhierarchical, time-varying routing.[15]

## REFERENCES


1. A. E. Ritchie and J. Z. Menard, "Common Channel Interoffice Signaling: An Overview," B.S.T.J., *57*, No. 2 (February 1978), pp. 221–24.
2. P. J. Burke, "Automatic Overload Controls in a Circuit-Switched Communications Network," Proc. Nat. Elec. Conf., December 1968, pp. 667–72.
3. D. G. Haenschke, D. A. Kettler, and E. Oberer, "Network Management and Congestion in the U.S. Telecommunications Network," IEEE Trans. Commun., *COM-29*, No. 4 (April 1981), pp. 376–85.
4. L. A. Gimpelson, "Network Management: Design and Control of Communications Networks," Elec. Commun., ITT Corp., *49*, No. 1 (1973), pp. 4–22.
5. Y. Nakagome and H. Mori, "Flexible Routing in the Global Communication Network," Seventh International Teletraffic Cong., Stockholm, June 1973, Paper 426.
6. J. H. Weber, "Some Traffic Characteristics of Communications Networks and Automatic Alternate Routing," B.S.T.J., *41*, No. 2 (March 1962), pp. 769–96.
7. J. H. Weber, "A Simulation Study of Routing and Control in Communications Networks," B.S.T.J., *43*, No. 6 (November 1964), pp. 2639–76.
8. T. V. Greene, D. G. Haenschke, B. H. Hornbach, and C. E. Johnson, "No. 4 ESS: Network Management and Traffic Administration," B.S.T.J., *56*, No. 6 (September 1977), pp. 1169–202.
9. D. G. Haenschke, "Network Management—An Overview of Controls and Systems," Proc. National Electronics Conf., October 1980, pp. 364–67.
10. E. Oberer, "The Role of the AT&T Network Operations Center," Proc. of the

Comm. Techniques Seminar, Princeton University, March 1980, pp. 1-1 through 1-6.
11. W. S. Bartz, and R. W. Patterson, "Total Network Data System: National Network Management," B.S.T.J., this issue.
12. R. M. Averill, Jr. and R. E. Machol, "A Centralized Network Management System for the Bell System Network," Int. Switch. Symp. Record, Japan, October 1976, pp. 433-3-1 through 433-3-7.
13. D. G. Haenschke, "Network Management Systems in the United States," Proc. 9th Int. Teletraffic Cong., October 1979, Torremolinos, Spain, Paper 632.
14. R. L. Bennett, "EADAS/NM—The Computer System Supporting the Area and Regional Centers," Proc. Nat. Elec. Conf., October 1980, pp. 370-5.
15. G. R. Ash, R. H. Cardwell, and R. P. Murray, "Design and Optimization of Networks With Dynamic Routing," B.S.T.J., *60*, No. 8 (October 1981), pp. 1787-820.

## AUTHORS

**D. G. Haenschke,** Diplom Vorpruefung, 1951, Diplom Ing., 1953, Technical University, Vienna: Institute for Telecommunications, Vienna University, 1953-1955; Bell Laboratories, 1955—. Mr. Haenschke has planned communications systems for air traffic control, performed traffic studies applied to data communications, and engaged in long-range planning studies for the telecommunications network. In 1968 he became Supervisor of a group responsible for performance appraisals of operator services traffic systems, and the evaluation of a *Picturephone*® service trial. Since 1969 he has been responsible for the planning of traffic network management and overload control features for the telecommunications network and its interfaces with network management operations support systems. He is currently supervising a group concerned with network management capabilities for new switching and signaling systems and for new services for the Stored Program Control Telecommunications Network. Senior member, IEEE; member, AAAS, Verband Deutscher Electrotekhniker.

**George C. Ebner,** B.S.E.E., 1964, M.S.E.E., 1965, Massachusetts Institute of Technology; Ph.D. (Electrical Engineering), Ohio State University; Bell Laboratories, 1965—. Mr. Ebner's initial work at Bell Laboratories was in the exploratory development of switching processors and their associated peripherals. Later he was involved in the development of CCIS equipment. In 1973, Mr. Ebner became Supervisor of the CCIS Engineering and Current Planning Group, a responsibility whose major thrust was systems engineering for the rapidly evolving CCIS network. In 1979, he became Supervisor of the Network Management System Group, which develops generic software for the EADAS/NM. Member, Eta Kappa Nu, Tau Beta Pi, Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

*Total Network Data System:*

# National Network Management

By W. S. BARTZ* and R. W. PATTERSON*

(Manuscript received June 14, 1982)

During periods of network stress, network management (real-time monitoring and control of the network) assures optimum call-carrying capacity. To assure timely reactions to network overloads and failures, the national Network Operations Center coordinates the activities of a set of network management operations centers. Supporting the Network Operations Center, the Network Operations Center System collects data for major network elements and, within minutes, provides a national overview of exceptional conditions.

## I. INTRODUCTION

Real-time monitoring and control of the network through a coordinated set of network management operations centers assures optimum call-carrying capacity during periods of stress caused by overloads and failures. Switching systems are controlled and monitored in detail in specific geographical areas at the 27 Network Management Centers (NMCs) that blanket the country. Regional Operations Centers (ROCs) coordinate NMC activities within each of the ten switching regions, and monitor and control the regional switching and signaling systems. To coordinate and manage the activities of the NMCs, the ROCs, and the two Canadian regional management centers, the na-

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

tional Network Operations Center (NOC) at AT&T Long Lines' headquarters was placed in operation in 1977.[1]

The Network Operations Center System (NOCS), a one-of-a-kind system, supports the NOC. Twenty-four hours a day, seven days a week, NOCS collects data for the major elements of the telecommunications network, thousands of trunk groups, and hundreds of switching systems. It then checks these data for exceptional conditions, and displays information for these conditions within minutes of the occurrence. This national overview allows personnel at the NOC to coordinate appropriate reactions to national network situations, assuring uniformity of control.

The preceding paper in this issue of the *Journal*[2] explains the reasons for network management of the telecommunications network and describes the system that provides area and regional support for this function. In this paper, we describe the functions of the NOC, its support system, NOCS, and network management coordination for the North American telecommunications network.

## II. NETWORK MANAGEMENT COORDINATION

### *2.1 The network management operations structure*

As described in the preceding paper in this issue,[2] network management centers are organized in a three-level hierarchy. The national NOC is at the top level of this hierarchy and is supported by NOCS. Personnel at the NOC have ultimate responsibility for coordinating network management activities for situations that transcend regional boundaries. By assuring that actions taken optimize the use of network facilities, they protect areas hit by disasters and extraordinary calling volumes, coordinate the use of idle capacity when failures cut the capacity of normal routes, and assure consistent, compatible control application and removal.

While the NMC/ROC structure covers the entire domestic network, this center structure does not extend overseas. Most of the information about overseas destinations comes from data collected at the international gateway switching systems. Because no more than one gateway switching system is covered by any one NMC, the NOC plays a more direct role for overseas network management. Data from all gateways come together at the NOC. Thus, personnel at the NOC are in a unique position to see how calls from the domestic network as a whole are completing to overseas points. Because of this, NOC personnel take a more direct role in coordinating management activities at the international gateways and in working with overseas partners to assure consistency in methods and procedures.

### *2.2 Network management methods*

In addition to monitoring the network and coordinating management activities in real time, the NOC is responsible for establishing a unified network management methodology. NOC personnel work with AT&T and Bell Laboratories to determine future directions and capabilities for network management. Guidelines for the selection of automatic control responses and for manual control strategies are sent to the ROCs and forwarded to the NMCs. The NOC organizes preplanning for anticipated problems and post-event analysis and critiques. Finally, the NOC is responsible for training network managers by arranging on-site rotational assignments, administering a training school, and issuing newsletters covering recent developments affecting network management.

Putting the responsibility for network management methods at the NOC assures a consistent, unified application of network management techniques throughout the domestic telecommunications network.

## III. THE TELECOMMUNICATIONS NETWORK

The telecommunications network for which the NOC personnel are responsible comprises three distinct but highly interdependent networks.

First there is the top portion of the North American network hierarchy of toll switching systems and the trunk groups that interconnect them. This includes all ten United States regional (Class 1) switching systems and the two Canadian regional switching systems, about 70 sectional (Class 2) systems, 200 primary (Class 3) systems, and some toll (Class 4) systems. In total there are about 300 to 350 switching systems of interest to the NOC. The NOC does not monitor the local switching systems or their associated trunk groups.

The second network that the NOC personnel monitor is the overseas network, which comprises seven gateway switching systems through which all overseas traffic flows, trunk groups to several hundred overseas destinations, and a few trunk groups in the United States dedicated to overseas traffic between gateway systems. The seven gateway systems also serve as switching systems in the domestic network, where they range from Class 1 to Class 3 switching systems. The several hundred overseas trunk groups terminate in over one hundred foreign countries. There may be one or many trunk groups to any given country. Three different types of trunk groups within the United States are dedicated to overseas traffic. First, there are intergateway trunk groups reserved for overseas calls that overflow a particular gateway's trunk group to the destination country and must be routed to another gateway to use its trunk group to the desired

destination. Second, there are intergateway trunk groups that are used exclusively for incoming overseas calls that are merely transiting the U.S. to get to another overseas point. Third, there are trunk groups that connect the gateway systems to the overseas operations at the international operator centers.

The third network monitored by the NOC personnel is the Common Channel Interoffice Signaling (CCIS) network.[3] This high-speed, highly reliable signaling medium provides for fast call setup, efficient use and control of the network, and network features that require access to centralized databases. It is composed of several pairs of Signal Transfer Points (STPs) and thousands of signaling links that are of interest to the NOC. The signaling links connect the STPs to each other, to the switching systems, and to centralized databases called network control points.

Collectively these three networks are composed of many elements. To receive data on all of them constantly at one location would be an enormous task. If it were done, only a small part of the data collected would be used most of the time. Because of these two facts, the NOCS does not receive data on all elements all the time. Rather, it receives data primarily on an exception basis: data are transmitted to the NOCS when a problem develops in an area. While there are a few items that are always reported, such as data on interregional final trunk groups and overseas trunk groups, data for most entities are rarely reported.

## IV. OVERVIEW OF THE NETWORK OPERATIONS CENTER

The philosophy behind the Network Operations Center is exception alerting, followed by detailed reporting on the upper levels of the network. Because status information on all parts of the network would be much too voluminous to sort through manually, NOCS analyzes and displays information based on calculation results that exceed normal thresholds. Managers were alerted to these exceptional conditions in the network primarily by a wall display. This wall display, partially shown in Fig. 1, is visible from all operating positions. It is divided into network management and facility management sections. The display is functionally separated into two pieces; the right portion of the network management section displays information on the North American network, while status of overseas trunking is displayed on the left. The facility management portion of the wall display and the facility management function of the NOC are not discussed in this article.

The main part of the North American display is arranged in a matrix wherein each column represents data from one of the 12 switching regions. Conditions in any particular region affecting the

Fig. 1—NOC network management wall display.

rest of the network can be quickly deduced by scanning the appropriate column of the matrix. Similarly, conditions affecting a region can be detected by scaninng the appropriate matrix row. Thus, the network manager can quickly assess the location and scope of a network problem by viewing the domestic part of the wall display.

The main part of the international wall display is also laid out in a matrix fashion. The columns of this matrix represent the seven international gateways. The rows represent individual foreign countries. For backup and record-keeping purposes, wall display information is also printed on exception printers at the NOC.

For focus on individual exception conditions, CRT terminal displays provide resolution of data that triggers wall display exceptions. Other displays allow managers to analyze traffic patterns and aid in selecting and monitoring routes outside the normal routing chain when the normal in-chain routes cannot accommodate the traffic offered to them.

Analysis of troubles in the CCIS network is handled by an interactive, color graphics display, which provides information for both an overview and a detailed analysis of CCIS network problems.

## V. NETWORK OPERATIONS CENTER SYSTEM

The primary need of the personnel at the NOC in carrying out their job is information—data telling them about problem conditions in the telecommunications network and helping them decide what, if anything, they can do to alleviate the problems. Making this information available quickly and in the most usable format possible is the task of the NOCS.

### *5.1 System architecture*

Reference 2 describes how area and regional network management is supported by the Engineering and Administrative Data Acquisition System for Network Management (EADAS/NM). One function of these systems is data collection and screening for the NOCS. These systems collect data for all entities in the domestic network that are of interest to the NOC personnel. (The two Canadian regions are not monitored through EADAS/NM; data collection for those switching systems is provided by an E2 telemetry system.) The EADAS/NMs perform calculations on the data they collect and compare the results to thresholds established by the NOC personnel. For those entities that have exceeded one or more thresholds and for those entities that the NOC personnel have scheduled, the calculation results are transmitted to the NOCS.

Working in conjunction with NOCS, the inter-EADAS/NM network collects data from the various EADAS/NMs. This network

consists of the Data Transfer Point (DTP) together with the set of links to the EADAS/NMs over which information is sent. The EADAS/NM links to the DTP operate asynchronously at 1800 baud. As we will note later, these links are used in both directions and do more than just transmit results of real-time data calculations to the NOCS.

Figure 2 shows the configuration of the multiprocessor NOCS with the DTP. Because of its important function in the network management environment, it is a duplicated system. It is possible to switch

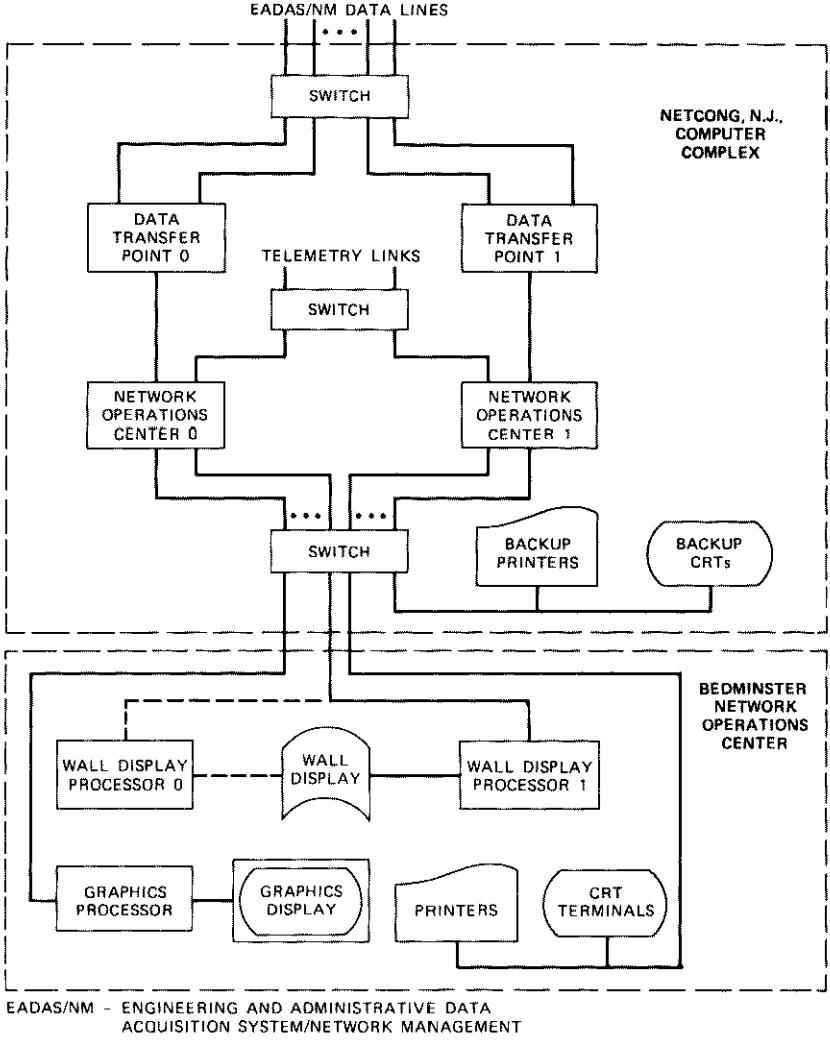



Fig. 2—System configuration for NOCS.

to the backup system within 15 minutes if a failure occurs on the active system. To achieve this, the backup system is always kept operational and is identical to the on-line system. Manual switches allow all EADAS/NM lines and all input/output (I/O) devices to be switched from one system to the other.

The primary computers in the system are not colocated with the primary display devices. As we stated earlier, the NOC is located in the Long Lines headquarters building in Bedminster, New Jersey, where the primary set of display devices and the small peripheral processors are located. The main computers, though, and the terminations of the EADAS/NM links are located about 25 miles away in Netcong, New Jersey. Also located at Netcong are a minimal set of backup display devices that can be used if a failure prevents the use of the facilities in Bedminster.

A simplex configuration of the systems would consist of the following elements:

1. The Data Transfer Point
2. The Network Operations Center System (NOCS) processor (labeled NOC in Fig. 2)
3. The Wall Display Processor (WDP) connected to the wall display
4. The Graphics Processor (GP) connected to the graphics display and the joystick
5. The Receive-Only Printers (ROPs)
6. CRT terminals.

Each DTP-NOC pair has a permanent high-capacity link joining the two processors. The WDPs are connected through the switch arrangement to a NOCS processor via a 4800-baud link. The wall display is 12 feet high, about 85 feet long, and uses luminescent, electromagnetically controlled indicators. These indicators are about one-inch square with black on one side and a reflective, highly visible color on the other side. Most of the wall display surface is actually taken up with labels. These identify the entities associated with the conditions represented by the indicators. The GP is also connected through a switch to a NOCS processor via a 4800-baud link. The ROPs are 4800-baud printers. The system supports both 1200-baud and 4800-baud CRTs. The telemetry links shown in Fig. 2 collect data from the two Canadian regions.

### *5.2 DTP functions*

Although the DTP performs several functions in the network management support system environment, its primary function is transferring data among the links that terminate on it. These links connect with the EADAS/NMs and the NOCS processor. Address information received in each message enables the DTP to determine on which link

or links the message is to be transmitted. Sometimes the message cannot be routed because of insufficient address information. In these instances the DTP must retrieve additional address information from its database before it can transmit the message to its proper destination(s). Keeping this additional address information in the DTP database eliminates the need to maintain it at all the EADAS/NMs.

In Section 5.1 we stated that the EADAS/NMs determine what data the NOC personnel want and then transmit that data to the NOCS. This information arrives at the DTP addressed to the NOCS, which is where the DTP routes it. In addition to this type of message the DTP also receives messages from the NOCS processor addressed to one or more EADAS/NMs. These messages supply information such as thresholds to be used to determine what data to send to the NOC. The DTP also receives messages from EADAS/NMs addressed to other EADAS/NMs.

The basic data update interval used by the network management centers and the NOC is five minutes. Although some events are reported as they occur or soon thereafter, the bulk of the network management data consists of counters scored locally by the switching systems or STPs. These are read and zeroed once every five minutes. The timing of these five-minute intervals is another function that the DTP performs. When a period begins, the DTP sends a message to the NOCS processor and to all the EADAS/NMs. As soon after the beginning of the period as possible, the EADAS/NMs start collecting data from the entities for which they are responsible, performing the calculations on these data, and sending the exceptions and scheduled data to the DTP. The DTP receives this five-minute data and passes it on to the NOCS processor until all the EADAS/NMs have reported that they have sent all their data. In the event that a cutoff time near the end of the five-minute interval is reached, the DTP stops the EADAS/NMs from further transmission and forwards the available data to the NOCS processor. When all EADAS/NMs have completed data transmission or the cutoff point is reached, the DTP sends a message to the NOCS and all EADAS/NMs to inform them that this period is over. Five-minute data for a period received by the DTP after the cutoff point is not passed on to the NOCS processor.

The DTP also monitors the inter-EADAS/NM links for problems and keeps its database up to date. If any of the links from the EADAS/NMs are having trouble, the DTP disconnects those, and then tries to reconnect them to clear the condition. The current status of all links is constantly available and can be accessed by the NOC personnel. The DTP receives all its database information automatically from the NOCS processor. Whenever the NOCS processor or the DTP is restarted, or the NOC personnel install new information in the NOCS

database that could affect the DTP, the NOCS processor updates the DTP database without any manual intervention.

### *5.3 Reference data*

The NOCS must have a large amount of reference data to display information. This reference data reflects the characteristics of all the switching, trunking, and signaling entities discussed in Section III. The NOCS reference data must be consistent with the reference data used at the EADAS/NMs. To achieve this consistency, and to handle the many database changes that occur, NOCS derives a major portion of its reference data from reference data tramsmitted by the EADAS/NMs to the NOCS processor over the inter-EADAS/NM links. It generates the remainder from data that are input by the NOCS database administrator.

The reference data transmitted from the EADAS/NMs to the NOCS processor are in binary format. When these data are received, they are converted to an ASCII format and stored in the NOCS. The NOCS database administrator can then modify and augment this information as required before using it to create the active NOCS database. Some information that is needed in the NOCS database is not received from the EADAS/NMs and must be manually entered and updated at the NOCS. The NOCS database administrator has a set of commmands to create, modify, examine, and audit additional reference information. The creation and installation of the active database from the combined EADAS/NM and locally supplied reference information are also under the control of the NOCS database administrator.

In addition to the reference data sent to the NOCS from the EADAS/NMs, there is also reference data sent from the NOCS to the EADAS/NMs. These include such items as thresholds to be used by the EADAS/NMs and a common set of location (switching system and STP) identifiers. To ensure that all EADAS/NMs are using the same version of these tables, each version has a unique identifier sent along with it. Whenever an EADAS/NM is reconnected to the DTP, it sends the NOCS a list of its current table identifiers. If they are not the correct set, the NOCS sends the EADAS/NM the proper tables automatically.

### *5.4 Real-time data*

The types of real-time data that the NOCS receives from the EADAS/NMs include:

1. Five-minute data calculation results
2. Switching system and STP status indicators (hereafter referred to as machine status indicators)
3. Control activations and deactivations.

We discussed the five-minute data calculations in Section 5.1. The EADAS/NMs collect data, perform calculations on the data, compare the results to NOCS established thresholds, and transmit exceptions and scheduled data to the NOCS. The EADAS/NMs collect data on trunk groups, signaling links, switching systems, STPs, and destination codes. These raw data are used by the EADAS/NMs to calculate rates and percentages, which can then be compared to thresholds. Whenever it is determined that a calculation result exceeds its threshold, all the data for the entity concerned are sent to the NOCS.

The NOCS stores all the five-minute data calculation results that it receives in an interval for twenty minutes. Therefore, at any time network managers can retrieve the data for the four most recent five-minute time periods. When data for a new five-minute period are available, exceptions are immediately displayed and automatically recorded for the network managers. We discuss this further in Section 5.5.

NOCS uses the machine status indicators, received as often as every 30 seconds, to notify the NOC personnel of any significant switching system or STP problem as quickly as possible. The events that are reported include congestion and failure conditions measured at switching systems and STPs. Whenever new machine status information arrives, NOCS immediately displays it to the network manager. In addition, a history of all machine status changes is kept and is available to the network managers (see Section 5.5).

The third type of real-time data that NOCS receives is control activations and removals. NOCS receives information on code controls and trunk group cancel, skip, trunk reservation, and reroute controls. At the time that any of these controls are applied to or removed from a NOC interest trunk group, the NOCS is informed. Furthermore, the NOCS control audit capability permits it to request an update of all the controls that are active in all the switching systems monitored by an EADAS/NM.

### *5.5 Data display*

Sections IV and 5.1 described briefly the functional characteristics of the NOCS I/O devices. This section describes how these devices provide an interface to the user.

#### *5.5.1 The wall display*

The primary objective of the wall display is to alert the network managers to problems in the network. In general it is not used to determine exactly what a problem is or how severe it is; this is done by other I/O devices available on the system. The indicators on the wall display are used to inform the NOC personnel that some calcu-

lation result has exceeded its threshold, some system condition has activated a machine status indicator, or some control has been invoked. About 30 percent of the wall display on the left is used for overseas items, about 40 percent in the middle is used for domestic conditions, and the remaining portion on the right is used for a display of transmission facilities. This right portion is not controlled by the NOCS and is not discussed in this article.

The domestic section has four general areas, as shown in Fig. 3. Area 1 is the largest portion. Its 12 × 12 matrix display of intraregional and interregional trunking conditions was described in Section IV. Each column has the regional center name at the top and the names of all the regional and sectional centers below it. To the left of each switching system name are four indicators that reflect conditions from the regional center listed at the top to that particular system. Area 2 also lists all the regional and sectional centers. To the left of the names this time are seven indicators that reflect code controls or trunk group controls activated at the named systems. To the right of these names are six (for sectional centers) or seven (for regional centers) more indicators showing the machine status indicators. Area 3 contains more specific information on which machine status indicators are active, as well as some miscellaneous items about the NOCS, threshold tables in use, and other conditions. Area 4 is similar to Area 1 in that there is a column for each of the twelve regions. Each column shows the status of the STPs in the region and the signaling links that emanate from them.

The overseas portion of the wall display is shown in Fig. 4. The layout in this portion is straightforward; all the countries that have trunk groups are listed in three columns. Each country also has the names of its gateway switching systems listed. To the left of the country names two items are shown: (1) a label listing the International Operator Center(s) that serves the country, and (2) indicators displaying code controls in effect. To the right of the country names

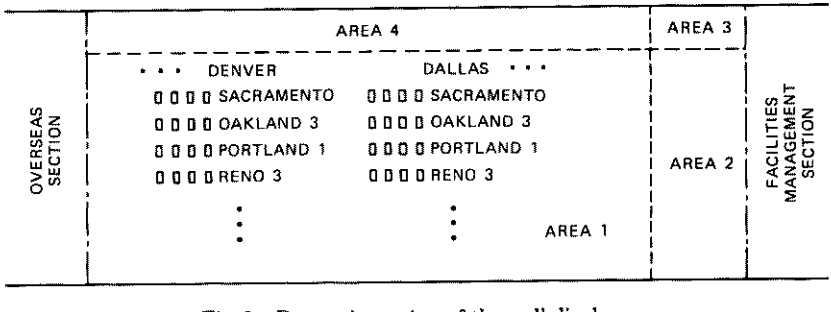



Fig. 3—Domestic section of the wall display.

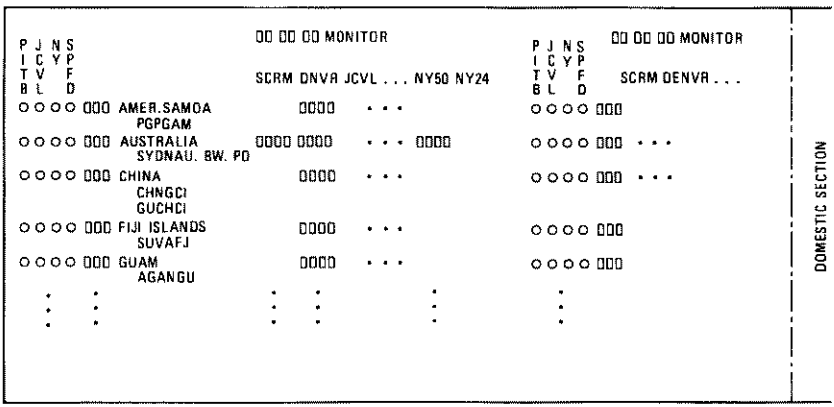



Fig. 4—Overseas section of the wall display.

are seven columns for the seven U.S. gateway switching systems. If a gateway serves a country, there are four indicators that show conditions to that country.

### *5.5.2 The CRTs*

The CRTs are the primary tools that enable network managers to retrieve switching and trunking information from NOCS. Once alerted to a problem by the wall display, network managers analyze details of the problem using the CRTs. Since the network managers at the NOC have to obtain information from the system as fast as possible, a series of fixed format displays have been developed for use on the CRTs. They provide quick access to almost all the data in the system with a minimum of input. These displays or "pages" are divided into the following categories:

1. Analysis pages, for analyzing traffic data patterns
2. Exception pages, for resolution of data-triggering exception indicators on the wall display
3. Input pages, for some of the database modifications.

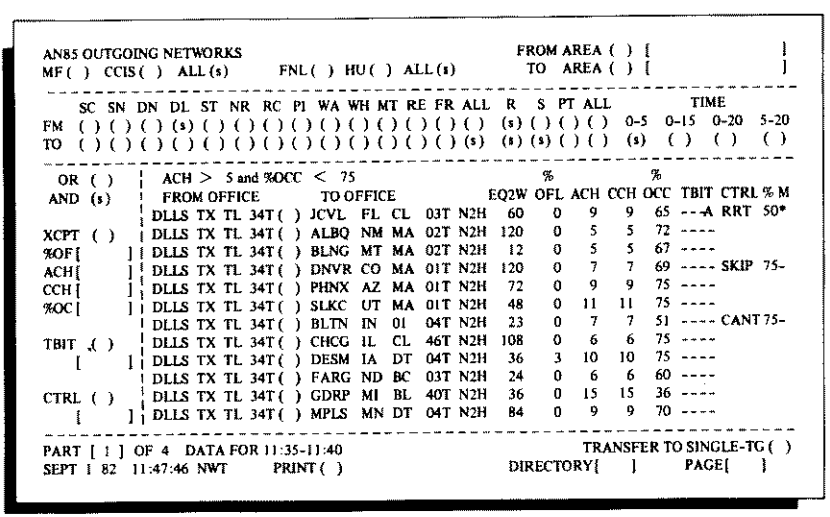
The pages provide access to data for the four most recent five-minute periods, current machine status indicators, and current controls. A page consists of up to 68 lines, 80 characters in length. It contains fixed information, known as "background" and variable input and output areas, referred to as "windows." Input windows enable network managers to select parameters quickly by entering a single character designate (+), or a short string of characters, such as the first few characters of a Common Language Location Identifier. After designating the appropriate input parameters or using the default values where appropriate, the network manager depresses the SEND

key. The system then responds by filling in the output windows with the desired information.

Of the eleven analysis pages in the system, seven are for analyzing domestic trunk group conditions, such as reroute controls in effect or interregional trunk group congestion. The other four are for looking at overseas trunking conditions, such as the traffic from a gateway to a country. The 13 exception pages provide data for machine exceptions, domestic and overseas trunk group exceptions, and code-related completion problems.

Figure 5 is an example of an analysis page. This page is for viewing trunk groups from one geographical area to another. In the area at the top and along the left edge, the network manager enters selection parameters in input windows. The remainder of the page is the display area (output windows) for the trunk groups that meet all the input selection criteria. This area identifies the trunk groups by the Common Language Location Identifiers of the switching systems at each end, and data shows calculation results and conditions affecting the trunk group.

In addition to providing access to current data, status, and controls, the CRTs serve other purposes. The NOCS maintains a history of machine status indicators. Using a system command entered on any of the CRTs, network managers can obtain specific subsets of this file tailored to their current interest. A report may cover from 30 minutes to several days of data, and may specify a single machine or all machines, as well as one or more specific status indicators. The data in the report can be ordered by time or by machine.



```
AN85 OUTGOING NETWORKS                                  FROM AREA ( ) [          ]
MF( ) CCIS( ) ALL(s)      FNL( ) HU( ) ALL(s)            TO  AREA ( ) [          ]

    SC SN DN DL ST NR RC P1 WA WH MT RE FR ALL  R  S PT ALL        TIME
FM  ( ) ( ) ( ) (s) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (s) ( ) ( ) ( )  0-5  0-15  0-20  5-20
TO  ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (s) (s) (s) ( ) ( )  (s)  ( )   ( )   ( )

 OR ( )  |  ACH > 5 and %OCC < 75                                  %          %
 AND (s) |  FROM OFFICE        TO OFFICE                  EQ2W OFL ACH CCH OCC TBIT CTRL % M
         |  DLLS TX TL 34T( ) JCVL  FL CL 03T N2H   60    0   9   9  65 --A RRT 50*
XCPT ( ) |  DLLS TX TL 34T( ) ALBQ  NM MA 02T N2H  120    0   5   5  72 ----
%OF[    ]|  DLLS TX TL 34T( ) BLNG  MT MA 02T N2H   12    0   5   5  67 ----
ACH[    ]|  DLLS TX TL 34T( ) DNVR  CO MA 01T N2H  120    0   7   7  69 ---- SKIP 75-
CCH[    ]|  DLLS TX TL 34T( ) PHNX  AZ MA 01T N2H   72    0   9   9  75 ----
%OC[    ]|  DLLS TX TL 34T( ) SLKC  UT MA 01T N2H   48    0  11  11  75 ----
         |  DLLS TX TL 34T( ) BLTN  IN 01 04T N2H   23    0   7   7  51 ---- CANT 75-
TBIT ( ) |  DLLS TX TL 34T( ) CHCG  IL CL 46T N2H  108    0   6   6  75 ----
   [    ]|  DLLS TX TL 34T( ) DESM  IA DT 04T N2H   36    3  10  10  75 ----
         |  DLLS TX TL 34T( ) FARG  ND BC 03T N2H   24    0   6   6  60 ----
CTRL ( ) |  DLLS TX TL 34T( ) GDRP  MI BL 40T N2H   36    0  15  15  36 ----
   [    ]|  DLLS TX TL 34T( ) MPLS  MN DT 04T N2H   84    0   9   9  70 ----

PART [ 1 ] OF 4  DATA FOR 11:35-11:40                    TRANSFER TO SINGLE-TG ( )
SEPT 1 82  11:47:46 NWT      PRINT ( )                   DIRECTORY[   ]   PAGE[   ]
```

Fig. 5—Example of an analysis page.

A large set of system commands is available to administer and monitor the NOCS and its database. These commands are also entered on the CRTs and can be directed to either the NOCS processor or the DTP. The DTP commands permit such functions as the attaching and detaching of EADAS/NMs; the starting and stopping of exception transmission from an EADAS/NM; and the printing of DTP data tables, inter-EADAS/NM network status, and various other network statistics. The NOCS processor commands permit such functions as initializing and testing the wall display and graphics processors, administering the database (as discussed in Section 5.3), auditing EADAS/NMs for current status indicators and controls, and controlling the playback function (to be described in Section 5.6).

#### *5.5.3 The graphics displays*

The graphics subsystem of the NOCS provides an interactive, color display that is used as both an alerting device and a detailed data source. It is interactive in that its fast update rate permits the user to receive near instantaneous response to information requests. It is capable of displaying 17 colors in 15 different shades and producing two- and three-dimensional shaded displays. This device permits adjustment of three input coordinates and the selection of items in the displays. Cursors on the monitor show the current values of the three coordinates.

The operating system is designed to support up to 15 functions, or collections of closely related displays. These can show information from the four latest five-minute data periods along with the current controls and machine status indicators. The operating system can also automatically update the current displays with new data when the NOCS processor indicates it is available. Currently, the functions available on the graphics system are:

1. CCIS monitoring
2. Domestic trunk group monitoring and analysis
3. A geographic editor.

Since the graphics system is a relatively recent addition to the system, this list of functions is expected to grow.

The CCIS display pictorially represents all the elements in the CCIS network, and color-coded symbols alert the network manager to exception conditions. The display shows the latest sample of five-minute data and current CCIS-related machine status indicators for the entire CCIS network. This display allows the network manager to determine the general nature of CCIS problems. At that point the network manager can request that particular link sets with exceptions be expanded and that labels and specific data calculation results be shown. The domestic trunk group display graphically represents the

switching and trunking entities to be monitored and color codes problems to allow the network managers to focus quickly on the worst problem areas or to determine the type of problem that exists. The geographic editor is used for database administration only.

#### *5.5.4 The printers*

The receive-only printers on the NOCS are used as exception printers, as monitor subsystem printers, and for database work. The exception printers exist to maintain a permanent record of all domestic and overseas exceptions that are reported by the EADAS/NMs. Trunk group, machine, and signaling link exceptions are recorded.

The monitor subsystem gives the network manager a printed chronological record of data for up to 150 trunk groups. The trunk groups for which print criteria are selected and set are listed on a separate printer called the monitor printer. Furthermore, the network manager can activate special indicators on the wall display or set off an audible alarm when a monitor system threshold is exceeded. These wall display indicators exist on a per-region basis on the domestic part of the wall display and on a per-gateway-system basis on the overseas part of the wall display.

### *5.6 Playback*

The playback subsystem of NOCS allows the status of the switching network, as reported to the NOCS by EADAS/NM and the E2A telemetry interface of NOCS, to be recorded and later replayed on the system. The recording is done at any time on the active NOCS processor. All the data messages that come over the DTP-NOCS link and all the data that comes from the E2A telemetry are copied to a file on the disk. This file is large enough to hold more than a day's worth of data at normal system load. There are two such files on the disk so when one is filled, the other can be activated. Either file can be copied to magnetic tape for later analysis if desired. The recording of incoming data in the playback system disk file does not affect normal operation of the NOCS. When the record mode is activated, all pertinent system information that will be required to replay the data is also recorded.

Previously recorded network data can be replayed at any time on the off-line NOCS processor. If the data to be replayed are not currently in a playback system disk file, they are read in from magnetic tape. Using the switches, the system administrator can connect any combination of I/O devices, i.e., wall display, CRTs, graphics, or printers, to the system running replay. These devices will then appear as though they are connected to a NOCS processor that is receiving live data. When the playback system is running in the replay mode, it

retrieves the data stored on the disk file and feeds it to the NOCS as if it had just come from the EADAS/NMs or the E2A telemetry. Several options are made available to the replay operator. Replay can be made to run the system at a "normal" speed or faster or slower than normal. It can also be made to start at a certain point in the file and to continue from there, or to read in one period of data and not advance.

### *5.7 NOCS software features*

Both the DTP and NOCS processor run *UNIX** operating systems and all the application code in both systems is implemented in the C programming language.[4] This paper will not attempt to go into detail on the software architecture of the NOCS. Rather, a few salient points of the implementation will be discussed briefly because of the part that they play in making the NOCS a successful real-time operations support system.

DTP software takes the data from the EADAS/NM lines and "funnels" it into one data stream for internal processing. Most of the messages are destined for the NOCS processor, so the DTP "blocks" as many as possible of the short EADAS/NM messages together into larger messages for efficient transfer through the DTP and to the NOCS.

Figure 6 shows the primary subsystems in the NOCS processor. All data coming in from the DTP or the E2A and all data going out to the DTP must pass through the Message Processing System (MPS). The MPS sorts incoming data into two basic categories: reference data and real-time data. It passes the reference data on to the Reference Data Update System (RDUS) to be processed as discussed in Section 5.3 while the MPS enters the real-time data into the database as they arrive. These processes also make certain data items immediately available to other processes so that the network managers can be alerted as fast as possible to potential problems. When all the data for a five-minute period are in, the MPS notifies all interested subsystems.

A prominent part of the NOCS software is the Data Access System (DAS), which handles all interactions with the database. The DAS is a large set of subroutines that provide common, standard access to the real-time and active reference database files. This single point of access to the system database has many benefits; the two most prominent ones are:

1. Standardized, centrally maintained retrievals that can be used by a multitude of processes

---

* Trademark of Bell Laboratories.

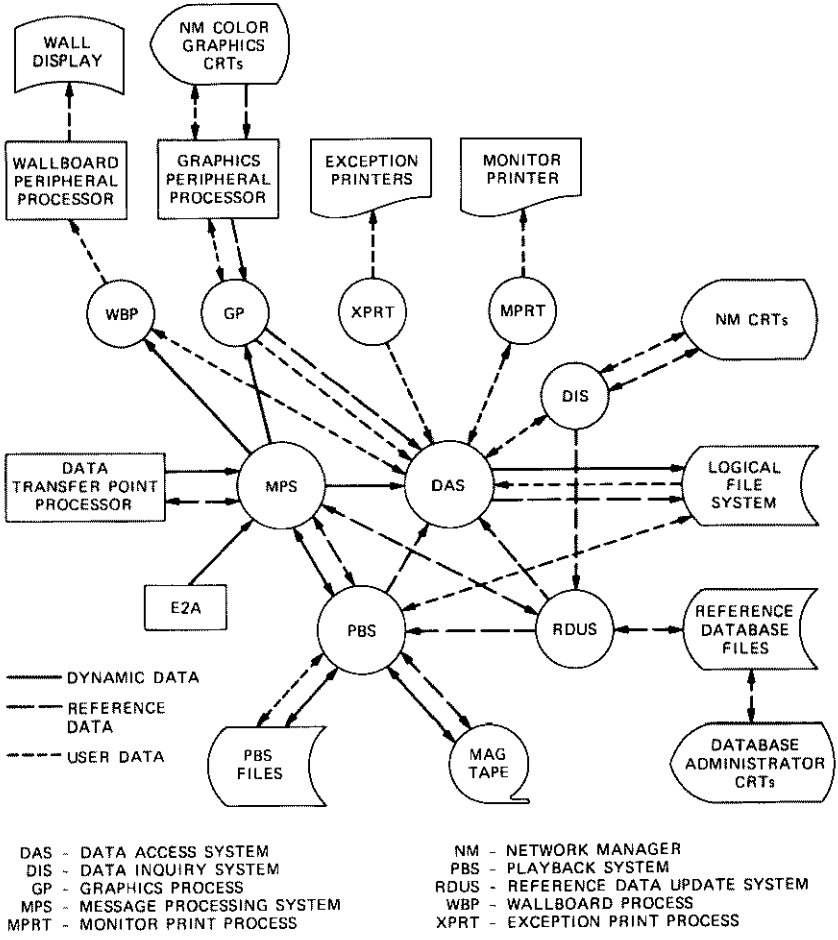



Fig. 6—Major NOCS software systems.

2. Control of the reading and writing of individual files by a multitude of processes.

In addition to these benefits, the DAS also provides relief on system disk I/O through a per-process cache buffering mechanism that allows frequently accessed disk sectors to be kept in core, thereby avoiding additional disk reads.

In the NOCS processor the real-time data and active reference database are not kept in the standard *UNIX* file system but rather are maintained in a special file system called the Logical File System (LFS). The LFS is implemented using the raw I/O facility in the *UNIX* operating system. It provides a fast excess file system that:

1. Isolates the applications programs from having to know the physical disk location of files.

2. Enables application programs to create, delete, read, write, switch, and copy logical files.

3. Contiguously stores LFS files. This results in reduced disk I/O transfer time.

4. Transfers LFS files directly between the disk and the application data area, thereby reducing system overhead on each disk transfer.

The portion of the grahics subsystem that resides in the GP (Graphics Processor) runs under a specially designed operating system. The system provides functions required by the applications processes that drive the graphics displays. Most of the code that resides in the GP is implemented in C language; the remainder, though, is implemented in assembly language or is microcoded. This non-C code exists only in the operating system and is used to achieve the speed required to produce the interactive displays.

## VI. CONCLUDING REMARKS

The NOC, supported by NOCS, plays a key role in coordinating and managing the North American telecommunications network. Its role is evolving from coordination of different geographical areas of the MTS network toward coordination and management of different subnetworks and service capabilities of the telecommunications network. We discussed the North American message network, the CCIS network, and overseas trunking. Another function of the NOC, not yet supported by NOCS, is monitoring and control of System 800, as noted in Ref. 2. As the network evolves with topological changes, new approaches to common channel signaling, and new service capabilities, we expect coordination and management of new service capabilities and the interactions of different network components to become an increasingly important part of the NOC's job.

## REFERENCES


1. E. Oberer, "Role of the AT&T Network Operations Center," Proc. Comm. Techniques Seminar, Princeton University, March 1980, pp. 1-1 through 1-6.
2. G. C. Ebner and D. G. Haenschke, "Total Network Data System: Network Management," B.S.T.J., this issue.
3. A. E. Ritchie and J. Z. Menard, "Common Channel Interoffice Signaling: An Overview," B.S.T.J., *57*, No. 2 (February 1978), pp. 221–4.
4. H. Cohen and J. C. Kaufeld, Jr., "The Network Operations Center System," B.S.T.J., *57*, No. 6, Part 2 (July-August 1978), pp. 2289–304.
5. K. E. Greisen, "Keeping the NOC Informed—The Computer System That Provides the Data," Proc. Nat. Elec. Conf., *34* (October 1980), pp. 382–6.


## AUTHORS


**Robert W. Patterson,** B.S.E.E., 1976, University of Tennessee; M.S.E.E., 1978, M.I.T.; Bell Laboratories, 1976—. At Bell Laboratories, Mr. Patterson

initially worked on network management surveillance and control capabilities for electronic switching systems and new services. He currently supervises the systems engineering for network management operations centers and support systems. Member, IEEE.

**William S. Bartz,** B.S.E.E., 1970, Manhattan College; M.S.E.E., 1971, Cornell University; Bell Laboratories, 1970—. From 1970 to 1978 Mr. Bartz worked in the Toll Crossbar Systems Department on the Peripheral Bus Computer system development. Since 1978 he has supervised the Network Operations Center System group in the Switching Operations Systems laboratory.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Equipment Systems

By N. D. FULTON,* J. J. GALIARDI,* E. J. PASTERNAK,*
S. A. SCHULMAN,* and H. E. VOIGT*

(Manuscript received March 10, 1983)

This article describes a set of five systems collectively called the Total Network Data System Equipment Systems, which support a number of network administration and engineering functions in the operating telephone companies. These systems are run by the operating telephone companies on large mainframe computers. In an average telephone company they support the management of the collection and distribution of over 50 million individual items of network data each week. They also provide reports to assist in the engineering and administration of No. 5 Crossbar switching offices, and the load balancing of all local switching systems. Initial development of these systems occurred in the early to mid-1970s. They are installed in all Bell System operating telephone companies and Bell Canada. Currently, on-line record base updates and inquiries and on-line viewing of report outputs are being developed.

## I. INTRODUCTION

### *1.1 Role of TNDS/EQ in support of network functions*

Basic to the operation of the telephone business is the collection, processing, distribution, and analysis of traffic data. The usage and availability for service of each of the many millions of components in

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

the telephone system are measured by evaluating these data. In a typical week, the average Operating Telephone Company (OTC) collects over 50 million separate traffic measurements. Aided by the reports formulated from these data, telephone company personnel monitor how well customers are being served, place work orders as needed to optimize usage of currently installed equipment, and formulate traffic trends to forecast future needs. Almost all major decisions in the managing of the telephone network are influenced by the conclusions derived from traffic data.

The group of systems referred to as the TNDS Equipment (TNDS/EQ) systems primarily support data administration and central office switching engineering and administration functions in network organizations. Included in data administration is the collection, verification, and delivery of traffic data, both to the network and to other users. (As an example of the latter, local/toll call separations data are delivered to Division of Revenues.) One function of central office switching engineering and administration is to monitor usage patterns of the many individual traffic-sensitive components in a central office to ensure good customer service along with proper equipment utilization.

### *1.2 Organization of article*

The remainder of Section I provides an overview of TNDS/EQ and discusses some of the history and decisions involved in its evolution. Section II describes the data collection environment, which serves as the input to the TNDS/EQ systems. Section III discusses the computer operational environment under which these systems run, and describes some common controls that were initiated to improve this environment. Sections IV through X describe in detail each of the TNDS/EQ components, and Section XI indicates some of the future directions for TNDS/EQ.

### *1.3 Overview of TNDS/EQ*

TNDS/EQ comprises five component systems (see Fig. 1) and two special facilities. As discussed in Section 1.4, planning for these systems goes back to the mid-1960s with most initial development occurring in the early to mid-1970s. They are currently used throughout all Bell System operating telephone companies and Bell Canada. TNDS/EQ operates systems on local IBM 370 or equivalent mainframe computers, and are run weekly in a batch processing mode. Comprising over one-half million instructions, the programs have been written in a combination of COBOL, FORTRAN, PL/1, and Basic Assembly Language. Products delivered to the operating companies include executable load modules, IBM job control language, installa-

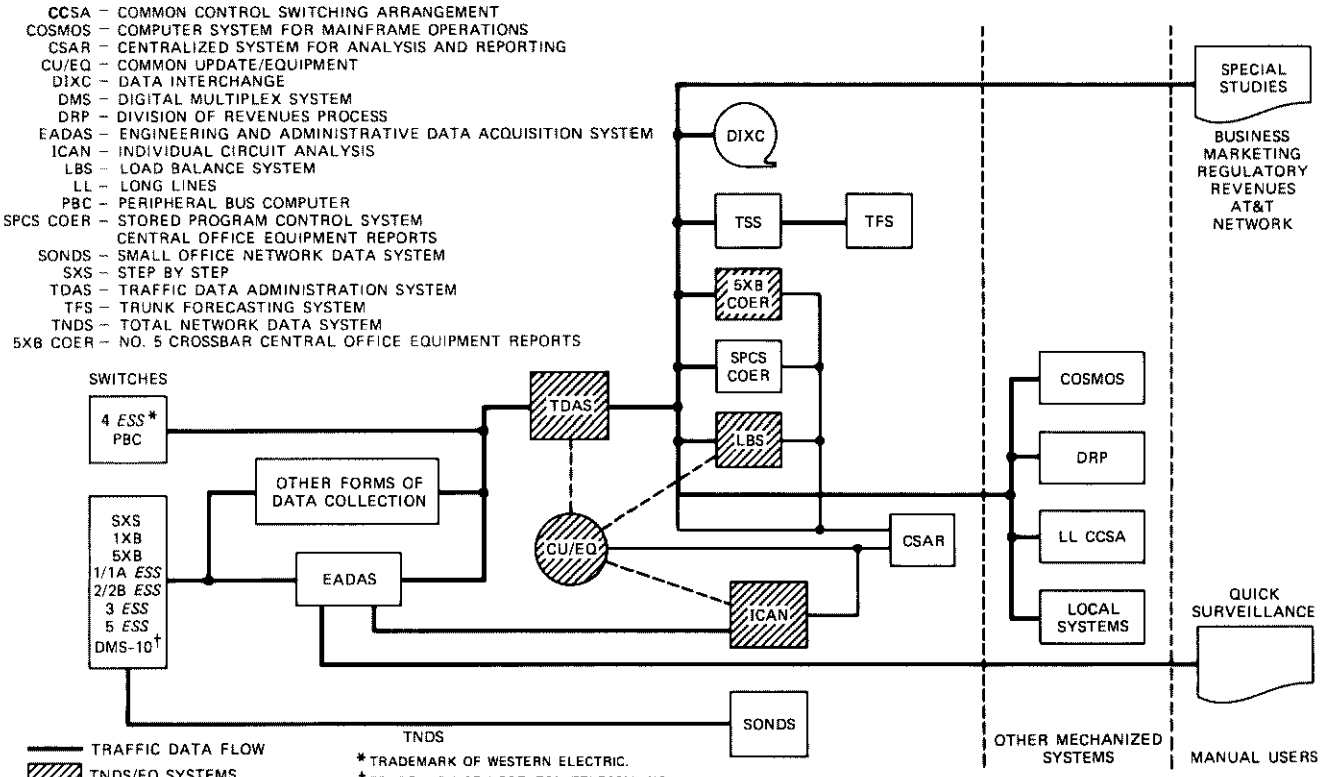



Fig. 1—TNDS data processing.

tion and conversion instructions, and documentation to support both computer center operations personnel and network users.

Three other systems are sometimes considered part of TNDS/EQ—the Stored Program Control System Central Office Equipment Reports System (SPCS COER),[1] the Centralized System for Analysis and Reporting (CSAR),[2] and the Small Office Network Data System (SONDS).[3] Because of their different operating environment—running on a single, time-shared computer for all using companies, rather than being deployed to each operating telephone company—they are presented in other articles in this issue.

One of the TNDS/EQ components, Common Update (CU), maintains master files used by a number of the other TNDS component systems (see Section IV). These files contain the basic records needed to process the data. As an example, the CU master files contain information that defines what data are to be transmitted to each of the downstream systems.

The Traffic Data Administration System (TDAS) is the key TNDS element serving the data administration functions in the operating companies (see Section V). TDAS accepts traffic data from most data collection sources, including paper tape, punched card, and magnetic tape. Among the sources of data are the Western Electric Engineering and Administrative Data Acquisition System (EADAS),[4] several non-Bell System data acquisition systems, and those switching systems, such as the Bell System 4 *ESS** switching equipment, which internally summarize traffic data. Using CU files, TDAS identifies, verifies, and labels each individual incoming data item, saving those items wanted for further processing and discarding those that are not wanted. On a weekly basis, scheduled data are transmitted to the requesting downstream systems. These output (downstream) interfaces are unlimited, as TDAS can effectively fill any order for data provided the data have been collected and delivered to it.

Included among the downstream systems are two component systems of TNDS/EQ: the Load Balance System [(LBS), see Section VIII], and the No. 5 Crossbar Central Office Equipment Reports System [(5XB COER, see Section VI]. LBS reports assist the network administrator in assigning customer lines to the switching machines in a manner that balances the traffic load on the switch. It also provides telephone company management personnel with quantified measures (known as the Load Balance Index) on how well the office balance is being maintained. 5XB COER reports on weekly, monthly, and annual traffic load for each individual component in a No. 5 Crossbar switching office.

* Trademark of Western Electric.

Another component of TNDS/EQ, the Individual Circuit Analysis System (ICAN), serves two basic functions (see Section VII). First, it provides reports to help administer the record base for the Individual Circuit Usage Recording (ICUR) option of EADAS; second, it analyzes individual component usage data to identify unusual measurements or patterns that could be indicative of central office or trunk maintenance problems.

In addition to these five component systems, TNDS/EQ has two special facilities, the Report Distributor (Section IX) and the Management Reporting System [(MRS) see Section X]. The former helps to deliver outputs to the network users and produces both paper and microfiche output. The MRS enables an operating company to access selected portions of the record base and output files to produce its own specialized reports.

### *1.4 History and major decisions*

Elements of TNDS/EQ originated in several different development organizations within Bell Laboratories, AT&T and the Operating Telephone Companies. Most of the components were developed prior to the overall concept of a coordinated Total Network Data System. As a result, many activities over the past decade have been directed at integrating the systems. Some of this history has been addressed in an earlier article.[5] Even though there originally was not a complete plan, the TNDS architecture quickly and naturally fell into an "upstream" and "downstream" split. Placed upstream were the functions associated with data collection and real-time exception reporting, activities suitably accomplished with minicomputer technology. On the other hand, the heavy processing functions involved with crunching tens to hundreds of millions of weekly data items into usable engineering and administrative reports readily fell into a downstream processing environment of large mainframe batch computers.

The history of downstream TNDS components goes back to 1966 (see discussion in Ref. 5), with a decision to centralize the development of a computerized Trunk Facilities System. This decision led to a major development program in the Business Information Systems (BIS) area of Bell Laboratories. Included in the program were three systems that became part of TNDS—The Trunk Servicing System (TSS), The Trunk Forecasting System (TFS), and TDAS. The former two are now part of the Trunking System (TNDS/TK).[6] These three systems, together with Common Update as a supporting record base system, became known as Servicing and Estimating, under Bell Laboratories BIS development. Around the time Servicing and Estimating got its start, a stand-alone reporting system for No. 5 Crossbar central office data was developed by New Jersey Bell. It was standardized and

integrated into the TNDS downstream structure by AT&T and a mechanized interface to TDAS was implemented in 1972.

One of the earliest and perhaps most important architectural decisions was the placement of TDAS as a single point of entry for all data to the "downstream" world. In this position, TDAS served as a data warehouse—receiving data on its input side from many different data collection processes and distributing data on its output side to the many different users (both individuals as well as other systems). This basic architecture, established when most data collection was manually transposed and keypunched from Traffic Usage Recorder film outputs, has been extremely robust under the stress of rapidly changing data collection methods and data processing needs. On the output side, new downstream processing systems have been able to satisfy their needs for traffic data by simply placing orders, known as Traffic Measurement Requests (TMRs), on the TDAS data warehouse. And on the input side, new collection systems have been able to deliver their data by defining interfaces to TDAS.

In the mid-1970s, ICAN and LBS were developed and added to the TNDS downstream systems. ICAN was developed by the Network Planning area of Bell Laboratories, while LBS was developed by the BTL Servicing and Estimating organization. Shortly after ICAN was developed, all TNDS downstream systems (Equipment and Trunking) migrated to the Bell Laboratories BIS organization. By 1977 5XB COER, ICAN, and CSAR (which had also been developed by Bell Laboratories Network Planning) had been moved. Around this time, because of the growing size of the development organizations, the downstream systems were split into two BIS departments, along the lines of network user communities (trunking and equipment), to form TNDS/TK and TNDS/EQ as separate system groupings.

Combining all the downstream TNDS systems (TNDS/EQ and TNDS/TK) into one organization enabled common standards and features to be established including standard run control procedures (discussed in Section III), and a report distribution facility (discussed in Section IX).

With the expansion in the mid-1970s of both the number of systems as well as the number of operating telephone company installations, it became necessary to provide strong controls over the release of program modifications—both fixes and new features. Coordination was necessary from many standpoints. For developers, the incorporation of a new feature often required simultaneous changes to several systems. For system maintainers, keeping track of each of 20 operating company configurations of approximately 200 program modules required mechanized aids. And for each operating company, where hundreds of individuals depended on TNDS/EQ outputs to carry out

their daily jobs, there was a need to plan for new features and to be kept informed of the status of fixes to problems.

To cope with these needs that are common to large systems developments and large user communities, three major changes were instituted:

1. A coordinated annual release was established for all TNDS/EQ systems for packaging of new features. The scheduling of the release was set to occur during periods of slack telephone traffic so that any disruptions resulting from installing the new release would not interfere with processing busy season data.

2. A development environment was established based on the *UNIX** operating system, utilizing the Source Code Control System (SCCS) and Modification Request Control System (MRCS). These provided management control over different program versions and provided status tracking of trouble reports (modification requests).

3. A functional organization was established with separate requirements, development, test, installation, and consultation groups. Along with this organization, standard procedures and schedules were established for internal development activities. Also, greater emphasis was placed on formal documentation, both to support the internal functional organization and to support telephone company personnel in their planning and training for the annual TNDS/EQ releases.

Starting in the late 1970s attention was directed to the need for modernization of TNDS/EQ, in line with technological advances that had occurred over the past decade. Of all the decisions made, this one was the most difficult. Over the years, the major design focus for TNDS/EQ had been to keep up with network changes, to improve the efficiency of its heavy-duty data processing activities, and to streamline the flow of data from data collection systems, through TDAS to the downstream processing programs. For those TNDS/EQ functions associated with this processing of traffic measurement data, batch processing remained appropriate, even with current state-of-the-art computers.

Batch operations were awkward in areas involving direct user interfaces: inputs to update the record base (Common Update), and deliveries of output reports. On the input side, Common Update's weekly processing cycle introduced long delays in making record base changes and verifying their correctness. When errors occurred, several weeks could elapse before they would be corrected. On the output side, since the batch process does not retain outputs, telephone company personnel would often obtain output volumes far exceeding their needs "just in case" some information might be needed at a later date. It is evident

* Trademark of Bell Laboratories.

that an on-line, interactive interface with the user would significantly improve the environment. But in spite of these opportunities for improving human interfaces, net benefits were difficult to quantify. On the one hand, costs for terminals, printers, telecommunications lines, and computer on-line processing are easily quantified and generally are expensive. On the other hand, savings gained in personnel time from improved input error processing and less paper shuffling are very subjective.

Several activities were conducted to obtain a better view of benefits. In 1980, time-motion studies were conducted at two operating companies to obtain a view on how people, particularly clerical personnel, spent their time. The results of these two studies were consistent and showed that there were significant opportunities for savings through modernization. As a result of these studies, an experimental on-line system was installed in the second half of 1981 at a Chesapeake & Potomac Telephone Company network location and remained in place for about one year. Even though limited in scope, the on-line capabilities were received enthusiastically by the C & P Company personnel, and comparative time-motion studies conducted both before and during the experiment again showed that there were opportunities for savings. As a final activity, each Bell Operating Company (BOC) was requested to conduct its own cost-benefit study to determine whether on-line features would be economically beneficial. These studies were conducted during the summer of 1982 and resulted in the decision to proceed with an on-line system for record base inputs and output report viewing. This system has been named On-line Records and Reporting System (ORRS). The current plan is to provide these capabilities over the 1983 to 1986 time period, with first operating company application occurring in 1983.

## II. DATA COLLECTION

Because TNDS/EQ (specifically TDAS) needs to interface with multiple sources of data, a short discussion on data collection is presented here. Methods of data collection as well as types of data collected have changed greatly over the years. TNDS/EQ serves the vital function in the overall TNDS design of translating these multiple inputs to a common output format and identifying the data in terms of Bell System Common Language. The full list of measurement types supported by TNDS/EQ is given in Table I and were discussed in an earlier article.[7]

There was a need to collect telephone traffic data well before there were mechanized systems for collecting the data. In the early 1900s operators who manually completed calls recorded data on the frequency of calls to specific locations. This information was used for

Table I—TNDS/EQ measurement types

| Measurement Type | Description | Switching Machines Applicable |
|---|---|---|
| PC | Peg count | All |
| USG | Usage | All |
| OVF | Overflow | All |
| MTU | Maintenance usage | 1E, 2E, 5XB, 1XB, XBT |
| INU | Incoming usage | 1 *ESS*, 4A, 2 *ESS* |
| ATB | All trunks busy | SXS |
| LTB | Last trunk busy | SXS |
| CMP | Completions | SXS, 5XB, XBT, 1XB |
| IPC | Incoming peg count | 4E, DMS |
| MBC | Maintenance busy count | 2E, 3E |
| DGU | Detector group usage | SXS, 5XB, XBT, 1XB |
| RTS | Reroute to seizure | 4E |

establishing new routes between cities and resizing existing trunk groups. With the development of electromechanical switching systems, the collection of counts was mechanized. Counts were accumulated on mechanical registers. Periodically, the register readings were recorded by clerical personnel and compared with earlier readings to obtain a difference count. The manual recording of traffic register values was later replaced by automatically photographing these registers at predefined intervals.

During the 1960s, the rapid introduction of computer technology brought about three major changes in the data collection process. First, general-purpose computers were used to calculate the register differences. Second, the camera-register method began to be replaced by centralized data collection hardware, designed to collect measurements directly from the switching machines and to record the counts on magnetic tape for later processing by general-purpose computers. The first system of this type was the Traffic Data Recording System (TDRS). This system, implemented as a hard-wired computer, was soon found to lack the flexibility and capacity of the rapidly evolving minicomputer technology. The third major change in the 1960s was the development of electronic switching machines, which could perform their own internal data collection.

During the 1970s, the collection of real-time data introduced new capabilities to support office maintenance and network management. Also during this period, various support systems were centrally developed to run on general-purpose computers, providing features and flexibility applicable to all companies. These centrally developed systems standardized the format and labeling of traffic data so operating telephone companies could conveniently interchange common data.

As a result of this evolutionary process, data are collected many different ways in the Bell System. Table II lists the 35 different collection types currently supported. One of the major ongoing activ-

Table II—Traffic data input formats

| Data Collection Type | Description | Switching Machines Supported |
|---|---|---|
| R | Register readings | Electromechanical |
| D | Register differences | Electromechanical |
| S | Minicomputer scanned data | Electromechanical |
| A | Minicomputer accumulated data | Electromechanical |
| P | Minicomputer polled data | Electromechanical |
| I | Minicomputer individual circuit data | Electromechanical |
| X-A | 4A trunking data | 4A |
| X-D | 4A central office data | 4A |
| 1-C | 1/1A *ESS* switch continuous data | 1/1A *ESS* |
| 1-H | 1/1A *ESS* switch hourly data | 1/1A *ESS* |
| 1-W | 1/1A *ESS* switch weekly data | 1/1A *ESS* |
| 1-D | 1/1A *ESS* switch daily data | 1/1A *ESS* |
| 1-T | 1/1A *ESS* switch traffic separations data | 1/1A *ESS* |
| 1-E | 1/1A *ESS* switch extreme value data | 1/1A *ESS* |
| 2-C | 2/2B *ESS* switch continuous data | 2/2B |
| 2-H | 2/2B *ESS* switch hourly data | 2/2B |
| 2-W | 2/2B *ESS* switch weekly data | 2/2B |
| 2-D | 2/2B *ESS* switch daily data | 2/2B |
| 2-R | 2B *ESS* switch record verification data | 2B |
| 2-E | 2B *ESS* switch extreme value data | 2B |
| 3-C | 3 *ESS* switch continuous data | 3 *ESS* |
| 3-H | 3 *ESS* switch hourly data | 3 *ESS* |
| 3-W | 3 *ESS* switch weekly data | 3 *ESS* |
| 3-D | 3 *ESS* switch daily data | 3 *ESS* |
| 4-5 | 4 *ESS* switch trunking data | 4 *ESS* |
| 5-C | 5 *ESS* switch continuous data | 5 *ESS* |
| 5-H | 5 *ESS* switch hourly data | 5 *ESS* |
| 5-D | 5 *ESS* switch daily data | 5 *ESS* |
| 5-R | 5 *ESS* switch record verification data | 5 *ESS* |
| 5-E | 5 *ESS* switch extreme value data | 5 *ESS* |
| M-C | DMS-10 continuous data | DMS-10* |
| M-H | DMS-10 hourly data | DMS-10 |
| M-D | DMS-10 daily data | DMS-10 |
| M-E | DMS-10 extreme value data | DMS-10 |
| J-C | DMS-200 continuous data | DMS-200 |

* Trademark of Northern Telecom Inc.

ities of TNDS/EQ is to keep current with the collection environment as it continues to evolve.

## III. OPERATIONAL ENVIRONMENT AND CONTROLS

### *3.1 Introduction*

When TNDS/EQ and TNDS/TK were introduced in the early 1970s, the OTC computer operating environment was frequently not prepared to handle them. For some companies the TNDS systems were the first centrally developed BIS product to be installed. Interfacing with Bell Laboratories for reporting troubles, transmitting diagnostic data, and receiving corrections represented new concepts and procedures. Of even greater difficulty, each operating company had its own local rules for documentation and computer job operations. Conventions for computer job termination, job completion, and error detection, recovery, and restart varied. In some cases local procedures

required extensive manual verifications to ensure proper program job execution. From these early dissimilarities, a number of activities emerged. Working with the central development organizations and the operating telephone companies, the AT&T Information Systems organization established basic rules. And within the TNDS project, a very important activity was the establishment of "run control" standards for the TNDS/EQ and TNDS/TK systems. These controls were designed to detect error conditions as they occur, inform operations personnel of their existence, and inhibit further processing until the problem can be corrected. Once the problem is corrected, these controls permit simple restart with minimum reprocessing. Run control has been implemented in all TNDS/EQ systems and has been effective in minimizing the overhead associated with operating centrally developed software in remote batch production environments. The remainder of Section III provides details on the interface of run control with the operating environment.

### *3.2 Run control*

The attributes of the TNDS run control standard are described in this section.

#### *3.2.1 Load module execution sequence control*

To carry out a specific program function, program modules (known as load modules) that are delivered by Bell Laboratories to the processing sites are grouped into larger entities, called jobs. The execution sequence of jobs and of the load modules within jobs usually follows a predefined arrangement. Communications between programs within a job and between separate jobs are carried out by use of intermediate files stored on tape or disk. If through operational error, a defined job sequence is violated, run control detects and inhibits further processing until the proper sequence is resumed. Before run control was used, the jobs would have continued either to completion or until an error was detected. Because of the delay in detecting the trouble, location of its source and subsequent recovery was often difficult.

#### *3.2.2 Load module termination conventions*

When a load module termination occurs, either because it has completed its task or because of an abnormal condition, it is necessary to identify the reason for its termination. This information is needed both internally by other load modules executing in the same job and externally by the computer operations staff. The run control standard specifies use of a mechanism called the condition code to indicate normal load module termination and an "abort end" or abend code mechanism to indicate abnormal load module termination. The oper-

ations staff are thereby notified of abnormal terminations in a consistent way and job processing is stopped whenever a load module terminates abnormally. Similar to the previous attribute, early detection of a trouble condition is therefore improved.

#### *3.2.3 Restart aids*

A third aspect of run control ensures that restarts after a failure are handled as simply as possible. To help achieve this, run control:

1. Automatically selects the restart load module among the set of load modules within a job and alerts operations staff in the event of an error.
2. Provides a standardized interface to the operating system "checkpoint/restart" facility, which under certain operating conditions restarts the job at an intermediate point in the processing.
3. Automatically removes those intermediate files that must be recreated as part of the restart.
4. Automatically restores the contents of update files to their pre-execution contents before beginning the restart.

Figure 2 illustrates these concepts by showing a hypothetical job sequence and listing the items that must be performed to complete a restart.

#### *3.2.4 Logging system processing activities*

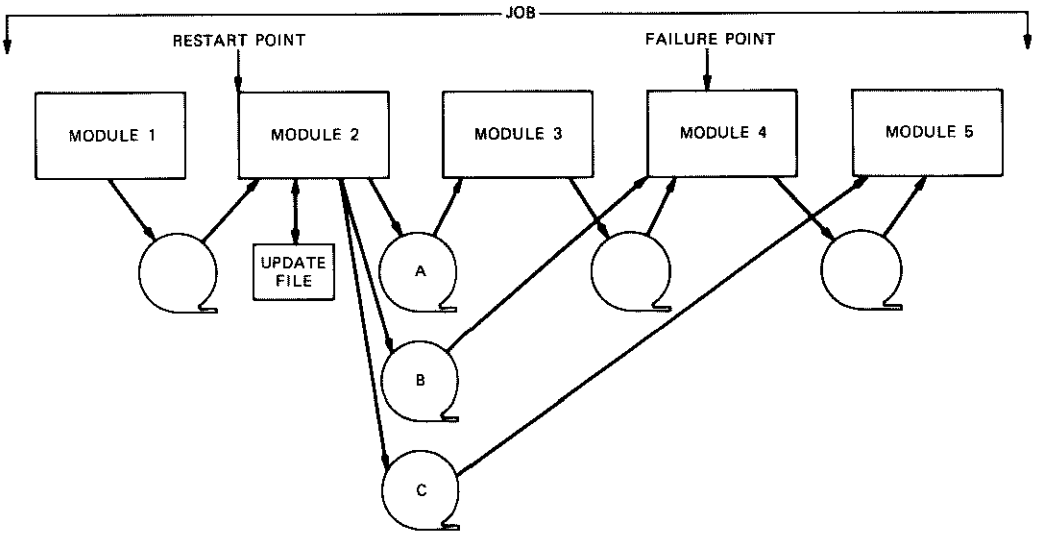
Run control also specifies standards for logging information about the processing activities of the system. This information is a summary of processing activities and has been very helpful later in diagnosing problems. The following information is logged for this purpose:

1. Indication of every module executed, recorded by date, time, module name, and unique version identification.
2. Indication of when the execution of the load module was completed, and whether such completion was normal or abnormal.
3. Identification of every file processed, recorded by its name and unique version identification. In addition, for sequential files, the number of records processed is recorded.
4. Recording of any anomalous conditions encountered by the program.
5. Recording of the Central Processing Unit (CPU) time and wall clock time needed to execute the load module.
6. Changes in value of any run control information.

The log can be readily transmitted back to the central development facility for detailed analysis.

#### *3.2.5 Automatic verification of file versions and sequential file record counts*

The fifth attribute of run control has improved the integrity of the




STATUS: SUCCESSFUL PROCESSING THROUGH MODULE 4, WHICH IS ABNORMALLY TERMINATED.

TO RESTART A JOB AFTER FAILURE:
1. DETERMINE RESTART POINT, SAY MODULE 2.
2. RESTORE UPDATE FILE TO PRERUN CONTENTS.
3. REMOVE INTERMEDIATE OUTPUT FILES (A, B, C) BEFORE INITIATING RESTART.
4. SPECIFY JOB RESUMPTION AT THE BEGINNING OF MODULE 2.

Fig. 2—Restart after failure.

file system. During a production cycle of TNDS/EQ, permanent files are modified to reflect current information. In addition to retaining the updated file, the operating system also retains the version of the file as it existed before the cycle so that a rerun can be performed if needed. The operating system provides a mechanism for using a family of files [called Generation Data Group (GDG)] to maintain information across multiple production cycles. The input to a program would be generation zero (0) and the output would be generation plus one (+1). At the completion of execution, the system records the new output as (0) and the original input as (−1). Thus generations are "rolled" from run to run. Figure 3 illustrates the use of GDGs.

Unfortunately, referencing a specific member of a GDG family requires manual intervention and can lead to accessing an incorrect member of the family. Run control detects such situations, alerts the OTC computer operations staff, and halts further processing pending correction of the problem.

Another file problem that can occur, particularly with the processing of sequential files, is loss of some of the records. A common way for this to occur is through inadvertent deletion of one of the middle reels of a multireel tape file. Run control detects this condition by maintaining knowledge of the number of records placed on the file at creation time versus the number actually processed when the file was accessed. Prior to run control, this verification was often done manually by computer operations personnel matching file counts.

### *3.3 Run control example*

The following example illustrates how the parts of run control work together to ensure correct processing. Consider a job consisting of four load modules (LM1, LM2, LM3, and LM4). Each module creates one intermediate output file that is passed to the next load module in

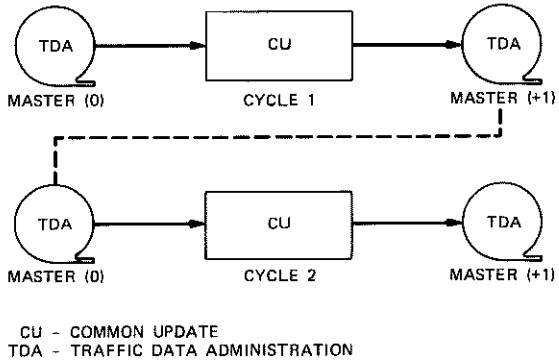



Fig. 3—Use of generation data groups.

sequence. Table III provides the contents of the run control external file. This file indicates that the current production cycle has been stopped, with only LM1 having completed execution. In this state, it is possible that LM2 might have been started, but was stopped because of an error such as a program problem, a hardware difficulty, or an incorrect tape mount.

Given the control information in Table III, when the job is executed again (the problem having been corrected), the following events occur as part of the run control functions:

1. Run control restart module executes.
   (a) A log entry is made showing run control module name, date, time, and version identification (id).
   (b) The name of the program to be executed next (LM2) is obtained from the run control external file.
   (c) The intermediate work file associated with LM2 (F2) is erased. This information is obtained from the table entitled "Association of Load Modules to Files Created" (see Table IV).
   (d) A condition code is issued to cause module LM1 to be skipped by Job Control Language (JCL) processing.
   (e) A log entry is made showing the run control module name, date, time, CPU time, and normal completion.
2. Load Module LM1 is skipped by the operating system JCL processor.
3. Load Module LM2 executes.
   (a) A log entry is made showing module LM2 name, date, time, and version id.
   (b) Run control verifies that it is proper for LM2 to execute by comparing its name (LM2) as recorded internally within the module with the contents of the run control external file.
   (c) LM2 opens file F1, extracts header information, and compares

Table III—Contents of run control external file after abnormal termination of a job

| Module | Execution State | File | Date Last Executed | Time Last Executed | No. of Records |
|---|---|---|---|---|---|
| LM1 | Executed | F1 | 8-4-81 | 13:00 | 1000 |
| LM2 | Not executed | F2 | 8-3-81 | 09:30 | 2500 |
| LM3 | Not executed | F3 | 8-3-81 | 09:45 | 5000 |
| LM4 | Not executed | F4 | 8-3-81 | 09:50 | 200 |

Table IV—Association of load modules to files created

| Module | LM1 | LM2 | LM3 | LM4 |
|---|---|---|---|---|
| Files Created | F1 | F2 | F3 | F4 |

it against the audit values in the run control external file recorded when file F1 was created. Let's assume that an earlier version of F1 was mounted and LM2 reads from the file header "8-3-81 9:00." This does not match the time and date logged in the run control external file "8-4-81 13:00." The error condition, file name, and audit values are logged.

(d) Execution of LM2 is terminated by an abend.

4. The operating system JCL processor flushes the job.

This scenario illustrates run control processing under an error condition. If file F1 had not failed its audit, then processing would have continued with modules LM3 and LM4 and actions similar to those recorded for LM2 (without the abend) would have occurred.

Program recovery prior to implementation of the controls typified by this example was often difficult, requiring extensive manual intervention. Run control software has provided dual benefits to the TNDS user community: first, it has reduced the level of direct involvement needed for centralized maintenance personnel to provide operational support; and second, it has made available better diagnostic information for handling problem situations to both the OTC operations personnel and to the central maintenance personnel.

## IV. COMMON UPDATE (CU)

### *4.1 Introduction*

Section IV describes the details of the data processing functions carried out by TNDS/EQ components. Because Common Update (CU) provides record base support to a number of the other components, it is described first. It is an example of the integration that has been achieved within the TNDS/EQ and TNDS/TK systems. In particular, CU fully supports the TNDS/EQ systems TDAS and LBS, as well as the Report Distributor facility, and provides partial support of No. 5 Crossbar Central Office Equipment Reports (5XB COER) and ICAN. It also fully supports the TNDS trunking systems TSS and TFS.

Since CU supports both switching equipment and trunking support centers, it has been subdivided into independent subsystems, CU/Equipment and CU/Trunking. This division allows an operating company to have the flexibility of providing multiple installations of TNDS Equipment (TNDS/EQ) systems, including CU/EQ, while maintaining a centralized TNDS Trunking (TNDS/TK) system, including CU/TK. The remainder of this section will focus on CU/EQ, hereafter referred to simply as CU. The article on trunking systems (Ref. 6) discusses CU/TK.

To support the Network Administration personnel who maintain the record base, CU:

1. Validates input record base transactions according to predetermined parameters and ranges, and performs intrarecord and interrecord validations.

2. Outputs activity or addenda reports after each CU cycle to provide updated record base status.

3. Provides messages and reports on transactions found in error.

4. Responds to input requests for full record base listings of a given report type.

CU executes as a series of jobs or runs. It is designed to run on an "as needed" basis, i.e., whenever transactions are created to add, change, or delete information in the record base. In most operating companies, CU executes on a weekly basis with all other dependent TNDS systems running subsequent to its successful completion.

The following four support run groupings are associated with the mainline runs of CU:

1. New TNDS/EQ installation—One-time runs needed during an initial installation of the system at a new site.

2. Release conversion—One-time runs to convert databases resulting from implementation of new TNDS/EQ features in a major new release.

3. Switching machine installations or conversions—Runs to provide CU record base support of changes in the switching or data collection environment. These include automatic database generation and conversions associated with new switching machine generics.

4. Recovery and restart—Runs that allow the database files to be restored to their original content in the event of an abnormal termination.

### *4.2 System inputs*

CU activity is driven by batch input transactions. Each transaction is identified by a unique three-digit number, referred to as the transaction code or type. CU supports 50 different transactions, grouped into series by the TNDS system that they support. For example, transactions in the 700 series (742, 743, 750, 751, $\cdots$) support TDAS, while the 600 series transactions (600, 601, 620, $\cdots$) support LBS. Common information, which is used by multiple systems, is stored in files called tables and their transaction series number is 100 (e.g., 100, 105, 180, 181, $\cdots$).

To accommodate the current CU batch inputs many of the operating companies have developed their own front-end input processors. These systems typically are minicomputers or intelligent terminal configurations that provide screen formats, perform on-line intrarecord validations, and allow geographical dispersion of input sources.

As a future CU replacement, the On-Line Records and Reporting

System (ORRS) discussed in Section 1.4 is being developed. It will have all the benefits of the current front-end systems, as well as immediate interrecord validations, on-line record base inquiry, and on-line report viewing.

### *4.3 System outputs*

#### *4.3.1 General*

There are two categories of outputs from CU, record base files and reports. The files support the other TNDS systems, which rely on CU for their record base. The reports support the network administrator's job of maintaining the record base so that it accurately reflects the physical telephone network.

#### *4.3.2 Record base files*

User data are entered into CU to establish an up-to-date description of the network. Information relevant to a particular processing system is isolated into one or more record base files. For example, the information that describes the data collection environment needed by the TDAS system is stored in the Traffic Data Administration master file, while the information that describes a switching entity for load balance purposes is stored in the LBS master file.

As updates are introduced to the system in the form of add, change, and delete transactions, the current version of the master file is updated to produce a new version of the file. Generation Data Groups (GDGs) are used to help simplify the updating of these files and the maintenance of adequate backups, as described in Section III.

#### *4.3.3 Reports*

CU provides a number of reports to help maintain the record base. The two main categories are error reports and reference listings.

Error reports are issued whenever CU determines that an input transaction is invalid. In addition to these reports of specific errors, other reports provide record base status and error statistics. A number of error conditions do not result in complete rejection of the transaction. Rather, to reduce subsequent data entry, the data are posted in the record base, and an error indicator is turned on. Record base status reports inform the user after each run of all records that remain invalid and require correction. Operating telephone company managers receive error statistics by transaction type and origination, which enables them to analyze error rates and spot training deficiencies. Errors that predominate in a particular geographic region of the company can thereby be recognized and corrected with additional training. Error statistics also enable managers to recognize deficiencies in system documentation.

Reference listings provide a readout of the record base. These reports take the form of addenda and full listings. Depending on the particular report, they can be generated in various formats. For example, the Data Collection Environment report can be sorted and printed three different ways, depending on the needs of the particular user.

As we described in Section IX, CU reports are directed to the report distributor for printing or microfiching and distribution.

### *4.4 CU processing flow*

The processing flow of CU is conceptually quite simple. First, inputs are sorted by transaction type. Invalid transaction numbers are directed to an error file. The sorted transactions are then validated, starting with the 100 series transactions.

Following basic intrarecord validations, the transactions are segregated by series and directed to files for subsequent processing by individual master file update modules.

These modules update the following master files:

- Circuit group master and translation files (CU/TK module)
- Load balance master files
- Traffic unit master file (CU/TK module)
- Traffic data administration and traffic measurement requests master files
- Common access tables.

Errors detected during the update process are directed to error files. Activity and demand requests result in master file records being directed to report files. After all the files have been updated, report and error records are sorted and formatted into standard output reports.

The final step in the CU mainline process is to generate backup tapes for CU table files. These tables are all random access files updated in place in response to their corresponding input transactions. Since destruction or loss of any of these tables would be very detrimental to the entire TNDS downstream operation, the files are automatically copied during the main cycle of the system.

## V. TRAFFIC DATA ADMINISTRATION SYSTEM

### *5.1 Introduction*

As shown in Fig. 1, the Traffic Data Administration System (TDAS) is the first of the TNDS/EQ systems to process traffic data. The data are received from upstream collection systems and can be processed immediately, but typically they are processed in batches in weekly cycles. The traffic data shown in Fig. 1 are input to TDAS from a number of different sources. Among the "Other Forms of Data Collection" are vendor-supplied data collection systems and keypunched

data collected in a camera/register environment. In response to specific requests for data from the users of various downstream systems, TDAS formats the data as required by those systems. As we described previously, TDAS acts as a centralized warehouse and distribution facility for traffic data.

The primary objectives of TDAS are to:

1. Accept traffic data from any standard data collection source.
2. Edit, validate, and adjust data.
3. Identify the origin of data, i.e., what the measurements represent.
4. Summarize and, if necessary, store the data, to satisfy weekly requests.
5. Transform the data into standard formats that are acceptable to the downstream systems.
6. Provide traffic data reports that primarily satisfy special study requests.

Although Network Administration is the major direct user of TNDS-collected traffic data, there are many other users. Each has different needs and, therefore, may require different subsets of the data in different physical forms and formats. The output reports of TDAS range from machine-readable files for mechanized interfaces to paper and microfiche reports.

Within the total data provisioning process, TDAS is positioned to control the data flow from the collection sources to the end users.

The remainder of this section describes how TDAS functions and its outputs, inputs, and processing flow.

### *5.2 TDAS outputs*

#### *5.2.1 General*

TDAS supplies data to a large audience of traffic data users. Some of these are supported by computer systems with which TDAS interfaces via magnetic tape files, while others use reports created directly by TDAS. Thus, the major outputs of the system are interface files and reports.

#### *5.2.2 Interface files*

A standard interface file has been defined for each distinct system with which TDAS interfaces. This definition contains file characteristics and record formats. Associated with the record format is a detailed definition of every field within the record and its data attributes. File characteristics include definitions such as file format and block size. As user requests are processed, TDAS directs the appropriate data to the corresponding physical file, which is typically a tape but can be any IBM-compatible secondary storage device. At the completion of a weekly TDAS run, the interface files are transported

to their respective data processing sites for further processing by the downstream systems. TDAS has the continuing requirement to ensure that the output formats remain stable as future changes occur in the data collection process. Note that because each class of end user has a distinct and different interest in the data, the formats serving these end users are different. For example, the format of the Trunk Servicing System (TSS) interface file is different from the Load Balance System (LBS) interface file.

#### *5.2.3 Reports*

TDAS provides data summary reports for users of traffic data not served by a downstream data analysis system. For these outputs, TDAS can perform some formatting and processing functions on the data to assist the user. These include editing, validating, adjusting for improper sample (or scan) rate, summarizing, and identifying with the appropriate Equipment Measurement Code (EMC) or Trunk Group Serial Number (TGSN) designation.

In addition to reports oriented to end users, TDAS also produces a series of reports to aid in the data provisioning and tracking process. These include:

1. Data edit error reports, tailored to each data collection source,
2. Data log reports, which relate data requests to data availability, and
3. Input data tape processing reports, which account for the many input tapes processed by TDAS in a given week.

All of these reports are written to a report file for input to the Report Distributor (Section IX) for final distribution and printing (or microfiching).

### *5.3 TDAS inputs*

#### *5.3.1 Record base*

To achieve its objectives, TDAS relies on CU for maintenance of its record base. The TDAS record base consists of two primary inputs—a definition of the data collection environment, called the Traffic Data Administration (TDA) master file, and a list of all user requests, called the Traffic Measurement Request (TMR) master file. Using these two basic inputs, together with traffic data itself, TDAS can treat the data provisioning job as a basic order inventory problem. Orders, or TMRs, for data are compared with the inventory of traffic data. Based on prescribed rules and the use of the TDA master file for identification purposes, the orders are filled by TDAS.

#### *5.3.2 Traffic data*

Concurrent with the evolution of the switching network has been

the evolution of data collection hardware. As the means of collecting data have changed, so has the format of the data itself. For example, data that originates from the old camera/register environment, which still serves some electromechanical switches, are in the format of keypunched card input. Another old form of data collection is 1 *ESS* switch data in the form of paper tape. All other modes of data collection are in machine-readable magnetic tape form but in many different formats. *ESS* switch data differ in format from electromechanical data; 4 *ESS* switch data are different from all other *ESS* switch data. Even the same switch can have different data formats based on the software version of the switching machine. One of the important functions performed by TDAS is to mask these format differences and present the data to the end user as though they were collected in a common manner.

Based on its format, the traffic data are read into the appropriate TDAS data entry module for editing, sorting, and converting them into a common format for further processing. The following input data formats are currently defined by TDAS:

1. Keypunch data in support of camera/register data collection.
2. Paper tape data in support of 1 *ESS* switch paper tape.
3. ASCII-encoded data in support of older EADAS machines; the Peripheral Bus Computer (PBC), which collects 4A toll machine data; and various outside vendor collection machines.
4. 4 *ESS* switch formatted data.
5. Binary-coded data in support of current EADAS machines and various outside vendor collection machines.

The last format, the Binary Interface, is a flexible data format intended to standardize the data coming from current stored program control systems. This interface has been documented as a General Trade *Technical Advisory* and has been distributed by AT&T through the United States Independent Telephone Association (USITA). TDAS will process data collected by any data collection system meeting the interface.

#### *5.3.3 System options*

To satisfy the variability in data collection environments and other factors that make each operating telephone company unique, TDAS accepts a number of user options, which are stored in its parameter file. One such option is the expiration date offset value for each mode of data collection. It indicates the maximum number of days from time of collection that the system will wait for all data to be received for a given type of Data Collection Unit (DCU). If all data are not received by the calculated expiration date, the TMRs will be processed

with whatever data are currently available from that collection machine. This option is particularly important for camera/register collection, since each data item for this type of DCU is received as a separate and independent input. On the other hand, data collected via mechanized methods, such as EADAS, are usually received on a single input tape.

### *5.4 TDAS processing flow*

#### *5.4.1 General*

A complete execution of TDAS, called a cycle, is performed on a weekly basis. To simplify its operation and scheduling for the computer operations personnel, TDAS is divided into a number of discrete jobs.

The mainline TDAS programs (that is, those that are run every production cycle) are divided into three functional entities: CU/TDAS File Interface Process (FIPS), Data Entry, and Data Analysis. FIPS extracts the records required from CU and formats them for efficient TDAS processing. It is a multistep run and is executed once per cycle. Data Entry serves as the front-end process of TDAS. There are five Data Entry modules (hereafter called editors), one for each of the data entry formats, as listed in Section 5.3.2. Their frequency of execution depends on the data collection environment and computer scheduling of the individual OTC. Some OTCs run daily editors, while others batch the daily data tapes into a weekly editor run. TDAS allows both or any combination in between. The editors prepare the traffic data for the Data Analysis programs, which are executed once each production cycle. This overall TDAS processing flow is shown in Fig. 4.

The primary responsibility of TDAS is to process traffic data, and that it does—in excess of 50 million data items per week in an average size OTC. It is this magnitude of data that makes TDAS the longest running system in TNDS/EQ. Run time, therefore, has been a continuing area of concern.

One of the techniques that were introduced to improve processing times is data screening. Soon after TDAS was introduced it was recognized that much of the data collected was not needed by any end user. Of the 50 million data items received by TDAS, only about 25 percent are sent downstream—the rest are excess! The varying data requirements of the different end users may result in upstream data collection for almost 24 hours of every day, even though only a small portion is needed for the entire period. Figure 5 shows in a typical example the individual needs of the downstream systems that TDAS serves and the resulting collection of unrequested data. Because of the needs of one downstream system, it becomes necessary to collect the full increment of 1000 data items for 24 hours. The function of data

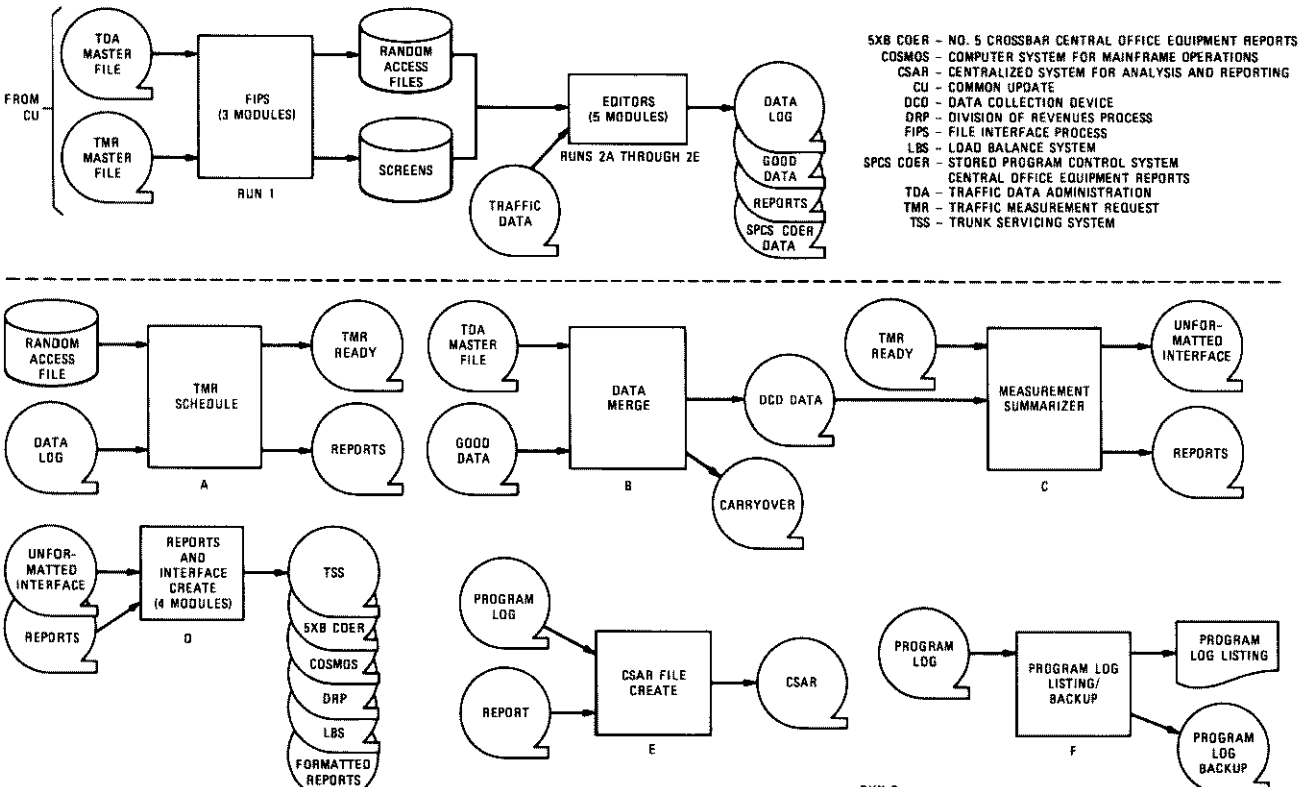



Fig. 4—TDAS processing flow.

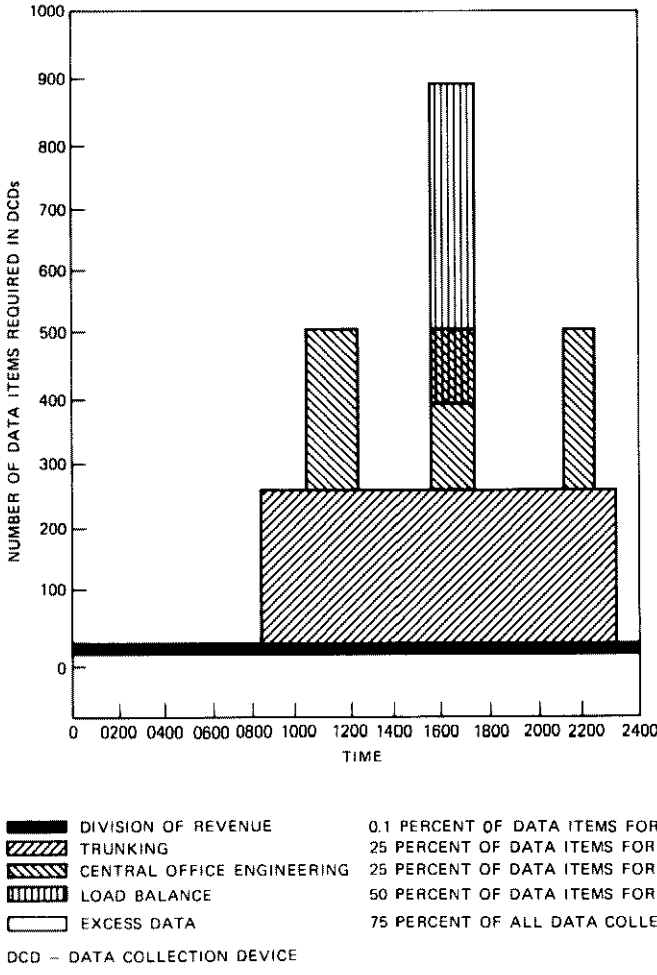



Fig. 5—Example of excess data.

screening, then, is to discard excess data at the earliest possible point in the process (the editors). It should be noted, however, that the function of screening itself incurs an expense, and for that reason it has been implemented only in those editors that process large volumes of data. The camera/register data editor, for example, does not perform automatic screening because much of the screening is done concurrently with the manual keypunching activities.

The remainder of the discussion on TDAS will focus on each of the runs that encompasses the mainline system. A number of additional support runs that perform peripheral functions, like file recovery, will not be discussed here.

### 5.4.2 File Interface Process

To perform its processing, TDAS requires certain information from the Common Update TDA and TMR master files to be accessed randomly. FIPS, then, is the first run in a TDAS cycle and is responsible for converting the TDAS master files into various random access files for efficient processing. Some files use the Virtual Sequential Access Method (VSAM), where the file is created sequentially based on a defined key and the access method routines retrieve data records randomly using indexes and pointers. Other files use the Basic Direct Access Method (BDAM), where the file is created and randomly accessed using a calculated key.

FIPS creates a two-level screen with which the editors screen the data. The programs first relate TMRs to Data Collection Units (DCUs) on a day and time basis to indicate for a given day and time whether any requests exist for data from the DCU. This first screening level is accomplished by creating a VSAM file keyed by DCU and date. The second level of screening is an interrogation of a BDAM bit map file consisting of bits for each data item, called Data Collection Device (DCD) within the data collection unit. Each bit indicates whether the data item (DCD) has been requested (1) or not requested (0). The VSAM record for a DCU/date contains calculated keys that point to maps of 1000 DCDs in the BDAM file for particular times of the day. Figure 6 shows this screen creation process.

FIPS is run following the weekly CU cycle so that the latest record base changes can be captured for the current TDAS cycle.

### 5.4.3 Data entry

After FIPS has completed its run successfully, the editors as listed in Section 5.3.2 begin execution. Each of those five editors is run separately. They can be executed in any sequence but not concurrently. An editor is responsible for detecting errors, and screening, sorting, and reformatting the acceptable data into a common output format. The editor also maintains a log of all input data. To account for lost data and the status of individual input data tapes, each editor produces a Data Errors report and a Tape Processing report. In addition, if data are lost because the collection machine goes out of service, the cause of that loss is indicated on the data tape when the collection machine returns to service. This indication is displayed on the Tape Processing report by TDAS and the same information is also forwarded to the Centralized System for Analysis and Reporting (CSAR)[2] for its total system data tracking responsibility.

The editors that process data from Stored Program Controlled Systems also strip off a portion of that data and write it to a file for transmission to the centrally located SPCS COER[1] system. The

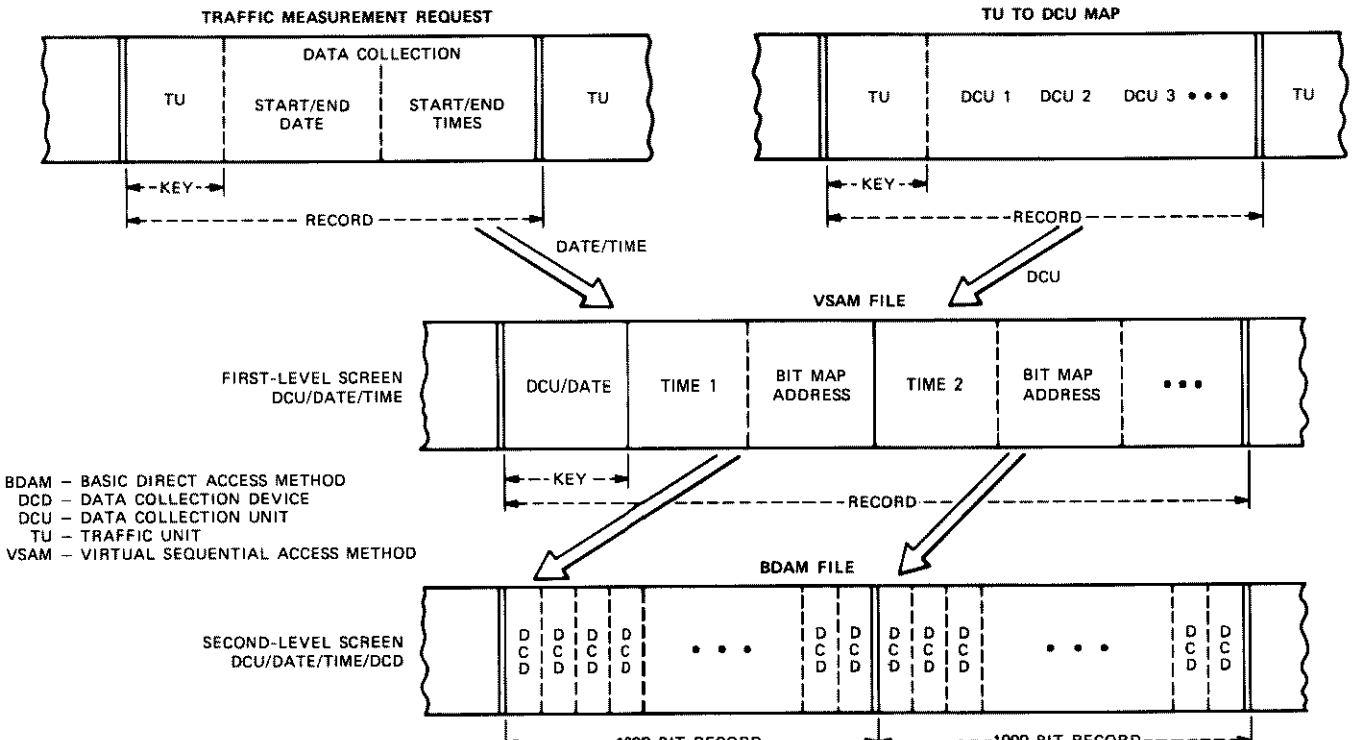



Fig. 6—Data screen creation.

TDAS-to-SPCS COER interface is the same as the collection-machine-to-TDAS interface. As a result data directed to SPCS COER are output by the editors (after being screened). Some OTCs that input daily tapes to the editors transmit the daily outputs to SPCS COER for overnight processing.

A problem inherent in the current collection-machine-to-TDAS interface is the mechanics of tape transport. In a typical OTC, 30 to 50 tapes are transported between these systems weekly. The administration and transportation costs as well as loss of tapes incurred under this arrangement have stimulated investigation into a direct collection-machine-to-TDAS data link. A future release of these systems will include the capability to transmit the data over either a dedicated or dial-up link.

#### *5.4.4 Data analysis*

After submitting all the desired data to the TDAS editors, the next and final run is Data Analysis. It is a multistep process that performs most of the work of the system.

The first step, TMR Scheduling, determines which TMRs are ready for processing in the current TDAS cycle. It compares the list of TMRs to the available data, and then based on rules and options governing calculation of expiration dates, releases whatever data are available.

The Data Merge module then merges the data with identification information from the TDA master file. The data that remains (for example, data that was input to TDAS for a partial week and does not fully satisfy the TMRs on file), along with the unexpired TMRs, are held for processing in a subsequent cycle.

With TMRs now ready for processing, the Measurement Summarizer module verifies that the collection equipment was operating correctly at the time the data were gathered. As an example, usage data are expected to be collected at a sampling rate of 36 scans per hour. If the rate falls outside acceptable limits, between 32 and 37 for electromechanical switches and exactly 36 for stored program control switches, it is marked invalid and not passed downstream. Otherwise the data are considered acceptable. If the user requests that the data be adjusted, they are then adjusted to the hourly (36 scan) equivalent. The data are then summarized into intervals as requested on the TMR and written to the appropriate interface and report files. Throughout this process statistics are compiled on invalid data and missing data.

The interface files are then sorted and formatted by the Interface Create modules for delivery to the downstream systems. Next, the Reports Create modules format the various TDAS-generated reports,

and the CSAR Create module formats the CSAR interface file. Finally, the run ends with system cleanup functions.

## VI. NO. 5 CROSSBAR CENTRAL OFFICE EQUIPMENT REPORTS SYSTEM (5XB COER)

### *6.1 Introduction*

The No. 5XB COER system was one of the earliest TNDS component systems developed. It was designed to assist the network engineers in forecasting future equipment requirements in 5XB central offices. 5XB COER summarizes, over a one-year period, traffic usage data for each individual 5XB equipment component. This interval of time is referred to as the engineering year. Maintenance of this full-year history is one of the distinguishing features of 5XB COER, as compared with other systems, such as EADAS, which provide quick looks at the traffic data over short intervals of time (e.g., weekly).

The summarized data are organized into a report, referred to as the Machine Load and Service Summary (MLSS). This report is the key output of the system. The year's worth of data summarized in this report, combined with past years' data and other forecast information, enable the engineer to determine future equipment requirements.

The following sections will discuss the MLSS reports as they apply to the 5XB engineering function, and will discuss the responsibilities of Network Administration in ensuring the quality of those reports. Finally, a description of the software implementation is also provided.

### *6.2 MLSS reports in support of engineering*

"Load" and "Service" in the title describe the content of the MLSS report: load, referring to the traffic carried by a particular component; and service, the measure of performance associated with that particular load. The engineer determines what equipment will be needed for the probable level of service to meet a projected load.

Data are summarized in High Day (HD), Ten-High-Day (10HD), and Average Busy Season (ABS) results for a Time-Consistent Busy Hour (TCBH). TCBH implies the single hour (e.g., 9 a.m. to 10 a.m.) in which the particular equipment component (e.g., originating registers for *Touch-Tone** telephone) consistently experiences its heaviest traffic load. Within this busy hour, HD, 10HD, and ABS can best be described with the help of Figs. 7 and 8. Assume in these figures that dots represent traffic data values of load on which a component is to be engineered. Figure 7 plots these values for each day (typically only business days) over the 12 months of the engineering year. As expected,

* Trademark of AT&T.

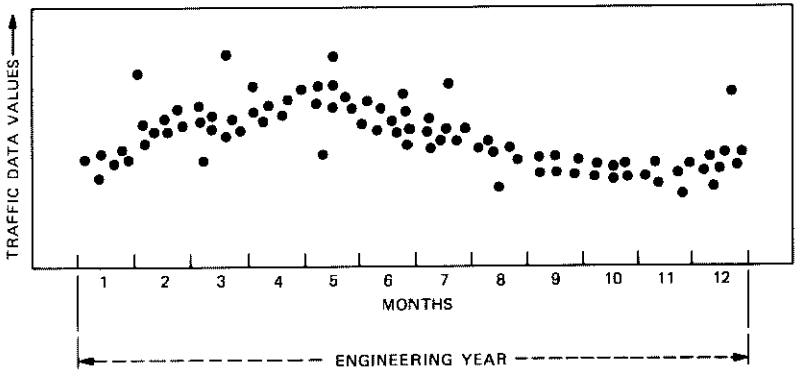



Fig. 7—Typical traffic profile.

some days have higher than normal values, and there are extended periods where the general average is higher than the rest of the year. Figure 8 shows this data summarized in terms of monthly averages for each of the twelve months, and also shows the selection of the ten highest days and three busiest months. The average of the ten highest days forms the 10HD value, and the three busiest months the ABS value.

The common format of the MLSS report used for all 5XB components, shown in Fig. 9, incorporates all the engineering requirements. Header information identifies details such as office, component, and time span. The columns (left to right) list the date, key column value* (used to engineer the component) and support columns (significant calculations such as percent overflow/blocking, holding time, percent occupancy/capacity, along with the actual values used in their calculations). The Ratio to Busy Season (RA/BS) and Fitted Ratio (FIT RAT) enable the engineer to determine if a particular reported high-day key column value is a yearly recurring event. High days typically not expected to recur (e.g., heavy telephone traffic due to a severe storm) should be excluded from the engineering base. RA/BS is the ratio of the key column value to the ABS value. FIT RAT is a statistical quantity relating this key column data point to the total year's values, assuming a gamma distribution. The RA/BS value should be very close in value to the corresponding FIT RAT value if the data point is in fact a legitimate recurring high day. The rows display the highest day (HD), the next nine highest days, the 10HD average, the next 15

* Typically, this quantity is a usage measurement normalized with respect to an office parameter such as main stations (MS) or trunks (TRK), depending on which is applicable.

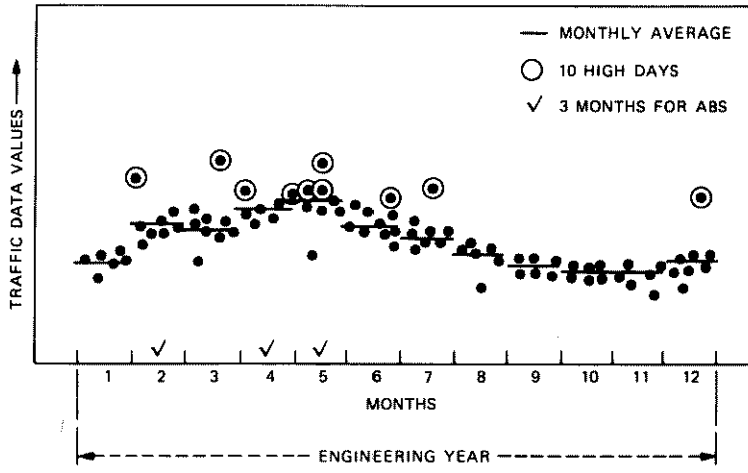



Fig. 8—Summary of traffic data.

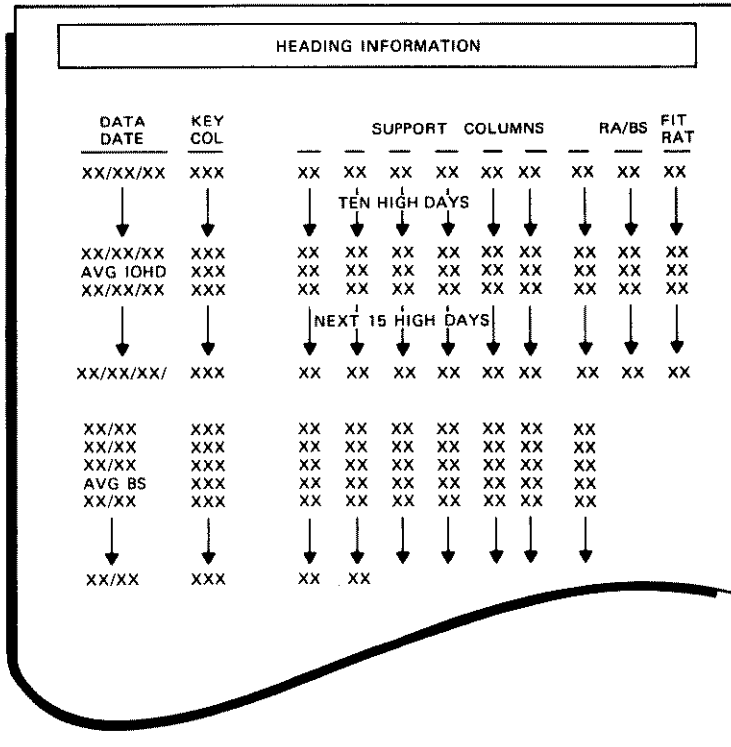



Fig. 9—MLSS report format.

highest days (displayed for their potential use as one or more 10HD replacements), the three highest monthly averages, the ABS calculation, and the monthly averages for the remaining nine months.

As traffic data are received by the system, typically in weekly batches, the MLSS report is automatically developed for each component and hour requested.* Among the components are line link frames, markers, originating registers, outgoing senders, incoming registers, trunks, and transverters.

### *6.3 Network administrative support*

The network administrator is responsible for supplying a complete, accurate, and timely set of MLSS reports to the engineering staff. This job function fits in with the overall responsibility for monitoring the daily performance of the central office to offer the best quality of service possible with the currently installed equipment. To perform this function, equipment utilization information is gathered by coordinating the collection, report generation, and analysis of the traffic data.

To achieve the prime system objective of producing usable MLSS reports, the No. 5XB COER system supplies extensive aids to the network administrator. Among these are:

1. Busy hour determination
2. Data validation
3. Data surveillance
4. Data management.

Each will be described in the following sections.

#### *6.3.1 Busy hour determination*

Integrated into the 5XB COER system is the ability to perform a Busy Hour Determination (BHD) study. A BHD study identifies the busiest hour(s) for each engineerable component. Those hours will then be the ones reported in the next year's MLSS. Any particular component may have many busy hours contending for the heaviest load. Busy hour contention normally results from heterogeneous traffic (e.g., business versus residential). The selection of the correct hour(s) is critical because data collection and processing for the coming engineering year will be based on that decision. Selection of the wrong MLSS hour(s) can result in invalid engineering decisions.

A BHD study is conducted at least once a year, usually just prior to entering the busier part of the year. The study period typically spans

* The system will process up to five hours per component. In general, only the busy hour is studied, but one or two side hours may also be studied if uncertainty exists concerning the busy hour.

two weeks. For this study period, up to twenty-four hours per day of data can be collected in half-hour increments. The half-hour totals are summed by TDAS into hourly totals, ending either on the hour or the half hour or both. This results in up to 48 traffic data totals per day for each component measured.

Once all the study data have been collected and processed, a BHD report is produced for each component, as shown in Fig. 10. The key and support columns in this report parallel the MLSS report.

The system-selected busiest hour (based on key column) is identified with a "BH" to the left of the hour. Two additional columns, RA/BH and DAYS = BH/#, are computed to help the user understand and interpret the results, as well as to assess the degree of confidence in selecting the busy hour. RA/BH is the ratio of the individual hour to the selected busy hour. This column reveals hours that are close in load to the selected busy hour. The next column displays the actual number of days in which that hour was the busy hour (DAYS = BH), compared to the number of days that had valid data (#).

### *6.3.2 Data validation*

In order for the summaries in the MLSS reports to help the engineers make equipment forecasts, the basic measurement data must

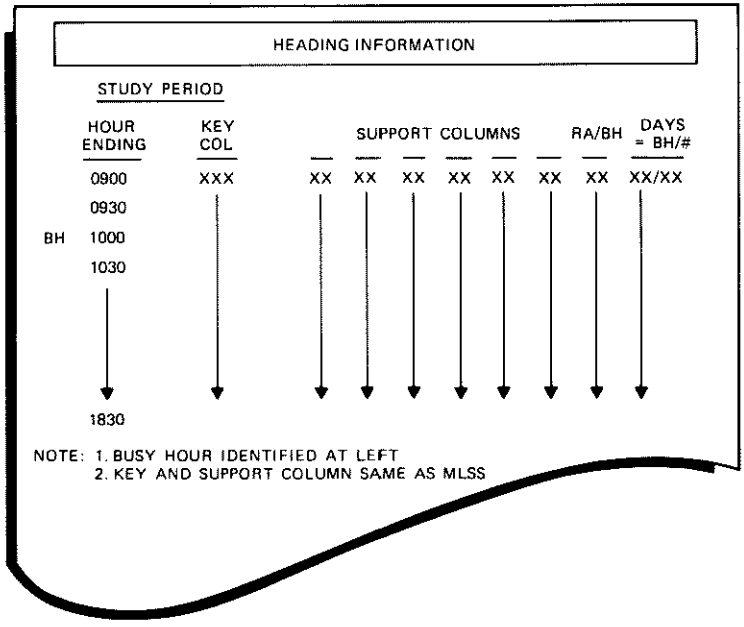



Fig. 10—Busy hour determination report format.

be a true representation of what occurred physically in the central office. An extensive set of data validation tests and associated failure reports are therefore provided.

The tests can be classified into four major categories, and they are usually performed in the following order:

1. Sanity checks
2. Peg count balance tests
3. Holding time tests
4. Historical tests.

The order of performance is based on the relative "strength" of the test. For example, sanity check failures carry a 100-percent confidence factor that the tested data are in fact incorrect. At the other end, failures in the historical tests should be investigated further before the data are declared invalid.

***6.3.2.1 Sanity checks.*** These checks are simple and detect impossible conditions. Two examples of test failure conditions for originating registers (OR) and incoming registers (IR) are:

1. OR total usage > 36 ccs × no. of ORs in group
2. IR maintenance usage > total usage.*

***6.3.2.2 Peg count balance test.*** This type of test defines certain relationships that must exist among several peg count (PC) measurements. For example, the Dial Tone Marker (DTM) peg count (which is scored on each call origination) should be equal to the sum of those available peg counts that indicate call disposition. In this case there are three dispositions:

1. Call successfully seized an originating register, causing the OR peg count to score.
2. Call encountered an "All OR Busy" (AORB) condition, causing the AORB peg count to score.
3. Call was not able to find an idle path through the switch, thereby causing a "Dial Tone Marker Failure to Match" peg count (DTM FM PC) to score.

The form of this test would be:

$$\text{Lower Range} \leq \frac{DTM\ PC}{OR\ PC + AORB + DTM\ FM\ PC} < \text{Upper Range}.$$

Because of the complexity of the 5XB equipment interrelationships, these types of tests often involve a large number of data items. Exact balances generally cannot be achieved because of the lack of complete measurements and the time skew for recording the various peg counts. Therefore, a range is defined around the theoretical value of 1.0.

* Total Usage means traffic plus maintenance usage.

Normally the PC balance test is performed on a component group. However, the precision of the test frequently can be refined by applying diagnostic tests to each individual in a failing group.

*6.3.2.3 Holding time tests.* Based on the specific equipment design, the average Holding Time (HT) of a component, e.g., Outgoing Senders (OSs), should be within a certain range of theoretical values. The range is usually broad to account for variable central office characteristics. For any given central office, the user can override and redefine the range test to strengthen the validation for that office.

*6.3.2.4 Historical reliability test.* The Historical Reliability test is used for data that on a statistical basis fall within a range based on past history. An example of this is the office calling load per main station, CCS/MS. The value of CCS/MS will vary according to telephone calling fluctuations but usually within a historically determined range. For these cases, tests must be designed based on past data by using a range tracking procedure.

There are many statistical techniques used for this type of validation. The method implemented by 5XB COER has produced good empirical results, was simple to implement, and required little storage. Three parameters are stored for this test: an estimated value based on past history, a variance, and a residual. The residual helps to adjust the acceptable range as the data value changes. This is particularly important in minimizing false alarms as busy seasons are encountered. The valid range for the next data point is the estimated value plus or minus one standard deviation. Based upon a comparison of the residual to the standard deviation, additional adjustments are applied to the estimated value.

### *6.3.3 Data surveillance*

The ability to detect subtle patterns in equipment performance can mean the difference between the correct diagnosis of an equipment malfunction and the incorrect conclusion that an office is underprovisioned. Real-time surveillance is provided by systems such as EADAS, which produce exception reports based on exceeded thresholds. To perform a comprehensive analysis of total office performance, component information for the entire office must be available for the same hour. 5XB COER produces an optional report package consisting of detailed component data, validation failures, and raw measurement listing for the entire office for one or two individual hours from a week's worth of data. Network administrators select the hour(s) to be reported, and the criteria on which the system will select "the highest hour(s)" from those received to process. One of four criteria may be selected for each hour to be reported: Terminating Peg Count, Originating Peg Count, Originating plus Terminating Peg Count, Originat-

ing plus Terminating Usage. The selected reports are produced each processing cycle.

Another data surveillance feature of 5XB COER supports the annual MLSS product. As the system receives and processes the MLSS data, normally weekly, it produces output reports titled Weekly Machine Administrative Reports (WMARs) and Monthly Machine Administrative Reports (MMARs) for each component and hour requested. WMARs are optional reports. The format of these reports parallels the associated MLSS reports, except that it displays daily data for each day of the week or month rather than 10HD and ABS data. This permits a review of the daily results to ensure their appropriateness for later inclusion in the MLSS results.

#### *6.3.4 Data management*

To permit user judgment and knowledge to be reflected in the results, several capabilities can alter the contents of the MLSS report. These capabilities are called data management and include the ability to change an input measurement, change a day in the high day list, change a month in the average busy season list, or change a validation failure flag. Any changes made are identified on the MLSS so the engineer receiving the report is aware of the changes and can question their reason. The changes are made by input transactions that direct the system to modify the history for the current engineering year. At the end of the engineering year, the system produces the completed MLSS report, which reflects the entire year's data along with all data management changes. After this report is produced, the system retains the history for up to two additional months to allow for further changes.

### *6.4 No. 5XB COER software*

#### *6.4.1 Overview*

No. 5XB COER software consists of two main procedures and four support procedures. The four support procedures perform:

1. File initialization—Establishes the record base for new installations of the system.
2. File conversion—For each new software release that affects the record base, this procedure converts from the old to the new.
3. Parameter file maintenance—The parameter file contains the information that controls the processing order of the system. This procedure can modify that information.
4. Program log list—As described in Section 3.2.4, the program log is a continuous record of system execution activities. This procedure produces a report of that information and is used primarily as a debugging aid.

The two main procedures, Control File Update and Process Traffic Data, will be described in the following sections. An overview of the processing steps and interactions of these two procedures is shown in Fig. 11.

There are two primary files in 5XB COER. The Control File contains all the user-supplied control information and office parameters. It serves a function for 5XB COER that is analogous to that provided by the CU files for TDAS and other systems. The History File contains all the processing results being accumulated over the engineering year for MLSS reporting. The data management transactions are depicted in Fig. 11 from a logical point of view as going directly to the history file; in reality the transactions pass through the Control File.

#### *6.4.2 Control file update*

The Control File Update process incorporates numerous input transaction checks to ensure database accuracy. There are two levels of checks: syntactic checks that ensure correct data formats and values, and semantic checks that perform further edits to ensure the records are logically consistent. Errors result in one of two actions: either the transaction is rejected outright, or the affected office is placed in an "error" state. The error state prevents further processing of traffic data for the office until the error is corrected. Transactions may be effective immediately or they may become effective in the future, depending on the date supplied by the user. Since this is a file update process, it is designed to run as often as necessary prior to a single execution of the Process Traffic Data run.

#### *6.4.3 Process traffic data*

Process Traffic Data is the main batch procedure of the 5XB COER system, generally executed weekly. The major functions of this process are:

1. Data Edit—Screen out all but the valid data requested for processing
2. Validate—Assess each measurement and assign a degree of confidence tag
3. Calculate—Formulate all required numerical results
4. History
   (a) Mesh new results with historical base
   (b) Perform special BHD and data management processing (if applicable)
   (c) Assemble output results file for reporting
5. Reporting—Format the output results for input to the Report Distributor for printing.

These functions are described in the following paragraphs.

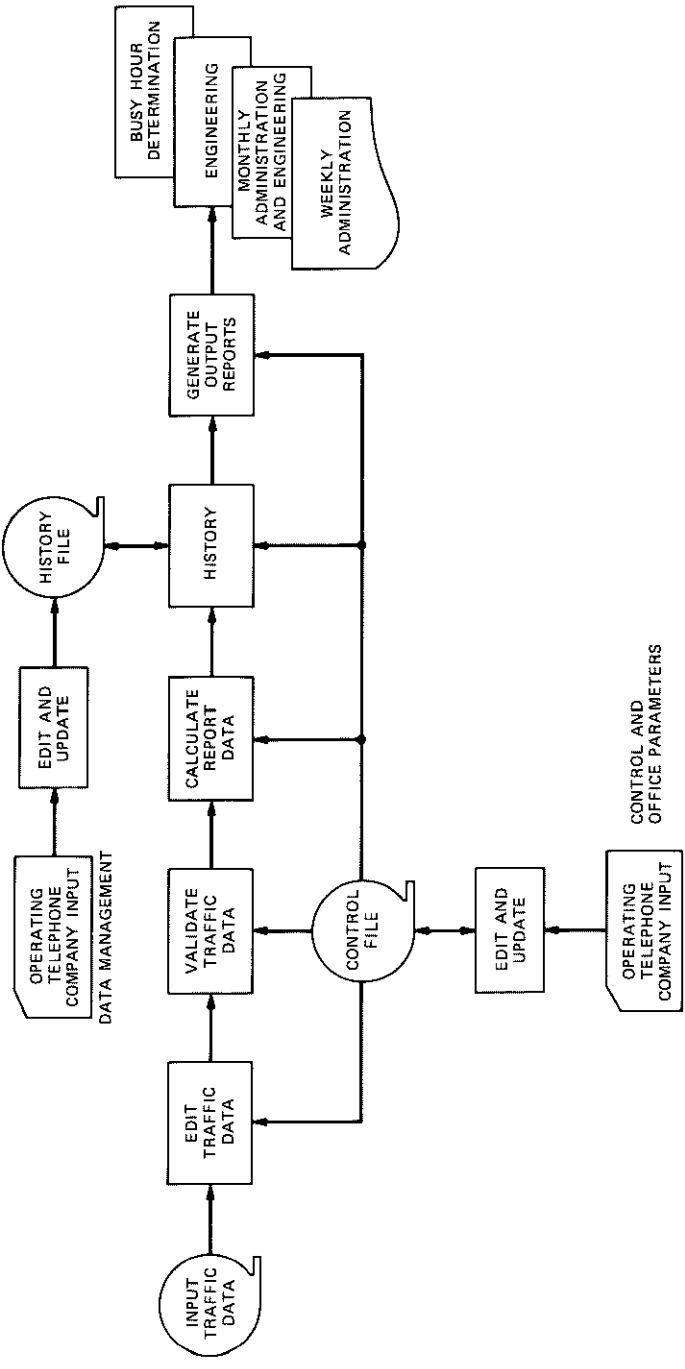



Fig. 11—Processing overview of 5XB COER.

*6.4.3.1 Data edit and validate.* The Edit function performs syntactical checks on new traffic data received from TDAS, places data in a Save Data File for any office indicated in the control file as being in "error", and combines new traffic data with any saved data for further processing. All input edit errors are detailed in an error report that includes a statistical log reflecting the current year's activity.

The Validation function makes a comprehensive assessment of the quality of the data by employing the extensive set of tests described in Section 6.3.2. All validation test failures are reported, identifying the data as well as test parameters involved.

*6.4.3.2 Calculate.* The Calculation function is the next major task. The validation tags are propagated into the calculations so that a calculation result receives the lowest tag of any of its input data items.

The calculation module has incorporated a particularly robust design that is described further because of its usefulness in other applications. The design centers on a common computation subroutine driven by tables that define the calculations to be performed. Calling routines are organized to parallel the component report structure, such that for a particular summary there is one calculation routine.

The information in the control tables is stored as a stack in symbolic postfix form. The computation subroutine evaluates the stack in two passes. The first pass checks the stack for syntactic and semantic errors and builds a work table of operand values. The second pass performs the calculation and checks for overflow and transaction errors. The control information is organized in five tables, one of which provides the calculation formulas in postfix form, the other four of which provide operand values.

Seven permissible operators have been incorporated in the calculation table: the five normal arithmetic operations of addition, subtraction, multiplication, division, and exponentiation; and two end of stack symbols, one of which indicates that a percent check should be made to ensure that the resultant value is less than 100.

This design has enabled the calculations to be easily maintained. To correct or modify any calculation, all that is required is a table update. To add a new report requires the addition of a new calling routine in which the most difficult part is the table definition.

Up through the calculation function, all the modules have operated on the current cycle's traffic data. The remaining functions, History and Report, integrate that data with previously processed data and report the results to the user.

*6.4.3.3 History.* The History Update module performs three major tasks. First, it merges the current traffic data with the stored traffic data in the history file. To do this, the current daily data records are sequentially added to the appropriate section of the file; the high day

list is reorganized with respect to the new data with new days inserted and old days deleted; and once sufficient daily records exist, the monthly average is computed (or recomputed if it already exists). Also, if a BHD study is active and new data are received, the merging of the new hourly results by component must be completed. Second, any user data management transactions are applied to the high day and monthly average records. Third data appropriate for the WMAR, MMAR, or MLSS reports are released.

*6.4.3.4 Reports.* The last function executed in the process traffic data procedure is Report Formation. This module formats and merges all the outputs from the previous modules into a file for input to the Report Distributor System (Section IX). There are currently more than fifty basic report formats and over 200 variations within these formats. A multilevel table-driven design provides a generalized module that is simple to modify, maintain, and test.

## VII. INDIVIDUAL CIRCUIT ANALYSIS SYSTEM

### *7.1 Introduction*

Maintaining wiring and assignment records for traffic data collection is a complicated task. It is quite common to sample more than 10,000 points in an electromechanical switching machine. These sampled points are combined to provide accumulated counts in typically a thousand registers or data collection devices (DCDs) used for engineering and administering the office. This accumulation of sampled points into DCDs greatly simplifies the engineering and administration functions by reducing the amount of data to be analyzed. However, the record-keeping and physical wiring associated with this grouping function is a major source of error that can propagate into costly engineering mistakes.

To understand the Individual Circuit Analysis Program (ICAN) and how it assists in this record-keeping process, it is necessary to understand the functions of EADAS/ICUR. This subject is covered in detail in Ref. 4. However, Fig. 12 presents a brief overview of EADAS/ICUR.

With EADAS, traffic data counts occurring in the central office are encoded and sent via a data link to the collection machine. In basic EADAS (i.e., without the ICUR option) the usage data are collected at the output of the Traffic Usage Recorder (TUR), where the measurements have already been grouped by wiring on the cross-connect field of the TUR frame. This grouping function allows the accumulation of related information into a single DCD (e.g., usage data collected from 10 individual trunks into one trunk group). With the ICUR option, which can be applied to electromechanical switching offices, the data are grouped via software in EADAS rather than via wiring

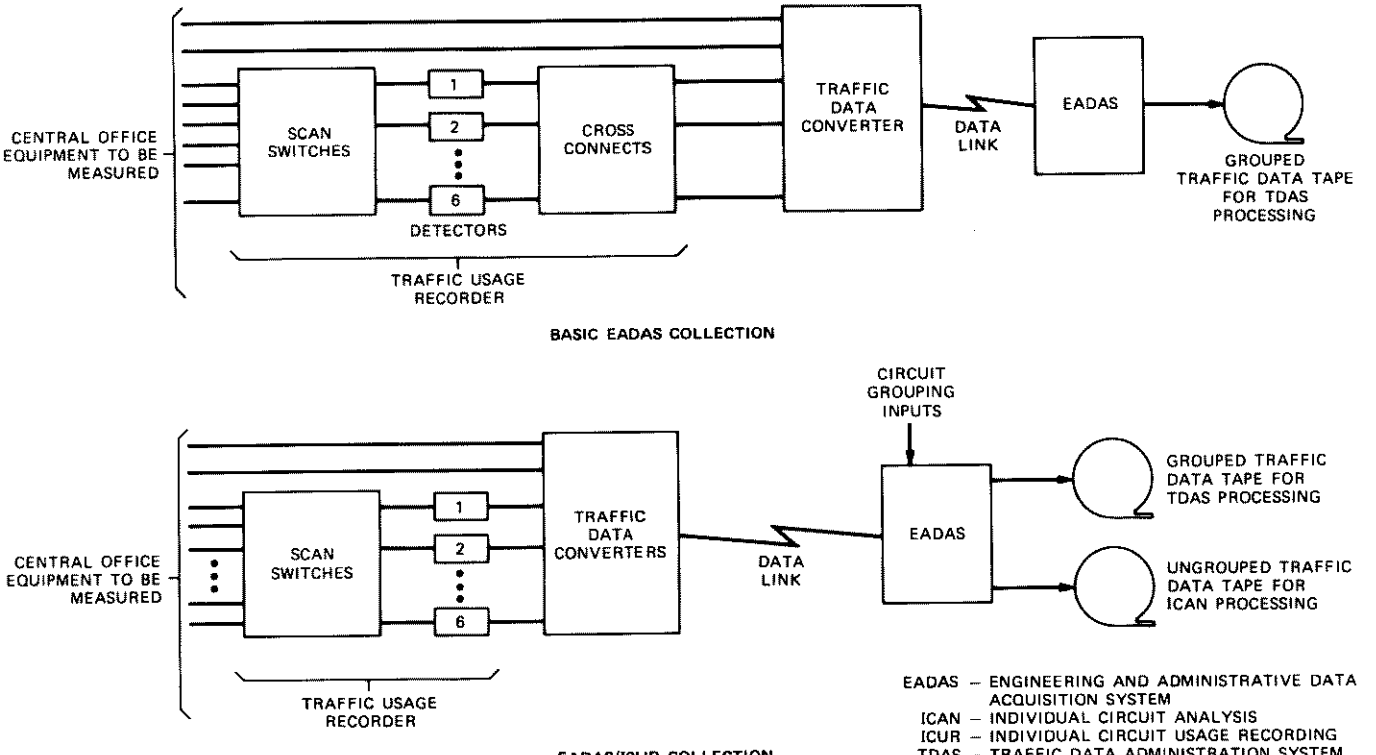



Fig. 12—Overview of EADAS/ICUR.

on the TUR cross-connect field. Software grouping simplifies the TUR administration function by replacing manual TUR cross-connect logs. In addition, the ICUR option allows network administrators to observe the usage being accumulated on each of the individual TUR inputs within a group. These ungrouped data can be analyzed to detect switching equipment problems in the central office, data collection equipment problems, and various database errors.

EADAS/ICUR produces a magnetic tape for ICAN processing. This tape contains the ungrouped usage data and additional information necessary to administer the TURs. The ICAN process is unique among TNDS/EQ components in having access to individual circuit usage information. This makes the system highly valuable in detecting central office operational problems that otherwise could go unnoticed.

This overview of ICUR operation will lend perspective to the description of the inputs, outputs, and processing flow of ICAN that follows.

### *7.2 ICAN inputs*

ICAN's purpose is twofold: to aid in the administration of TUR frames through the use of the EADAS/ICUR circuit grouping map and to help detect faulty central office equipment items. To accomplish these functions, inputs are needed from three sources: EADAS/ICUR, the Common Update record base, and the network administrators.

EADAS/ICUR provides the ungrouped traffic data measurements, the schedules on when these data were collected, a map on how the individual measurements are grouped into DCDs, and the updates to this map since the last ICAN tape was written.

Descriptive information for each of the DCDs is obtained from the Common Update record base. Finally, the network administrators provide control information specifying the reports to be produced and the individual circuit data to be purged from the ICAN history file.

### *7.3 ICAN outputs*

With the basic inputs described above, ICAN can produce a series of output reports to aid network administration. These reports can be produced on either a demand or exception basis. The reports are divided into the categories of record base listing reports, error reports, equipment surveillance reports, and system administration reports.

#### *7.3.1 Record base listing reports*

ICAN maintains the record base describing how individual TUR inputs are grouped into DCDs. Various views of this record base can be produced in the form of report listings tailored for a particular network administration task.

TUR administration listings provide a complete layout of the TUR inputs. Each input is identified by physical location of the central office equipment being measured, along with the associated DCD accumulating the information. To aid in the assignment of new TUR measures, a separate report can be produced itemizing those inputs that are unassigned or unequipped.

Other views of this record base list the individual DCDs and specify the associated TUR inputs that are accumulated in each DCD count. This view aids in troubleshooting various validation test failures as well as in making new DCD assignments.

A third view of the record base is organized by the equipment location being measured and identifies the associated TUR input and associated DCD. This view is primarily used in central office maintenance activities for troubleshooting reported data collection problems.

Individual TUR inputs are associated with a primary DCD. However, the scorings can also contribute to a secondary device. These secondary devices, called load grouping codes, typically accumulate total office input equipment utilization broken down by load division (see LBS, Section VIII). This secondary method of accumulation simplifies various office engineering functions by providing a single DCD representing the total usage accumulated over hundreds of other devices. Load grouping code listings show this type of DCD summarization.

In addition to the various TUR and DCD administration listings mentioned above, there are two other ICAN record base listing reports: the TUR schedule report, which identifies the ICUR data collection periods, and the Site Definition listing, which identifies all EADAS/ICUR data collection machines within the company.

#### *7.3.2 Error reports*

The ICAN error report series produces various exception reports itemizing discrepancies found in processing inputs and updating the TUR record base maintained in ICAN. The reports specify the various input transaction errors, and the discrepancies found between Common Update, EADAS/ICUR, and ICAN record bases. The record base auditing functions incorporated in ICAN correlate various parts of the TNDS record base and verify that these record bases are kept in synchronization.

#### *7.3.3 Equipment surveillance reports*

Of particular interest in the ICAN system are the three central office equipment surveillance reports. Because information is available on the basis of an individual TUR input, additional verification can

be performed to assure the central office equipment is operating properly.

*7.3.3.1 Abnormally Short Holding Time reports.* The Abnormally Short Holding Time (ASH) report is useful in detecting a "killer trunk." A killer trunk is a circuit that reflects an abnormally short holding time because customers abandon it almost immediately to escape noise, bad transmission, or other problems. Because these trunks are quickly returned to an idle state, the switching equipment is likely to pick them more often than other trunks within a group. One faulty trunk may cause numerous failures in attempts to use the group, even though the group is well below capacity. The ASH tests detect probable killer trunks by a statistical analysis of the individual usage data.

Following is a simplified example demonstrating the working of ASH. Consider the sequence of busy/idle indications for two circuits of the same group connected to a 100-second scan TUR:

Circuit A: 111001100
Circuit B: 101001101,

where: 1 = busy condition when sampled, and 0 = idle condition when sampled. As you can see, both circuits were busy five of the nine times they were scanned by the TUR. However, Circuit A has only three transitions between busy and idle status, while Circuit B has six. If we assume normal call duration of 200 to 300 seconds, it is probable that Circuit A is carrying only two calls of at least a 200-second duration in this time period. It is also probable that Circuit B is carrying at least four distinct calls (busy indication followed by idle indication) of much shorter duration. Circuits displaying these characteristics are likely to be ASH circuits.

As noted above, many busy-to-idle transitions are indicative of a killer trunk. A busy-to-idle transition is considered a positive ASH factor. Similarly, a trunk that is detected as being busy for a number of successive TUR scans is more likely to be operating normally. A trunk appearing busy for two successive TUR scans is therefore considered a negative ASH factor.

To determine which trunks have abnormally short holding times, ICAN first weights these positive and negative ASH factors by the group occupancy rate (total CCS for the DCD divided by the number of TUR scans) and by the individual trunk occupancy for all other trunks. These weighted factors are accumulated until their sum (called the "ASH statistic") exceeds a positive or negative threshold.

If the positive threshold is reached, a message is output on the ASH report indicating that the trunk has an abnormally short holding time.

Thus, ICAN infers through a statistical algorithm that a trunk has

an abnormally short holding time. Even though the ASH detection method is statistical, it is designed to be 98 percent reliable in reporting true office problems.

*7.3.3.2 Unusual usage report.* ICAN analyzes the usage data received and reports on TUR inputs that show usage but are unassigned or unequipped, as well as TUR inputs that are assigned but appear inactive.

Unassigned/unequipped analysis is quite simple. With each transmittal from ICUR, the unassigned/unequipped TUR inputs are analyzed for transitions between busy and idle states. If a transition has occurred, the TUR input is flagged for exception reporting. In the exception report, the amount of circuit occupancy is computed and displayed.

Inactive circuit analysis is also quite simple. If the TUR input shows no transitions between busy and idle, it is flagged for exception reporting. The state of the circuit (100 percent busy or 100 percent idle) along with the date of last activity are displayed on the exception report. Care must be taken to avoid reporting false alarms for those measurements that are typically inactive. The set of measurement types that are typically inactive are automatically identified by the unusual usage software and further analysis is bypassed.

*7.3.3.3 Individual circuit usage displays.* The individual circuit usage display is a valuable tool available with ICAN. At the user's prerogative, usage on selected circuits can be accumulated and displayed to help analyze problems too subtle to be detected by the ASH algorithms. These displays are also used for tracking specific troubles indicated on central office equipment. For the selected DCDs, the occupancy of each of the individual measurement components that accumulate data for the DCD are displayed in a graphical form. This enables easy comparison of measurement components with similar characteristics. For example, the occupancy over a selected time period of each of the trunks in a given trunk group can be displayed for analysis. In essence, the display report shows a view of the central office equipment operation that is similar to a time-lapse photograph.

### *7.3.4 System administration reports*

The final set of ICAN reports is designed for administration of the ICAN System itself. This set of reports depicts activity that has occurred, and summarizes processing results for the individual ICUR input tapes on an individual DCU basis. The majority of these reports are intended for use by network administration personnel. Additional reports list computer resource usage and record base activity. The latter would be needed in the event of a system recovery.

### 7.4 Processing flow

#### 7.4.1 General

The ICAN System consists of a single mainline program and several support utilities. This single job approach simplifies OTC scheduling, operation, and control. Based on control inputs, the mainline ICAN process validates inputs, processes ICUR tapes, produces periodic and exception reports, purges unneeded data, and backs up the database.

Because of the large volume of data and the complexity of some of the analysis algorithms, the system has been designed for efficient data processing. A typical company uses ICAN to administer 300 TURs, requiring the weekly processing of over 10 million individual circuit usage measures and over 30 million individual validations.

The flow of mainstream ICAN and the various utilities are described in the following sections.

#### 7.4.2 Mainline process

The functions performed in the mainline ICAN process are shown in Table V. A single execution consists of four phases. The first, the card input processing phase, analyzes and validates the input requests, produces various error reports, and sets up the controls for the additional phases as requested by the input transactions.

The second phase, tape processing, is the most complex of the mainline ICAN and includes:

1. Schedule processing—Analyzes the ICUR tape writing schedules, produces the associated master record base updates, and outputs any appropriate error messages.

2. Card image processing—Analyzes the Circuit Grouping Map (CGM) updates produced since the last ICUR tape was written. The updates are validated and the CGMs are updated. Any discrepancies are noted on the appropriate error reports.

3. Map processing—Compares the updated ICAN CGM with the current ICUR CGM within this function. Again, after analysis, the appropriate error messages are generated and the ICAN CGMs updated to reflect current assignments in EADAS.

4. Circuit data processing—Analyzes individual circuits for ASH and unusual usage, stores data for display reports and long-term analysis, Pending Status Flag (PSF) processes, and deletes old, unneeded data currently resident in the record base.

The reporting phase, three, formulates and produces both requested and exception reports.

In the final phase a backup tape of the record base is made so that restart and recovery steps are simplified if errors are encountered in future runs of mainline ICAN.

Within mainline ICAN processing, a checkpoint feature is imple-

Table V—Mainline ICAN processing phases

| I: Card Input | II: Tape Processing | | | | III: Reporting | IV: Backup |
|---|---|---|---|---|---|---|
| | Schedule Processing | Card Image Processing | Map Processing | Circuit Data Processing | | |
| Validate input cards and report discrepancies | 1. Compare schedule with existing schedule<br>2. Report schedules if changed<br>3. Determine number of tape writes on tape | 1. Validate and report discrepancies<br>2. Update ICAN maps or discrepancies | 1. Compare 1-for-1 with existing maps<br>2. Report discrepancies<br>3. Change ICAN maps | 1. Store data<br>2. Unusual usage-inactive circuit<br>3. PSF change<br>4. Produce ASH report<br>5. Display report<br>6. Delete data if requested | Assemble and produce reports | Generate backup tape |

mented so that if processing problems are encountered later in the same job, job restart will begin automatically at the point immediately after the last successfully processed ICUR tape.

#### *7.4.3 Support process*

To supply needed DCD descriptive information to mainline ICAN, a single support run is periodically executed to extract the DCD information from the Common Update record base, identify discrepancies between the record bases, and produce the necessary record base updates.

#### *7.4.4 Utilities*

A number of support utilities are available to be used when needed. These utilities create and modify various portions of the ICAN record base, provide recovery capabilities of the ICAN record base, and serve as a backup in case problems occur in the EADAS/ICUR record base.

## VIII. LOAD BALANCE SYSTEM

### *8.1 Introduction*

A principal Bell System goal has always been to offer the best possible service at the lowest possible cost. For the network administrator, this objective translates into making the most efficient use of existing switching facilities while maintaining an acceptable level of service to all customers. To achieve this goal, network administrators must distribute the telephone traffic load evenly across the machine's switching (or load) units. This principle is called load balance. This section will describe a tool available to help achieve that goal—the Load Balance System (LBS). LBS is a integrated component of TNDS/EQ. It relies on CU for maintaining its record base and on TDAS for supplying the weekly traffic data with which it does all its processing. Distribution and printing (or microfiching) of all LBS reports are the responsibility of the Report Distribution Facility (see Section IX).

LBS supports all 1 *ESS* switch, 2 *ESS* switch, No. 1 Crossbar (1XB), 5XB, Crossbar Tandem (XBT), and Step by Step (SXS) machines for which data are collected. Network administration functions are supported by calculating a Load Balance Index (LBI) and providing assignment and balance guides.

### *8.2 LBS outputs*

#### *8.2.1 Index reports*

Because of the importance of load balance, an administrative index was devised in the mid-1960s to measure the effectiveness of load balance procedures. LBS mechanizes the implementation of the re-

vised Load Balance Index Plan, as prescribed in AT&T administrative practices. Index reports are produced on a traffic unit (TU) to indicate the level of service and balance achieved by that TU. From these measures indexes are aggregated into the hierarchy of district, division, area, and company scores. On a Service Observing Month (SOM) basis, the company-level report is forwarded to AT&T.

For those offices in which the index is not calculated by LBS, typically smaller, rural SXS machines, the index can be entered into the LBS record base (via CU) for inclusion in higher-level LBI reports.

#### *8.2.2 Data summaries*

Data summaries provide detailed information on the data used in calculating the Load Balance Index. These reports require constant attention. Identified on these reports are load units that have rising or falling load trends, unusually high or low loads, or penalty points assigned from the index calculations. These are signs of either current or potential trouble spots in the switching machines and indicate not only where corrective action may be needed but where additional studies may be appropriate. These reports should be used in association with the line assignment aids (see Section 8.2.4) to achieve an increase of traffic in the more lightly loaded load units while diminishing traffic in the more heavily loaded load units. This is typically accomplished through normal disconnect and assignment activity. In drastic cases, reduction of heavy loads is accomplished through specific transfer of customer lines out of heavily loaded units into lightly loaded ones. This procedure to improve line load balance (and thus the Load Balance Index) should result both in increasing the machine's effective capacity and improving service to the customer.

#### *8.2.3 Second Session Studies reports*

Many switching entities experience a secondary daily busy hour during some part or all of the year, due to the calling habits of a community of customers with different calling patterns than those responsible for the primary busy hour. While the LBI is calculated using only primary busy hour data, it is also important to provide quality service during the secondary busy hour. To identify secondary busy hour imbalances, LBS produces Second Session Study reports. The same principles apply to use of second session data summaries as to the busy hour data summaries. The reports are similar, with the exception that penalty point and index values are not calculated for second session studies.

#### *8.2.4 Line Assignment/Transfer reports*

LBS produces two basic aids for line assignment. Line Assignment

Guides provide lists of load units, with the most lightly loaded ones listed first, thus appearing in order of preference for assigning new lines. Line Equipment Transfer Guides start at the other end of the list; the most heavily loaded units appear first and are thus in order of preference for transfer of lines. The user may request a Line Assignment Guide or Transfer Guide in any length desired. These reports assist the manual process of determining new line assignments and, if necessary, transferring current assignments to other load units. As telephone companies introduce mechanized line assignment processes, such as the Computer System For Main Frame Operations (COSMOS), manual assignments from LBS reports are discontinued. For that reason Line Assignment Guides are produced only on demand on a TU basis.

#### *8.2.5 Nonindexed balance aids*

There are some components such as trunks and service circuits that are also sensitive to load balance but are not included in the LBI calculation. To assist in achieving proper balance of those components, LBS produces balance aids that contain corrective load values.

### *8.3 LBS inputs*

LBS requires three categories of input to perform its functions: traffic data, traffic unit characteristics, and a history base. The history base is a file internally maintained by the system itself. The other two inputs are provided by TDAS and CU, which operate with LBS in a tightly integrated fashion.

#### *8.3.1 Traffic unit characteristics*

For LBS to judge the degree of balance within a traffic unit and to provide aids in achieving or maintaining proper balance, the system requires a detailed description of certain TU characteristics. This information is provided by the user via CU. Through the input of various transactions, the user builds a Load Balance Master File, which contains a description of all TUs that LBS supports, specifying the traffic measurements to be processed for each TU. Included in such TU descriptive information is common language identification, number of main stations, destination of output reports, assignment and loading division information, theoretical capacities, and average holding time (AHT).

#### *8.3.2 Traffic data*

In addition to TU characteristics, LBS requires data that represent the traffic load being offered to the TU. The source of the data is, of course, the switch itself. However, before reaching LBS, the data are

first obtained by a data collection system and passed through TDAS. Using the traffic measurement definitions and traffic measurement requests, TDAS validates, adjusts, and summarizes the data before writing them to an interface file for LBS processing. TDAS also applies an Equipment Measurement Code (EMC) label to each data item, permitting LBS calculations to be driven by internally interpreting the EMCs.

#### *8.3.3 History files and system parameters*

Since some of the computations involve averaging and trend analysis, it is necessary that history files be maintained for reference and updated on each run of LBS. Included in the history files are the previous week's loads and penalty points, average holding times, percent capacities, and total-to-sample ratios. The latter is a means of determining total usage on a load unit from a sampling of two of its links.

Certain numerical values are required in the LBI algorithms used by LBS. These include default values for average holding time and total-to-sample ratio, Quality Control Limits (QCL), designation of threshold values for "hot-spot" determination, and specifications of LBI correction values. The QCL is a measure of how far the load on a load unit is allowed to deviate, without penalty, from the average load for the loading division.

### *8.4 LBS processing flow*

#### *8.4.1 General*

A complete execution of LBS is performed on a weekly basis following the running of TDAS. The system is divided into six discrete runs, one for mainline LBS and the other five serving as support runs. The remainder of this section will address the mainline run of the system, as shown in Fig. 13.

#### *8.4.2 Study formation and validation*

The Study Formation and Validation module receives the input traffic measurements from TDAS. It performs validation tests on the traffic data and combines it with office characteristic information obtained from the CU files for further processing. In addition, if the appropriate peg count data are available, average holding time (AHT) and total-to-sample (T/S) ratios are calculated for later use in index and corrective CCS calculations.

Validations are categorized as follows:

1. Study identification—Data are checked to ensure that only valid combinations of studies have been requested.
2. Data volume—The number of hours requested is not always the

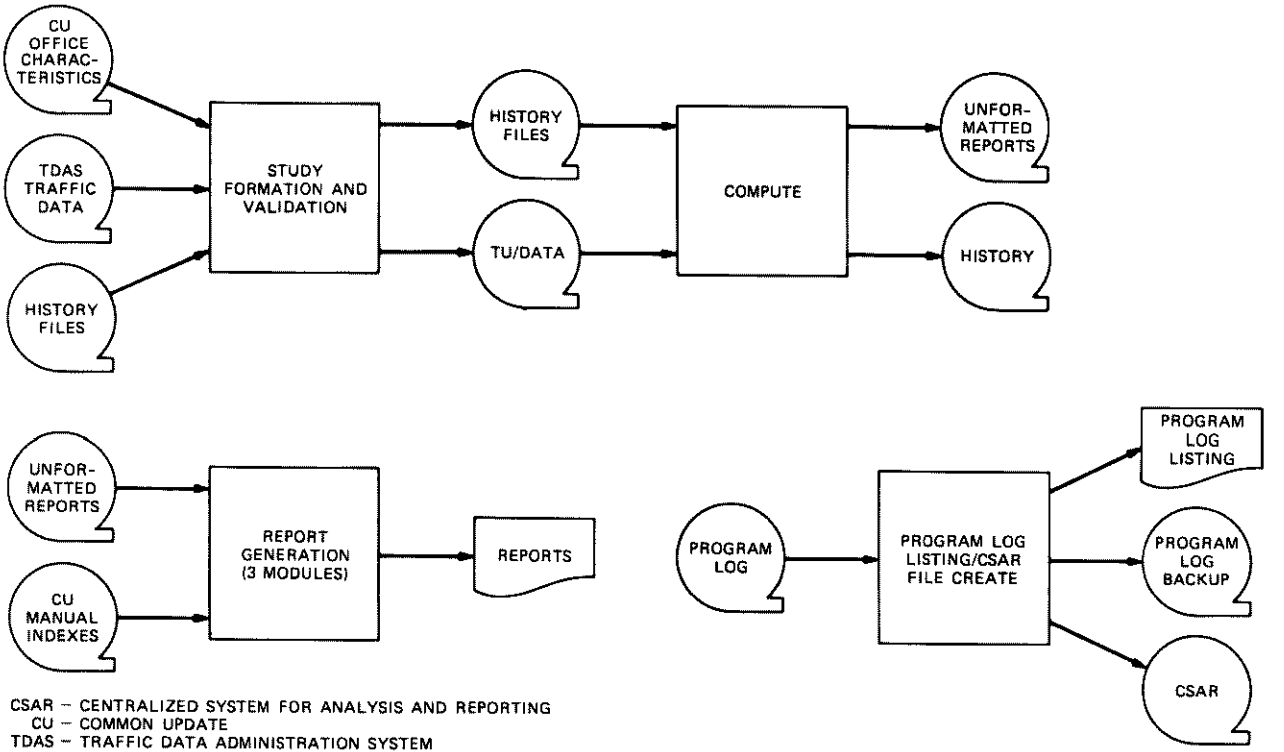



Fig. 13—Load Balance System processing flow.

number received. Acceptable ranges for the different studies have been established to allow for reasonable variation in equipment performance and to retain the statistical reliability of the study. If sufficient data are received, the study is abandoned. In addition, usage data for any hour having either zero or more than 36 CCS are rejected.

3. Data identification—Data are validated to contain acceptable EMC and measurement-type designations.

4. AHT and T/S Ratio—Calculated AHT and T/S ratios are compared to historical values. If they differ by more than a fixed percent, they can be discarded, based on the judgment of network administration personnel.

5. Study criteria—To perform an index study, more than 75 percent of the load units in a TU must have valid data and the loading divisions must be at greater than 30-percent capacity. If both conditions are not satisfied, the study is aborted.

#### *8.4.3 LBS compute*

Following the above data validations, the Compute module performs the many calculations necessary to satisfy the study requests. The first function is to react to history change requests to recompute, delete, and reinstate data from history.

Component calculations that contribute to the LBI are performed next. Using AHT and T/S ratio, the quality control limits are determined. Scores are then developed depending upon the indicated deviation of actual usage from the average usage. Load units operating at loads above predetermined thresholds are assigned a "hot-spot" penalty. The index is then calculated using weighted linear regression techniques that utilize up to 12 weeks of past data. Since load imbalances affect service more severely as the total traffic handled by the office increases, the percent load capacity is another factor used in determining the final index.

Finally, corrective CCS values are derived from linear regression estimates of what each load unit's deviation from the average will be at the time of the next study. These values are displayed on the balance guides in the appropriate order to facilitate manual corrections to imbalance.

#### *8.4.4 Reports formations and system cleanup*

The remaining modules assemble, sort, and format the various reports produced by the system, and perform cleanup functions.

## IX. REPORT DISTRIBUTOR

### *9.1 Introduction*

For the effective utilization of the TNDS downstream systems, it is essential that the reports be distributed in an efficient and timely

manner to the users, most of whom are located physically remote from the computation center executing TNDS. The TNDS/EQ components alone can produce several hundred different report formats for distribution to as many as 600 distinct locations within an operating telephone company. To complicate this process further, selected reports are needed on microfiche produced by equipment manufactured by several different vendors, and other reports require multiple copies.

Early in the evolution of TNDS, a Report Distribution and Printing Facility was developed. This capability is utilized today for all the downstream TNDS systems. In addition, it is available as a standalone package and can be used by non-TNDS systems to solve their report distribution needs. The following sections describe the capabilities of the Report Distributor, its inputs, outputs, and processing flow.

### *9.2 Report Distributor capabilities*

The design of the report distribution process is based on two standards implemented throughout the downstream TNDS systems:

1. Every report type is uniquely identified with a five-character report identification specified in the report header.

2. Every report utilizes a four-character responsibility code indicating the specific user who is to receive the report. This responsibility code is also identified in the page header.

With these two basic standards, the report distribution process was designed to produce:

1. Microfiche—For long-term storage, or cost reduction reasons, users are able to select particular reports for output to microfiche. The Report Distributor has the capability of producing outputs that meet the specifications of several microfiche hardware vendors.

2. Multiple copies—Users are able to request multiple copies of selected reports without multiple passes through the distribution process.

3. Proprietary marking—The users can specify that the appropriate proprietary notice be printed on the individual report pages.

4. Monitoring distribution—Although the primary individual is identified with the report responsibility code, alternate distributions can be identified, usually for monitoring purposes.

5. Burst pages—Before a printout can be distributed, it must be broken down by responsibility code. To make this easier and to reduce errors, burst pages are optionally printed (three pages of output with characters printed over the fold) between reports where the responsibility code changes.

6. Remote print—Where the report destinations are physically remote from the computer center running TNDS, time and cost

savings are realized by producing report tapes according to destination and having them printed locally at the remote site. This facility allows selected reports to be routed to numerous locations.

7. Selective retrieval—To recover lost listings, or for historical checking of a report, selective retrieval can be made of an individual report or all reports for a particular responsibility code from a report tape.

8. Print restart—Following a system failure or job cancellation, the report printing program is able to restart at a point that results in minimal duplication of printout. Typically, report print programs are run with a dedicated printer, so that restart points can be communicated to operations through the system console. The frequency of the checkpoints are at the discretion of the OTC. When restarting is needed where multiple print tapes are involved, the restart does not have to spool through complete tapes that have already been printed.

### *9.3 Report file inputs*

The report distribution facility imposes two requirements on every system providing a reporting file for distribution:

1. The report file must be organized in responsibility code/report number order.

2. Key records must be included at specific points within the report file. These records identify the number and responsibility code of the report following the key record. At a minimum, a key record is specified on each change in responsibility code and report number. In the worst case, a key record is provided for every report page. (This is necessary if the report is on microfiche and an index to the fiche is provided on a page basis.)

The report distribution process detects these key records and through the use of Common Update files determines the proper output distribution device.

### *9.4 Common Update inputs*

Inputs through Common Update define the pattern for the distribution process and the types of proprietary messages to appear on reports. Three types of transactions are necessary to define these characteristics:

1. One transaction defines the report output device. These devices are used to break up the distribution process into parts. An individual device may be designated for transmission to a remote computational center where the reports on that device are further subdivided for distribution. A device may be designated for input to a particular type of microfiche equipment, or it may represent the set of reports to be distributed to an individual user.

2. The second type of transaction specifies which report number/responsibility code combinations are to be distributed to each of the numerous output devices.

3. The third transaction specifies by report number the proprietary message to be printed at the bottom of each report page.

The current contents of the various report distribution tables maintained by Common Update can be examined on request.

### *9.5 Outputs*

For problem-tracking considerations, output reports are produced that summarize the information written to each of the output devices during every execution of the report distribution process. These reports identify each responsibility code/report number change, along with any further report content information contained in the key record.

### *9.6 Report distribution process*

The report distribution process is implemented by a combination of three programs (Fig. 14): the merge program, the distributor program, and the printer program.

#### *9.6.1 The merge program*

All TNDS report generation programs produce report data sets ordered by destination (responsibility code), similar to the outputs from the distributor and printer programs. When multiple report tapes are to be processed by the distribution facility, a merging of the reports must be made. This is the function of the merge program. It is designed as a separate run because (1) its output may be input to either the distributor or printer program, and (2) to put the merge functions in the distributor program would increase the number of devices required to run it.

#### *9.6.2 The distributor program*

The distributor program must be used when there is a need for the generation of remote print tapes and/or microfiche tapes. It uses as input the TNDS keyed report tapes, a key definition file, and the report distribution table. It produces any combination of three outputs: keyed report tapes, nonkeyed report tapes, and microfiche tapes. Keyed report tapes are printed at a local or remote site and can be fed to the printer program directly or further processed at the remote site by another copy of the distributor program before being printed. Nonkeyed report tapes are printed at a local or remote site, and microfiche tapes can be processed by any of the supported microfiche systems.

Report processing by the distributor is key record driven. That is,

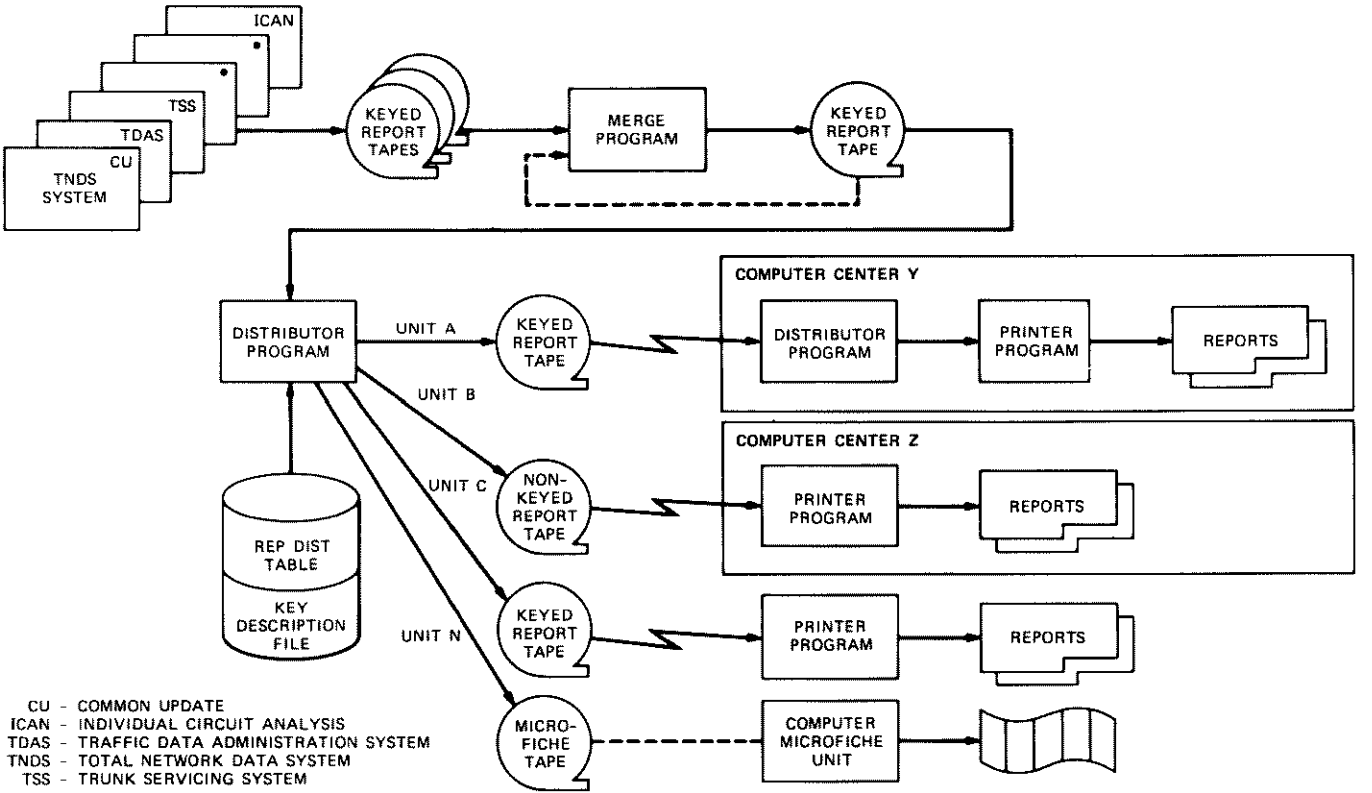



Fig. 14—Report distribution process.

when it encounters a key record on a report tape, it matches the responsibility code and report number against the selection masks in the report distribution table, identifying those report devices that are to receive the report. It then treats all report records up to the next key record as one report, spooling each record to the appropriate data set(s). If no report selection mask matches the report, the report is written to a default report device. When the report is completed, it produces its own reports summarizing the processing that has occurred.

To accommodate changes easily in the meaning of key record fields for reports, allow for addition or deletion of reports by a system, and permit use of the distribution facility by other than TNDS systems, the distributor program has no system-specific internal information. All report-dependent information (i.e., legal report numbers, algorithms for formatting key record data on microfiche, index lines, and distributor reports) is kept external to the program, in a Key Definition file constructed by the using systems.

There are four options supported in the distributor program for microfiche output, which are provided by the device definition record in the report distribution table. These options deal with blank microfiche separators, magnification factors, forms flash, and microfiche sequencing. The blank microfiche option, if used, specifies that a blank microfiche is to be generated when the responsibility code changes. This allows operations to more easily cut the microfiche film for distribution. The magnification factor parameter specifies the magnification to be used in generating the microfiche. The forms flash option enables the user to have a preprinted form superimposed over a microfiche frame. Finally, the microfiche sequencing option determines whether the sequence number as printed on the microfiche is to be reset to one when a new responsibility code is encountered. Each report number or responsibility code change causes a microfiche advance.

#### *9.6.3 The printer program*

The printer program can be used both at the OTC central computer center and at a remote print site to produce hard copy listings for distribution. It has options for burst pages and printer restart (options under control of EDP operations) and selective report retrieval and multiple copies (under control of the TNDS EDP coordinator).

## X. MANAGEMENT REPORTING SYSTEM (MRS)

### *10.1 Introduction*

In the past, the OTCs originated many requests to develop specialized reports in addition to the current fixed-format reports available

through the various TNDS components. The Management Reporting System (MRS) was developed to answer this need. MRS enables both TNDS/EQ and TNDS/TK users to extract information available in various TNDS files and format this information to satisfy their local reporting needs.

Besides customized report generation, MRS offers two other major capabilities for its users. First, MRS can be used to mechanize the production of Common Update input transactions. This capability is especially useful for initializing an office database and for various office conversion activities.

Second, MRS can serve as an interface between software processes. Data may be extracted and made available to OTC-developed systems without the user having to know the actual structure of the TNDS source, nor care about future revisions to this structure.

In its implementation, MRS is an adaptation of the MARK IV™ software package of Informatics, Incorporated, Canoga Park, California. Using this package, Bell Telephone Laboratories personnel provide the necessary file definitions, interfacing software, and associated documentation required to produce reports, transactions, and system interfaces.

### *10.2 File definitions*

More than twenty major TNDS files are defined for MRS use. The file definitions describe the record characteristics and define the structure of each segment within a record. These definitions allow a convenient interface between the TNDS products and the flexible reporting capabilities made available to the user community through MRS. BTL provides the file definitions and updates them with each major release. Because the Mark IV definitions describe the data information items independent of file structure, OTC-designed programs need not be modified as the basic files are restructured with new TNDS releases.

### *10.3 Mark IV interfacing software*

In addition to the basic file definitions and documentation, BTL provides various MARK IV interfacing software. This software ranges from catalogued requests to restructure the TNDS files for MARK IV use to BTL written code that is integrated with MARK IV providing specialized translation operations. Examples of these specialized translation operations include:

1. Conversion of Trunk Group Serial Numbers to the Common Language Circuit Identification (CLCI).
2. Conversion of Location Machine Processing Codes to Common Language Location Identifications (CLLI).
3. Date and time format conversion routines.

### *10.4 Using the system*

To illustrate the use of MRS, consider a request by a TNDS/TK user to identify, list, and count all trunk groups that use a particular group as an alternate leg in completing a call. Using one of the MRS reference manuals, the MARK IV definitions of the required information are identified. In this example seven distinct field definitions are used:

Field definition 1—The Trunk Group Serial Number (TGSN) of the group being investigated.

Field definitions 2 through 4—The TGSN of the first, second, and third alternate legs at the A end of the circuit group.

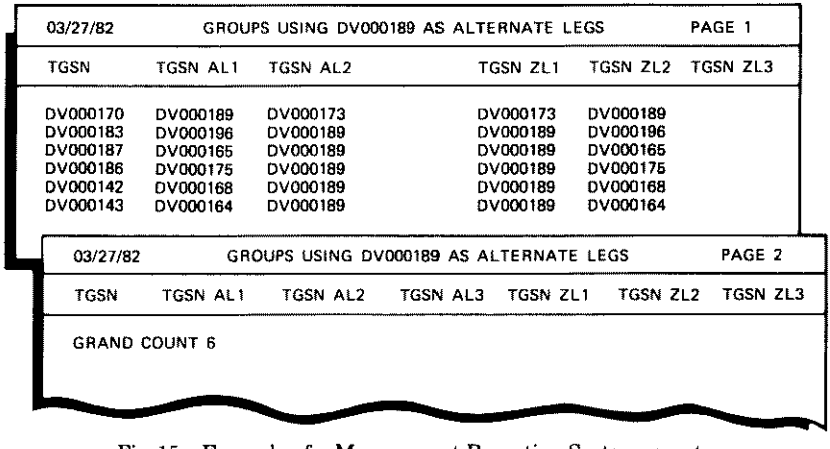
Field definitions 5 through 7—The TGSN of the first, second, and third alternate legs at the Z end of the circuit group.

A MRS request is now coded on standard MARK IV forms. The forms specify the comparisons to be made on the input files, the sort sequence of the data appearing on the output report, and the format and field titles of the output report.

In the above example a scan is made of the A and Z alternate legs for the particular TGSN in question (DV000189). If DV000189 appears as an alternate leg, the trunk group is listed in TGSN order with all legs identified. Finally, a count of the TGSNs using DV000189 is computed and displayed. Figure 15 shows an example of the resulting output report.

### *10.5 Processing flow*

The generic flowchart for an MRS run is shown in Fig. 16. Standard TNDS files and MARK IV user source are submitted as input to the run. The specified fields are identified and analyzed in Step A, during



03/27/82 GROUPS USING DV000189 AS ALTERNATE LEGS PAGE 1

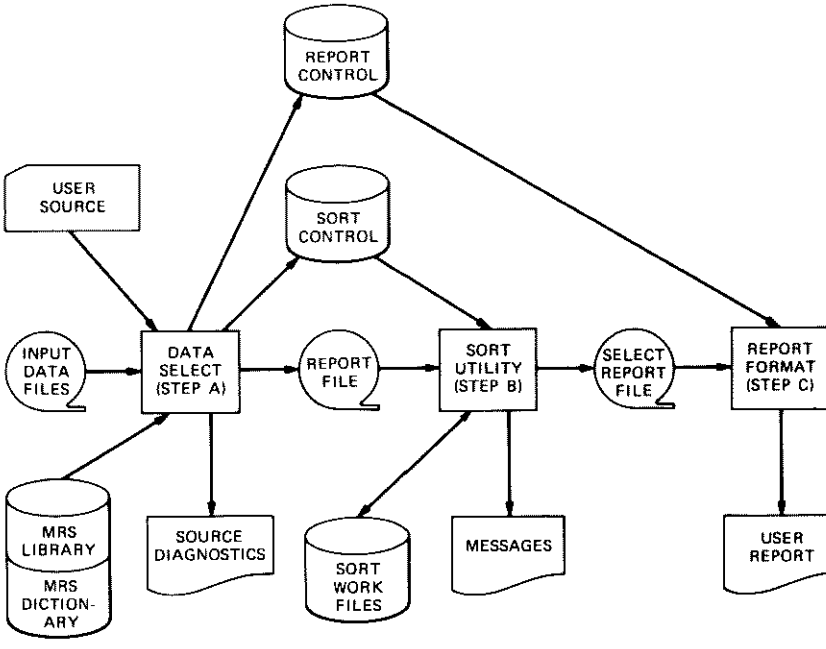
| TGSN | TGSN AL1 | TGSN AL2 | TGSN ZL1 | TGSN ZL2 | TGSN ZL3 |
|---|---|---|---|---|---|
| DV000170 | DV000189 | DV000173 | DV000173 | DV000189 | |
| DV000183 | DV000196 | DV000189 | DV000189 | DV000196 | |
| DV000187 | DV000165 | DV000189 | DV000189 | DV000165 | |
| DV000186 | DV000175 | DV000189 | DV000189 | DV000175 | |
| DV000142 | DV000168 | DV000189 | DV000189 | DV000168 | |
| DV000143 | DV000164 | DV000189 | DV000189 | DV000164 | |

03/27/82 GROUPS USING DV000189 AS ALTERNATE LEGS PAGE 2

| TGSN | TGSN AL1 | TGSN AL2 | TGSN AL3 | TGSN ZL1 | TGSN ZL2 | TGSN ZL3 |
|---|---|---|---|---|---|---|

GRAND COUNT 6

Fig. 15—Example of a Management Reporting System report.




MRS – MANAGEMENT REPORTING SYSTEM

Fig. 16—Generic flowchart for the Management Reporting System.

which the basic report file is produced. Step B is a sort utility that organizes the report file into the sequence requested by the user. Step C takes the organized report file and formats the output report requested by the user. If the objective was to generate a TNDS transaction file or create a file for input to other mechanized systems, the sort output file is used directly, bypassing the third step.

## XI. SUMMARY AND FUTURE DIRECTIONS

Components of TNDS/EQ have been in use since the early 1970s. These systems play a key role in the overall administration of network data. The systems are in production use by all Bell System operating telephone companies and Bell Canada. Extensive controls have been implemented to ensure smooth and reliable operation of the data processing center. Through standardization of the data collection environment, interchange of data among companies of common items has become possible. Development and user support activities are carried out by Bell Laboratories and Western Electric central developers, with enhancements to the systems provided by annual releases.

For the future, the major work for TNDS/EQ will be twofold: to maintain support of the continuously evolving central office and trunking network, and to provide more responsive service by using the

on-line record base and reporting features of the On-line Records and Reporting System (ORRS). This replacement to Common Update will be developed during the period from 1983 to 1986. The modernization of TNDS/EQ will include establishing telecommunications links between collection machines and TDAS, integrating the No. 5XB COER Control File into the ORRS record base, and assimilating functions currently carried out by CSAR into ORRS. In assessing future telephone company needs to be supported by TNDS/EQ, the key will be responsiveness. The use and expected number of users of traffic data will increase, along with demands for more timely, reliable, and accurate data. The modernization program for TNDS/EQ will be essential for these new business requirements.

## REFERENCES


1. R. F. Grantges, V. L. Fahrmann, T. A. Gibson, and L. M. Brown, "Total Network Data System: Central Office Equipment Reports for Stored Program Control Systems," B.S.T.J., this issue.
2. D. R. Anderson and M. J. Evans, "Total Network Data System: Performance Measurement/Trouble Location," B.S.T.J., this issue.
3. D. H. Barnes and J. J. O'Connor, "Total Network Data System: Small Office Network Data System," B.S.T.J., this issue.
4. C. J. Byrne, D. J. Gagne, J. A. Grandle, Jr., and G. H. Wedemeyer, "Total Network Data System: Data Acquisition and Near-Real-Time Surveillance," B.S.T.J., this issue.
5. M. S. Hall, Jr., J. A. Kohut, G. W. Riesz, and J. W. Steifle, "Total Network Data System: System Plan," B.S.T.J., this issue.
6. P. V. Bezdek and J. P. Collins, "Total Network Data System: Trunking Systems," B.S.T.J., this issue.
7. W. S. Hayward and J. Moreland, "Total Network Data System: Theoretical and Engineering Foundations," B.S.T.J., this issue.


## AUTHORS


**N. Dudley Fulton,** B.S.E.E. (Electrical Engineering), 1967, Purdue University; M.S. (Computer Science), 1972, Ohio State University; Bell Laboratories, 1972—. Mr. Fulton has worked on several components of the Total Network Data System family including TDAS, SXS COER, 5XB COER, and TRFS. He is currently Supervisor of the Trunk Forecasting Software Development group. Member, Tau Beta Pi, Eta Kappa Nu, IEEE, and ACM.

**James J. Galiardi,** B.S.E.E. (Electrical Engineering), 1966, Illinois Institute of Technology; M.S.E.E., 1968, Ohio State University; Bell Laboratories, 1966—. From 1966 to 1970 Mr. Galiardi was located in the Columbus Laboratories and was involved in the initial hardware and software design of the 4A ETS toll switching machine. In 1970 he became involved in mechanizing various switching machine laboratory support functions in minicomputer hardware and software. From 1971 to 1973 his work responsibilities included hardware diagnostic programming for Common Channel Interoffice Signaling. In 1973 he was promoted to a development supervisor position in the Total Network Data System Project and relocated to New Jersey. From 1973 to the present Mr. Galiardi has held various supervisory positions within TNDS, ranging from system planning through the development process to system test. Member, Eta Kappa Nu, Tau Beta Pi.

**Edward J. Pasternak,** A.B. (magna cum laude, Engineering Sciences), 1957, Harvard University; M.S.E.E., 1962, E.E. (Professional), 1966, Columbia University; Bell Laboratories, 1962—. Mr. Pasternak was initially assigned to an Exploratory Development group which was examining stored program control methods for modernization of SXS. With the inception of the Traffic Service Position System (TSPS) project, he was assigned development responsibilities for the TSPS Audit programs. In 1966 he was promoted to Supervisor of the TSPS Audit and Recent Change Programs Group. Following cutover in 1969 of the first TSPS office in Morristown, New Jersey, Mr. Pasternak was appointed Head of the Stored Program Control (SPC) Development Department with hardware and software design responsibilities for the SPC 1A processor. In 1973 Mr. Pasternak was transferred to the Business Information Systems area of Bell Laboratories as Head of the BISCUS/FACS System Test Department. In 1974 he became Head of the Servicing and Estimating Department, with design responsibilities for several systems that were subsequently incorporated into the Total Network Data System. Mr. Pasternak currently has responsibilities for the TNDS Equipment Systems component of the Total Network Data System. Member, IEEE, ACM.

**Stewart A. Schulman,** B.S. (Computer Science), 1972, City College of New York; M.S. (Computer Science), 1974, Stevens Institute of Technology; Bell Laboratories, 1972—. Mr. Schulman began his career at Bell Laboratories in the System Test area of the Service and Estimating (S&E) department, which later evolved into the Total Network Data Systems—Equipment (TNDS/EQ) development organization. Following his initial assignment, he was involved in the development of various TNDS/EQ component systems. In 1979, Mr. Schulman was promoted to Supervisor, responsible for ongoing development of the Traffic Data Administration System (TDAS) and the Load Balance System (LBS). His current assignment is development Supervisor for the On-line Records and Reporting System (ORRS), an on-line replacement for an existing TNDS batch system and the vehicle for future TNDS/EQ modernization.

**Herman E. Voigt,** B.S.E.E., 1962, Polytechnic Institute of Brooklyn; M.S.E.E., 1964, Columbia University; Bell Laboratories, 1962—. Mr. Voigt's initial work was in exploratory development of Common Control for Step-by-Step local switching offices. From 1963 to 1973 he was involved in the development of the first Traffic Service Position System (TSPS). His prime areas of contribution were in the software supporting the operator actions call processing functions, for which he received two patents, systems test and eventually field support of all installations. Since 1973 he has been supervising the development of various operation support systems that provide administrative and engineering equipment utilization reports for local electromechanical and electronic switching offices. Member, Eta Kappa Nu, Tau Beta Pi, IEEE Computer Society.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Trunking Systems

By P. V. BEZDEK* and J. P. COLLINS*

(Manuscript received August 3, 1982)

The Total Network Data System/Trunking (TNDS/TK) systems are those modules of TNDS that support the engineering and administration of the message trunk network. TNDS/TK consists of the Trunk Forecasting System (TFS), the Trunk Servicing System (TSS), and Common Update/Trunking (CU/TK). Using representative trunk group loads and switching system growth data as input, TFS forecasts trunk needs for five future years. TSS fine tunes the first year of the forecast by showing where to rearrange the network to meet current demand. And CU/TK supports TSS and TFS with a record base that contains a description of the network and user-stated parameters. We describe TNDS/TK from the standpoint of its environment, functions, system internals, and future direction. We also present a high-level view of its algorithms. For more detail, the reader is referred to the "Theoretical and Engineering Foundations" article and its references in this issue.

## I. THE CIRCUIT ADMINISTRATION CENTER

### *1.1 General*

The Total Network Data System/Trunking (TNDS/TK) systems support the activities of people in the Circuit Administration Centers (CACs). These are work centers in the Bell Operating Companies (BOCs) that are responsible for engineering and administering the

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

message trunk network. Each BOC has at least one CAC; large companies may have several CACs with responsibilities divided geographically.

The CAC functions consist of (1) forecasting the required size and placement of trunk groups over five future years (trunk forecasting), (2) determining how best to modify the existing network to satisfy current demand (trunk servicing), and (3) initiating and monitoring the work orders that result in rerouting traffic from one trunk group onto another (route administration). TNDS/TK directly supports trunk forecasting and trunk servicing. However, it does not directly support route administration, although it does provide a major input to routing by defining, in forecasting, where reroutes are planned.

For a description of trunk servicing and trunk forecasting that is more comprehensive than that given below, the reader is referred to the "Environment and Objectives" article in this issue.

### *1.2 Trunk servicing functions*

By sending Traffic Measurement Requests (TMRs) to TNDS/EQ, the trunk servicer indicates needs for the collection of traffic data on each trunk group throughout the year and frequently during the expected busy season. Traffic measurements that do not satisfy validation tests or represent normal customer behavior must be excluded from use. The servicer must initiate action to correct problems in the data collection process so that subsequent weeks' data can be collected successfully. The accepted measurements are used to estimate each trunk group's offered loads, current service levels, and the number of trunks required to provide objective service.

The servicer issues and tracks Message Trunk Orders (MTOs) to add or disconnect trunks to maintain objective levels of service and utilization. Representative busy-season loads, called "base" loads, must also be selected for each trunk group to support the forecasting process. Annually, Trunk Administration Measurement Plan (TAMP) reports are produced, which describe the adequacy of trunk-group data collection and compare the actual trunk network with a theoretical, low-cost network that produces the desired level of service and utilization.

### *1.3 Trunk forecasting functions*

Trunk forecasting determines the future size and location of trunk groups as a major input to the construction program. Although the generation of the forecast is mechanized by TNDS/TK, the forecaster must manually determine and monitor much of its input. The activities this involves are described below.

The forecaster must define central office growth, typically in terms of main stations and traffic volume per main station. This person must specify the presence or absence of trunk groups, based on switch plans, homing arrangements, switch capacities, costs of facilities, tariff changes, and marketing demand forecast changes. Also based on these sources, the forecaster must indicate where the major shifts in traffic load will occur and select the appropriate load projection and sizing algorithms. The forecaster must perform these functions for scheduled forecasts and, often on short notice, for unscheduled ones in support of new switch plans or other construction program constraints. Finally, the forecaster must make input corrections that will reconcile future forecasts with current load variations that servicing reacts to with unplanned MTOs.

## II. A FUNCTIONAL DESCRIPTION OF TNDS/TK

### *2.1 Overview*

Figure 1 shows that TNDS/TK is a batch system made up of three component systems. The Trunk Servicing System (TSS) supports the servicing functions described above. Except for that part of the system related to annual TAMP, TSS is usually run once a week. The Trunk Forecasting System (TFS) performs most of the calculations involved in producing a General Trunk Forecast. It consists of several modules that may operate independently, though all are run between two and four or more times per year. Supporting TSS and TFS is Common Update/Trunking (CU/TK), which maintains a record base that stores a network description and associated parameters.

Separate installations of TNDS/TK are present in each BOC. Depending on the size of the company, an installation may process data for as few as 4000 to more than 100,000 trunk groups.

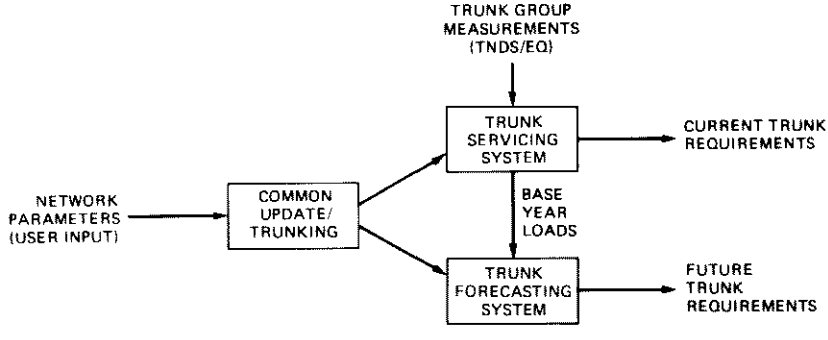



Fig. 1—TNDS/TK components.

### 2.2 Common Update/Trunking (CU/TK)

#### 2.2.1 General

CU/TK is the interface between the user (the servicer or forecaster) and the rest of the system. Its record base stores a description of the network, engineering and administrative parameter values, some intermediate calculations, revisions to calculations, and some traffic loads. CU/TK provides 27 reports that describe its content. It also validates the input and advises the user of actual and potential errors.

#### 2.2.2 Record base content

CU/TK data fall in two broad categories: those that apply to the company as a whole and those that apply to a portion of it, possibly to a single switch or trunk group. Normally, the first type is created and maintained by a person with company-level responsibility. The second is the obligation of the individual servicers and forecasters.

The company-level information is global in nature and serves several purposes. It may specify company policy or provide a key processing control. A single company input may also obviate numerous identical inputs by individual users. Among other things, company-level inputs define Bell System Common Language name change associations, report formatting and distribution criteria, engineering options, and the forecast years for TFS.

The servicer/forecaster-level information is more detailed and network dependent. *Traffic Unit Records* define the nodes in the network. They give the start and end dates of switching systems or NXX's that terminate trunk groups or for which growth data are to be stored. These records also allow forecasters to override the company-level specification of minimum trunk group size, described later. *Circuit Group Records* define the links in the network. They specify the life spans of trunk groups, load projection and sizing algorithms, trunk servicing criteria and defaults, base year loads, and alternate routes. Figure 2 shows a sample Circuit Group Record (TU500). *Growth Records* contain the main station and traffic forecast data that apply to central offices and are used in projecting trunk group loads. *Traffic Transfer Records* contain a description of the network impacted by area transfers, rehomes and reroutes, and the loads involved. In addition, there are inputs to revise intermediate TFS calculations, request nonautomatic reports to analyze the forecast, and specify that TSS ignore certain weeks of nonrepresentative traffic data.

#### 2.2.3 Input validation

CU/TK verifies that the inputs contain the proper character set (field validation) and that all data on a record are consistent (intra-record analysis). Consistency *between* records is verified downstream

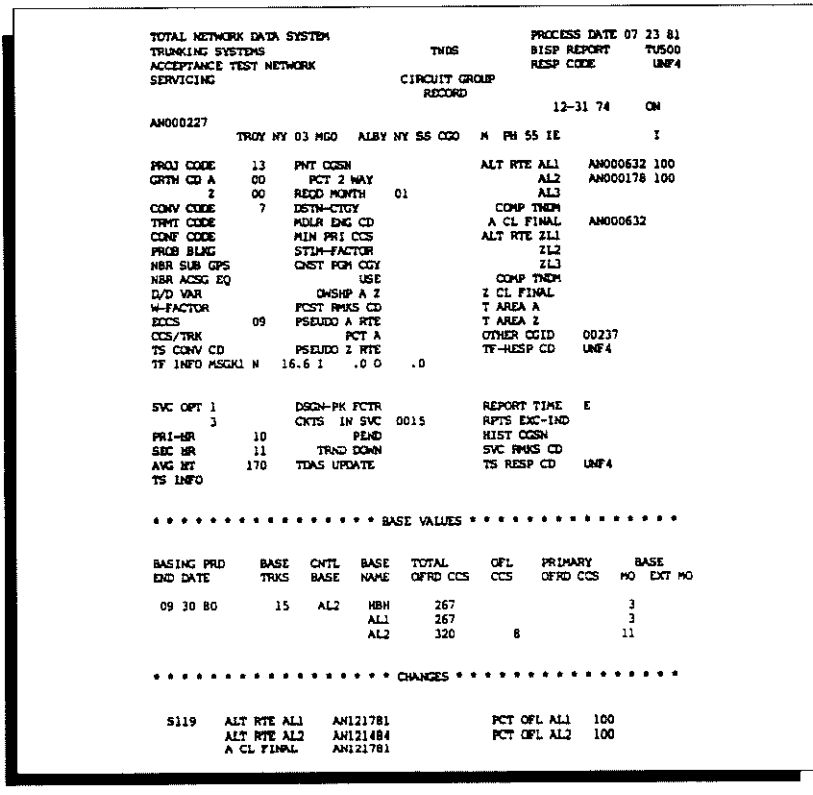



Fig. 2—Circuit Group Record (TU500).

in TSS and TFS. CU/TK rejects inputs that fail its analysis and notifies the user with a Record Update Errors Report (TU001) and several statistical summaries.

### *2.3 The Trunk Servicing System (TSS)*

The following sections outline the major functions of TSS. Representative processes within these functions are described in some detail, illustrating the types of processing that TSS performs.

#### *2.3.1 Cycle setup*

Although a TSS cycle can only process traffic measurements taken during a single week, the data received from the Traffic Data Administration System (TDAS) can pertain to several weeks. (The different technologies employed in data collection, ranging from photographic to electronic, produce varied delays before the measurements are made available to TSS.) The TSS cycle-setup function therefore separates the measurements that pertain to a user input "study week" from

those that pertain to other weeks. The study-week measurements are passed to the measurement validation function; the others are held for later runs of the system. Measurements that pertain to those weeks before the study week are reported to the users so that back-dated cycles can be run as appropriate.

The CU/TK circuit group records contain (at one time) a time-varying description of the trunk network. The cycle setup function also extracts a static description of the trunk network, appropriate for the study week to be processed.

#### *2.3.2 Measurement validation*

The systems that collect traffic measurements and deliver them to TSS do not eliminate machine or human error. TSS must attempt to keep erroneous measurements away from its load-estimation processing. Although certain measurement errors generate absurd data, other errors can create apparently normal data or unlikely but theoretically possible data. The TSS validation function screens out measurements statistically likely to be in error. Rejected and missing measurements are reported to the users, so that problems in the measurement collection systems can be identified and corrected.

A few sample validations are described below. These validation tests are performed for each hour in which the appropriate measurements are available. The measurements discussed here are defined in the "Theoretical and Engineering Foundations" article earlier in this issue.

The peg count ($PC$) and overflow ($OFL$) measurements taken at one end of a trunk group are compared. If

$$OFL \geq PC > 0,$$

then both measurements are rejected. It is not valid for $OFL$ to exceed $PC$. Although they can theoretically be equal and positive, most cases of such measurements are caused by measurement error.

When usage ($U$) is measured at both ends of a trunk group, the measurements need not agree, because of the sampling technique used. Based on a few assumptions about holding times and random calling patterns, each measurement should be an approximation to the true usage on the group, and hence to the other measurement. Specifically, if

$$(U_1 - U_2)^2 > (200 \text{ seconds})(U_1 + U_2),$$

TSS determines that the two measurements are not sufficiently close for both to be acceptable. It rejects the lesser measurement because typical problems in collecting usage data produce undervalued measurements. If, however, the two measurements are sufficiently close, TSS uses their average as its estimate of true usage.

### 2.3.3 Load estimation

TSS can estimate the average load offered to a trunk group from any of five measurements, or from two particular combinations of these measurements. The specific calculation used for a trunk group depends on trunk group type and the availability of measurements after the validation function. In most cases, several parameters that describe the traffic on a group and the group's performance are calculated, rather than average offered load alone. These parameters are determined from up to five hours' data, representing a fixed clock hour for five days of a business week. Thus, up to 24 sets of estimates, one per measured clock hour, are produced for the business days of the study week.

If, for example, usage, peg count, and overflow measurements are available from the same hour for the appropriate hours, TSS estimates the offered load ($a$) for each such hour as

$$a = \frac{PC - OFL \times R}{PC - OFL} U,$$

where $R$ is based on trunk group type and accounts for customer retrial of blocked calls. This equation is based on the assumption that calls blocked by the trunk group and not retried would have the same average holding time as completed calls. These estimates of offered load are averaged over five days to produce study week hourly average offered loads. The variance among the daily offered loads in each clock hour is computed and stored, as well as estimates of blocking, average holding time, and peakedness.

If only usage measurements are available for a trunk group because of equipment limitations or lost or rejected data, TSS executes a more time-consuming algorithm that produces fewer, less reliable results. First it averages the usage measurements in each clock hour over the measured days. Then it uses standard numerical-analysis convergence techniques around a program that calculates expected usage, given offered load, to find an offered load that corresponds to the average measured usage. Blocking is computed as a by-product of this process. But average holding time and peakedness cannot be computed in this case. In fact, a user estimate of peakedness (or, if necessary, a TSS default value) is needed as an input to the process.

### 2.3.4 Study period formation

To increase the reliability of its traffic estimates, TSS must produce averaged parameter values that cover study periods of up to four weeks, again for each measured clock hour. The values from the current study week are averaged with values from preceding study weeks, weighted by the number of measured days for each hour in

each week. As a result, the measured days are effectively averaged with all equal weights. The function outputs up to 24 sets of averaged estimates, one per measured clock hour.

Many groups are not measured every week. If a group is measured during the study week, TSS normally uses prior weeks' data (at most eight weeks older) to form averages of four measured weeks. Servicers can optionally submit Administrative Period Control inputs to CU/TK, inhibiting TSS from using past data, in cases where the traffic offered to the group is changing significantly (caused, for example, by seasonal variations in demand). If a group is not measured during the given study week, but during one or more of the preceding three weeks, TSS estimates "current" study period loads equal to the previous study period loads, for use in selecting busy hours.

#### *2.3.5 Busy hour selection*

From these hourly study-period load estimates, TSS must select a particular hour's load for which to size each trunk group. The network would obviously satisfy service objectives in all hours if each group were sized to satisfy its own service objective in its own busy hour (where "busy" is suitably defined). But because traffic can overflow from one trunk group to another, and unused capacity in one group can relieve a nearby overloaded group, a network engineered in this way would be more costly than necessary. For economic reasons, the hour for which a trunk group should be sized, the administrative hour, depends on loads offered to that group and the surrounding network.

The TSS process for selecting busy hours, Significant Hour Engineering, involves clusters of trunk groups as well as individual groups. A cluster is a final trunk group, together with all the surrounding high-usage groups that (1) share one (fixed) endpoint with the final trunk group, and (2) overflow traffic directly or indirectly to the final trunk group based on the alternate-routing logic of the switch at the shared endpoint.

In cases where all the groups in a nontrivial cluster (i.e., more than one trunk group) have study period load estimates available, TSS computes a cluster load for each hour by summing the carried load on the high-usage groups and the offered load on the final group, in corresponding hours. The hour with the greatest average cluster load per measured trunk is identified as the cluster's busy hour and the "control hour" of the final group. And, the hour with greatest measured blocking on the final group is termed the final group's "service busy hour." These two selected hours may coincide.

Selecting busy hours for high-usage groups is, from a logical view, recursive. Each high-usage group has a set of "significant" related groups defined in the CU/TK circuit group records, consisting of (1)

the groups in its alternate route(s) and (2) its cluster final(s) (the plurals here are applicable to two-way high-usage groups only). The set of control hours of the significant groups is computed and then identified as the set of significant hours for the high-usage group itself. Then the significant hour with the greatest load on the high-usage group becomes the group's control hour.

For example, in Fig. 3, there are three clusters: groups AB and AD, BC and BD, and CD. Based on greatest average cluster load per measured trunk, the control hours for groups AB, BC, and CD might be hours 7, 8, and 12, respectively. The significant hours for group BD are now known to be 8 and 12. If the load on BD in hour 8 exceeds that in hour 12, then the control hour for BD is hour 8. The significant hours for group AD are now known to be 7 and 8. If the load on AD in hour eight exceeds that in hour seven, then the control hour for AD is hour eight. An even greater load on AD in hour 9 or 12 would not be considered.

#### *2.3.6 Trunk group sizing*

TSS computes the number of trunks required to provide objective service on final groups, based on average offered load, peakedness, and day-to-day variation. Required-trunks values are calculated for the traffic in the group's control hour and in the group's service busy hour. TSS chooses the larger required-trunks value as the value for the current study period, and the associated hour is called the group's administrative hour.

High-usage groups are sized for economic, rather than service-based, criteria. Using control-hour traffic data, TSS determines the number of high-usage trunks required to minimize the combined cost of the direct and alternate routes.

The Circuit Group Servicing Record report shown in Fig. 4 is produced after the trunk group sizing process. The bottom part of the report page shows a rough graph of trunks in service and study-period trunks required against time, for the past 65 weeks. The top left gives

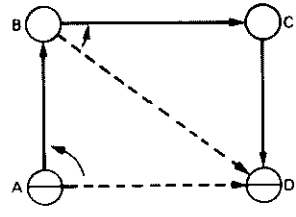



Fig. 3—Sample network for busy hour selection. Solid lines indicate final trunk groups, and dashed lines high-usage trunk groups. Curved arrows show the alternate routing used for calls overflowing a high-usage group.

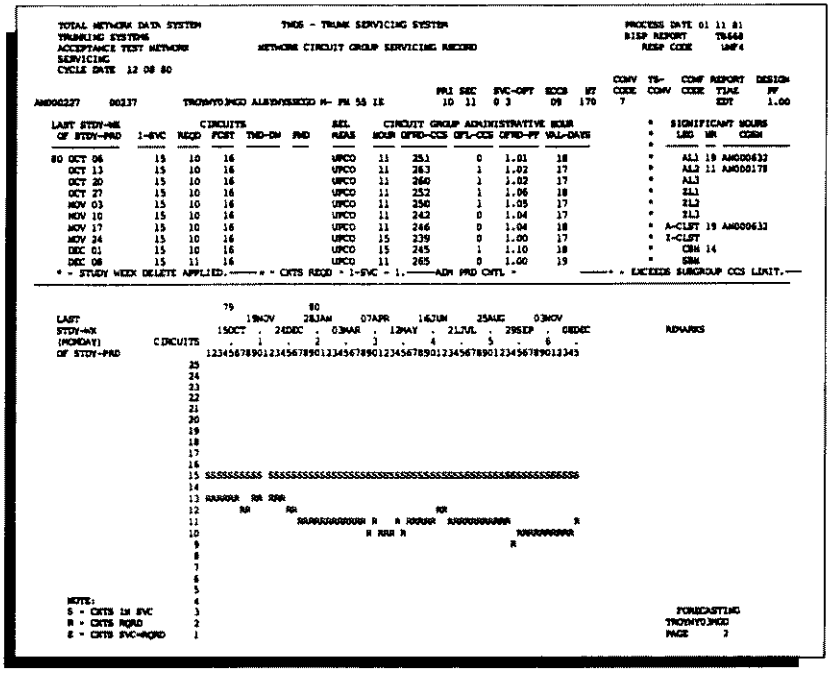



Fig. 4—Circuit Group Servicing Record.

more detail on administrative hour load conditions for the past ten study periods. The top right shows the significant hours identified for the current study period.

### *2.3.7 Trunk group banding*

Based on observed blocking on final groups, and comparison of trunks in service with trunks required on high-usage groups, TSS assigns a "service band" to each measured group for each study period. The five bands identify groups that are overprovided (underloaded), possibly overprovided, close to objective, possibly underprovided (overloaded), and underprovided. The thresholds that separate these bands are adjusted for the statistical reliability of the data available for a trunk group. For example, small trunk groups with a few days' data available would be expected to have wide variations in their TSS-calculated load parameters. For such groups, the limits of the "close to objective" band are wider than for other groups.

In each study period groups judged to be overloaded are reported to the servicers. However, groups may appear underloaded simply because they have (currently) spare capacity needed for an upcoming busy season. Therefore, TSS only displays underloaded groups for

corrective action if such groups have been underloaded for their entire banded history (up to 65 weeks).

The reports of overloaded or underloaded trunk groups from this function advise the users of significant service or utilization problems. Servicers must then determine what actions are appropriate, if any, or if only a transient condition in loads or measurements has occurred. The Circuit Group Servicing Record, as well as reports of weekly average load data, daily measurements, and measurement validation results, are available to support the servicers in this task.

#### *2.3.8 Base selection*

Just as TSS must select an appropriate current load from all the hourly loads in a study period to do trunk group sizing, so must TFS have an appropriate future load from all the hourly loads in a future forecast year to forecast future trunk requirements. The TSS base selection function executes a process similar to busy hour selection, starting with the hourly loads in a user-specified span of study periods. It outputs the most significant trunk group loads and hours in the specified base period, so that TFS can later estimate the loads in corresponding hours of future years.

Servicers can input Extended History Delete transactions to CU/TK to exclude a specified study period's data for a specified trunk group from the base selection function. This feature allows the user to prevent data that represents unusual traffic, or errors in measurement, from affecting the trunk forecast. The user can also replace any program-generated base load with a human-generated load before TFS is run, after reviewing the Base Selection Details Report.

#### *2.3.9 Network optimization*

After base selection has determined the significant trunk group loads from a twelve-month historical period, the TSS network optimization function determines a low-cost trunk network that should have been in place to handle these loads. Unlike the trunk group sizing function, which calculates the required size of each group for its actual, measured load, the optimization function estimates the theoretical load on overflow-receiving groups, to adjust for proposed trunk changes in the surrounding network, and sizes these groups accordingly. Optimization can therefore suggest widespread, related changes in trunk-group sizes but it does not suggest adding or removing trunk groups.

#### *2.3.10 Trunk Administrative Measurement Plan (TAMP) reporting*

In support of TAMP, the AT&T plan to index the performance of the trunk administration process, TSS produces reports annually from

base selection and network optimization outputs. Each final group is assigned to one of the five bands previously described, according to its blocking in its annual busy hour. In addition, every group is assigned to a band, based on a comparison of trunks in service (in the annual busy hour) and optimized required trunks. The distribution of trunk groups among these bands is reported, by administrative responsibility and by trunk group type (end office, operator connecting, tandem, tandem connecting, and auxiliary services). Also, the TAMP reports list and summarize the number of days' data available to compute each group's busy hour load.

### *2.4 The Trunk Forecasting System (TFS)*

The principal output of TFS is the General Trunk Forecast, a document that is a major input to the BOC construction program. A company-created forecast period table controls the span and complexity of the forecast. Through the table, the company defines the five forecast years, the first four of which must be consecutive and are normally chosen to be in the immediate future. For planning purposes, the company may choose a more distant fifth year, perhaps 20 years ahead.

The company also uses the table to define "base year subdivisions" and "forecast periods." These are partitions of the base (current) year and forecast years that tell TFS how much detail to create. Up to 20 forecast periods (normally quarters for each forecast year) and four base year subdivisions may be specified. TFS assumes conditions that exist on the last day of a forecast period exist throughout the period, so the importance of defining more than one forecast period per year is clear. Single annual periods would result in an improperly sized network if major events such as area transfers did not coincide with busy seasons, as frequently occurs. Although a company may opt for numerous forecast periods, it does so at the expense of computer run time which is proportional to the number of periods chosen.

The sections below describe the major functions of TFS, including user interactions, reports, and a high-level view of the calculations. Following these is a description of the sequencing of operations considering both TFS and CU/TK.

#### *2.4.1 Growth factor computation*

To forecast traffic load on trunk groups, TFS requires knowledge of the growth in load of the switches that the trunk groups terminate on. For each such switch, the forecaster enters growth data through CU/TK. The data are usually in the form of main stations (MS) and hundred call-seconds per main station (CCS/MS), as of a given date. The data may represent all customers served by the switch, or a subset

of the customers according to NXX or class of service, e.g. coin, residence, etc. The data are entered normally by year to cover the base year and all forecast years.

TFS uses these data to compute growth factors in the form of ratios of future load to present load. First, TFS assures that data are available for each base year subdivision and forecast period end date; when the data have not been expressed to coincide precisely with the needed dates, TFS derives what it needs by linearly interpolating from surrounding data. Next, the data are aggregated to the appropriate level for forecasting trunk group loads. For example, if the forecaster input data by NXX but required growth factors for only the total switch (comprising several NXX's), then the NXX data would be summed for the entire switch.

Products of the like-dated individual growth data terms are then formed, e.g., MS and CCS/MS are multiplied to form CCS. Finally, by simple division of a future product by a base year product, growth factors are produced. For each switch, one growth factor is computed for each forecast period relative to each base period (as defined by the base year subdivisions). The forecaster has the option of bypassing this process by manually stating growth factors, if computed ones are inappropriate.

With calculations complete, the system outputs a Growth Factor Report. At this point, errors in the input data are most evident. To avoid the need to rerun the process with new inputs, the system allows the forecaster to substitute new growth factors for the incorrect ones through CU/TK.

#### *2.4.2 Network disassembly*

TSS base selection provides TFS with trunk group loads that represent busy hour and busy season conditions during the base year. TFS must convert these loads to a form appropriate for the projection function. For the most part, this amounts to removing overflow traffic from the loads of groups that are alternate routes for other (high-usage) trunk groups. This "disassembly" is necessary, since the overflow is a function of a previous network structure and must not be projected with the (network-independent) first-route traffic.

For each high-usage group, TSS base selection identifies the load offered to the group and its overflow. Using the alternate route information on Circuit Group Records, TFS associates the overflow with the appropriate alternate route groups. (When the overflow is fragmented over multiple alternate routes, TFS allocates a portion of the overflow to each route according to user-specified percentages.) TFS then subtracts the overflow values from the offered loads of the receiving group.

Because of differences in data collection schedules and measurement errors, a group's received overflow may appear to exceed its offered load. In such a case, TFS will assume a 0 value for the first route load. If the forecaster anticipates this and concludes that a 0 value is an understatement, the forecaster may input a "minimum primary CCS" that will override any lower computed value.

#### *2.4.3 Projection*

With first route loads and growth factors as input, TFS estimates future load. The first step is to compute projection ratios. These are created by substituting the growth factors into a formula that the forecaster has selected for each group. For example, if the AT&T recommended formula $(A + Z)/2$ were selected, the projection ratio would reflect the average growth of the originating end ($A$-end) and terminating end ($Z$-end) of the group, as determined by $A$'s and $Z$'s growth factors.

Of the total set of growth factors available for a specific $A$ and $Z$, the ones used in the formula are those developed for the base year subdivision and forecast period containing the measurement period (busy month) of the base load. So if a trunk group had a base load with a May busy month, and the base year and forecast years were calendar years with a quarterly base and quarterly forecast periods, then the growth factors used would be those that apply to the second quarter of each forecast year relative to the second quarter of the base year.

The multiplication of the base loads by the projection ratios completes projection. In a similar manner, the loads stated on traffic transfer records are projected.

The system allows several options that the forecaster can exercise in advance of a projection run. Instead of relying on growth factors, the forecaster may state manually derived projection ratios or specify an annual percent for compounding. The forecaster may supply a stimulation factor or select a projection formula that accounts for community-of-interest considerations or regional growth. For a specific $A$ or $Z$, the forecaster may request the use of growth factors that apply to only a part of the switch. Using "pseudo" trunk groups, described later, the forecaster may project separately the individual traffic items on a group. Finally, if all the system-provided methods are inappropriate, the forecaster may state future loads that are externally derived.

#### *2.4.4 Sizing*

Disassembly, projection, and sizing are run as a single process; the forecaster has the opportunity to review all calculations only after

required trunks are computed. The same blocking objectives and economic criteria are used in TFS sizing as in its TSS counterpart. In general, TFS computes one trunks-required value per group per forecast year based on the load in the forecast period that contains the group's busy month. For office(s) affected by major event(s) such as cutovers, however, the forecaster may specify that required trunks be computed as of the date of the event(s). This will result in more than one trunks-required computation per year (except where busy months and event dates fall in the same forecast period).

As a first (logical) step, the process adds together (1) projected first route loads, (2) projected traffic transfer CCS values (+ or −), and (3) user-stated load adjustments (+ or −), if any. Transfer loads are needed since projected trunk group loads alone may not compensate for the effects of reroutes and new or deleted groups that occur after the base year. Next, the process sizes those groups that receive no overflow (only-route and primary high-usage groups). Overflow from the primary high-usage groups is then computed.

The rest of the procedure is the inverse of disassembly. Overflows are added to the offered loads on the alternate routes. When a group receives the overflows from all expected sources, the process computes the peakedness of its load and its required trunks. If the group is high usage, overflow and variance are computed. The previous steps are then repeated. The result is that the network is sized iteratively, bottom-up through that portion of the five-level network hierarchy included in the company's database. The process will add to the computed size of a group a trunk adjustment (+ or −), if any is stated; for high-usage groups this is performed before overflow is computed.

To facilitate more economic purchasing of trunk equipment, TFS will convert the above values to modules of 12 or 24 trunks. The forecaster, however, must request this action on a group-by-group basis. Which of four modular sizing procedures is invoked depends on whether the group is high usage, one way or two way, or uses digital facilities or digital terminal equipment. Modular sizing takes place before computing overflow from high-usage groups.

A major function imbedded in sizing is the determination of where new trunk groups are warranted. Although TFS mechanizes the test for new groups, the forecaster must identify candidates in advance by creating what are called "pseudo" Circuit Group Records in CU/TK, complete with base loads. To test fully all possibilities, the forecaster should input large numbers, perhaps thousands, of pseudo groups. But the practical limitations of deriving base loads and specifying all the necessary parameters reduce the number created to a small subset of those possible.

TFS projects the pseudo trunk group base load as if the pseudo

group were an actual group. The sizing process calculates required trunks for it and determines whether a minimum-trunk-group-size threshold is exceeded. If so, the pseudo group is retained as a planned group (one that will appear in some future year of the forecast). If not, the pseudo group's loads are distributed onto existing groups specified by the forecaster. This is necessary since the pseudo group's loads represent actual traffic that is not accounted for elsewhere and would otherwise be lost.

Apart from planning new groups, pseudo groups give the forecaster more flexibility in projecting traffic. The forecaster may remove the load of one or more traffic items from an existing group's base load, define pseudo groups for them, and project them by different methods. If the forecaster indicates that the pseudo groups were created solely for separate projection, sizing will suppress the minimum-size test and add the pseudo groups' projected traffic back to their existing route(s).

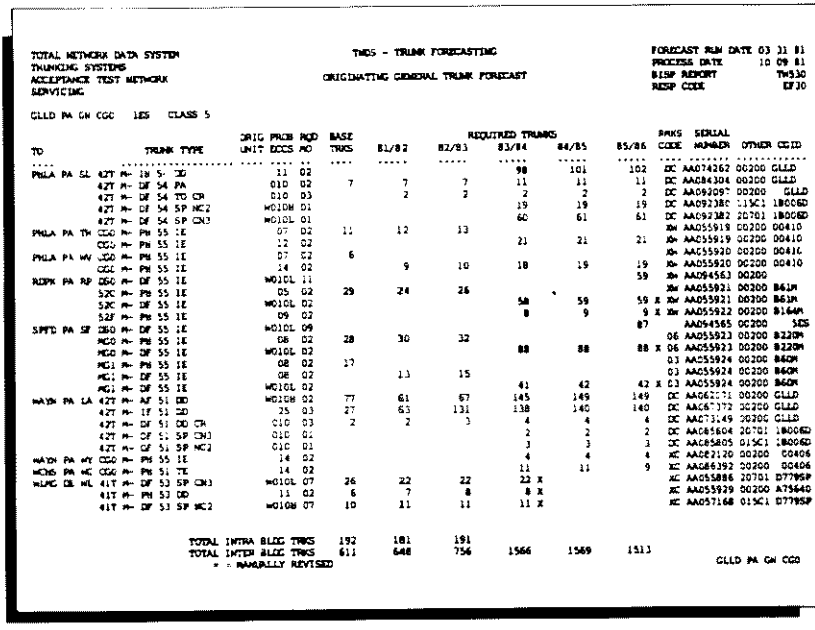
The principal user output from sizing is the Preliminary General Trunk Forecast (PGTF). In addition to trunks, it displays several of the major intermediate calculations: base and future offered CCS, projection ratios, and peakedness. For each group, the report indicates, among other things, whether it was a pseudo, is modularly engineered, received overflow, or includes transfer loads. At this point, the forecaster may revise only future offered CCS and required trunks, as none of the other items appear on the final TFS reports. Since the analysis may be complicated, the forecaster may request up to three additional reports. The Forecast Detail Report, Detail Traffic Transfer Summary, and Overflow Summary supplement the PGTF with the information their titles imply.

With revisions complete, TFS produces its final reports. The General Trunk Forecast (GTF) may be issued in several forms, combining the newly computed trunk values with absolute differences or percent differences from previous forecasts. Figure 5 shows a GTF. The Office Trunk Studies display load and trunk values in a way useful for central office equipment engineering, and the Construction Program Worksheets provide a starting point for preparing AT&T construction program reports.

TFS also produces a computer tape of forecast results that is input to another Bell Laboratories product, the Facility and Equipment Planning System (FEPS), and BOC-developed programs. FEPS is a module of the Trunks Integrated Record Keeping System (TIRKS).

#### *2.4.5 Sequencing with CU/TK*

Figure 6 shows the relationship between CU/TK and TFS in a complete forecast run. The diagram refers to two TFS functions that have not been described to this point. Growth Factor Analysis (GFA)



TOTAL NETWORK DATA SYSTEM
TRUNKING SYSTEMS
ACCEPTANCE TEST NETWORK
SERVICING

TNDS - TRUNK FORECASTING

ORIGINATING GENERAL TRUNK FORECAST

FORECAST RUN DATE 03 11 81
PROCESS DATE 10 09 81
BISP REPORT TM530
RESP CODE EF30

GLLD PA GN CGO 1ES CLASS 5

| TO | TRUNK TYPE | ORIG UNIT | PROB ECCS | RQD AD | BASE TRKS | REQUIRED TRUNKS 81/82 | 82/83 | 83/84 | 84/85 | 85/86 | MARKS CODE | SERIAL NUMBER | OTHER CCID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PHLA PA SL | 42T M- IW S- 3D | | 11 | 02 | | | | 98 | 101 | 102 | DC | AA074262 | 00200 GLLD |
| | 42T M- DF 54 PA | | 010 | 02 | 7 | 7 | 7 | 11 | 11 | 11 | DC | AA084304 | 00200 GLLD |
| | 42T M- DF 54 TO CR | | 010 | 03 | | 2 | 2 | 2 | 2 | 2 | DC | AA093097 | 00200 GLLD |
| | 42T M- DF 54 SP MC2 | | W010M | 01 | | | | 19 | 19 | 19 | DC | AA092380 | 11SC1 1B006D |
| | 42T M- DF 54 SP CN3 | | W010L | 01 | | | | 60 | 61 | 61 | DC | AA092382 | 2D701 1B006D |
| PHLA PA TH CGO M- PM 55 IE | | | 07 | 02 | 11 | 12 | 13 | | | | XM | AA055918 | 00200 00410 |
| | CGO M- PM 55 IE | | 12 | 02 | | | | 21 | 21 | 21 | XM | AA055919 | 00200 00410 |
| PHLA PA WV JGO M- PM 55 IE | | | 07 | 02 | 6 | | | | | | XM | AA055920 | 00200 00410 |
| | CGO M- PM 55 IE | | 14 | 02 | | 9 | 10 | 18 | 19 | 19 | XM | AA055920 | 00200 00410 |
| RDPK PA RP DBO M- DF 55 IE | | | W010L | 11 | | | | | | 59 | XM | AA094563 | 00200 |
| | 52C M- PB 55 IE | | 05 | 02 | 29 | 24 | 26 | | | | XM | AA055921 | 00200 B61M |
| | 52C M- DF 55 IE | | W010L | 02 | | | | 58 | 59 | 59 | X XM | AA055921 | 00200 B61M |
| | 52F M- PM 55 IE | | 09 | 02 | | | | 8 | 9 | 9 | X XM | AA055922 | 00200 B164M |
| | | | | | | | | | | 87 | | AA045465 | 00200 SDS |
| SPFD PA SF IBO M- DF 55 IE | | | W010L | 09 | | | | | | | 06 | AA055923 | 00200 B220M |
| | MCO M- PB 55 IE | | 08 | 02 | 28 | 30 | 32 | | | | | | |
| | MCO M- DF 55 IE | | W010L | 02 | | | | 88 | 88 | 88 | X 06 | AA055923 | 00200 B220M |
| | MC1 M- PM 55 IE | | 08 | 02 | 17 | | | | | | 03 | AA055924 | 00200 B60M |
| | MC1 M- DF 55 IE | | 08 | 02 | | 13 | 15 | | | | 03 | AA055924 | 00200 B60M |
| | MC1 M- DF 55 IE | | W010L | 02 | | | | 41 | 42 | 42 | X 03 | AA055924 | 00200 B60M |
| MAYN PA LA 42T M- AF 51 OD | | | W010M | 02 | 77 | 61 | 67 | 145 | 149 | 149 | DC | AA062251 | 00200 GLLD |
| | 42T M- IF 51 OD | | 3S | 03 | 27 | 65 | 131 | 138 | 140 | 140 | DC | AA067172 | 00200 GLLD |
| | 42T M- DF 51 OD CR | | C1C | 03 | 2 | 2 | 3 | 4 | 4 | 4 | DC | AA073149 | 00200 GLLD |
| | 42T M- DF 51 SP CN3 | | C1C | 01 | | | | 2 | 2 | 2 | DC | AA085604 | 2D701 1B006D |
| | 42T M- DF 51 SP MC2 | | C1C | 01 | | | | 3 | 3 | 3 | DC | AA085805 | 01SC1 1B006D |
| MAYN PA MY CGO M- PM 55 IE | | | 14 | 02 | | | | 4 | 4 | 4 | XC | AA062120 | 00200 00406 |
| MCHS PA MC CGO M- PM 51 TE | | | 14 | 02 | | | | 11 | 11 | 9 | XC | AA086392 | 00200 00406 |
| WLMG DE ML 41T M- DF 53 SP CN3 | | | W010L | 07 | 26 | 22 | 22 | 22 X | | | XC | AA058886 | 20701 D77958P |
| | 41T M- PM 53 OD | | 11 | 02 | 6 | 7 | 8 | 8 X | | | XC | AA055929 | 00200 A75640 |
| | 41T M- DF 53 SP MC2 | | W010M | 07 | 10 | 11 | 11 | 11 X | | | XC | AA057168 | 01SC1 D77958P |
| TOTAL INTRA BLDG TRKS | | | | | 192 | 181 | 191 | | | | | | |
| TOTAL INTER BLDG TRKS | | | | | 611 | 648 | 756 | 1566 | 1569 | 1513 | | | |

* = MANUALLY REVISED

GLLD PA GN CGO

Fig. 5—General Trunk Forecast.

and Circuit Group Analysis (CGA) extend the validations of CU/TK to include interrecord error checks. The philosophy behind this sequence, referring to the numbered steps in the diagram, is as follows:

1. Computer runs of CU/TK and GFA are needed to establish or correct the database for this forecast view. The forecaster receives error reports and, with this step and the ones below, is allowed time to submit corrections.

2. CU/TK is run to accept the corrections.

3. CU/TK is run to allow corrections to erroneous inputs in Step 2, GFA is run to freeze the database, and growth factors are computed.

4. CU/TK and CGA are run to accept growth factor revisions and identify any remaining interrecord errors.

5. Similar to Step 2.

6. Similar to Step 3, concluding with all remaining calculations.

7. CU/TK accepts requests for additional analysis reports, issued by TFS.

8. CU/TK accepts the forecast revisions. These are merged with the preliminary results and final outputs are produced.

Rarely, however, does a company execute the full sequence as described. To do so, allowing for forecaster interactions, would require two or three months. Typically, the company will omit Steps 2 and 5 and will merge GFA and CGA into a single step. A company may also

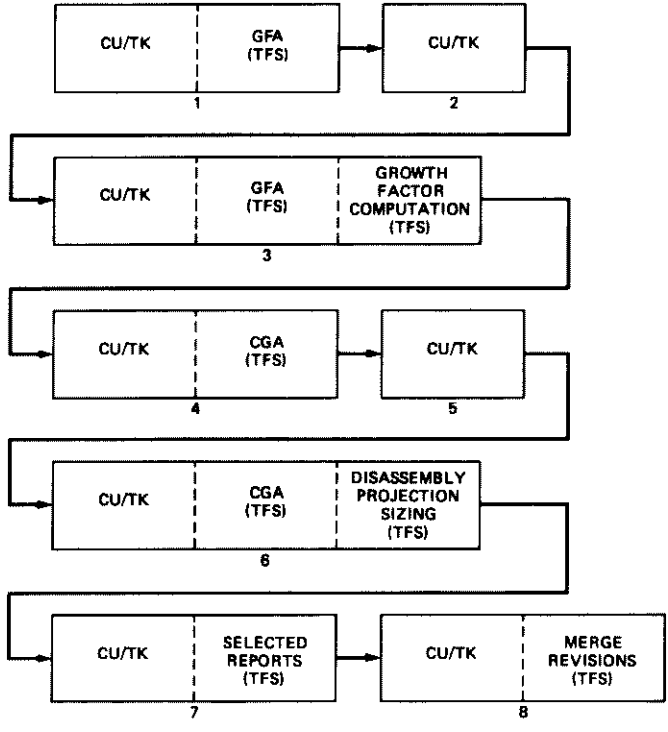



Fig. 6—Sequencing TFS with CU/TK. Arrows show the sequence of operations, not flow of data, and indicate pauses in computer run. During pauses, the forecaster reviews the intermediate output, prepares collections to the output and database, and inputs collections to CU/TK.

overlap the early steps of one forecast run with the concluding steps of the previous run. In some cases, too, only the forecast that pertains to a few switches is of interest. The company may then run TFS in the "Area Forecasting" mode, against only a selected subset of the database, and speed up the process accordingly.

## III. SYSTEM INTERNALS

The TNDS trunking systems are designed to run in batch mode on an IBM 370-series computer system or equivalent, supported by the Bell System Standard Operating Environment software. The systems are coded primarily in COBOL, with some programs in FORTRAN or PL/I. Table I describes the size of these systems in several dimensions.

The source code and development documentation for these systems

Table I—Approximate system size

| | Lines of Source Code | Compiled Source Parts | Delivered Programs | Data Files |
|---|---|---|---|---|
| CU/TK | 34,000 | 25 | 14 | 15 |
| TSS | 77,000 | 100 | 58 | 170 |
| TFS | 85,000 | 110 | 68 | 135 |

are created and maintained using the *UNIX**/Programmer's Workbench (PWB) operating system and text-processing facilities on a minicomputer system. Multiple releases of a system, in simultaneous use at different installations, are maintained conveniently using the Source Code Control System. The Change Management Tracking System monitors the status of Modification Requests through various phases of investigation and resolution.

The programs are converted to executable format, tested, and transmitted to the users' computation centers by use of a telecommunications software system for computer-to-computer data exchange with operating telephone companies (TTRAN).

## IV. LOOKING TO THE FUTURE

### *4.1 With TSS*

With a planned trunk demand servicing policy feature, TSS will more effectively separate service problems that require corrective action from transient conditions associated with normal fluctuations in load. This feature will also select the best groups for corrective action within overloaded clusters, subject to user-input facility constraints. A second feature, providing a short-term forecast of loads, will then allow TSS to recommend cost-effective solutions to service problems before they occur.

An extensive revision of the TSS load estimation and study period formation functions will provide more accurate load data and trunk requirements through (1) calculation of loads from individual rather than averaged measurements, (2) inclusion of more days' data in weekend study-period averages, and (3) greater use of calculated parameter values (such as holding time) in preference to user-estimated or theoretical values.

TSS reports will be restructured so that a lesser amount of data is output automatically to the servicers. Servicers will be able to receive more detailed supporting data on request. Currently, TSS is designed to produce all of its reports automatically, thereby providing servicers with an excess of data from which they must extract the information of interest.

* Trademark of Bell Laboratories.

### 4.2 With TFS

Over the next few years, significant attention will be devoted to improving the user interface for forecasting. There are major implications in this for both TFS and CU/TK. One thrust will be to replace a large portion of the batch operation with on-line capability. Candidates for on-line use include database update, validation and inquiry, report retrieval, revisions to calculations, and the calculations themselves.

Another major enhancement that would be particularly useful in an on-line environment is a "what if" capability. Currently, TFS produces a single forecast with a database that gives a single consistent (though evolving) picture of the network at any given time. A "what if" capability would allow the user to produce several forecasts, each for a different switch and homing configuration. The best forecast, by criteria to be defined, would result in a database update with the preferred configuration.

The companies will also be able to exchange network information in a more mechanized way than is now possible. The exchange will include CU/TK records, intermediate TFS calculations, and forecasted trunks. To do all three will require the companies to coordinate their forecast schedules. The design of this feature will be robust enough to apply to the current network partition or to other partitions required by Dynamic Non-Hierarchical Routing (DNHR) or legislation.

Other proposed enhancements include the Sequential Projection Algorithm (SPA) and a Trunk Implementation Plan (TIP). SPA will replace the current projection method with one based on Kalman filter prediction theory. TIP is a method to convert the demand forecast from TFS into an administered one, considering forecast uncertainty, expense and capital costs, trunks in service, and facility availability. DNHR is also under study as it applies to TFS.

## AUTHORS

**Paul V. Bezdek,** B.S. (Mathematics), B.A. (Physics), 1969, Wayne State University; M.S. (Computer Science), 1976, Steven Institute of Technology; Bell Laboratories, 1972—. Mr. Bezdek is currently Supervisor of the COEES/DOPS Systems Engineering Group. Previously he was responsible for developing functional requirements for the Total Network Data System/Trunking Systems. Member, Phi Beta Kappa.

**Julius P. Collins,** B.S., M.S. (Mathematics), 1974, Polytechnic Institute of New York; Bell Laboratories, 1974—. Mr. Collins has worked on design, implementation, and testing of several operations support systems for engineering and administration of the public telephone network. He is currently working on an engineering system for corporate telecommunications networks. Member, Pi Mu Epsilon.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Central Office Equipment Reports for Stored Program Control Systems

By R. F. GRANTGES,* V. L. FAHRMANN,* T. A. GIBSON,* and L. M. BROWN*

(Manuscript received February 16, 1983)

Stored Program Control System Central Office Equipment Reports (SPCS COER) is one of the earliest of the family of Operations Systems. Deployed on a centralized time-shared system, it produces engineering and administrative reports for Stored Program Central Office Equipment. This paper describes the user needs that motivated the design of SPCS COER. It traces the development of the system from a joint experiment of Bell Laboratories and the New York Telephone Company's Manhattan Engineering Department to the present system serving over 2800 electronic central offices from three centralized Amdahl V6 computers. We explain the functional system design from the user's point of view and outline the structure of the software, emphasizing the "tools" approach that is central to the development. The project management featured a research and development style, which we call the "whole-job" approach and discuss in some detail. Finally, we explore possible future directions for the project.

## I. INTRODUCTION AND HISTORICAL PERSPECTIVE

When the first 1 *ESS*† electronic central office cut over into service on May 30, 1965, at Succasunna, New Jersey, the practice of traffic

* Bell Laboratories.
† Trademark of Western Electric.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

data collection took a large step forward. Peg count, overflow, and usage measurements were made by the stored program control of the switching machine and kept in memory until their scheduled printout on a remotely located traffic teletypewriter.[1] No longer was there any need for the old electromechanical traffic registers, for clerks to read them, or for cameras to photograph them. Since the traffic teletypewriter could easily perforate a machine-readable paper tape with the measurement data, the tapes could be collected and read into a computer whenever convenient. It seemed clear to AT&T and Bell Laboratories planners that telephone company written computer programs would soon be processing the measurements into statistical information that would be unprecedented in accuracy, reliability, and timeliness.

### *1.1 Traffic data needs*

Central office traffic measurements tell a Bell Operating Company (BOC) what is going on in the switch and the network. The company needs to know about call load, customer service, and equipment utilization. Information needs vary from short to long term. Network administration people are responsible for ensuring that valid, complete, and timely data are properly collected and reported for administration, business, engineering, maintenance, and other purposes. If something is wrong with the data, or the switching office, the administrators need to know quickly so they can fix it quickly, thus minimizing the effect of the problem on data and service quality.

Network administrators use the data to trend and balance loads and determine exhaust dates for central office equipment components (i.e., dates when the expected call loads will exceed the traffic capacity of the installed equipment). Network design personnel have more long-term needs, using the data to determine equipment capacities and the size and timing of future office additions. Marketing people help determine how well customers' needs are met by existing equipment arrangements.

Because telephone use varies widely (even wildly) with time, most data needs are best served by statistical information (e.g., averages) rather than by individual measurement readings. This means arithmetic operations (data processing) must be done on the measurements collected. It also means that—unlike, for example, the banking industry—a few missing measurements may not be serious. Some data may need to be excluded from the averages to keep the results representative. A good example of this was the day it snowed in Miami in 1977. Or the day a telephone building caught fire in New York City. Or the time when an equipment trouble caused all calls to certain areas to be set up so that no one could talk. Thus, data cannot be accepted

uncritically but must pass the test, "Is this measurement both accurate and representative?"

To meet these needs in the years before electronic switching had been difficult and labor intensive. Data systems were semiautomatic at best. For example, measurement registers (mechanical counters that resemble automobile odometers) could be grouped together in large arrays and photographed periodically by automatic cameras. But, after development, the film had to be manually read and keypunched before it could be used by data processing machines. It was particularly difficult to obtain timely information in quantity. Thus, the electronic switching systems' promise of an economical, completely mechanized collection and data processing capability was truly an important and exciting step forward. Increased mechanization was going to permit improved information quality and reduced processing time and save a lot of manual effort as well.

The expected increase in mechanization was not intended to permit force reduction but to prevent a large force increase. Electronic switching systems were going to produce a lot more traffic measurements than previous electromechanical systems. This was because of the increased complexity of the switch (requiring new measurements), the ease and low cost of making the measurements, and the plans of the Bell System to offer new and special services using the capabilities of the stored program control. Indeed, within 10 years a typical 1 *ESS* central office was estimated to be producing perhaps 18,000 register readings daily or some 4.5 million per year. As the Queen said in Lewis Carroll's *Through the Looking Glass,* it was going to take "... all the running you can do, to keep in the same place."

### *1.2 An experiment begins*

By 1969, little of the expected telephone company development had taken place. A few companies were running programs on time-shared computers that rearranged the traffic measurements into neatly labeled columns. Many companies were using transparent overlays to locate a few key measurements in the teletype printout, and were then recording and processing those manually. Some 50 *ESS* central offices were in service and new offices were being installed at an increasing rate that would soon reach one a week.

In mid-1969, work to design an experimental traffic data processing system for 1 *ESS* offices was begun as a joint project of Bell Laboratories Traffic Studies Center and the Manhattan Engineering Department of the New York Telephone Company under the auspices of the Traffic Division of AT&T. The system was experimental because, from the outset, the intention was to use technology radically new for the time in an attempt to build a practical and economical data system

uniquely responsive to the needs of the telephone company users. Those needs, of course, were incompletely understood at the time, a further reason for considering the system experimental.

After two years of initial development, overlapped with about a year of trial use in several companies, AT&T in October 1971 announced a new time-shared computer program to process and manage traffic measurement data in 1 *ESS* offices. The new system, "PATROL" (Program for Administrative Traffic Reports On Line), provided an uncommon combination of features of significant help to the network administrator and network design engineer in the management of data. The announced features included the following:

1. On-line access to large quantities of traffic data.
2. Daily, weekly, monthly, high day, and busy season reports on demand in nearly real time.*
3. Automatic selection of high-day data.
4. Built-in data validation and exception flagging capabilities.
5. Single end-user control of traffic data entry, data retention, report generation, and data management in general.
6. A forgiving, interactive user dialogue that makes it easy to use with minimum computer knowledge and little or no formal training.

(Later experience proved two more features important:)

7. Reliable operation with rapid, low-cost system maintenance and feature development.
8. No hardware investment required of the user community for the experimental system that used only widely available teletypewriters with paper tape readers.

PATROL was initially implemented in a combination of FORTRAN and EXEC language on an IBM 360/67 computer operating under a modified Cambridge monitor. The vendor, Computer Software Systems, Inc. (CSSI) of Stamford, Connecticut, was chosen for the unique capability of its service at the time. There were no plans to make PATROL portable.

### *1.3 The first system*

After initializing the system with a detailed description of a particular switching system, a network administrator would dial up the CSSI machine over the regular telephone network and transmit hourly ("Block H") traffic register counts from the paper tape punched by the traffic teletypewriter. PATROL would immediately make data validation tests, and items exceeding predetermined tolerance limits would be called (flagged) to the administrator's attention, along with

* In this context, "near-real time" refers to reports obtainable with little delay (perhaps minutes) after the end of the period being measured.

a brief summary of office performance data. The measurements would be accumulated in history files of data on an hourly basis. As many hours of daily data as desired could be processed, with costs increasing in proportion. Because of cost, most offices processed only one or two hours of data a day.

The history files were arranged for the generation of reports on a daily, weekly, and monthly basis, or for any desired span of dates at user request for each individual switching office. The reports were usually printed off-line and mailed to the administrator, but when time value justified higher cost, the reports could be printed on-line at the administrator's (or network design engineer's) terminal.

Probably little in PATROL was startling or new from a computer science point of view (a term just then beginning to come into vogue). What was new was the application of the technology to assure maximum capability and ease of use for end users with little training. A single time-shared machine serving thousands of offices spread over a continent was a concept still considered a bit adventurous (perhaps even foolhardy) in the early 1970s, although there were several precedents in the 1960s.[2,3] An on-line, interactive development facility was still a wonderful novelty for program designers and, of course, proved efficient and cost effective.

One design approach that was not recognized as new at the time was to provide each switching office with a database distinct from every other office and to conduct almost all processing as though only the one office and the one administrator existed. The (perhaps concurrent) processing of other offices for other administrators was generally hidden in the design, it being left to the operating system to sort out. This single-office approach greatly simplified development and maintenance by reducing the complexity of the software compared to other Total Network Data Systems (TNDSs) [e.g., Traffic Data Administration System (TDAS) and No. 5 Crossbar Central Office Equipment Reports (5XB COER) described elsewhere in this issue] whose designers had to plan for the sequential processing of perhaps hundreds of different offices in the same batch run.

### *1.4 Growth and evolution*

In the years that followed the initial introduction of PATROL, most of the original features were preserved, while many features and systems for other types of electronic switching systems were added. At the end of 1971, 27 offices in 9 companies were using PATROL.

During 1972, the participation of the New York Telephone Company was gradually phased out and Bell Laboratories continued the experiment, working closely with AT&T and representatives of the BOC

network administrators.* Many report types were added, an on-line documentation feature, Lessons, was produced, and processing costs were reduced. By March 1973, there were more than 165 offices on the system.[4] In subsequent years, versions of PATROL were developed for 2 *ESS*, 3 *ESS*, Remote Switching Systems (RSS), Voice Storage System (VSS), and 5 *ESS* machines.[5]

When an in-house (AT&T) computer service offering nearly the same software features [Virtual Machine/Conversational Monitor System (VM/CMS)] as our original vendor became available in 1974, PATROL was the first large project to move to that facility. It has remained there since.

In 1975, when the Engineering and Administration Data Acquisition System/Traffic Data Administration System (EADAS/TDAS)† data collection portion of TNDS was available in most BOCs, PATROL developed a major new feature called Batch Data Entry (BDE). Data could be collected by EADAS, sent by magnetic tape to TDAS, and relayed from TDAS to the PATROL computer over the high-speed corporate data network (see Fig. 1). Compared to paper tape data entry, BDE significantly reduced computer expense at the sacrifice of two initial PATROL features. A single end user could no longer control the whole system alone, as coordination with EADAS, TDAS, and the corporate data network was required. And, because of the delays inherent in the new batch process, results from PATROL could no longer be obtained in "near-real time." However, because of the data processing and reporting capability of EADAS, the near-real-time feature of PATROL was no longer needed where BDE could be used.

In 1977, a major rewrite of PATROL changed its basic orientation from processing clock hours for all office components to processing selected busy and side hours for each office component. This component busy hour feature was followed in 1979 by a Busy Hour Determination System. By the end of 1978, there were more than 1600 offices using PATROL.

### *1.5 The experiment ends*

During its early years PATROL was enthusiastically received by network administrators and network design engineers. However, upper BOC management, often advised by computer experts that time shar-

---

* As has often been our experience, the collaboration with a BOC, here New York Telephone, had been richly rewarding. The "real world" expertise and "get it done now" attitude of the New York people really helped move the project along. By the end of 1972, however, the on-line system was providing feedback from the whole country.

† EADAS is the subject of the article "Data Acquisition and Near-Real-Time Surveillance," this issue.

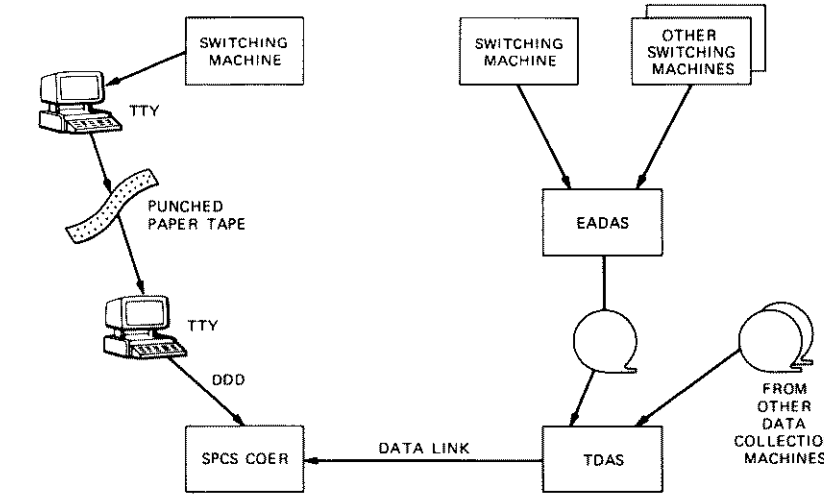



Fig. 1—On-line and batch entry of data in SPCS COER.

ing was always more expensive than batch, feared that perhaps the system was too expensive. In 1976, these fears were laid to rest by two events. A major survey of computing costs showed that batch operating costs were much higher than was widely believed—sometimes 10 times higher. Finally, an extensive multilevel, multidepartment survey of every BOC, including three that at that time were building or operating their own traffic data systems, was conducted. It found that the PATROL system, of all other candidates, had the best match of features to the future data needs of the BOCs. With these and other studies concluded, it was finally clear that the PATROL experiment could be considered at an end. Continuing standard development under the Business Information Systems (BIS) funding agreement was begun in January 1978.

### *1.6 Continuing development*

The BIS Advisory Committee asked that the name of the system be changed from PATROL to *ESS* COER to better agree with other TNDS component system names. This was later amended to SPCS COER when we realized that the system would be used for Stored Program Control System Reports not involving switching. However, because the term PATROL had become extensively embedded in Bell

System literature and had been used by the BOCs for many years, and because two syllables are much easier to say than six,* the term PATROL is still frequently used by the field forces.

The change from experimental to standard status made it possible to emphasize system design considerations in SPCS COER over simple feature availability because long-term planning and investment of resources were now warranted. The original version of PATROL, with substantial amounts of interpretive code, had been replaced by a much more efficient FORTRAN version by the end of 1972. As an experimental system, the software emphasis was on achieving features quickly, while steering a prudent course between run-time inefficiency and high performance. With the prospect of an assured future, the development effort began to emphasize new approaches to the design in 1977. A modular, generic-tools approach would achieve both high performance and reduce development and maintenance effort in the long term. This new approach is described in Section III.

In 1979, a new experimental feature was designed for AT&T. The Data Profiles feature consolidates selected information from all the individual 1/1A *ESS* office databases for the whole Bell System. Elaborate retrieval, computation, and reporting facilities are provided in an English-like language for the use of personnel engaged in studies of various types.

During the 1980–81 busy season (January to March), SPCS COER began to strain the resources of its host (Amdahl V6) computer, with the traffic data load from nearly 2400 *ESS* machines. Therefore, in September 1981, after about a decade of continuous operation on a single computer, the user community was divided up among three computers (at the same location).

In December 1982 a new feature was introduced that allows printing of reports in the user's local company data center. In addition to existing options that allow reports to be either printed on-line or off-line and mailed, the user may now have the reports sent to the nearest BOC data center over the corporate data network or via magnetic tape. This new option was provided to allow users willing to incur additional delay in the distribution of reports the opportunity to reduce centrally billed expense for SPCS COER by as much as one third.

During its first decade of operation, SPCS COER compiled an excellent record of availability. While few detailed records were kept, the longest recorded nonscheduled total system outage (no access to reports) was about six hours. Some individual users or BOCs experienced longer outages while software errors were being repaired.

At the end of 1982, there were more than 2800 offices using SPCS

* The preferred pronunciation of SPCS COER is S-P-C-S KO-AIR.

COER. The program had grown to 225,000 lines of FORTRAN code with an additional 105,000 lines of comment. There were about 1000 pages of on-line user lessons for all the component systems. Users were logging on to the system at a rate of about 25,000 per month and issuing about 1,000,000 commands a year. About 3,000,000 hours of data were being handled annually and 1.2 million pages of reports produced per month. One third of the report output is produced on fiche, the rest on paper.

## II. FUNCTIONAL OVERVIEW

This section presents a functional overview of SPCS COER from the user's point of view.

SPCS COER is really a family of on-line data management systems that process traffic data from the class of switching systems called sorted program control systems. This includes 1/1A *ESS*, 2/2B *ESS*, 3 *ESS*, 5 *ESS* and RSS switching entities.

Each switching entity, or office, served by SPCS COER has its own distinct, independently managed database. This implies separate data collection, reporting, and data management for individual offices. This single-office view is incorporated in the software design and the user interface. It reflects the usual approach taken by network administrators and network designers in examining central office traffic data. A consequence of this design is that most of the SPCS COER systems cannot summarize data across multiple entities. There is, however, one SPCS COER system, described in Section 2.6, that provides data summaries across offices.

### *2.1 User interface and documentation*

A paramount goal in designing the SPCS COER systems is that they be easy to use by network administration clerks as well as managers. An interactive English-like dialogue, on-line documentation, and a generally forgiving design help support this goal. Since SPCS COER is an on-line system, it gives the user direct control of the system functions. When something unexpected happens, for example, an invalid request is submitted, the user is notified and given an opportunity to take immediate corrective action.

The user interface takes the form of a question and answer type of dialogue. The user types a command in response to a REQUEST = prompter. Any parameters required before the command can be executed are then prompted for by the system. The user need type only a carriage return to ask for help in responding to a particular prompter. The system will then print a list of valid responses or response formats. This list is also printed if the user types an invalid reply. The user can type CLEAR in response to any prompter to abort the command and return to REQUEST =.

A complete set of on-line user documentation is available with each SPCS COER system. The user may access the documentation on request and have one or more sections of it typed at the terminal or printed off-line and sent by mail. This allows changes in the documentation to be made available to the users quickly and easily by the developers. It also assures the user of an easily accessible source of the latest information. The combination of readily changeable on-line documentation with a hot line open 24 hours a day (by answering machine at night) has proven effective in keeping the documentation error free and the users informed, confident, and satisfied. One finding of many studies is that software system problems are often due to the unavailability of up-to-date documentation at the user sites. SPCS COER totally avoids this problem by placing the most current information in the hands of the user whenever the user asks.

News messages may be printed on request as well. News items are put on the system by the developers to announce new features, revised documentation, or the existence of (or, preferably, the solutions to) newly discovered common problems. Users are alerted to news items by one-line flashes whenever they log on the system. Two levels of flashes and news items are used depending on the intended audience for the item—all users or only users of a particular component system.

Since costs of any operations support system need to be monitored, SPCS COER provides estimated expenditures to the user for all on- and off-line requests. This feedback has helped users manage their use of the system effectively.

An important part of the SPCS COER user interface is a network of "SPCS COER coordinators," one or more in each BOC. These coordinators directly assist the BOC users in operating and managing the SPCS COER system. Usually it is the company coordinator who contacts the SPCS COER staff over the hot line about problems. The coordinators help the users identify and rectify problems. Over the years, the coordinators have asked for and been given the tools to identify and correct system problems that initially only the PATROL central staff could fix.* These tools have been grouped into a separate interactive system, modeled on SPCS COER, called COMP for Coordinators' Management Package. COMP even has its own set of on-line lessons.

### *2.2 Office description file management*

Before any SPCS COER system can process traffic data for an office, an Office Description File (ODF) must be established for that

* That is, when the generally forgiving design had stopped being forgiving to some user.

office. The ODF contains information describing the office and the traffic measurements that are to be processed. This information includes:

1. The size of the office
2. The equipment or service components of the office
3. Characteristics of each component
4. Measurements to be processed and components to which they correspond
5. Hours of traffic data to be processed for each component.

An SPCS COER user establishes, or activates, an ODF through an on-line request. The user must supply to COER all the information needed for the ODF. This information is used by the system to interpret, validate, and make calculations on the traffic input data and to store and retrieve processed data. Therefore, it is essential that the ODF be current and correct.

A validate function ensures that the ODF information is internally consistent and that the user-specified values fall within appropriate bounds (e.g., the information represents a valid *ESS* office configuration). Validation is automatically done when an ODF is first activated and whenever the information in it is updated. If errors are found in the ODF, the user is alerted and COER will not process any new traffic data until the errors are corrected.

The on-line update command allows the user to examine the ODF and change it. All changes are validated, and any problems immediately reported to the user on-line. The user can then fix any errors by issuing update commands. Commands are also available to print the contents of an ODF on-line or off-line, and to deactivate an ODF (remove the ODF and its associated database from the system).

### *2.3 Data entry*

To enter data into the database for an office, an SPCS COER system must first read the raw data values put out by the SPCS on what is known as a traffic schedule. A traffic schedule contains measurement data for a particular office, the collection date, collection interval, and end-of-interval time. The system uses ODF information to associate measurement values with their corresponding components. Format checks are made on the values, and traffic statistics are calculated for each component based on the raw data values and ODF information. These processed component values are then stored in the database for the office.

As part of the data entry process, SPCS COER makes various data reliability tests. These tests are intended to pinpoint instances of traffic measurements that are invalid, for whatever reason. Invalid measurements are caused by equipment or software malfunctions in

the switching machine or data collection system, highly unusual and unrepresentative traffic conditions, or (most of the time) errors in the ODF information. If data for a component fail one or more reliability tests, they are flagged, or marked as questionable, in the database. Flagged data are processed specially during data retrieval. These data are marked on all reports on which they appear and are excluded from report averages. The user has a recourse if data are inappropriately flagged, as described in Section 2.5. The results of the data reliability tests made during data entry are saved and made available to the user as exception reports.

Traffic data may be entered into SPCS COER in one of two modes, on-line or batch, as shown in Fig. 1. On-line data entry, the original PATROL data entry mode, involves using punched paper tape generated at the traffic teletypewriter connected to the switch. These tapes, containing the required traffic measurements, are transmitted to SPCS COER on-line over the switched telephone network via a terminal with a paper tape reading facility. The user must monitor the process while the tapes are being read. For large volumes of data, this is a tedious and time-consuming process that also involves substantial computer connect time charges.

The newer batch data entry facility reduces the cost and clerical effort associated with on-line data entry. It is now used for most of the offices served by SPCS COER. In designing BDE, the idea was that a user could set things up once, and then data would flow from the switch into SPCS COER almost unattended.

Batch data entry may be used by offices supported by EADAS and TDAS,[6] or their equivalents. Under this method of data entry, data for many offices and for several days are transmitted from a telephone company TDAS computer site to the SPCS COER computer site over the corporate data network. The SPCS COER system automatically places this data into a common disk storage area as pending files. An individual pending file is created for each block of data that corresponds to a particular office, date, and collection time. These pending files are organized on a per-user basis.

At night, an automatically scheduled job is run for each user who has pending files. This job processes the data from the pending files into the databases for the user's individual offices. It also generates messages for the user that indicate the status of the batch data entry job. These messages may be printed on request when the user next logs on to the system. In this way, a user knows exactly what data were entered into the database, or why any data were not entered.

Pending files that are not successfully processed are retained on the common disk area for two weeks. They are also available for 45 days from a backup tape. The user can obtain a listing of all pending files

for a given office. Users are provided with on-line commands for managing these pending files. Pending files can be erased from the common area or restored from backup tape. Certain errors in the header information for pending files or in the traffic data values can be corrected interactively to enable the system to process those data.

### *2.4 Reports*

To retrieve data from an office database, a user requests traffic reports from the system on-line. The user has the option of having them run on-line and typed directly at the terminal, or run in a batch mode, printed off-line, and mailed to the user.

There are two basic types of traffic reports produced by the SPCS COER component systems: daily reports, which produce individual daily data and data averaged over a given span of days; and Machine Load and Service Summary (MLSS) reports, which make available year-to-date summaries of monthly averages and the highest traffic days.

Daily reports may be requested in weekly, monthly, or intermediate form. Weekly and monthly reports provide data for all central office equipment components and all hours. Intermediate reports may cover any span of days and any selected components and hours. Each section of a daily report is devoted to one component in the office. It shows the daily statistics as well as the averages of the values over the report period. Flagged data are excluded from the averages. Monthly reports are useful to the network administrator for analyzing the overall service in the office, for trending office loads, and for data validation (as described in the next section).

The MLSS reports are used by the network designer in capacity determination and in planning for future central office equipment needs. An MLSS report contains, for each component and hour requested, data for the 15 high load days of the year to date, together with the average of the 10 highest unflagged days. Also included are the monthly averages for the year to date, together with the average of the three highest months. At the end of the data collection year, this last average represents the Average Busy Season (ABS) value for the component.

The formats of the traffic reports produced by the SPCS COER component systems follow those produced by the 5XB COER system wherever possible. The 5XB COER reports are covered in some detail in the article "Equipment Systems," in this issue.

The Busy Hour Determination (BHD) systems (1 *ESS* BHD and 2 *ESS* BHD) of SPCS COER produce a different set of reports from the ones just described. The BHD systems help the network administrator determine, for each component in an office, which hour has,

on average, the highest traffic load. This is called the time-consistent busy hour for the component. A BHD study for an office is usually done once or twice a year, for two to four weeks. During that time, data are collected and processed by the BHD systems for 12 to 24 hours per day. The output of this study helps the network administrator determine which hours of data should be processed for each office component in 1 *ESS* COER or 2 *ESS* COER. There are five sets of reports produced by the BHD systems. They range from detailed hourly reports to a high-level summary report listing the suggested COER collection hours. Because RSS and 5 *ESS* switches are Extreme Value Engineered (EVE),[7] instead of time-consistent busy hour (or average busy season busy hour) engineered, they do not require busy hour determination studies.

### *2.5 Database management*

SPCS COER makes available to the user several data management capabilities. There are on-line requests that tell the user what data are available in the database and allow the user to flag and unflag data, remove unwanted data, and initiate (or terminate) a data collection year.

The on-line flag and unflag requests allow the user to override the data reliability flagging decisions that were automatically made by the system during data entry. A user can selectively flag or unflag data for any component, hour, and date. This is usually done before requesting a monthly report.

After a monthly report is run, the monthly averages are stored with the MLSS data, and users have received their paper or microfiche reports, there is usually no need for SPCS COER to keep the detailed daily data for that month. The SPCS COER has a remove command that lets the user remove daily data for any span of days from the database. Data can also be removed selectively from the MLSS high day lists. A cycle function lets the user remove all MLSS data from the database at the end of the year.

The database management features, along with ODF validation, are key parts of the success SPCS COER has achieved in producing high-quality results. However, the degree to which user actions are required to manage the database might seem inefficient. Why not further mechanize these data management processes to require less user intervention? Let us look at why this user control is so important.

The user, usually a network administration clerk, is responsible for producing accurate and timely information. But it appears to be a fact of life (based on some 20 years of data processing experience) that bad data often does not become apparent until after it is used to produce reports. And every new type of report produced seems to uncover new instances of bad, or at least suspicious, data.

When this happens in SPCS COER, the user, after suitable investigation to learn the facts, simply flags or unflags data as appropriate and reruns the report. Only after the reports have been thoroughly examined is it generally safe to remove the data from the system. Data processing systems without these features often leave their users with no usable results and/or the task of reprocessing the data by hand.

It is also a fact of life that users make mistakes. While this may be regrettable, it is no excuse for producing poor results. Thus, almost every action that a user can take in data management can be reversed if the action proves to have been a mistake.

User control of the data management has a cost to the user in data processing and storage charges, of course, But if the user feels that exercising the recovery features is not worth the cost, then the feature is not used and the cost avoided.

There is an overhead cost in software for having some of these options available, even if they are not exercised by individual users. However, the system design has assumed that these are far offset by several types of savings that their presence makes possible. First, manual data processing by clerks is eliminated for this purpose. Second, results are sometimes achievable that could not possibly be obtained in other designs (for example, when a serious error is discovered some months after the data has been discarded and a large monetary decision hangs in the balance). Third, formal training expense is avoided by virtue of the new system design (e.g., users can learn by doing with minimal risk of data loss through user error).

Since data retention affects storage requirements (and costs), the system monitors storage usage and alerts the user when the amount of storage consumed exceeds a predetermined threshold. The system also provides the user with a storage command that reports the current storage consumption.

### *2.6 Data profiles*

The functions and capabilities described so far in this section apply to most of the family of SPCS COER systems. However, one system, called Data Profiles, serves a special function. Data Profiles, unlike the other systems, is not restricted to a single-office view of data. It provides a facility for summarizing traffic data across many offices (and also across more than one service year).

This system collects preselected MLSS data from each SPCS COER office database and stores this information in a single hierarchical database. In a hierarchical database, data are divided, subdivided, and sub-subdivided in a treelike fashion as many times as needed to permit convenient access to its parts. Data Profiles divides and subdivides the MLSS data by BOC, type of *ESS* switch, particular office, and

collection period (current year, latest complete year, and archived older years).

The data for an office are collected by Data Profiles when the SPCS COER user requests a monthly report for that office. The database currently contains data for all 1/1A *ESS* offices. Data exist for each year, starting with 1978. Users can access any or all these data on an on-line demand basis.

Data Profiles uses a generalized database system/tool called the Off the Shelf System (OTSS),[8] which provides data management routines and a query language. This query language allows Data Profiles users to retrieve any items or combination of items stored in the database. The query language statements are simple. They consist of a few verb and modifier expressions that specify the part of the database to be traversed and what action should be taken.

By typing a single OTSS sentence, a user can ask, for example, "What was the average busy hour CCS per main station for *Touch-Tone** dialing digit receivers in New York Telephone offices during 1980?" Users can set screens on data so that the system will retrieve only data that satisfies certain conditions. For example, including the phrase, "when average busy season percent capacity is greater than 90," in the above query would cause the system to include in the average only data for receiver groups that were operating at more than 90 percent of capacity. The query language has commands that distribute, plot, print, alter, and make statistical computations on the data. Because queries may become complex and may be useful to many users on many occasions, a query library is maintained by the system.

Data Profiles provides a powerful tool for studies since it contains data for an entire area and so can be used by telephone engineers for many different planning purposes.

## III. SOFTWARE STRUCTURE

The first SPCS COER system was hard coded with data structures and algorithms specifically designed for the 1 *ESS* central office application. The resulting system, although remarkably popular, was also inflexible and hard to enhance. What might seem a small conceptual change (the component busy hour feature) required a major rewrite of the system. Even the seemingly trivial change to compress the batch reports seven spaces to allow for hole punching required an embarrassing amount of reprogramming. In this section, we describe a software design method that led to SPCS COER modules that are easier to develop and maintain.

---

* Registered service mark of AT&T.

### *3.1 The view from the back room, or a hacker's view of history*

One motivation for this design philosophy was the rapid proliferation of SPCS COER subsystems (serving different switching systems) during the mid-1970s. The 2 *ESS* COER was the first offspring of the initial 1 *ESS* system. This system was produced by the venerable red pencil technique. Engineers poured over the ponderous listings of 1 *ESS* COER (60K lines) deleting and changing sections of code that were not quite right for the new application. The resulting system worked but became difficult to enhance. Then 3 *ESS* COER was produced from 2 *ESS* COER in much the same way. The prospect of making parallel enhancements in these systems was awesome in much the same way that the Okefenokee Swamp is awesome. The laws of genetics suggested that this inbreeding would have to stop or the species would become extinct.

The crux came in the summer of 1977 when the SPCS COER team was faced with the task of developing four new subsystems (Busy Hour Determination, Voice Storage System, Remote Switching System, and Advanced Communication System). It was clear that the red pencil technique was no longer tenable. It was not so clear how these ambitious systems could be produced without a large development effort and substantial duplication of code. Fortunately, this crisis coincided with the emergence of the software-tools approach to system development. This method advocates developing larger systems out of a kit of small general-purpose tools rather than building monolithic special-purpose systems. This philosophy was an enormous help in managing the large development task. Rather than rushing off in four separate development efforts, time was taken to define a set of tools that could be useful in all four systems. The result was a marked reduction in development time and lines of code. In addition, the resulting systems are simpler and easier to maintain.

### *3.2 Tools plus*

As valuable as the tools approach was, it was not the whole answer to the problem. Each SPCS COER system contains a fair amount of domain-specific knowledge both about traffic engineering and about a particular switching system. Without further philosophical underpinnings, it would be difficult to capture this knowledge in simple, generically useful tools. What was needed was some way to factor this specialist's knowledge out of the tools. The goal was to make the tools simple enough to be flexible and yet sophisticated enough to be knit into a useful system. The solution was to capture the knowledge of traffic engineering in a data model that can be used by the tools and to capture the switch specific knowledge in a set of tables that drive

the tools. As a result, the systems developer's job changed. Previously, it was to write or change large amounts of code to meet the requirements of the system. With the new method, it is to first select the proper tools for the system, then define the tables specifying the new system to the tools, and finally to write the controlling code which welds the individual tools into an effective system. The advantage is that the developer spends less time in low-level development and is free to devote more time to the design of an integrated system meeting the customer's need. In effect, with intelligent tools we have raised the developer's perspective from that of programmer to that of system designer.

The heart of the new approach is the traffic data model. The model consists of both a view of the traffic data and the routines that manipulate the data in accordance with this view. The traffic data that SPCS COER deals with come in two varieties—daily and MLSS.

### *3.3 Daily data manager*

The SPCS COER Daily Data Manager (DDM) captures the notion of daily data in an abstract model. To DDM, an individual traffic measurement is represented by a tuple of five elements (day, hour, equipment tag, measurement tag, value). The quadruple (day, hour, equipment tag, measurement tag) is an index to the data values. Any measurement in an office's database can be accessed by specifying the identifying index.

The DDM provides standard database operations. The user of DDM can create and destroy databases, and add, delete, retrieve, and update items in an existing database. But DDM is a powerful tool because it understands the particular needs of an SPCS COER system. For one thing, it understands traffic data. It understands that the data are collected as a block of measurements for all the equipment in the office for a given interval. The data structures and algorithms of DDM are carefully tailored to efficiently organize and store data collected in this way. The DDM also understands that these measurements are subject to error and provides routines to flag questionable data items.

The DDM is particularly useful to SPCS COER systems because it understands how the systems are likely to use the data it manages. Report generation provides a convenient example. An SPCS COER report is a formattted output of the traffic statistics for a specific piece of equipment at a particular hour over a span of days. The DDM provides two special interface routines to aid in report generation. The first takes an hour, an equipment tag, and a range of days as an argument to prepare DDM to retrieve data for the report page in a systematic way. Each time the second routine is called, it returns a

new line of data for the report which includes the day on which the data were collected.

### *3.4 MLSS data manager*

The MLSS database contains the same measurements as the daily database, but instead of keeping a day-by-day account, the MLSS database contains rank-ordered lists of the busiest days and the monthly averages on the equipment in the office. The MLSS Data Manager (MLSM) is an SPCS COER tool for managing MLSS databases. It views an MLSS data item as a five element tuple (hour, rank, equipment tag, measurement tag, value). The MLSM provides SPCS COER systems with the same sorts of services for managing MLSS databases as DDM provides for daily databases. In particular, MLSM maintains the rank-ordered lists.

DDM and MLSM free the individual system developer from worrying about the complexity of data management. They take care of storing and retrieving data on disk and worry about such issues as concurrency control and recovery. But more importantly, they enforce a standard data model across all the SPCS COER subsystems. This has two important ramifications. First, since all the traffic data are stored in a common manner, they can be analyzed by common software. Second, since the data are stored by a common data manager, higher-level tools can be built that use DDM and MLSM to fetch and store the data.

### *3.5 RSL*

The SPCS COER Report Specification Language (RSL) is a good example of a high-level tool that uses DDM and MLSM. It also illustrates how a system designer can customize a generic tool to fit specific subsystem needs. The RSL is a simple but powerful language which allows the user to specify the format of a wide class of SPCS COER reports. Figure 2 shows a sample RSL program and Fig. 3 shows a report that was generated from this specification.

A brief study of Fig. 2 illustrates how RSL works. The first section of the report specification is the heading. This is the information to be printed at the top of the report. Information in quotes is printed here as it is written on the report. The key words offnam, datnam, schnam, and stunam specify translations to be made when a specific report is printed. The key word offnam will print the name of the office for which the report is being generated. Key word schnam will print the hour for which the data were taken, stunam will print the name of the equipment, and datnam will print the Gregorian representation of the first or last day in the report.

The supertitles section presents titles that span more than one

```
daily report 2 for (s52, s53, s54, s55, s56);
        flag line on position 9;
        heading
            offnam;
            '';
            'NO. 1 ESS SERVICE CIRCUITS AND SPECIAL TRUNKS';
            'DETAILED DATA FROM', datnam (fstday), 'TO', datnam (lstday);
            'ALL COMPONENT HOURS';
            '';
            '';
            'ITEM NO.', %d stutag, ':', stunam, '          ', schnam;
            '';
        end;
        supertitles
            'PER STATION' 14-24;
            'USAGE' 29-33;
        end;
        columns
            datnam (curdat) 1-8%s" no average;
            i1  12-18%6.4f* 'CCS' exclude on '*%&';
            i2  20-25%5.3f* 'CALLS';
            i3  26-31%5d* 'TRFFC';
            i4  32-35%3d* 'MB';
            i5  36-41%5d* 'PEG COUNT';
            i6  43-48%5.2f* 'HT SEC.';
            i7  50-54%4.2f* 'OVFL %';
            i8  55-58%3d* 'CAP %';
            i9  59-63%4d* 'NCI';
            i10 64-69%5d* 'CAP CCS';
            100* (i3+i4) / 36*i9  70-72%3d  'OCC %';
        end;
end report;
```

Fig. 2—Sample RSL program.

0000013H MLDNMAELCGO SP CTX-8 3.5 COMP. ID: 001
NO. 1 ESS SERVICE CIRCUITS AND SPECIAL TRUNKS
DETAILED DATA FROM 07/01/79 TO 07/09/79
ALL COMPONENT HOURS

ITEM NO. 54: T-T MEMRY P 10:00

| | PER STATION | | USAGE | | PEG | HT | OVFL | CAP | | CAP | OCC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | CCS | CALLS | TRFFC | MB | COUNT | SEC. | % | % | NCI | CCS | % |
| 07/01/79 | 0.0029 | 0.058 | 40 | | 792 | 5.04 | 0.00 | 3 | 55 | 1256 | 2 |
| 07/02/79 | 0.0041 | 0.047 | 55 | | 634 | 8.72 | 0.00 | 4 | 55 | 1256 | 3 |
| 07/03/79 | 0.0493+ | 0.047 | 66 | | 643 | 10.26 | 0.00 | 5 | 55 | 1256 | 3 |
| 07/04/79 | 0.0077 | 0.052 | 105 | | 705 | 14.89 | 0.00 | 8 | 55 | 1256 | 5 |
| 07/05/79 | 0.3333* | 0.039 | 62 | | 531 | 11.73 | 0.00 | 5 | 21* | 1256 | 3 |
| 07/06/79 | 0.0070 | 0.066 | 2* | | 897 | 10.64 | 0.00 | 8 | 55 | 1256 | 5 |
| 07/07/79 | 0.0050 | 0.001* | 68 | | 547 | 12.36 | 0.00 | 5 | 55 | 1256 | 3 |
| 07/08/79 | 0.0054 | 0.051 | 73 | | 693 | 10.59 | 0.00 | 6 | 55 | 1256 | 4 |
| 07/09/79 | 0.0060 | 0.043 | 81 | | 581 | 13.96 | 0.00 | 6 | 55 | 1256 | 4 |
| AVERAGE | 0.0109 | 0.050 | 69 | | 669 | 10.91 | 0.00 | 6 | 55 | 1256 | 4 |

Fig. 3—Sample report generated from an RSL program.

column in the report. The quoted string gives the title, and the numbers give the character positions over which the title should be centered.

The columns section specifies the main body of the report; each line

in this section specifies a column in the report. The first entry is the name of the data item to be printed in this column and the numbers show the character position boundaries for the column. Information after the percent sign tells how the value should be printed (s = string, d = integer, f = floating point, n.n = how many total digits in the number and how many digits after the decimal point. Compare with the C language function, printf). The asterisk shows that the data item can be flagged, and the string in quotes is the column heading. The key words no average indicate that there should be no average taken over this column. The key words exclude on followed by a list of flags indicate the data items flagged with any of the listed flags should be excluded from the column averages.

The system developer writes the report specification when designing the system. The RSL compiler converts the report specification into intermediate code to be interpreted at report generation time—that is, when the SPCS COER user requests a report of this type. At report generation time, the SPCS COER subsystem calls the RSL run-time interpreter with a short description of the desired report. This description includes the report type, the office, the equipment tag, the hour, and the range of days. The RSL interpreter then prints the report according to the specification and data. First, RSL prints the heading, supertitles, and column headings in an attractive fashion. It then calls DDM or MLSM to get the data for each line of the report, which it prints according to the column specifications. Finally, it prints the appropriate averages for each column.

RSL takes the drudgery out of report generation. Instead of writing code to fetch, format, and print reports, the system designer writes a few simple specifications that describe the required reports. RSL is able to do the rest because it understands what is generic in SPCS COER reports.

This section has described only a few of the tools used in the construction of new SPCS COER subsystems. The present arsenal contains many other tools both of the type providing some generic service, such as DDM, and of the table-driven type, such as RSL. Moreover, each new component SPCS COER system that is built suggests new tools so that the arsenal continues to grow. The net result is that the system designer can concentrate more and more on the problems of each system, with less and less bogging down in the details of coding.

The savings can be considerable. Studies on the RSS COER system indicated that tools have resulted in a 65-percent reduction in the code required. Tools have also made SPCS COER an easier project to manage. Since a large portion of the code is generic, it is well established and tested. This substantially reduces the maintenance burden.

In addition, better service is given to system users since new features and component systems can be developed more quickly.

## IV. THE MANAGEMENT APPROACH TO SPCS COER SPECIFICATION AND DEVELOPMENT

Because of its origin as an experimental system, the SPCS COER project was managed with a research and development style,[9] instead of the classical software design style. Although many software projects seem to use this approach, it does not appear to be discussed anywhere in the software management literature. Weinberger hints at it in various places in his excellent book.[10] On the other hand, there is vigorous literature on research and development management methods.[9,11–14] We find this literature applicable to software projects and have coined the term "whole-job software design method" to describe it.

### *4.1 Whole-job team*

The "whole-job" software design method puts all responsibility for a project, including requirements and development, on a small design team. The principal advantage of this approach is that the developers themselves have a close relation with the clients. This results in other derivative advantages: high motivation, flexibility, and low overhead. The team is responsible for:

1. Studying and understanding previous work.
2. Meeting with clients and understanding their views.
3. Proposing a solution. This includes selling the proposal to the BOC's, to AT&T, and to Bell Laboratories affected organizations and management.
4. Designing the system.
5. Implementing and testing the system.
6. Writing user documentation.
7. Conducting a field trial.
8. Managing the general release.
9. Following up on user problems.

Steps 1 to 3 are systems engineering functions, and 4 to 9 are development. In the whole-job approach, they are all done by the same team.

### *4.2 Structure and responsibility in the whole-job team*

A team consists of one to a few members of technical staff. A team member may belong to more than one team. Typically, we have used people with Master's degrees in computer science. The teams are flexible; they reform from time to time as people come and go, but maintaining continuity of people assigned to a given project is a

primary goal of whole-job management. This has not been difficult to achieve.

For multiperson teams, the supervisor sometimes appoints a "team leader." Other times, team members are free to organize themselves as they see fit. They devise and tell the supervisor their plans and a method for reporting progress. The supervisor must choose between imposed and free organization. This choice is based on a variety of factors: the needs and expertise of the team members being the principal ones. In any case, the team plans its own work.

The team does not make commitments on its own to AT&T or BOC clients; management does that. With experience, this hard line can be relaxed somewhat. Team members learn management's views and make minor commitments. This smooths client relations because clients get quick answers.

The whole-job team tells management what it believes it can deliver and when. Management may adjust these estimates a little, sometimes a lot. But even if the dates are adjusted, the team members are usually confident they own them. This has led to much hard work and overtime, not because it was demanded by management, but because the team members assumed a responsibility to deliver to *their* clients what *they* had promised.

#### *4.3 High team motivation*

The whole-job approach places virtually all project responsibility on the shoulders of a few people. It is not divided among several organizations (e.g., a requirements group, development group, test group, maintenance group, etc.). The team members can immediately learn about and satisfy customer needs. This is a motivator.

High motivation arises from all the advantages of this approach: low overhead, closeness to clients, career flexibility, and broadening of team members. Small teams are easier to motivate than large teams. Self-organization is a motivator, a required motivator among professionals.

Team members develop a sense of pride in their project that we call "parental motivation." This carries into parts of the project that are less enjoyable, for example, documentation and maintenance. In SPCS COER, there have been few complaints about documentation and maintenance; they just get done. User documentation is delivered to end users at least a month before general availability of the software.

#### *4.4 Low overhead*

Low overhead arises for two reasons:

1. There are no "handoff ceremonies" from requirements to development, and from there to other follow-on organizations.

2. There is only one chain of command within Bell Laboratories that is responsible for the technical project management.

The whole-job approach allows the team to begin some design work even before requirements work is completed. There is a tight feedback loop from design to requirements because few people are involved. When a plan is generally accepted by clients, and details of formats, formulas, thresholds, etc., are all that remain, the planner's mind can gradually shift to design. Coding of stable or generic parts may even begin. Thus, a smooth and natural transition occurs.

#### *4.5 Better understanding of the client's problem by developers*

Requirements writers are usually close to the client. As they work their part of the problem, they can understand the client's point of view. They can see possibilities for trade-offs that will not compromise real objectives.

If the requirements people are the developers, then it follows that the developers understand the client's problems, have empathy, etc.

If there is a formalization of requirements and a handoff to developers, then the developers are once removed from the client. Many developers minimize this distance by attending some requirements meetings and getting to know the customers. Nonetheless, there is a distance between developers and clients when developers are once removed from the clients by project structure.

In general, the whole-job approach avoids the problem of distance between developers and clients. There are only two groups of people who must thrash things out, see each other's points of view, and agree: the whole job team and the client. One is trying to provide something that the other needs. Relations are simple. Add a third organization, requirements, and you must choose between the two communications networks shown in Fig. 4. The left plan has distance between client and developer, and slow feedback loops; is inflexible; and has difficulty in exploiting opportunities. The right has overhead, tri-team committee meetings, and is a little more flexible and capable of exploiting opportunities. The whole-job diagram is shown in Fig. 5. It has direct feedback and is maximally suited to exploiting opportunities.

Some developers see distance between development and client as helpful. It gives the developers more freedom to pursue design without reference to the customers' opinions. It protects developers from the well-known human trait of clients asking for more and more features on the original development schedule. Experience on the SPCS COER project indicates that neither of these conditions is a problem if the whole-job team works closely enough with the client.

The whole-job approach does have at least two problems. Close ties to the client tend to emphasize short-term over long-term goals and

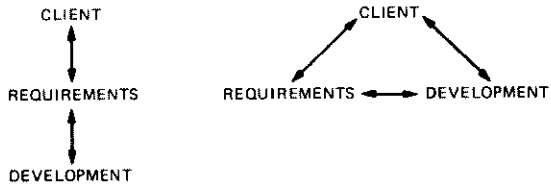



Fig. 4—Communications in a multiorganization software project.

Fig. 5—Communications in a whole-job software project.

activities. Overcoming this calls for very close ties and a willingness to invest time in explanations and discussions. When a client is obdurate, perhaps it really is better to emphasize the short-term goal. A second potential problem is that projects staffed by one or two people would seem extremely vulnerable to sickness and accidents. SPCS COER experience over 10 years shows that this is not a serious problem. Because of the multiply overlapping team design and because people develop friendships with their peers and like to talk, several members of other teams always seemed to know what a missing team member had been doing so that gaps could be quickly closed.

### *4.6 Career flexibility*

System developers often view systems engineers as plodding writers of dull requirements. Systems engineers in turn frequently perceive development folk as narrow crank turners, plagued by self-imposed inflexible deadlines. While there may be something to be said for both points of view, both are distorted views and do not encourage communication and personal growth.

The whole-job approach ensures that individuals are exposed to the realities of both systems engineering and development—both the good and bad of each. This is a broadening experience. Entering the project with a development orientation, as most fresh college graduates do, many whole-job people develop a strong desire to move into full-time systems engineering. In our experience, this rarely happens in conventionally structured developments. Graduates of the whole-job experience have gone on to do well in both systems engineering and development areas. We view this as being greatly to the benefit of both the individual and the organization.

### *4.7 Extending the scope of whole-job possibilities*

Whole-job projects are better for the customer than large multiorganization projects. Customers' needs are met more quickly and at lower cost. Long planning/development cycles are not needed. Customers' needs often change rapidly, and the whole-job approach is well suited to a rejuggling of priorities.

This conclusion seems evident. It is common knowledge that small, highly motivated teams can hit quicker and harder and more accurately than large organizations. In the military, commandos are considered elite. But commandos cannot win wars alone. And probably the whole-job approach cannot work on all software projects.

For small projects of two or three persons, most would agree it can be used. But size alone is not an adequate criterion. The method has worked on one medium-sized (10 to 20 person) project, SPCS COER, but may not work on others of this size. To find a criterion for where the whole-job approach can be applied, let us examine the structure of the project.

SPCS COER is organized horizontally instead of vertically, as shown in Fig. 6. SPCS COER is designed as a set of largely independent facilities. Of nine current and one future facility, their interdependencies graph is shown in Fig. 7. The asterisks represent customers. Most of the facilities depend only on BDE, which is a source of data from the world outside of SPCS COER. The 1 *ESS* subsystem supplies data to the world outside. Remove these two interfaces, and the graph reduces to Fig. 8.

What is remarkable here is the strong parallel between the customer's view of the project and its division into loosely coupled subsystems. Let's examine what is meant by the customer's view. Consider one of these subsystems, VSS COER. One person developed this subsystem. He consulted with the AT&T Network Design and Network Admin-

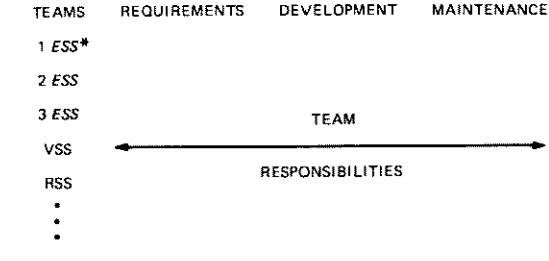
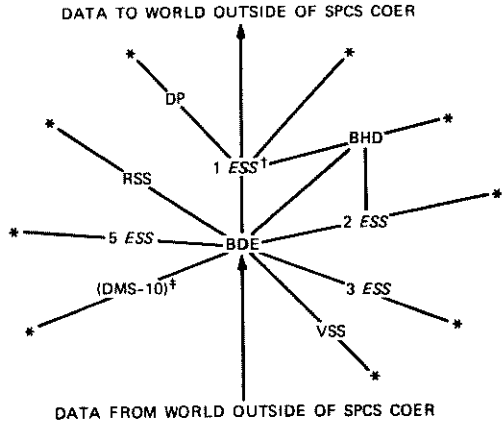


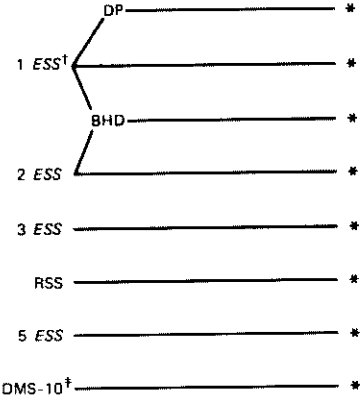

Fig. 6—SPCS COER team organization.




* – CUSTOMERS OF SPCS COER
BDE – BATCH DATA ENTRY
BHD – BUSY HOUR DETERMINATION
DMS – DIGITAL MESSAGE SWITCH
DP – DATA PROFILES
RSS – REMOTE SWITCHING SYSTEM
VSS – VOICE STORAGE SYSTEM

†*ESS* SWITCHING EQUIPMENT. TRADEMARK OF WESTERN ELECTRIC.
‡TRADEMARK OF NORTHERN TELECOM.

Fig. 7—SPCS COER subsystem interdependencies.




* – CUSTOMERS OF SPCS COER
BHD – BUSY HOUR DETERMINATION
DMS – DIGITAL MESSAGE SWITCH
DP – DATA PROFILES
RSS – REMOTE SWITCHING SYSTEM

†*ESS* SWITCHING EQUIPMENT. TRADEMARK OF WESTERN ELECTRIC.
‡TRADEMARK OF NORTHERN TELECOM.

Fig. 8—Simplified SPCS COER subsystem interdependencies.

istration departments in a small meeting. There was no representation from 1 *ESS*, 2 *ESS*, or any other parts of the COER project. The independence of the VSS module from other SPCS COER modules was clear from the customer's point of view. And this is reflected in the structure of this meeting. There are no customer requirements that the VSS database or processes interact closely with any other SPCS COER subsystem.

Consider 1 *ESS*, 2 *ESS*, and 3 *ESS* central offices, and the role of a real network administrator in a Bell Operating Company. One administrator may be responsible for all three types of offices. But this person usually works on offices one at a time. A network administrator may use 1 *ESS* COER for an hour or so examining a problem in one particular office, then switch to 2 *ESS* COER to examine the validity of some new data, and request a report. Although one person may use several modules of SPCS COER, the independence among these modules is still there from the customer's point of view. The independencies derive from the method of use.

So from both requirements and usage points of view, the customer's view of the project is a set of almost independent facilities. And each of these facilities is small enough to be planned and developed by a small team.

Underlying all SPCS COER, however, are common system design decisions: the on-line nature of the system, method of data entry, dialogue style, style of reports, etc. These can be viewed as system design decisions that affect all subsystems; but they were all made some years ago and evolve at a slow pace. It is these underlying design criteria that give SPCS COER a cohesiveness from the customer's point of view, i.e., make it one system.

So our hypothesis is this: It is not the size of a project, per se, that makes the whole-job approach applicable or not. The whole-job approach is applicable whenever:

1. A system can be divided into parts that are independent, from a customer's viewpoint.
2. The resulting parts are loosely coupled.
3. Each part can be planned and developed by a small, whole-job team.

Extending the whole-job concept to large projects requires that they be structured according to these criteria. Large projects that cannot be structured this way can probably only be done by the multiorganization approach.

### *4.8 Whole-job approach summarized*

The whole-job approach to software specification and development is a project method. It is described in the research and development

literature, but apparently not in the software literature. It is a proven technique and has been in use on software projects for at least 10 years. Its central advantage is that developers and customers build a close relation. This leads to realistic design by the designers, and high customer acceptance of the software products. Structure of a project, not size, determines whether this method can be used.

## V. FUTURE DIRECTIONS

SPCS COER, well into its second decade of service, is still growing vigorously in features, component systems, and entities served in response to the expressed needs of its users. But this is only to be expected of a successful product serving the needs of a growing market. We close this paper briefly citing four new and challenging areas toward which SPCS COER thinking is being directed. It is not possible to tell now which of these areas, if any, will bear fruit.

As mentioned earlier, an SPCS COER system was developed for VSS. While VSS COER is not now in operation for reasons having nothing to do with the present paper, this development was an interesting first application of SPCS COER technology outside the realm of switching systems. There appear to be many other potential applications beyond switching that require the capabilities of SPCS COER. Work is presently proceeding to expand our applications in this arena.

The reports produced by SPCS COER are laboriously worked out with user representatives to meet the reporting needs of the BOCs. But surveys have shown that fixed-format reports seldom meet all the needs of individual users. This leads to wasteful manual posting from different mechanized reports to the desired report format. Planning is well advanced, under the title of "Flexible Reporting," for new features that will allow users to alter the format and content of existing reports and create new reports from scratch.

While the development of generic tools, described in Section III, has greatly simplified and speeded the development of new SPCS COER systems, developing a system for each new application is still a significant and time-consuming effort for highly skilled and experienced program designers. Under the name of "Generic COER," also referred to as "User Programmability," planning is under way to allow users to (in effect) create their own component systems for new switching applications. Unlike flexible reporting, the user in "User Programmability" need not be the ordinary SPCS COER user. Most hands-on users of SPCS COER are clerks with little formal training. The user in "User Programmability" could well be a highly trained computer program designer employed by an individual BOC as long as such a resource were readily available to the operating users in the BOC.

Finally, in the 12 years SPCS COER has been operational, there has been a continuing advance in the computing facilities available to the BOCs in their data centers and elsewhere. It now seems possible and economically attractive to move the SPCS COER operation from its single central location for all BOCs into one or more installations in each BOC. Planning work is in progress on this future direction.

## REFERENCES


1. H. J. Dougherty, H. Ragg, P. G. Ridinger, and A. A. Stockert, "No. 1 ESS Master Control Center," B.S.T.J., *43*, No. 5, Part 2 (September 1964), p. 2286.
2. E. P. Gould and J. W. Mosior, "TELPORT - Time Shared Information Systems," Bell Lab. Rec., *46*, No. 6 (June 1968), pp. 197–202.
3. N. R. Sinowitz, "DATAPLUS - A Computer Language for English Speaking People," Bell Lab. Rec., *46*, No. 11 (December 1968), pp. 362–9.
4. M. S. Hall and G. P. Patti, "PATROL Surveys Traffic for No. 1 ESS Offices," Bell Lab. Rec., *51*, No. 3 (March, 1973), p. 72.
5. V. L. Fahrmann, "Reporting Central Office Traffic Information," Bell Lab. Rec., *57*, No. 4 (April 1979), pp. 99–104.
6. M. S. Hall, J. A. Kohut, G. W. Riesz, and J. W. Steifle, "System Plan," B.S.T.J., this issue.
7. K. A. Friedman, "Precutover Extreme Value Engineering of a Local Digital Switch," Proc. of the Tenth Int. Teletraffic Cong., Montreal, June 8–15, 1983.
8. L. E. Heindel and J. T. Roberto, "The Off-The-Shelf System—A Packaged Information Management System," B.S.T.J., *52*, No. 10 (December 1973), pp. 1743–63.
9. E. B. Roberts, "Stimulating Technological Innovation—Organizational Approaches," Res. Manage., *12*, No. 6 (November 1979), pp. 26–30.
10. G. M. Weinberger, *The Psychology of Computer Programming*, New York: Van Nostrand Reinhold, 1971.
11. T. J. Allen, D. M. S. Lee, and M. L. Tushman, "R&D Performance as a Function of Internal Communication, Project Management and the Nature of the Work," IEEE Trans. Eng. Manage., *27*, No. 1 (1980), pp. 2–11.
12. J. D. Goldhar, L. K. Bragaw, and J. J. Schwartz, "Information flows, management styles and technological innovation," IEEE Trans. Eng. Manage., *23*, No. 1 (1976), pp. 51–62.
13. T. J. Allen and S. I. Cohen, "Information flow in research and development laboratories," Admin. Sci. Quart., *14* (1969), pp. 12–19.
14. T. J. Allen and A. R. Fusfeld, "Design for Communication in the Research and Development Lab," Technol. Rev. *78*, No. 6 (May 1976), pp. 65–71.


## AUTHORS


**Richard F. Grantges,** B.S., B.E.E., 1953, University of Minnesota; Bell Laboratories, 1953—. Mr. Grantges was first assigned to long range engineering studies of transmission systems where, in 1954, he demonstrated the economic feasibility of Pulse Code Modulation for exchange carrier. After military service he was transferred to *ESS*® Systems Engineering in 1958 and promoted to Supervisor in 1961, responsible for switching network engineering and the mechanization of central office engineering practices. He was promoted to Head in 1968, responsible for appraisal studies in the Traffic Studies Center. In 1969, he went to AT&T as Traffic Operations Manager—Electronic Switching, responsible for the traffic engineering and administration of local and toll electronic switching machines. Returning to BTL in 1971 as Head of the Operator Services and Traffic Engineering Department, he was active in planning the mechanization of operator services and traffic data collection and processing, particularly the SPCS COER, Small Office Network Data System and EADAS systems. In June 1981, he became responsible for the

systems engineering of the Total Network Data System (TNDS) except for Network Management and Trunking. Member, Tau Beta Pi, Eta Kappa Nu, IEEE.

**Virginia L. Fahrmann,** B.A. (Mathematics), 1963, St. Mary-of-the-Woods College; M.S. (Mathematics), 1965, Purdue University; Bell Laboratories, 1965—. Ms. Fahrmann initially worked on traffic load simulations of 1 *ESS* and No. 1 Traffic Service Position System (TSPS). In 1971 she joined the AT&T Switching Engineering Department, with responsibility for providing technical support in long-range planning studies for local wire centers to Operating Companies. She returned to Bell Laboratories in 1973, and was involved in systems engineering and development for SPCS COER. In 1980 she became Supervisor of one of the two SPCS COER groups. Currently, Ms. Fahrmann supervises a group that works on requirements and development for the Plug-in Inventory Control System/Detailed Continuing Property Record (PICS/DCPR) system, which provides inventory control and maintains investment records for central office equipment.

**Thomas A. Gibson,** B.S.E.E., 1960, Kansas State University; Bell Laboratories, 1960–. At Bell Laboratories, Mr. Gibson did systems engineering studies on Electronic Switching Systems and developed a variety of software systems including the Master Links database management system and SPCS COER. He is currently Head of the Plug-in Inventory Control System/Detailed Continuing Property Record Development and Support Operations Department. Mr. Gibson has chaired a subcommittee of the American National Standards Institute. Member, ACM, Sigma Xi, Eta Kappa Nu.

**Laurence M. Brown,** B.A. (Classics), B.S. (Fundamental Sciences-Computer Science), 1975, Lehigh University; B.A. (Literae Humaniores), 1977, Balliol College, Oxford University, Oxford, U.K.; M.A. (Electrical Engineering), 1978, Stanford University; Bell Laboratories, 1977—. Mr. Brown first worked on the software tools of ESS COER. He is presently Supervisor of the Software Systems Design and Support Group.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Small Office Network Data System

By D. H. BARNES* and J. J. O'CONNOR*

(Manuscript received December 3, 1982)

The Small Office Network Data System (SONDS) is a part of the Total Network Data System (TNDS) designed specifically to meet the network data requirements for the small step-by-step offices of the Bell System. Its ability to accurately relate to the load and service characteristics seen in these small, mostly rural offices has ensured its success. It maintains low cost by measuring and recording a single observation per day for each traffic measurement. This is the daily peak hour load per group. A moving average of the mean and variance of peak hour measurements is interpreted by using a modified-Gumbel extreme value distribution function. This has been demonstrated to fit well within the actual distributions of daily peak hour loads. It is used to construct traffic capacity tables that better reflect customers service experience in these offices than do existing methods. It uses field-tested parameter assumptions of the underlying distribution from which peak values are sampled to estimate equivalent time-consistent busy hour loads relative to the observed peak hour loads. This allows service to be compared among SONDS and non-SONDS step-by-step offices. SONDS uses on-site data collection units, a single central computer for all Bell System offices, and provides local printing of output reports. Data are collected each night and daily, weekly, and monthly output reports are distributed over the switched network.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

## I. INTRODUCTION

The Small Office Network Data System (SONDS) is a miniature Total Network Data System (TNDS) designed to meet the unique and special needs of the Bell System's smaller, rural-type offices. In respect to SONDS a small office is one serving fewer than 5000 main stations (MSs). These comprise almost 40 percent of the Bell System's total switching entities but serve less than 8 percent of its customers. The offices that are candidates for SONDS are predominantly of the noncommon control, step-by-step type. They are generally located in remote rural locations and are unattended.

The amount of traffic and the service vary among these offices and among the serving groups within each office for several reasons. First, as random traffic theory predicts, there is a greater proportional variation in traffic with small versus large groupings of traffic sources. Observations have shown that the actual variations of source traffic are, in fact, much greater than assumed by the random theory underlying the probability tables normally used to size central offices. Second, the mix of customer types varies widely among these offices and this further affects the within-a-day and day-to-day traffic distributions in often unpredictable ways. In addition, as a result of the relatively small originating line and terminating connector groups of the small step-by-step offices, load balancing problems are accentuated. These offices are unattended, which means less frequent surveillance of central office equipment problems such as busy switches or trunks. On the average, there will be more such maintenance problems and they will be longer in duration than in larger, attended offices.

The small offices require special attention if both service and cost objectives are to be met. They are, however, too small and remote to justify support by conventional, continuous measurement systems. Typically, a few days of traffic data per year are available in these offices. Present standard dimensioning techniques and design service criteria are thus inadequate in the presence of minimal data and large traffic variation. We have demonstrated with SONDS how large office standards can be applied to these small offices at an affordable price.

## II. DIMENSIONING OF TRAFFIC SYSTEMS

Design service criteria for standard Bell System engineering are based on the concept of a Time-Consistent Busy Hour (TCBH). It is by definition that clock hour which has been observed by study to contribute the greatest amount of total traffic over some defined period of time. The sizing of a group of traffic servers is based on meeting service criteria for the highest controlling TCBH loads for the average busy season, average ten high day, or highest day.

The traffic theory used to calculate engineering capacity tables assumes the behavior of a constant system of chance causes ("statistical equilibrium"). To account for different levels of departure from truly random behavior, traffic capacity tables are adjusted to allow for options of low, medium, or high levels of day-to-day variation.

Time-consistent busy-hour practice is a compromise for data collection and processing costs. Simplifying assumptions are used, however, which lead, particularly in small offices, to overprovisioning to protect for variations not reflected in an average statistic. Due to lack of data, maintenance problems, and forecasting errors, underprovisioning can also occur.

TCBH practices assume that provision of objective service in the average busy hour will provide adequate service in all hours. Average service is not, however, the service seen by individual customers nor by groups of customers in offices with traffic imbalance. While TCBH service criteria have been shown by experience to be a reasonable approach to provide an adequate overall service control for large groups of customers, they are less valid for small groups because of the greater deviation from truly random behavior of such groups. For example, SONDS office studies reveal that the TCBH is the actual busiest hour of the day on only about 20 percent of the days.

## III. THE SONDS DEVELOPMENT

SONDS is the outcome of a series of Bell Laboratories' traffic studies and computer experiments that were initiated originally to explore the power of computers to completely control data collection from a central location. Included were studies of traffic characteristics and the evaluation of design service criteria that might be used to better control service levels and at the same time provide a more efficient allocation of resources. In recognition of the largely unsolved small office data and design problem, we initiated an experimental network data system designed to meet that need. These relatively simple offices, largely unmeasured, were judged to be a good environment to explore a completely mechanized system that would use time-shared computers to interface with smart microprocessor-controlled central office terminals. We were looking for a system free from the control of people other than the direct user. These offices were not directly coupled to TNDS and thus gave us a simpler environment in which to study innovative traffic systems.

The SONDS development, the formulation of analysis requirements, and the specifications of output reports were designed to incorporate:

- Low cost
- Service measurement
- A compact database

- A nearly paperless system
- Switched-network data communication
- An interactive computer interface
- Automatic report generation and distribution
- Complete access by the authorized user for:
  - —Nonscheduled data or special study requests
  - —Database control
  - —Trouble reports and data for analysis
  - —Remote testing of terminal equipment.

SONDS has met these desired objectives. Its success is due to a combination of the following features:

1. An Extreme Value Engineering (EVE) method has been formulated and adapted to the studied characteristics of the candidate offices. EVE uses order statistics of an extreme value distribution formulated from daily peak hour data observations. The need for simplifying assumptions using other traffic dimensioning models is largely reduced. Through use of the order statistics the computer is able to detect questionable data and to issue timely exception reports, including backup data, which are useful for analyzing and correcting trouble conditions.

2. New EVE design service criteria are based on a broad study of office characteristics and are selected to match the objectives of current dimensioning methods. On the average, with the same or less switching equipment and trunks, overall service will improve. With full-time data surveillance available for the first time and with a good system of trouble detection, there will be less need to play safe by overprovisioning.

3. A compact database is realized because only a single hourly value per day, the daily peak hour value, is recorded in the database for each traffic measurement. The extreme value distribution parameters are estimated from a moving average of the mean and variance of this measurement.

4. Automatic computer polling of office terminals during early morning hours is an efficient serial use of computer ports and the switched network.

5. A largely paperless system has been realized. The standard reports required in TNDS are automatically distributed immediately following the end of each reporting interval. Database updates and requests for special data and reports are made via an interactive interface with the computer and require no other (human) interactions.

6. No special monitoring or scaling is required to compensate for holiday or "odd-ball" traffic conditions in the office engineering because automatic outlier tests reject such data.

7. Dial tone delay service measurements and their reporting as a service index are contained in the system. Separate and costly dial tone test call facilities are not needed. The result is a more accurate measurement and includes a measure of group-to-group service differences not available otherwise.

## IV. FORMULATION OF THE EVE METHOD FOR SONDS OFFICES

The following is a short history of work that helped generate an interest in the EVE approach used in SONDS.[1-4]

### *4.1 Early work*

Karlsson[1] claimed that, particularly for international routes, the (time-consistent) busy hour is both unstable in amplitude and unfixed in time. He proposed the dimensioning of traffic routes from integrated information of all traffic peaks taken over a sufficiently long time interval $T$, typically six hours a day for five days, to give good statistical stability. This was an early attempt to recognize the importance of differences in the variances of traffic groupings with equal loads.

Gershwin, Laue, and Wolman[2,3] solved a problem related to the load administration of Subscriber Line Concentrators (SLC) using extreme value statistics. The problem was to administer line assignment far enough ahead so that concentrated lines could see nearly zero blocking in this additional switching stage. Extreme value methods were used to keep peak hour blocking less than 0.005 except for a few times per year.[4]

Unlike Karlsson, Gershwin et al. retained hourly measurements but dropped the concept of a time-consistent busy hour. They recorded only a weekly peak load. The mean and variance of these weekly peak loads were found to be good estimates of the fitting parameters of the Gumbel extreme value distribution function.[5] The plan was simple and inexpensive and served well to direct line assignments and to provide capacity estimates for planning purposes.

### *4.2 Daily peak values*

The initial work that led to extreme value engineering as used by SONDS is published in Refs. 6, 7, and 8. In SONDS it is important that traffic reports and service measurements be integrated into and be consistent with the rest of TNDS. It was thus important, in contrast to weekly values used for the SLC, to follow daily busy hour and busy season concepts and to furnish monthly traffic reports consistent with other TNDS reports. The use of daily peak values for design and administration of small offices was of particular interest because we had experimented with techniques to measure, collect, and summarize such data at a very low cost.

Day-to-day variation of TCBH originating traffic loads as observed in step-by-step offices is illustrated in Fig. 1. The coefficient of variation in percent is shown as a function of offered load in erlangs. Curve A is an approximate fit to the TCBH loads observed in many step-by-step offices. The data showed considerable scatter, indicating that day-to-day variation is more than just a function of the offered traffic. Curve C is a theoretical curve based on the assumption of constant system of chance causes ("statistical equilibrium") which is basic to the probability theory used to construct traffic capacity tables. Curve B depicts a possible location for daily peak-hour data. Its location between the theoretical curve C and the TCBH curve A will be a function of the number of hours in the day that have a load high enough to possibly be the peak load for the day. For calculation purposes we will develop an equivalent number of equally loaded hours with which to describe the underlying distribution from which peak hours come. We designate these as "Candidate Busy Hours".

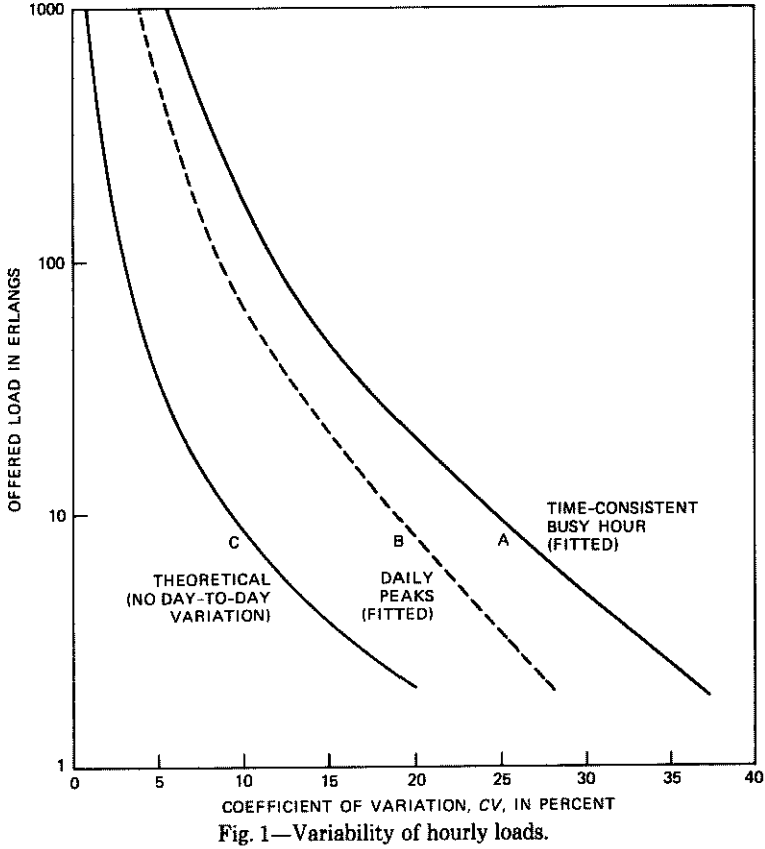



Fig. 1—Variability of hourly loads.

Figure 2 illustrates a comparison between TCBH and peak-hour concepts using actual data. These data were taken from a typical line finder group in a SONDS office, which had 13 line finders serving the group. The data are plotted on probability paper constructed to match the normal-to-the-sixth extreme value probability function derived in Section 4.4.

Curve A is a theoretical curve based on purely random, no day-to-day variation; it is the underlying distribution of the TCBH capacity tables. We could not plot this curve exactly as a straight line on this graph. To simplify the illustration we have approximated it as two line segments plotted between the 0.05 and the 0.95 probabilities points and the average value. This average value was scaled to be equal to the 242 hundred call seconds (CCS) measured average of the TCBH data. This gives the 0.95 point, equivalent to a 20-day return period load, a load of 286 CCS.

Curve B is a plot of the 20 TCBH loads observed in this office. It is seen that these data points significantly deviate from the theoretical curve A, and this is due in part to actual day-to-day variations. Note that the observed 20-day return period load, the 0.95 point, is 326 CCS during the time-consistent busy hour.

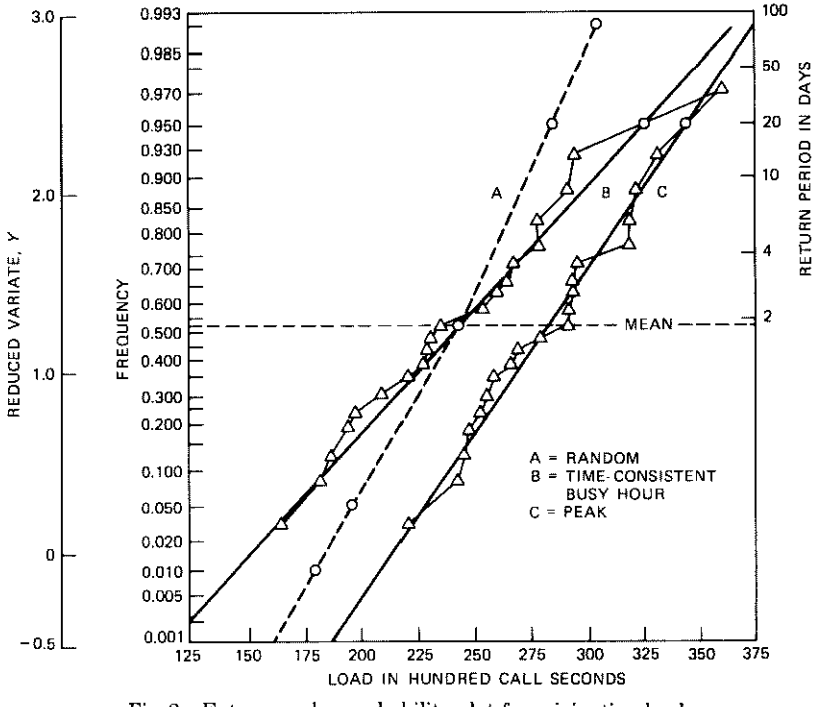



Fig. 2—Extreme value probability plot for originating loads.

Curve C is a plot of the 20 daily peaks taken for the same days as the TCBH data. The straight-line character of the data on this graph is typical of the agreement between daily peak data and the normal-to-the-sixth distribution function. Note that the mean value is 283 CCS and the 0.95 value is 346 CCS. While the mean and 0.95 values of the peak curve are larger than those of the TCBH curve, the larger slope of the peak curve indicates less variance, that is, daily peak values are less volatile than TCBH values!

These several load values obtained from Fig. 2 can be used to compare the dial tone delay service as given by current engineering capacity tables with the range of expected values that a customer will experience. This is given in Table I, which shows the service values for both mean and 20-day return period loads. Since the 20-day return period load $X_{20}$ is also interpreted as the load to be exceeded once a month, on the average, we can examine the service implications on this basis. Table I shows that a line finder group properly designed to give 1.5-percent dial tone delay during the time-consistent busy hour would be expected to exceed 4.3 percent once a month, even with purely random traffic. The day-to-day variation observed in Fig. 2 would increase this to 8.7 percent, and the use of the daily peak-hourly loads of Fig. 2 would predict a service level of 12.5 percent to be exceeded once a month.

Extreme value statistical methods can be applied to load balance. Loads are balanced through line assignment, which takes into account both additions and disconnects. Table II shows an office TCBH and peak load averages and their coefficients of variation, $CV$, using a month of data. The last column, $X_{20}$, is the 20-day return period load calculated from the mean and standard deviation of the daily peak values. As expected, we can see that the $CV$ of the daily peak loads (12 percent) is less than that of the TCBH loads (22 percent). Also note that the $CV$ of the peak loads (6 percent) across the loading division is less than that of the TCBH loads (7.3 percent). Even the $X_{20}$, once-a-month load, which includes the additional parameter of variance, has only a $CV$ of 7.9 percent.

Table III shows how the corrective action needed to balance load against load main station movement depends on the type of data used

Table I—Load service characteristics

| | Load in CCS | | Expected Dial Tone Delay (13 Line Finders) | |
|---|---|---|---|---|
| Distribution | Mean | $X_{20}$ | Mean | $X_{20}$ |
| A–Random | 242 | 293 | 1.5 | 4.3 |
| B–TCBH | 242 | 326 | 2.1 | 8.7 |
| C–Peak BH | 283 | 346 | 4.2 | 12.5 |

Table II—Originating load distributions

| Line Finder Group No. | TCBH CCS | TCBH *CV* (Percent) | Daily Peaks CCS | Daily Peaks *CV* (Percent) | $X_{20}$ CCS |
|---|---|---|---|---|---|
| 1 | 159 | 21 | 185 | 11 | 220 |
| 2 | 155 | 19 | 199 | 8 | 226 |
| 3 | 139 | 17 | 185 | 8 | 211 |
| 4 | 179 | 17 | 214 | 9 | 249 |
| 5 | 169 | 25 | 220 | 14 | 273 |
| 6 | 147 | 21 | 200 | 13 | 244 |
| 7 | 162 | 22 | 196 | 12 | 238 |
| 8 | 164 | 26 | 193 | 18 | 252 |
| 9 | 162 | 29 | 207 | 13 | 253 |
| Avg | 160 | 22 | 200 | 12 | 241 |
| Total percent of *CV* | | 7.3 | | 6.0 | 7.9 |

Table III—Originating main station balance

| Line Finder Group No. | TCBH | Daily Peaks | $X_{20}$ |
|---|---|---|---|
| 1 | −1 | −15 | −18 |
| 2 | −6 | −1 | −13 |
| 3 | −27 | −15 | −25 |
| 4 | +24 | +14 | +7 |
| 5 | +12 | +20 | +27 |
| 6 | −17 | 0 | +3 |
| 7 | +3 | −4 | −3 |
| 8 | +5 | −7 | +9 |
| 9 | +3 | +7 | +10 |
| Avg. | 11 | 9 | 12 |

from Table II. The best service balance will be accomplished using $X_{20}$ data. Note the very large differences by individual groups between the movement indicated by TCBH and that indicated by $X_{20}$ data.

### *4.3 Gumbel distribution*

Preliminary work on SONDS, following the direction of the previous work, used the Gumbel[5] extreme value distribution, a double exponential distribution, shown below.

$$G(x) = \exp\{-\exp[-\alpha(x - \mu)]\}. \quad (1)$$

$G(x)$ is the probability that the largest observation in a large set of observations is less than the value $x$. In this expression $\mu$ is a location parameter (the mode), and $\alpha$ is an inverse measure of dispersion. The mean of the distribution is:

$$E(x) = \mu + \gamma/\alpha, \quad (2)$$

where $\gamma = 0.57721$ (Euler's Constant) and the variance is:

$$V(x) = \frac{\Pi^2}{6\alpha^2}. \tag{3}$$

The Moments Method may be used to estimate $\mu$ and $\alpha$ by replacing $E(x)$ and $V(x)$ with their sample values from a sample of $N$ hourly peaks $x_1, \cdots, x_N$. The sample mean is:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4}$$

and the sample variance is:

$$\bar{s}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2. \tag{5}$$

Substituting eqs. (4) and (5) into eqs. (2) and (3), we have:

$$\hat{\mu} = \bar{x} - \gamma/\hat{\alpha} \tag{6}$$

and

$$\hat{\alpha} = \frac{\Pi}{\bar{s}\sqrt{6}}. \tag{7}$$

Data points fitting a Gumbel distribution function should plot as a straight line on Gumbel extreme value probability paper. Figure 3 shows an attempt to do this with hourly peak data obtained in a Private Branch Exchange (PBX) office serving a single business customer.[6] Notice that the data points do not fall within the control limits. A number of the low peaks were identified as holidays and days preceding holidays. By applying rejection test statistics to one low point at a time and recalculating the order statistics until the lowest point fell within control limits, it was possible to plot the data as shown in Fig. 4, after rejecting only nine of the original 80 data points. This method appeared to properly represent the high data points of most interest and automatically reject data of little interest, such as holidays.

If we reject the lowest few points to force a straight line fit of data to the double exponential probability plot, which is useful for eliminating holiday traffic, we also might eliminate some low data points necessary to identify problems in SONDS. This forced rejection of the lower data points makes the estimate of the mean of the daily peaks biased on the high side and the estimate of the variance biased on the low side, making the effect on the resulting estimated 20-day return period load indeterminate.

We realized that the problem arises because the daily peak loads do not satisfy the large sample assumption that leads to the double

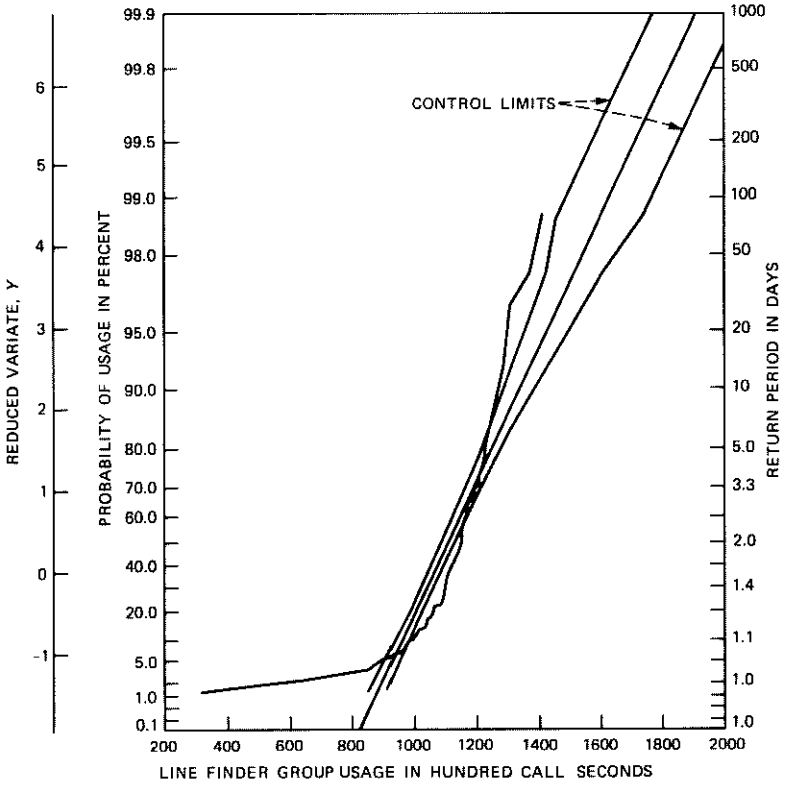



Fig. 3—Extreme value probability plot for 80 peak hours of data.

exponential distribution function of eq. (1). Instead of selecting a peak value from among a large number of candidate hours, we find that we are typically selecting daily peaks from a few to perhaps ten candidate busy hours.

### *4.4 Normal-to-the-sixth distribution*

An improved extreme value distribution function model was needed to fit the small office traffic data. A set of statistical test procedures was needed in the initial start-up testing of a SONDS office (see Section 9.1) to resolve installation problems and other central office problems not previously detected for lack of data.

The first step was to test the distribution of equipment loads across days for the busier hours of each of many offices. The findings were that the distributions satisfactorily tested for normality. The second step was to test whether round-the-clock hourly data can be modeled adequately by $h$ equally loaded candidate busy hours even though the actual clock hours do not have equal loads. These tests led to the

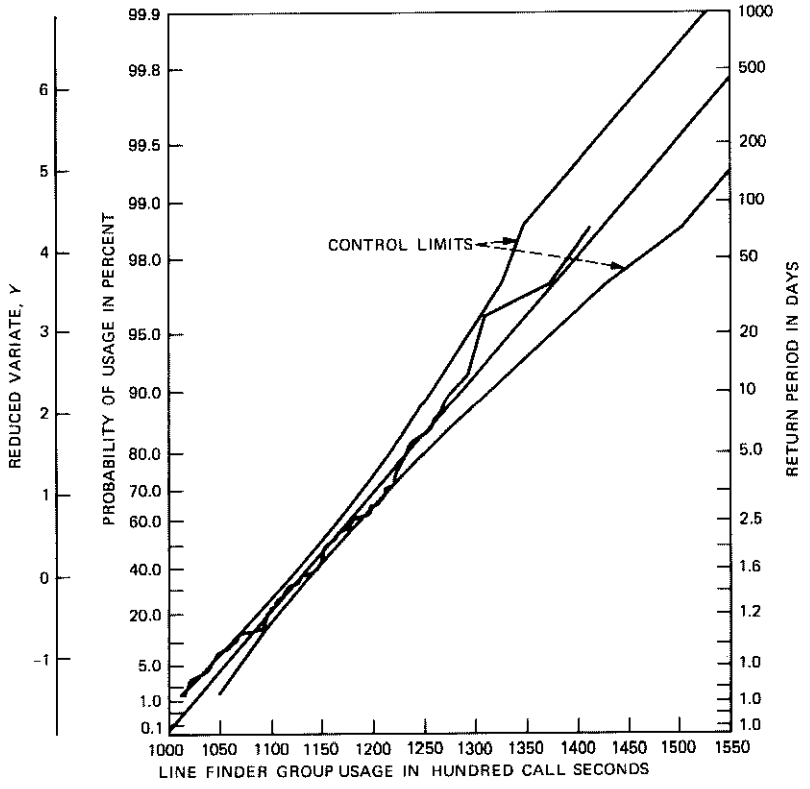



Fig. 4—Extreme value probability plot for 71 peak hours of data.

conclusion that a modified extreme value distribution function suitable for SONDS data is the normal probability distribution function raised to an optimum power of $h$. Table IV gives the results of three tests that resolved the optimum value of $h$ as 6.

Therefore, the modified extreme value distribution function for describing daily peak-hour usage data, $x$, is the normal distribution raised to the sixth power, $F(x)$:

$$F(x) = \left[\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x} \exp[-(t-\mu)^2/2\sigma^2]dt\right]^6, \tag{8}$$

where $\mu$ and $\sigma$ are the mean and standard deviation of the underlying equal "candidate busy hours" load distributions.

Tippett[9] numerically evaluated the probability of finding the largest value of $h$ samples drawn from a normal distribution. In particular for $h = 6$, in terms of the normalized variate, the mean is:

$$E(x) = \mu + 1.267\sigma \tag{9}$$

Table IV—Selection of number of candidate busy hours, $h$, in the (NORMAL)$^h$ model

| Tests | Ranges of $h$ Indicated by Tests |
|---|---|
| Distribution of daily peak loads among hours during the day | 6–10 |
| Validation and monitoring of lower daily peak loads | 4–6 |
| Estimating highest of 20 daily peaks | 6–9 |
| Collectively (all of the above) | 4–10 |
| Value selected | 6 |

and the variance is:

$$\mathrm{Var}(x) = 0.416\sigma^2. \tag{10}$$

In SONDS using the moments method, $\mu$ and $\sigma$ are estimated by matching the first two moments of the distribution of eq. (8) to their unbiased sample estimates. Replacing $E(x)$ and $\mathrm{Var}(x)$ by the unbiased sample estimates of the mean, $\bar{x}$, and variance, $\bar{s}^2$, the moments method equations are obtained:

$$\bar{x} = \hat{\mu} + 1.267\hat{\sigma} \tag{11}$$

and

$$\bar{s}^2 = 0.416\hat{\sigma}^2. \tag{12}$$

Manipulating eqs. (11) and (12), we have the parameters of the normal-to-the-sixth-distribution in terms of the underlying daily peak parameters as

$$\hat{\mu} = \bar{x} - 1.96\bar{s} \tag{13}$$

and

$$\hat{\sigma} = 1.55\bar{s}, \tag{14}$$

where $\bar{x}$ and $\bar{s}$ are the unbiased sample estimates, calculated directly from the daily hourly peaks $x_i$, as shown in eqs. (4) and (5).

The 95th percentile of $F(x)$ can be calculated directly from eq. (8) by setting $F(x) = 0.95$ to give

$$x_{20} = \hat{\mu} + 2.39\hat{\sigma}, \tag{15}$$

which by eqs. (13) and (14) can be stated in terms of the daily peak values as

$$x_{20} = \bar{x} + 1.74\bar{s}. \tag{16}$$

This $x_{20}$ is the load that is exceeded once, on the average, every 20 business days. It is also called the once-a-month load.

## V. EVE SERVICE CRITERIA

The use of daily peak-hour measurements to estimate a once-a-month load reflects to a greater degree the service viewpoint of customers than do present average time-consistent busy hour practices. The estimated once-a-month load is based on continuous monitoring of all hourly data, and as interpreted in SONDS by the $F(x)$ model of eq. (8), takes into account the variability of loads.

In order to use a once-a-month load for engineering and administration, new peak service criteria and corresponding load objectives must be determined. For small step-by-step offices, peak service criteria have been selected for line finders, connectors, selectors, and only-route trunk groups.

Peak service criteria are determined by calculating the expected blocking or delay that would occur in offices if the once-a-month loads were offered to the amount of equipment required by time-consistent practices. Continuous collection of hourly traffic loads in the study phase of SONDS provided the data needed for this comparison. The following are the objectives for selecting peak service criteria:

1. Provide overall service that is at least as good as that resulting from use of the current time-consistent criteria
2. Provide better service during peak load periods than results from using the current time-consistent criteria
3. Provide service to customers that is independent of the size of the serving office or equipment group
4. Require no more total equipment in the Bell System than is in current use.

Having resolved by field study the EVE blocking objectives for the various components relative to their present TCBH blocking objectives, it is a simple matter to extend the existing capacity tables to the new blocking values and calculate equivalent tables to replace existing TCBH tables. The existence of these extreme value capacity tables is necessary to apply EVE techniques to central office engineering, as discussed in the next section.

## VI. TRAFFIC ENGINEERING USING EVE

In the traffic engineering of step-by-step offices, it is necessary to distribute the forecasted originating, local, and incoming traffic loads and to provide equipment quantities to meet objective service criteria.

Customers are associated with a group of line finders that return dial tone and connect the customer to a first selector, which will respond to the first dial pulses. A selector steps vertically in response to each digit (0–9) dialed. It then searches horizontally up to ten terminals for an idle path to a following stage of selectors or for an outgoing trunk. Successive selector stages respond to successive dialed

digits to direct traffic toward outgoing trunks or intraoffice destinations. In the last switching stage a connector completes the call to the customer line corresponding to the telephone number. A connector responds vertically and horizontally in response to the last two numbers dialed. Incoming traffic from other offices is served by incoming selectors associated with incoming trunks. The number of selector switching stages and how the traffic mixes prior to switching to the customer's connector group depend on the numbering plan and size of the office.

To provide measurements for each part of the traffic flow would be too expensive. Using time-consistent practices it has been customary to estimate unmeasured traffic by adding and subtracting traffic measured in other parts of the network. With EVE this is not feasible because the peaks in the various parts of the traffic flow may not occur at the same time.

There are two relatively simple answers to estimating equivalent time-consistent traffic loads, which then may be apportioned as is done in existing step-by-step practices. Both depend to a degree on empirical relationships observed in small offices and relate to the candidate busy hour concept.

The first approach is based on analysis of field data, as was done to determine $h = 6$ of Table IV. In this case, the best fit between the daily peak-hourly values and the TCBH values is for $h = 2.5$. The mean and the standard deviation of the normalized variate for $h = 2.5$ are 0.725 and 0.790, respectively.[7] Using these values in eqs. (11) and (12) we calculate the TCBH equivalent load as:

$$\mu_{\text{TCBH}} = \bar{x} - 0.918\bar{s}, \tag{17}$$

where $\bar{x}$ and $\bar{s}$ are defined in eqs. 4 and 5.

A second method—more appropriate, we think, for estimating the TCBH mean value—is the use of the SONDS peak traffic capacity tables. These tables were obtained by deriving the equivalence over many offices between time-consistent busy hour service criteria and peak service criteria. Traffic capacity tables are available in the SONDS computer for both the peak loads and equivalent time-consistent loads. Using the load values in these tables, we can convert an observed peak load to a time-consistent load.

Table V shows an example comparing the estimates of time-consistent loads derived from once-a-month values with actual measured time-consistent loads for a typical mix of traffic. Both the normal-to-the-2.5 and the capacity table methods give satisfactory results. The table method is preferred, however, because it will come closer to correcting the unmeasured traffic to SONDS objective service levels.

Table V—Intraoffice* traffic estimates (in CCS)

| | Once-a-Month Actual | Time-Consistent Estimates | | TCBH Actual |
|---|---|---|---|---|
| | | $h$=2.5 | Table | |
| Terminating | 2789 | 2503 | 2418 | 2512 |
| Incoming | 1961 | 1737 | 1670 | 1757 |
| Local | | 766 | 748 | 755 |

* Intraoffice = Terminating-Incoming

## VII. SONDS DIAL TONE DELAY ANALYSIS

For SONDS offices, a dial tone delay service index is required for peak measurements, and it should be able to be compared directly with non-SONDS offices using TCBH measurements. The following paragraphs describe the procedures used to make this conversion.

In current practice the service index relates monthly average service results to objective capacity levels. SONDS introduces the concepts of the monthly Average Measured Peak Service (AMPS)[8] as the average of the peak values of the Dial Tone Delay (DTD) results for all the line finder groups in a loading division. An AMPS objective is chosen for the peak service criterion, thereby including the measured volatility of loads within the service objective.

The coefficient of variation, $CV$, of the current once-a-month load is found from eq. 16 to be:

$$CV = \bar{s}/\bar{x} = (x_{20} - \bar{x})/1.74\bar{x}. \tag{18}$$

An empirical relation between $CV$ and load has been determined.[8] The $CV$ of the capacity load is estimated from that of the measured load by:

$$CV_{CAP} = CV\left(\frac{x_{20}}{x_{CAP}}\right)^{0.432}. \tag{19}$$

The value $x_{CAP}$ is the $x_{20}$ value at capacity. Given the $CV_{CAP}$ and the number of line finders per group, the AMPS objective can be selected from Table VI.

These peak values of AMPS and AMPS objectives can be converted[8] to DTD TCBH equivalents from the following empirical relationship derived from data studies.

$$\frac{TCBH\ DTD}{TCBH\ DTD_{OBJ}} = 0.85\left(\frac{AMPS}{AMPS_{OBJ}}\right) + 0.17\left(\frac{AMPS}{AMPS_{OBJ}}\right)^2. \tag{20}$$

SONDS uses this equation to convert the measured AMPS values to equivalent TCBH DTD values. These values of TCBH DTD can be entered directly into current, accepted index tables.

Table VI—AMPS objective (percent) coefficient of variation of daily peak loads at capacity

| | | 0 | 0.05 | 0.10 | 0.15 | 0.2 | 0.25 | 0.3 | 0.35 | 0.4 | 0.45 | 0.5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of line finders per group | 1 | 9.0 | 8.3 | 7.6 | 7.2 | 6.7 | 6.3 | 5.9 | 5.6 | 5.3 | 5.1 | 4.8 |
| | 2 | 9.2 | 7.9 | 6.9 | 6.1 | 5.5 | 5.0 | 4.6 | 4.2 | 3.9 | 3.7 | 3.5 |
| | 3 | 9.3 | 7.8 | 6.6 | 5.6 | 4.9 | 4.4 | 3.9 | 3.6 | 3.3 | 3.1 | 2.9 |
| | 4 | 9.4 | 7.4 | 6.0 | 5.0 | 4.3 | 3.8 | 3.4 | 3.1 | 2.8 | 2.7 | 2.5 |
| | 5 | 9.6 | 7.4 | 5.8 | 4.8 | 4.0 | 3.5 | 3.1 | 2.9 | 2.7 | 2.5 | 2.4 |
| | 6 | 9.7 | 7.1 | 5.4 | 4.4 | 3.7 | 3.2 | 2.8 | 2.6 | 2.4 | 2.3 | 2.2 |
| | 7 | 9.7 | 7.0 | 5.3 | 4.2 | 3.5 | 3.0 | 2.7 | 2.5 | 2.3 | 2.2 | 2.1 |
| | 8 | 9.9 | 6.9 | 5.1 | 4.0 | 3.3 | 2.9 | 2.6 | 2.4 | 2.2 | 2.2 | 2.1 |
| | 9 | 9.9 | 6.8 | 4.9 | 3.8 | 3.2 | 2.8 | 2.5 | 2.3 | 2.2 | 2.1 | 2.1 |
| | 10 | 10.0 | 6.6 | 4.8 | 3.7 | 3.0 | 2.6 | 2.4 | 2.3 | 2.2 | 2.1 | 2.0 |
| | 11 | 10.1 | 6.6 | 4.7 | 3.6 | 3.0 | 2.6 | 2.4 | 2.2 | 2.1 | 2.1 | 2.1 |
| | 12 | 10.2 | 6.5 | 4.5 | 3.5 | 2.9 | 2.5 | 2.3 | 2.2 | 2.1 | 2.1 | 2.1 |
| | 13 | 10.3 | 6.4 | 4.4 | 3.3 | 2.8 | 2.5 | 2.3 | 2.2 | 2.1 | 2.1 | 2.1 |
| | 14 | 10.3 | 6.4 | 4.3 | 3.3 | 2.7 | 2.4 | 2.3 | 2.2 | 2.1 | 2.1 | 2.1 |
| | 15 | 10.4 | 6.3 | 4.3 | 3.2 | 2.7 | 2.4 | 2.3 | 2.2 | 2.2 | 2.1 | 2.1 |
| | 16 | 10.5 | 6.2 | 4.2 | 3.2 | 2.7 | 2.4 | 2.3 | 2.2 | 2.2 | 2.2 | 2.2 |
| | 17 | 10.5 | 6.1 | 4.1 | 3.1 | 2.6 | 2.4 | 2.3 | 2.2 | 2.2 | 2.2 | 2.2 |
| | 18 | 10.6 | 6.0 | 4.0 | 3.0 | 2.6 | 2.3 | 2.2 | 2.2 | 2.2 | 2.2 | 2.2 |
| | 19 | 10.7 | 6.0 | 3.9 | 3.0 | 2.6 | 2.4 | 2.3 | 2.2 | 2.2 | 2.2 | 2.2 |
| | 20 | 10.7 | 6.0 | 3.9 | 3.0 | 2.5 | 2.4 | 2.3 | 2.2 | 2.2 | 2.3 | 2.2 |
| | 21 | 10.8 | 5.9 | 3.8 | 2.9 | 2.5 | 2.3 | 2.3 | 2.2 | 2.2 | 2.3 | 2.3 |
| | 22 | 10.9 | 5.8 | 3.8 | 2.9 | 2.5 | 2.3 | 2.3 | 2.3 | 2.3 | 2.3 | 2.3 |
| | 23 | 10.9 | 5.8 | 3.7 | 2.8 | 2.5 | 2.3 | 2.3 | 2.3 | 2.3 | 2.3 | 2.4 |
| | 24 | 11.0 | 5.8 | 3.7 | 2.8 | 2.5 | 2.4 | 2.3 | 2.3 | 2.3 | 2.4 | 2.4 |
| | 25 | 11.1 | 5.7 | 3.6 | 2.8 | 2.5 | 2.4 | 2.3 | 2.3 | 2.4 | 2.4 | 2.4 |
| | 26 | 11.1 | 5.7 | 3.6 | 2.8 | 2.5 | 2.4 | 2.3 | 2.3 | 2.4 | 2.4 | 2.4 |
| | 27 | 11.2 | 5.7 | 3.6 | 2.8 | 2.5 | 2.4 | 2.4 | 2.3 | 2.4 | 2.4 | 2.5 |
| | 28 | 11.2 | 5.6 | 3.5 | 2.7 | 2.5 | 2.4 | 2.4 | 2.3 | 2.4 | 2.5 | 2.5 |
| | 29 | 11.3 | 5.6 | 3.5 | 2.7 | 2.5 | 2.4 | 2.4 | 2.4 | 2.5 | 2.5 | 2.5 |
| | 30 | 11.3 | 5.6 | 3.5 | 2.7 | 2.5 | 2.4 | 2.4 | 2.4 | 2.5 | 2.5 | 2.6 |

## VIII. SONDS LINE ASSIGNMENT AND LOAD BALANCE

Line assignment is an important task of the network administrator and a load balance index is part of TNDS. The procedures for both of these are well established but are based on TCBH techniques. Farel[10] derived new calculations to use the once-a-month loads obtained by SONDS. Consistent with the previous techniques, the SONDS line assignment recommendations are smoothed over time so that stable line assignment procedures are obtained. Also the load balance index reflects the level of load carried by the office; in particular, it recognizes that bad load balance does not produce bad service in a lightly loaded office.

Farel's line assignment algorithm[10] is based on the load deviation of each line finder group from the average load of the loading division, as follows:

$$\tilde{u}_i(t) = w_i(t)F \tag{21}$$

and

$$w_i(t) = a[\tilde{u}_i(t-1) + \lambda(k_i - \bar{k})] + (1-a)y_i(t), \tag{22}$$

where:

$\bar{u}_i(t)$ is the smoothed once-a-month load deviation for the $i$th line finder group for month $t$,

$\bar{u}_i(t-1)$ is the same quantity for the previous month,

$\lambda$ is the CCS/MS derived for the current month by dividing the sum of the once-a-month loads in hundreds of call seconds (CCS) for the loading division by the sum of the main stations (MS) in the loading division,

$k_i$ is the change in the number of main stations from last month for the $i$th line finder group,

$\bar{k}$ is the average of $k_i$,

$y_i(t)$ is the deviation of the once-a-month load for the $i$th line finder group from the average of the loading division,

$a$ is a constant that weights the previous month's data to those of the current month,

and

$$F = 1 - (N-3)(.034\bar{z})^2/\sum w_i^2, \tag{23}$$

where:

$N$ is the number of line finder groups in the loading division,

$\bar{z}$ is the loading division average of the line finder group once-a-month loads,

$w_i$ is an intermediate variable, given by eq. (22).

The line assignment recommendation for the $i$th line finder is:

$$q_i(t) = -\bar{\mu}_i(t)/\lambda \tag{24}$$

rounded to the nearest integer. A positive value of $q_i(t)$ indicates that lines should be added to line finder group $i$.

Examination of eq. (22) shows that the inclusion of the $k$ term accounts for the line assignment activity (connects and disconnects) of the previous month.

The value of $a$ has been set at 0.32 from data studies. This means that 68 percent of the weight of the current line assignment recommendation is based on the current month's data; 32 percent of the weight is based on history. This leads to consistent month-by-month line assignment recommendations.

The quantity $F$ of eq. (23) is the James-Stein factor and is a measure of the confidence to be placed on the calculations based on statistical volatility of the data. It is always less than unity and typically runs about 0.9.

The same quantities used to generate the line assignment recommendations are also used in the load balance index calculation.[11] First the nonstatistical deviation of the current month's loads $D(t)$ is calculated as:

$$D(t) = \sum w_i^2/N - (.034\bar{z})^2(N-1)/N. \tag{25}$$

This current month's value is used with that of the previous month $D(t-1)$ to derive a smoothed value $L$ by a constant $b$ as follows:

$$L = bD(t-1) + (1-b)D(t). \tag{26}$$

The value of $b$ has been set at 0.7 from data studies. This means that 30 percent of the weight of the smoothed deviation is based on the current month's data; 70 percent is based on history.

Finally, an imbalance factor is derived as:

$$(V/M)_{\text{OAM}} = L/\bar{z}. \tag{27}$$

While this factor is a direct measure of the load imbalance, the calculation of the index has to account for the level of load being carried by the office, called percent capacity $C$. It is calculated as:

$$C = \bar{z}/z_{\text{CAP}}. \tag{28}$$

The value of $z_{\text{CAP}}$ is obtained from the SONDS capacity table for the number of line finders in the line finder group.

The imbalance factor and the percent capacity are used to derive the load balance index, as shown in Table VII. This table shows that a given imbalance factor results in a lower index as the carried load approaches capacity. Therefore, the index is a measure of service as seen by the subscribers, while the imbalance factor is a message to the network administrator to bring the office into balance as the load builds up.

The imbalance factor is calculated for each loading division because the lines are assigned by loading division. The office load balance index is derived by combining the imbalance factors of all the loading

Table VII—SONDS load balance index

| Imbalance Factor $V/M_{\text{OAM}}$ | Percent Capacity | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ≥96 | 95–86 | 85–76 | 75–66 | 65–56 | 55–46 | 45–36 | ≤35 |
| 0–0.8 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 0.8–1.3 | 99 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |
| 1.3–1.7 | 98 | 99 | 99 | 100 | 100 | 100 | 100 | 100 |
| 1.7–2.1 | 97 | 98 | 99 | 99 | 100 | 100 | 100 | 100 |
| 2.1–2.6 | 96 | 97 | 98 | 99 | 100 | 100 | 100 | 100 |
| 2.6–3.3 | 94 | 96 | 97 | 98 | 99 | 100 | 100 | 100 |
| 3.3–4 | 92 | 95 | 96 | 98 | 99 | 99 | 100 | 100 |
| 4–5 | 90 | 93 | 95 | 97 | 98 | 99 | 100 | 100 |
| 5–6 | 85 | 92 | 94 | 96 | 98 | 99 | 99 | 100 |
| 6–7 | 75 | 90 | 93 | 95 | 97 | 98 | 99 | 99 |
| 7–8 | 63 | 87 | 91 | 94 | 96 | 98 | 99 | 99 |
| 8–10 | 45 | 82 | 90 | 93 | 96 | 97 | 98 | 99 |
| 10–13 | 18 | 68 | 87 | 92 | 95 | 97 | 98 | 99 |
| 13–16 | 11 | 57 | 84 | 91 | 94 | 96 | 97 | 98 |
| 16–25 | 1 | 20 | 64 | 87 | 92 | 95 | 97 | 98 |
| 25–33 | 0 | 11 | 48 | 83 | 91 | 94 | 96 | 97 |
| 33–50 | 0 | 6 | 28 | 74 | 89 | 93 | 95 | 97 |
| ≥50 | 0 | 1 | 14 | 61 | 86 | 91 | 94 | 96 |

divisions according to the relative number of main stations within each loading division.

## IX. ANALYSIS OF EVE DATA

One of the advantages of EVE techniques is that one can determine if a set of data points follows an extreme value distribution by establishing a set of quality control limits. It is seen that the data points of Fig. 3 violate their quality control limits. When nine low-side outliers were removed, the remaining data points, shown in Fig. 4, fall within their (updated) quality control limits.

This feature is used in SONDS in a powerful data validation test to determine if a new data point is an acceptable member of an established distribution. But first one has to establish the reference distribution for each usage measurement.

### *9.1 Start-up tests*

The reference distribution is established in SONDS by accepting 20 days of peak data with no screening. At the end of this time, called the start-up period, three low-side, followed by two high-side, outlier tests are applied to the data. These are called censored outlier tests because the contribution of the data point being tested is removed from the estimates of the mean and the variance. This is done so that a contaminated data point will not be accepted because of its influence on the test conditions.

The following equations are given by Friedman[8] for estimating the mean and the variance of the equal "candidate busy hour" load distribution from the censored sample set of peak data points via the moments method. The censored mean is

$$\hat{\mu} = \bar{x}_{n-1} - \frac{\hat{\sigma}}{n-1}\{1.267n - E[y_{(n)}]\} \tag{29}$$

and the censored variance is

$$\hat{\sigma}^2 = \bar{s}^2_{n-1}\,(n-2)/(0.416(n-1) - \frac{n}{n-1}\{E[y^2_{(n)}] - 2.534E[y_{(n)}] + 1.605\}) \tag{30}$$

where

$\bar{x}_{n-1}$ is the sample mean and $\bar{s}^2_{n-1}$ is the sample variance of the remaining $n-1$ data points as each highest or lowest data point is removed

and

the reduced variate $y = (x - \mu)/\sigma$.

Table VIII—Low-side and high-side censored means and variances

| | Low-Side | | High-Side | |
|---|---|---|---|---|
| $n$ | $E\|y_1\|$ | $E\|y_1^2\|$ | $E\|y_n\|$ | $E\|y_n^2\|$ |
| 17 | — | — | 2.515 | 6.50 |
| 18 | 0.182 | 0.114 | 2.535 | 6.60 |
| 19 | 0.168 | 0.108 | 2.554 | 6.70 |
| 20 | 0.156 | 0.102 | 2.572 | 6.79 |

$E[y_{(n)}]$ is the expected value of the largest of the $n$ data points, each of which has been selected as the largest of six samples drawn from a normal distribution with zero mean and unity variance. $E[y_{(n)}^2]$ is the expected value of the largest square of $n$ such data points. $E[y_{(1)}]$ and $E[y_{(1)}^2]$ are the equivalent values for the lowest of $n$ data points. Values for both sets of parameters as used in SONDS are listed in Table VIII as given by Friedman.[8]

Using the censored mean and variance of eqs. (29) and (30) with $E[y_{(1)}]$ and $E[y_{(1)}^2]$, we compute $F(x_L)$ from eq. (8). We then compute the low threshold $L$ as:

$$L = 1 - [1 - F(x_L)]^n \leq 0.06. \tag{31}$$

If the value of $L$ is less than 0.06, the data point is rejected and the next lowest data point is tested. This value of the threshold will reject six percent of valid lowest-of-twenty data points.

Using the censored mean and variance of eqs. (29) and (30) with $E[y_{(n)}]$ and $E[y_{(n)}^2]$, we compute $F(x_H)$ from eq. (8). We then compute the high threshold $H$ as:

$$H = 1 - [F(x_H)]^n \leq 0.01. \tag{32}$$

If the value of $H$ is less than 0.01, the data point is rejected and the next highest one is tested. This value of the threshold will reject one percent of valid highest-of-twenty data points.

If either three low-side data points or two high-side data points fail the threshold tests, the complete set of start-up data is rejected and a new start-up interval is automatically initiated. Restarts almost completely disappeared after the normal-to-the-sixth replaced the Gumbel as the assumed distribution.

If both thresholds are passed, the particular measurement is considered operational and a different set of data validation tests are applied, as described in the next section.

### *9.2 Operational tests*

When the start-up data set for a particular measurement is accepted, its mean and variance are used for the initial operational data valida-

tion tests. The values of the mean and the variance used for subsequent operational tests are continually updated as new data are accepted. The failures of certain of these operational tests will cause an exception report to be issued. Other test failures will increment a counter, which may or may not result in an exception report. If a given day's data contain too many exception conditions on individual measurements, the whole set of daily data is rejected and an exception report is issued.

The following are the operational tests applied to EVE data.

#### 9.2.1 Upper physical bounds test

The received data should not be greater than 36 CCS times the number of components. (This test is also applied to non-EVE usage measurements.)

#### 9.2.2 Lower physical bounds test

The received value must be greater than zero. A zero value for the daily peak usage measurement of a group of components to be subjected to EVE analysis would indicate a trouble condition or a mistake in measurement assignment. (Zero is allowed for both non-EVE usage measurements and peg count measurements.)

#### 9.2.3 Once-a-month load exceedance test

This test compares each new data point with the once-a-month load computed from eq. (16) with the current values of mean and variance. A data point higher than that value will increment an up-down counter; a data point lower than that value will decrement the counter (but not below zero). An exception report is issued at the count of three. In this way traffic trends or bad data can be detected.

#### 9.2.4 Coefficient of variation test

If a register or piece of equipment is inoperative, the coefficient of variation of the moving average of the mean and the variance falls below the threshold value of 0.025, and an exception report is issued.

#### 9.2.5 High and low outlier tests

Before a new EVE data point is accepted, it must pass both a low and high outlier test. The current values of the mean and the variance are used to compute $F(x_i)$ from eq. (8).

The low threshold $L$ is computed as:

$$L = 1 - [1 - F(x_i)]^{20} \leq 0.06. \tag{33}$$

If the value of $L$ is less than 0.06, the new data point is rejected and it is not used to update the mean or the variance. This threshold value will reject only about one in 300 valid daily peak values.

The high threshold $H$ is computed as:

$$H = 1 - [F(x_i)]^{20} \leq 0.01. \tag{34}$$

If the value of $H$ is less than 0.01, the new data point is rejected and not used for updating. This threshold value will reject only about one in 2000 valid daily peak values.

By solving eqs. (33) and (34) numerically and using eqs. (13) and (14), the acceptable range for a new data point in terms of the current values of the mean and standard deviation can be found as:

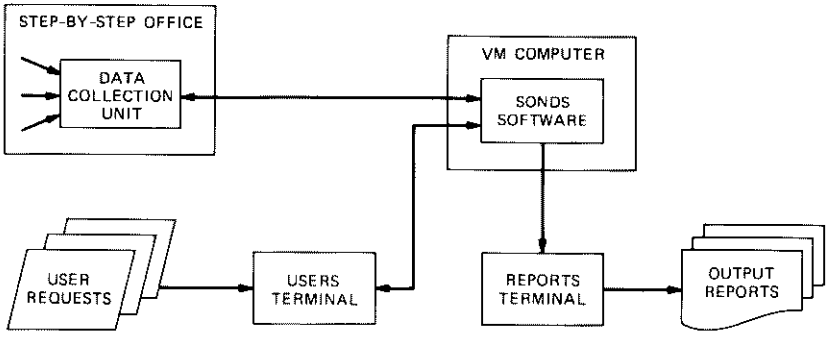
$$\bar{x} - 2.442\bar{s} \leq x_i \leq \bar{x} + 3.889\bar{s}. \tag{35}$$

## X. SYSTEM CONFIGURATION

A block diagram of SONDS is shown in Fig. 5. The most prominent feature is the single, central computer for the Bell System. The source of the traffic data is a data collection unit in each step-by-step office. Output reports are delivered to the user by shared access to a local reports terminal. The user controls the system by means of the user interface. All of these elements are connected together by the switched network.[12]

One of the advantages of the system is its use of the low-cost switched network instead of dedicated data links for the data collection and report distribution. The use of the switched network and computer resources is optimized by middle-of-the-night polling, followed by early morning reports transmission. The scheduled reports are available to the user at the start of business on the due date, and often contain results as recent as the previous day.

Consolidating the software in a single computer led to reduced development costs and timely revisions to correct bugs or to incorporate enhancements. The time-shared nature of the computer imple-




SONDS – SMALL OFFICE NETWORK DATA SYSTEM
VM – VIRTUAL MACHINE

Fig. 5—Block diagram of SONDS.

mentation made it natural to put the user manual on-line. This not only reduced the cost of documentation but made it more timely, user-oriented, and easily accessible. This type of documentation has reduced costs further by precluding the need for any formal training of the (many and casual) users.

Individual sections of the user manual can be obtained on-line. Also, the entire manual or any of its sections can be printed off-line and mailed to the user as the result of an on-line request. The date of the user manual is the date of its most recent change and the user is notified of this each time he or she logs onto the computer. The table of contents contains the date of the most recent change of each section. Each page contains the date of the most recent change incorporated thereon and the change itself is noted by marks in the margin.

To enhance the fully mechanized and less paper aspects of the system, the computer contains the capacity tables for the components to be engineered by use of the system's data. This allows the system to produce fully analyzed results, requiring no further manual manipulations by the user.

The user interface is the key to making the system easy to operate and completely under user control. The most important function of this interface is entering the office information that SONDS needs to produce the output reports. This information includes the component names that SONDS presents in the output reports. It also includes the number of the components that SONDS needs to access the capacity tables for those components. Further, SONDS must know about the units of measurement to convert the received data to the proper units: CCS for usage, units for peg counts, and percent for dial tone delays.

The interface contains a series of prompts that help the user enter the information. Out-of-bound values (probably typographical errors) are detected immediately and the user can correct them at this time or add them later if the proper value is unkown. Whenever the user does not know what to do next, typing a question mark will produce a list of acceptable inputs.

Another application of the user interface is to request previously generated reports that were either misplaced or not received because of a reports terminal problem. Typically, the most recent three or four scheduled reports are kept in this on-line archive; the (daily) exception reports are kept for two weeks. Also, the user can demand a current version of any of the scheduled reports. He will receive a partial version of the next scheduled report. This request, however, will not interfere with the regular generation of the scheduled report.

Perhaps the most powerful use of the interface is to command the system to go into hourly data collection. Without interfering with the regular nightly polling and report generation, the computer will initiate

a data poll each hour between 6 a.m. and midnight, or as specified by the user. Thus, time-consistent hourly data can be generated. This feature is needed for limited use to verify data for a newly equipped office, to troubleshoot a data problem, or to conduct special studies.

### *10.1 Data collection unit*

To obtain the desired data from the step-by-step offices, SONDS has specified a set of features, some of which are optional, for a Data Collection Unit (DCU). The three vendors and the four models of DCUs used on SONDS are Western Electric (PDT2A) (13), Tele-Sciences (TE450), and Alston (Model 565 and 566). Each microcomputer-controlled data terminal is connected to a data set that has an automatic answer capability. This is used for dial-up access by the SONDS computer to obtain the measurement data. This dial-up access can also be used by maintenance people for remotely running diagnostics, as specified by the vendor.

The DCU collects both peg count and usage data, the latter for either one-second or ten-second scan rates.

The output register set for each measurement contains three registers. One is the active register, which accumulates the counts during the measurement hour. At the end of each hour the active registers are reset automatically and the contents are made available to the previous-hour and long-term registers.

For each measurement, the new hourly value from the active register overwrites the value contained in the previous-hour register where it is retained for the next hour. The new hourly value also goes to the long-term register, which has been specified to be either a peak or an accumulative register. If the long-term register is peak, the new hourly value is compared to that stored in the peak long-term register and the higher of the two is retained in the peak long-term register. If the long-term register is accumulative, the new hourly value is added to its contents.

The long-term registers are reset after reading by the SONDS computer during the nightly poll. The values obtained from the long-term registers are either the highest hourly value experienced since the last resetting or the accumulations of the hourly values since the last resetting.

The DCU reads out the register contents on command from the SONDS computer. During the nightly poll, the computer requests the contents of both the previous-hour and long-term registers and then the computer resets the long-term registers. For the collection of hourly data during the daytime, the computer only requests the contents of the previous-hour registers and disconnects without any resetting. The SONDS computer will accept up to 200 register readings in response to a given request.

Consideration of mid-train engineering and the possible use of SONDS data for the pre-installation engineering of an electronic replacement office has led to the definition of two optional features. One is called the peak of the sum. What is desired here is the (24-hour) peak value of a measurement that itself is the sum of other measurements. Since the individual measurements might peak at different hours, the sum of the (24-hour) peaks will, in general, not be equal to the (24-hour) peak of the sum of the hourly values. Therefore, the hourly summing and (24-hour) peaking must be done in the DCU. The output is available in one of the regular peak long-term registers.

The other optional feature is called the peak of the difference. The desired result is the peak value of a measurement that itself is the difference between two sets of other measurements. Again, the peak of the difference is, in general, not equal to the difference of the peaks. The output is available in one of the regular peak long-term registers.

Besides these measurement features, the DCU satisifes several other SONDS requirements. Each response has a specific format that the SONDS computer expects. It includes the DCU time of day, which is resettable by the SONDS computer, and a status word to indicate several conditions within the terminal that might make the data suspect and that would aid in the trouble location of a DCU problem. The response also contains a checksum value of the response itself. The SONDS computer calculates the checksum of the received data and compares it to the received checksum. In this way the SONDS computer can be assured that it has an accurate copy of the register readings as they exist in the DCU.

### *10.2 Data calculations*

To generate the several types of output reports, SONDS has four different types of data analysis. In entering the office information, the user selects one type of analysis for each measurement.

#### *10.2.1 EVE analysis*

EVE analysis is used for the usage measurements in the main part of the SONDS monthly report and for selected measurements in the miscellaneous section of that report.

As seen in eq. (16), the once-a-month load is calculated from the mean and the standard deviation (the square root of the variance) of the daily peaks. Therefore, it is not necessary to retain all the received values; only an estimate of the mean and the variance is required. This is accomplished in SONDS as a continual updating of these values, once they are derived at the end of the start-up period.

The equations for this updating are as follows:

$$\bar{x}_i = \mathrm{p}x_i + (1 - \mathrm{p})\bar{x}_{i-1} \tag{36}$$

$$v_i = \mathrm{p}(x_i - \bar{x}_i)^2 + (1 - \mathrm{p})v_{i-1}, \tag{37}$$

where

$x_i$ is received value for the $i$th day,

$\bar{x}_i$ is the mean at the $i$th day,

$\bar{x}_{i-1}$ is the mean from the previous day,

$v_i$ is the variance at the $i$th day,

$v_{i-1}$ is the variance from the previous day,

and

p is a constant.

This updating method reduces the size of the database. Only two values need to be stored for each measurement; no raw data are retained. It gives an updated value for the mean and the variance, which is the exponential weighting of all the previously received data. The value of p was selected as 0.095 to make the average age of the weighted data the same as if the latest month of data itself had been used. One effect of this calculation is that 87 percent of the weight of the current estimate is derived from data received during the most recent month; 13 percent of the weight is due to data received in prior months. This approach, besides minimizing data storage requirements, is also compatible with current TCBH practice of reporting monthly results.

Another advantage of this updating technique described in eqs. (36) and (37) is that it makes the calculation robust for (occasional) losses of data. For any day that data are not obtained, the estimate is not updated, that is, the estimate is stale by that amount. The exponential weighting of previous data reduces this staleness, and the effect of a lost data point is soon eliminated.

#### *10.2.2 PA analysis*

Periodic Average (PA) analysis is used for dial tone delays and for those usage measurements in the Miscellaneous section of the SONDS monthly report that do not lend themselves to the EVE analysis.

The equation to update the estimate of the mean is:

$$\bar{x}_j = x_j/j + \bar{x}_{j-1}(j - 1)/j, \tag{38}$$

where

$j$ is the $j$th day during the current period.

This analysis type is called Periodic Average because it calculates the true, straight-line average during the period of the monthly report; the value is reset when the monthly report is generated. PA also retains the single largest value received during the calculation period.

#### *10.2.3 SMA analysis*

The third type of analysis is called the Service Month-to-date Accumulation (SMA). This analysis type is used for the Automatic Number Identification (ANI) measurements needed for the Network Switching Performance Measurement Plan (NSPMP). It basically gives the month-to-date summation of the daily values of the (24-hour) accumulative long-term registers. Similar to the PA analysis, the SMA analysis also produces the periodic average and the single highest received value.

#### *10.2.4 TOPC analysis*

The fourth type of analysis is called Total Office Originating Peg Count (TOPC). It is specified during the entry of the office information to be applied to originating PC measurements. These measurements are stored for the generation of the (calendar) month TOPC report in which certain weekday and weekend ratios and averages are calculated. Also certain adjustments are made for the loss of individual days of data.

### *10.3 Output reports*

As part of its goal to be a fully mechanized system, SONDS automatically calculates, generates, and transmits several output reports. The monthly report, the (monthly) Total Office Originating Peg Count Report and the weekly report are regularly scheduled. Exception reports are issued daily as needed. Only the hourly data report depends on user initiation (to request hourly polling). It is generated and transmitted daily to give the previous day's polling results.

#### *10.3.1 Monthly report*

The heading of the monthly report gives the office name and the date of the report, which is typically the 22nd of the month. The report heading also gives the date that the office information file was last updated; this is a message to the user because his/her monthly updates of office component counts can affect the numbers presented on the report.

Another part of the header information is the number of days of valid EVE data. Although SONDS collects data seven days a week, EVE data are only used for five days. Typically, these are the weekdays, but the user can include Saturday, Sunday, or both in the five days if weekends are high traffic days in a particular office. There are only 20 to 23 EVE days in a month and the number of valid days indicates the weight of the EVE results due to data collected during the current month. Fifty percent of the weight occurs in the current month if seven days of valid EVE data are obtained. Data for measurements

with fewer than seven days in the current month are shown on the monthly report, but they are flagged with a ?. Incidentally, if a measurement is in start-up, it is flagged with an asterisk.

The first part of the Originating Results section of the monthly report contains information on the Line Finder Groups (LFGs). This includes the items discussed in prior sections, such as the current load as percent capacity, main station assignment guide, and maximum value of Dial Tone Delay (DTD), all on a per-LFG basis. Also there are total office results in terms of the Dial Line Index (DLI) and Load Balance Index (LBI).

The Originating Results section of the monthly report also contains an outgoing trunk group section. Certain columns containing information not previously discussed were specifically designed for the trunking people. One such column gives the number of circuits required to carry the measured load at the once-a-month blocking objective; this quantity is equivalent to one of the main outputs of the Trunk Servicing System (TSS). The second column can be used as the TSS (TCBH) study period load; it is derived from the proper TCBH trunk capacity table, depending on the trunk accessing arrangement, for the number-of-trunks required specified in the previous column. SONDS retains these study period values for the most recent 12 months and each month prints the highest value in the third column. This is the base period load, which can be the input to the Trunk Forecasting System (TFS). The fourth column prints the month when this highest load occurred.

The Terminating Results section of the monthly report contains similar information for selectors, connectors, and incoming trunks.

The Miscellaneous Results section contains any other measurements the user has decided to add to the system. The user can select peg count or usage measurements, peak or (24-hour) accumulating registers, and has a choice of EVE, PA, or SMA analysis types. One of the main uses of this section has been to obtain the ANI counts for NSPMP reporting. Another use has been to obtain the results of the summing and differencing registers to obtain data for pre-installation engineering of an electronic replacement office. Of course, the user could also choose to obtain Last Trunk Usage (LTU) or Subscriber Line Usage (SLU) measurements.

#### *10.3.2 Weekly report*

The Weekly report is scheduled for Monday mornings; it gives daily results for seven days up to the previous Friday. The main part of the report gives the "always busy" data obtained by the SONDS computer during the nightly polling by reading the previous-hour registers of the data collection unit. Integer multiples of 36 CCS in these data are

indicative of busied-out or stuck components. The report also contains the daily peak values of dial tone delays so that the user can determine if high values of dial-tone delays were caused by maintenance conditions indicated by the always-busy data. The report contains the daily values of total office originating peg counts to make these data available to the Division of Revenues people before the end of the (calendar) month. The report contains the month-to-date results of the DTD (TCBH) and NSPMP measurements so the user can monitor the reportable service results during the month.

#### *10.3.3 Exception reports*

Exception Reports (ER) are issued daily as needed when abnormal conditions are experienced during the night's poll. These conditions might be a failure to connect (after four attempts), error bits set in the status word, and failed data validation tests. These ERs are short messages giving the ER code, the descriptive name, and the received data, if any. There are some 52 different types of ERs and each has a one- to three-page description in the SONDS User Manual to help the user locate the problem. Also, the user can access the SONDS computer to obtain a list of the ERs for the past two weeks.

#### *10.3.4 Total Office Originating Peg Count report*

The Total Office Originating Peg Count report has a calendar-like format showing the daily values with weekday values separated from weekend values. Weekday-to-weekend ratios are calculated as well as values of average business days and calendar days. The computer contains algorithms to adjust for isolated days of missing data and it computes adjusted averages.

#### *10.3.5 Hourly data report*

Hourly data reports are generated only when hourly polling has been requested by a user. The reports are received in the morning and contain up to 16 hours of polling results of the prior day. This produces time-consistent hourly data, including dial tone delays, to track the flow of traffic into, out of, and across the office. For each hour, the report also contains a traffic balance calculation to determine if permanent signals are inflating the measured value of line finder usage.

#### *10.3.6 Retention periods*

As mentioned previously, SONDS retains on-line for user access two weeks of exception reports and the last three or four reports of other types. The user can also receive on-line a demand version of any

of the reports, which can supply data as recent as that of the previous day.

## XI. COMPUTER AND SOFTWARE DESCRIPTION

The SONDS software consists of approximately 260 programs residing on an AT&T Information Systems computer located at the Corporate Computer Center in Piscataway, N.J. SONDS shares with many other corporate users an Amdahl 470/V8 processor under the Virtual Machine/Conversational Monitor System (VM/CMS) operating system.

### *11.1 VM/CMS computer facility*

VM/CMS manages the resources of a real-time computing system in such a way as to make them available to many users at the same time. Through VM/CMS, each SONDS user has a virtual machine composed of a virtual system console (a terminal), virtual CPU, virtual storage (disk), and virtual channels and Input/Output (I/O) devices.

The VM/CMS Auto-dial Facility allows application programs to initiate a direct distance dialing connection to a remote terminal properly equipped with an auto-answer feature. SONDS uses the Auto-dial Facility in two ways: It extracts data at a 300-baud rate from the data collection unit in each SONDS office by nightly polling, and it prints output reports on users' printing terminals at the rate of either 300 or 1200 baud. This report distribution process is faster and less expensive than mailing reports directly and can make use of terminal equipment outside normal working hours.

### *11.2 SONDS software*

Each of the major modules of the SONDS software is described in the following sections.

#### *11.2.1 User interface*

The user enters and maintains office information on the user interface. It can also be used to change the polling mode of the office: start (nightly) polling, stop polling, and schedule hourly polling.

The user accesses the Virtual Machine (VM) system over the switched network via any ASCII-compatible terminal at either 300 or 1200 baud. After entering a user identification (userid) and a password, the user is connected to the SONDS user interface. The system gives any short news items that the user should know, and the date of the last update to the User Manual. Whenever the user doesn't know what to do next in the interactive dialogue with the system, he or she can enter a question mark. The system will respond with a list of appropriate entries.

### *11.2.2 Users' database*

The office information file entered by the user via the user interface is stored in the users' database. Also the traffic information obtained from the data collection unit is stored here. All the data are stored on an office-by-office basis. Some administrative reports, however, do combine information on how SONDS is working for all the offices on a userid and all the offices for a company.

### *11.2.3 System files*

The system files primarily contain office status information to determine which activities should be started and when it is appropriate to start them. They are updated as the scheduled activities are performed. They thereby always contain the current information and can be queried by the SONDS Support Team to provide a current report of system status.

### *11.2.4 Master scheduler*

The master scheduler controls four main activities: nightly polling, hourly polling, report generation, and report transmission.

Nightly polling occurs seven days a week, including holidays, during low traffic periods, usually between midnight and 6:00 a.m. local office time. The polling load is distributed over as many hours as necessary, taking advantage of the different time zones across the country.

Hourly polling may be requested by the user through the user interface for as many as six days of polling. The user specifies the earliest hour and the latest hour for polling, between 6 a.m. and midnight. Hourly polling can be started on the next clock hour. Hourly polling increases SONDS costs and charges, which in turn have prevented abuse of this feature.

The master scheduler activates the report generator and report transmitter processors according to the schedule for each of the five output report types, described in Section 10.3. The master scheduler also establishes an efficient order of nightly polling, hourly polling, report generation, and report transmission by considering the number of Automatic Calling Units (ACUs), the number of offices per time zone, estimated time to perform each function, etc.

The master scheduler runs each hour, usually starting at one minute past the hour and running for 15 seconds to 2 minutes. In addition, one of the major objectives of the master scheduler is to invoke "batch jobs" as needed for the poller, the report generator, and the report transmitter. Once initiated, these batch jobs will run on their own without master scheduler intervention.

#### 11.2.5 Poller

The principal poller is activated during the night to place a call over the switched network to the data collection unit using the ACUs. The data as well as the status information of the DCU are sent to the users' database and the system files are updated. The action of the hourly poller is similar except that it is activated during daytime hours.

#### 11.2.6 Report generator and transmitter

When the report generator is activated by the master scheduler, it obtains the traffic data from the users' database, applies the necessary algorithms to calculate the numbers that appear on the report, and stores the report data. The report transmitter then places a call over the switched network to the user-specified reports terminal via the ACUs, retrieves and formats the report data, and transmits the report to the users' terminals over the switched network.

#### 11.2.7 Map Management System

The PDT2A data collection unit contains a software map to cross connect the input leads to the output registers.[13] This software map is created by the Interactive Map Assembly Program (IMAP), which exists on a time-shared DEC 10 computer at a Western Electric facility. Using the VM batch facility, the Map Management System will accept a user-generated request to place a call over the switched network via the ACUs to access IMAP and obtain the most recent edition of the software map, which is then stored in the users' database for down-loading to the PDT2A via the poller.

#### 11.2.8 Administrative System

The Administrative System is a group of programs that monitor the activities of all the other modules, recording their successes and failures. The system generates several types of detailed logs and summary reports, which are available to the SONDS Support Team to determine the health of the SONDS and VM systems and to identify and resolve problems in a timely manner. The user has access to a (pertinent) subset of these administrative reports via the user interface.

#### 11.2.9 Billing system

Billing for VM/CMS is done once a month and a separate statement is produced for each userid. Each statement includes a detailed listing of the charges for each off-line activity the userid had during the month, and a breakdown of the usage on the userid by accounting information, which identifies the initiator of the on-line activity and

the type of off-line activity. To increase the user's cost consciousness, the user interface prints an approximate charge, upon logoff, for that user session.

## XII. ACKNOWLEDGMENTS

The authors are indebted to K. A. Friedman, who was the main analyst throughout this project. He proposed and analyzed the normal-to-the-sixth distribution, the outlier detection tests, and the dial tone delay conversions. He collected and analyzed the vast quantities of field data needed to derive extreme value service criteria. He derived the new engineering tables used in SONDS for all the major components of the step-by-step office. R. A. Farel's work on the use of extreme values for line assignment and load balance, the results of which are published for the first time herein, was a significant contribution towards making SONDS a member of the TNDS family. We are particularly appreciative of the contributions, suggestions, and critiques of J. G. Kappel throughout the project. The experimental version of SONDS, which was used to develop many of the concepts reported herein, was possible due to the talent of the software team led by G. B. Foley and S. Chen and to the dedicated field support of R. R. Plum. The one thousand SONDS offices now operating on the VM computer would not have been possible without the dedicated effort of J. Schwartz and his software development team at AT&T Information Systems. Their generous review of the pertinent parts of this paper is gratefully acknowledged; the significant contribution of K. L. Dziedzic is specifically acknowledged.

## REFERENCES

1. S. A. Karlsson, "The Dimensioning of Telephone Traffic Routes for Measured Integrated Peak Traffic," Fifth Int. Teletraffic Cong., New York, June 14–20, 1967, pp. 7–17.
2. S. B. Gershwin, R. V. Laue, and E. Wolman, "Peak-Load Traffic Administration of a Rural Line Concentrator," Seventh Int. Teletraffic Cong., Stockholm, Sweden, June 13–20, 1973, pp. 146/1–7.
3. S. B. Gershwin, R. V. Laue, and E. Wolman, "Peak-Load Traffic Administration of a Rural Multiplexer With Concentration," B.S.T.J., *53*, No. 2 (February 1974), pp. 261–81.
4. R. V. Laue, "Extreme-Value Engineering Designs for Peak Traffic," Bell Lab. Rec., *56*, No. 2 (February 1978), pp. 38–42.
5. E. J. Gumbel, *Statistics of Extremes*, New York: Columbia University Press, 1958.
6. D. H. Barnes, "Extreme Value Engineering of Small Switching Offices," Eighth Int. Teletraffic Congr., Melbourne, Australia, November 10–17, 1976, pp. 242/1–7.
7. D. H. Barnes, "Observations of Extreme Value Statistics in Small Switching Offices," Ninth Int. Teletraffic Cong., Torremolinos, Spain, October 16–24, 1979, 9 pages.
8. K. A. Friedman, "Extreme Value Analysis Techniques," Ninth Int. Teletraffic Congr., Torremolinos, Spain, October 16–24, 1979, 7 pages.
9. L .H. C. Tippett, "On the Extreme Individuals and the Range of Samples Taken from a Normal Population," Biometrika, *17* (1925), pp. 364–87.
10. R. A. Farel, unpublished report.
11. R. A. Farel, unpublished report.

12. G. B. Foley, J. J. O'Connor, and J. Schwartz, "SONDS Says 'Sound Off' to Small Offices," Bell Lab. Rec., *58*, No. 10 (November 1980), p. 333–8.
13. J. Arthur Grandle, Jr., and Richard R. Plum, "A Versatile Data-Gathering Tool," Bell Lab. Rec., *57*, No. 8 (September 1979), pp. 227–31.

## AUTHORS

**Douglas H. Barnes,** B.S. (Industrial Engineering), 1938, Lehigh University. Mr. Barnes joined Ohio Bell Telephone Company in 1941 as an Assistant District Traffic Superintendent, where he worked on a variety of traffic assignments including operator force management, traffic staff, service observing, local and toll office traffic engineering, trunk engineering, and division of revenue studies. He came to Bell Laboratories in 1951 to work on requirements for traffic measurement needs and systems. He was appointed Head of the Traffic Systems Engineering and Analysis Department in 1961. He was responsible for systems engineering of all traffic data collection and processing systems. Some of these are: Traffic Usage Recorders, Traffic Data Recording System, Engineering and Administration Data System, Mechanized Service Observing, Force Administration Data System, Subscriber Line Usage System, and Small Office Network Data System. He was also responsible for various improvements in traffic methods, such as the adaptation of Extreme Value Engineering. He retired from Bell Laboratories in 1979 and has since been a Teletraffic Consultant.

**John J. O'Connor,** B.S.E.E., 1951, M.S.E.E., 1952, Columbia University; Ford Instrument Co., 1952–1954; Electronics Research Laboratories, C.U., 1954–1958; Hathaway Instrument Co., 1959–1961; Martin Co., 1961–1964; Bellcomm, 1965–1972; Bell Laboratories, 1972—. At Ford Instrument Co. Mr. O'Connor worked on terrain synthesis for an automatic radar map-matching guidance system. At Electronics Research Laboratories he studied and worked on simulating low-frequency (0 to 0.1 Hz) noise of radar glint in the Nike missile system. He worked on the man-rating of the Titan III launch vehicle at the Martin Co., and on the spacecraft attitude control propellent budget and structural oscillations (POGO) of the Apollo-Saturn launch vehicle for Bellcomm. In 1972 Mr. O'Connor joined Bell Laboratories, where until 1973 he worked on the maintenance aspects of electromechanical switching machines. From 1973 to 1976 he was involved with various traffic-related problems of toll switching machines. From 1976 until 1982 he worked on SONDS, and from 1982 until the present has been studying the project management aspects of the Dynamic Nonhierarchical Routing network.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Performance Measurement/Trouble Location

By D. R. ANDERSON* and M. J. EVANS*

(Manuscript received November 24, 1982)

The Total Network Data System (TNDS) includes a subsystem that measures TNDS performance and assists network administrators and those responsible for network traffic data collection in finding TNDS troubles. This interactive subsystem provides the telephone company managers with a versatile tool for a performance analysis of the many systems and organizations associated with TNDS. The troubleshooting features usually provide sufficient detail from which to specify corrective action. The performance information can be aggregated across all organizational levels from the single traffic unit to the entire Bell System.

## I. INTRODUCTION AND SUMMARY

Good service to the customer has always been a hallmark of the Bell System. This service is ensured by a number of measurement plans used by Bell System managers to monitor the quality of service to the customer. Operations such as those associated with the Total Network Data System (TNDS) do not all have an immediate effect on the customer, but if neglected for a long period of time, could result in costly engineering mistakes. Thus, a TNDS performance measurment plan is needed so that the efficiency and integrity of TNDS operations are maintained at a level sufficient to support continued good service

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

to the customer at reasonable cost. A TNDS Performance Measurement Plan (TPMP) was introduced in 1980. This measurement plan is highly automated, using a computerized support system called the Centralized System for Analysis and Reporting (CSAR).

In addition to highlighting performance, measurement plans often provide supplementary data useful to those responsible for assuring good performance. This function is especially important in TNDS operations because of the wide diversity in systems, organizations, geography, and timing associated with TNDS. Thus, assisting troubleshooting is an important feature of CSAR. It often allows the analyzer to identify the specific source of a data problem while still at the terminal.

## II. TNDS PERFORMANCE MEASUREMENT PLAN (TPMP)

### *2.1 Need and objectives*

In 1976, as the TNDS systems were being widely deployed in the Operating Telephone Companies (OTCs), TNDS project managers at AT&T and systems engineers at Bell Labs undertook a field study of TNDS Operations in South Central Bell. This company offered an ideal environment for observing TNDS operations. Its applications included rural areas and large metropolitan areas across five states.

With the help of South Central employees, personnel at Bell Laboratories and AT&T were able to obtain valuable insight into TNDS performance and typical operations. Over several months the AT&T/Bell Labs team observed just how TNDS traffic data reports were obtained, and how accompanying TNDS administrative information was analyzed and used in a variety of operating company offices. It was observed that managers often had difficulty obtaining an overview of performance from the administrative reports generated by the various systems of the TNDS. Also, the geographical diversity of these systems, the time interval over which traffic data progress through the TNDS, and the volume of the reports from the systems all made it difficult and time-consuming to detect and pinpoint data problems. These difficulties were the result of the increasing size and complexity of TNDS and its wide deployment.

This study clearly pointed out a need for a comprehensive, automated performance measurement plan that could help to localize traffic data problems. The reports associated with the measurement plan should provide a concise management overview with enough detail about the type and location of problems to allow managers to apply resources to bring about solutions. The system generating the measurements should also make available to the TNDS administrators selected, detailed information about data as the data pass through the

various systems in the TNDS. The measurement system should allow enough investigation, in increasing detail, to identify the location and time of problems. Such investigation should also identify enough about their cause to determine either the corrective measures or the testing and analysis that is required. The traffic data problems should be identified as quickly as possible so that critical traffic data studies are not jeopardized.

All of these requirements are best met by a system that obtains performance data directly from the other TNDS systems and provides a variety of reports on an on-line, interactive basis.

In considering measurement plans for TNDS, we noted a significant trend in other Bell System measurement plans. A measurement task force has recommended that measurement plans should avoid averaging results since they are aggregated together for several measured entities. This is important so that poor performance in one or two entities is not masked by good performance in a number of other entities. This principle is extremely valid in TNDS operations measurements where consistently poor performance in obtaining traffic data for an office could deny engineers the data needed to plan for its growth.

The method being adopted in Bell System measurement plans to preserve isolated indications of poor performance is to associate the results obtained for each measured entity into one of four bands: H, O, L, or U. The bands are designated as Higher than objective, Objective, Lower than objective, and Unsatisfactory. Then, aggregated results give the number of entity measurements in each of the bands. Thus, a poor performer is never lost in the crowd. In TPMP, there is usually no extra expenditure for flawless performance so there is no need for the Higher than objective category. Thus, TPMP only includes bands O, L, and U.

The measures important in monitoring TNDS performance are related to data completeness, data validity, and the accuracy of the record bases that associate the data with the telecommunications equipment. The TNDS Central Office Equipment Reports (COER) systems,[1,2] Trunk Servicing System (TSS),[3] and Load Balance System (LBS)[1] all present traffic data to end users (engineers or administrators) and, thus, can supply comprehensive information about TNDS performance. They all provide performance data to CSAR. The Trunk Forecasting System (TFS),[3] also an end-user system, does not provide data to CSAR at this time, but its inputs are directly dependent on the (CSAR-measured) TSS. The Traffic Data Administration System (TDAS),[1] which is an intermediate system in TNDS data flow, also makes available performance data to CSAR. This is done because TDAS is the final TNDS system for certain special study information

and equipment types, and because analysis of TDAS results can allow early detection and isolation of data-gathering problems.

The above systems generate performance data pertinent to data completeness, data validity, and record base accuracy. In addition, the Individual Circuit Analysis System (ICAN)[1] was designed primarily to help administer the record base for usage data acquired on individual circuits through the Individual Circuit Usage Recording (ICUR) version of the Engineering and Administrative Data Acquisition System (EADAS).[4] As such, ICAN is an important source of CSAR record base performance data for those switching systems served by EADAS/ICUR.

### *2.2 Performance measures*

Each system that furnishes CSAR with performance data has its own validity checks, record bases, and schedules. Each has its own way of measuring data completeness, data accuracy, and record base accuracy. Hence, the first level of the hierarchy for CSAR reporting is by the TNDS. Table I shows the TNDS performance measures generated by CSAR and gives a brief explanation for each. The results are reported as "problems," i.e., as percentage of missing data, validity failures, or record base errors. Two thresholds are assigned to each measure to distinguish between the three bands, O, L, and U. The threshold settings are determined differently for each measure because each is associated with a different part of the TNDS process, different operations, or different equipment. There is a guiding rationale, however, applied to all. The Objective band starts at zero and includes performance levels where minor flaws occur but are being attended to. The first threshold marks the transition to Band L, where the equipment errors and record base errors become significant and systematic. The upper threshold, which marks the beginning of band U, is set to indicate equipment problems that persist longer than is normally necessary for their remedy or to indicate the accumulation of record base errors.

### *2.3 Reporting*

TPMP is designed to assist operating company TNDS managers at all organizational levels. Their access to reports is through the on-line, interactive computer system, CSAR. As mentioned earlier, the first level in the CSAR reporting hierarchy is the TNDS, from which CSAR has obtained the performance data. The second level is the organization for which the manager is responsible. For a given request, results from all reporting units, switching offices, trunk services, etc., in that organization are summarized or delineated according to the request. The dialogue with the computer enables the manager to get a

Table I—TPMP/CSAR performance measures

| Measures | Performance Data Source | Description |
|---|---|---|
| DCU—Intervals missing | TDAS | The half-hour DCU* intervals that are required to satisfy all of the requests for obtaining data through TDAS, but where data are not recorded. |
| DCU—Intervals with any verification-device failures | TDAS | The half-hour DCU intervals where data are obtained but where one, or more, of the devices used to verify data accuracy have an imperfect score. |
| Detector test failures (only for TUs obtaining usage data through a TUR-traffic usage recorder) | TDAS | The number of discrepancies encountered when comparing a CU record base containing the number of circuits in each measured group with the usage inputs during a test scan when all circuits are simulated as busy. |
| Invalid detector test scans | TDAS | Discrepancies detected by the verification devices when the detector test is run. |
| ICAN/CU count mismatch | ICAN | The number of discrepancies encountered when ICAN compares the number of individual circuits in each circuit group in its record base with that in the CU record base. |
| ICAN/CU other error | ICAN | Any other discrepancies found when performing the record base comparison between ICAN and CU (e.g., circuit group type). |
| Suspect SCHV assignment | ICAN | The number of cases when ICAN reports an individual circuit assignment to a circuit group to be "suspicious" (these should be resolved by data administrators). |
| UA/UE SCHV usage | ICAN | The number of individual circuits that have measured usage but are designated in the EADAS/ICUR record base as being unassigned (UA) or unequipped (UE). |
| CGMT with insufficient data | TSS | The number of circuit group measurement (CGMT) types for which there was insufficient valid traffic data to use as a basis for trunk servicing or forecasting. |
| CGMT other invalid | TSS | Trunk group validity failures in CGMTs with sufficient valid data. |
| CGMT unassociated | TSS | The number of CGMTs for which traffic data were received but cannot be associated with a group in the record base. |
| LU invalid for LBI | LBS | The number of load units [(LUs) concentrators to which circuits are assigned] for which no valid data were provided for the load balance index (LBI) calculation. |
| LU measurements invalid | LBS | The number of load units where some data were obtained in the loading division but for which the LU had insufficient valid data (excludes cases where entire loading division missed data). |
| 5XB COER control file with error | 5XB COER | Indicates that data are not processed by 5XB COER because 5XB COER has detected an unresolved error in its record base. |
| 5XB COER end hour missing | 5XB COER | The number of hourly intervals for which 5XB COER expected data but did not receive it. |
| 5XB COER validation failures | 5XB COER | The number of failures detected when 5XB COER checked data validity. |
| SPCS COER missing data | SPCS COER | The number of hourly intervals for which 1 SPCS COER expected data but insufficient data were received. |
| SPCS COER sanity failures | SPCS COER | The number of data sanity check failures (gross data inconsistencies—tight thresholds apply). |
| SPCS COER cross-measure type test failures | SPCS COER | The number of cross-measure type tests that have failed (inconsistencies between data items—looser thresholds apply because peculiar switching environments may result in false alarms). |

* The physical embodiment of DCUs (data collection units) differ from one type of switching machine to another. However, a DCU may generally be thought of as an aggregate of traffic data items that are scheduled and collected as a set.

breakdown by the next lower organizational level. More will be said later about these reports and the manager's use of them. In addition, CSAR furnishes detailed information for trouble analysis.

An overall summary of results for each TNDS operating company is made available to AT&T every month. The percentage of reporting-unit measures (see Table I) in the bands L and U are included in a Bell System measurement summary called "Network Results," which is published monthly for Bell System use.

## III. THE CENTRALIZED SYSTEM FOR ANALYSIS AND REPORTING

CSAR is an on-line, interactive computer system that provides dial-up access six days a week to centralized databases holding over 250M bytes of information. The system combines the cost effectiveness of batch processing large volumes of data with the speed and convenience of on-line database access. CSAR can be accessed using almost any 110-, 300-, or 1200-baud asynchronous terminal. The daily operation of CSAR combines the resources of OTC computer facilities in 24 states, a Bell System data transmission network, a centralized computer system at AT&T, and the Direct Distance Dialing (DDD) network to support the TPMP and generate extensive information to assist in the operation and administration of TNDS. The reporting capabilities offer timely assessments of the performance of the traffic data collection and processing tasks, as seen by each system in the TNDS. Currently, eighteen OTCs use CSAR. Each TNDS installation within these companies becomes a remote source of TNDS performance data analyzed by CSAR and maintained in the central databases. Authorized users in each of these companies have dial-up access to the interactive component of CSAR, allowing flexible retrieval of information pertinent to their own company and certain Bell System results.

### *3.1 System configuration*

The major component in the CSAR system configuration resides at the centralized computer site and receives the majority of its data from the distributed OTC TNDS processing sites via the Bell System telecommunications software system for computer-to-computer data exchange with OTCs (T-TRAN) data transmission network. The CSAR users gain access to the central computer system from existing remote terminals using the standard DDD telephone network. Thus, the system includes five major components that effectively combine existing Bell System facilities:

1. Distributed OTC computer sites
2. A T-TRAN data network
3. A Central AT&T computer site

4. A DDD telephone network
5. Existing remote terminals.

Figure 1 shows the CSAR computer system configuration. The CSAR implementation utilizes and coordinates a number of different hardware and operating system environments. Brief descriptions of the first three system components listed above illustrate the overall CSAR operating environment.

#### *3.1.1 OTC computer facilities*

The CSAR measurement process begins at the OTC's computer facility when the batch TNDS subsystem modules are run. It uses the Standard Operating Environment of IBM 370 series mainframe or a compatible computer running the Multiple Virtual Storage (MVS) operating systems. The performance data are gathered by software embedded in the individual TNDS subsystem modules. Thus, the individual TNDS subsystems accumulate the majority of the raw data required for CSAR performance measurement and analysis. A standalone CSAR module executes under this environment at the remote computer facilities to merge the data files from the separate TNDS components and prepare a single standard interface for data transmission.

#### *3.1.2 T-TRAN data network*

Each OTC uses the T-TRAN network as a data link for the transmission of its TNDS performance data to the central computer site at an AT&T Corporate Computer Center in New Jersey. Weekly CSAR transmissions are an integral part of the TNDS operations in the OTCs. Backup data are retained in each company on disk or tape.

The receipt of a transmission at the AT&T T-TRAN site (an IBM 370 system running MVS with the time-sharing option) automatically invokes a CSAR program that initiates the CSAR batch processing sequence. This first step identifies the incoming CSAR data files and sends all necessary information to a second computer system that is the actual host for the primary CSAR batch and on-line interactive processes. The program under the Multiple Virtual Storage/Time-Sharing Option (MVS/TSO) communicates to the CSAR host machine via a local Inter-Processor Communication (IPC) network.

#### *3.1.3 Central host computer*

The heart of the CSAR system also resides on one of the many computers that comprise the AT&T Corporate Computer Center. The CSAR software is developed and centrally executed on an Amdahl 470 series mainframe running under the Virtual Machine (VM) operating system using the Conversational Monitor System (CMS). This VM/

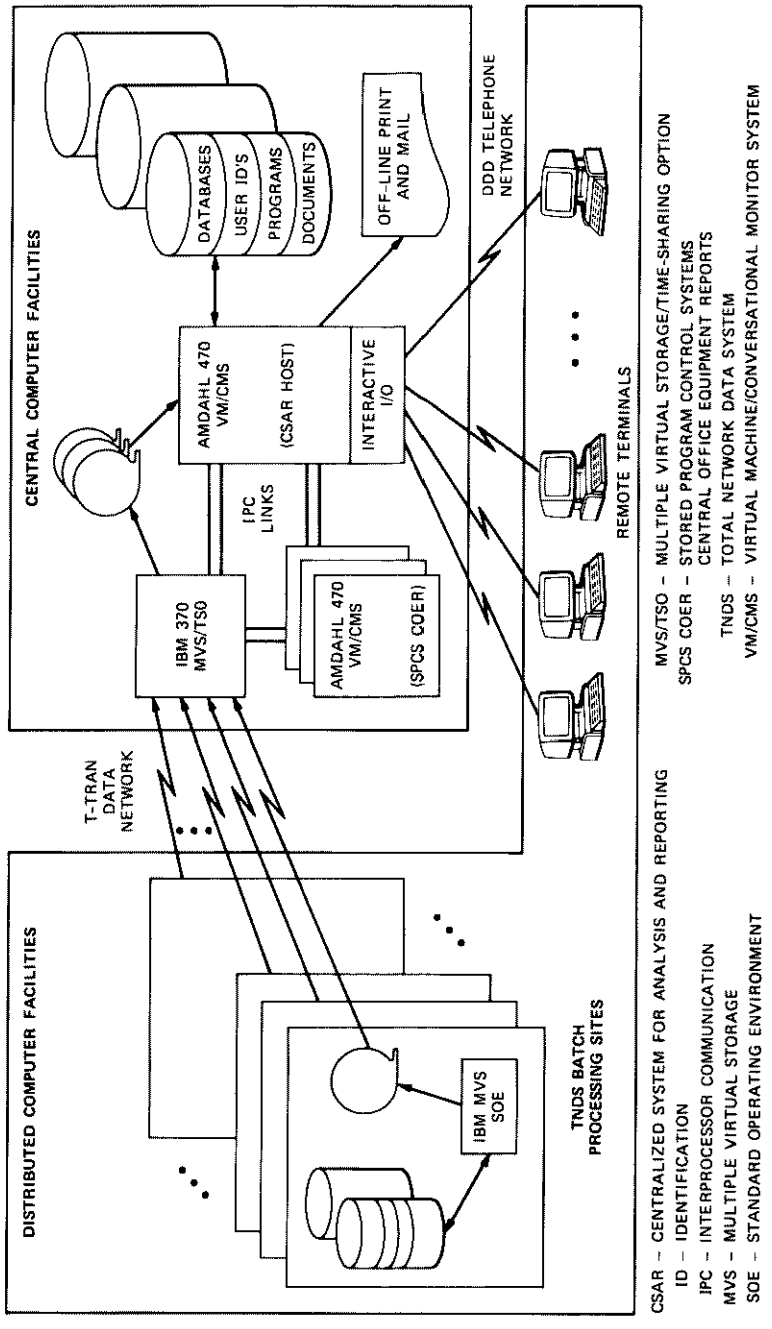



Fig. 1—CSAR computer system configuration.

CMS computer facility is the home of two other TNDS centralized operation support systems: the Stored Program Control System Central Office Equipment Reports system (SPCS COER)[2] and the Small Office Network Data System (SONDS).[5] Performance data are sent directly from SPCS COER to CSAR through a VM/CMS remote spooling interface.

This central host computer system offers the facilities and the processing capacity required by CSAR functions. The large volumes of data received from the 26 remote TNDS processing sites currently sending data to CSAR are scheduled for overnight processing using the VM/CMS batch facility. The analysis, correlation, and database loading occur during off-business hours at a reduced cost and without interfering with the typical interactive user.

## IV. CSAR ON-LINE FEATURES

The primary purpose of CSAR is to provide on-line interactive access to central databases that contain extensive TNDS performance information. The on-line CSAR features can be logically grouped into four general categories:

1. On-line user documentation
2. TNDS performance reporting
3. Database and processing controls
4. Administrative information reporting.

The many features work together, thereby enabling each OTC to tailor the database capacity and reporting strategy to its own data requirements, organizational structure, and management needs.

The following paragraphs present an overview of the CSAR features categorized above. The description will also highlight the on-line reporting techniques and selected features that distinguish CSAR as an innovative approach to special-purpose on-line database access and reporting.

### *4.1 Interactive user dialogue*

The CSAR interactive dialogue begins with the highest-level prompt: REQUEST =, which allows the user to select a system feature. The dialogue proceeds in a question and answer fashion until the user enters a specific request or inquiry.

The CSAR interactive user dialogue is designed to offer a user-friendly interface. The dialogue design gives the inexperienced user specific direction and guidance when questions and problems arise. For the experienced user, an abbreviated dialogue sequence can be strung together on a single line to save typing and interaction time. The user dialogue allows flexible inquiry and reporting. For frequent requests of tailored reports, the dialogue sequence can be defined once, saved, and activated when the need arises.

### 4.2 On-line documentation

A fundamental goal of CSAR is to be a totally on-line interactive tool that includes all necessary user documentation. Within CSAR, user documentation is available on-line in the form of 40 lessons that cover all aspects of CSAR use and administration. A particular lesson can be retrieved on-line at the user's terminal or printed off-line and mailed to specified address. This form of documentation is timely and easy to maintain, and eliminates the many problems associated with distribution lists and central reproduction.

### 4.3 Performance reporting

CSAR performance reports present data in predefined formats. Report content, however, is dynamically retrieved and prepared to satisfy specific user needs. The flexibility of the CSAR reporting mechanism lies in four levels of data selection available to the user. The first level, Report Type, identifies a particular enumeration or summarization of the performance data and implies an associated physical format. The second level, System, selects one, or possibly all, systems in the TNDS. The third and most versatile level of selection is Organization. The user can select one or more reportable entities, traffic units, by naming the organization to which they belong. The company organization structure (known as the organizational map) is defined to the system by the CSAR company coordinator or administrative user as part of start-up procedures and can be dynamically modified whenever necessary. The CSAR map is more fully described later in this section. The fourth level, Time, specifies the calendar dates for which results are desired. The performance report request, and the underlying database retrieval, are defined by the composite of these selection criteria: Report Type, System, Organization, and Time. Any additional specifications are supported by a list of Options that are different for each report type.

There are four general types of performance reports:

1. Performance Indicator reports
2. Performance Summary reports
3. Performance Monitor reports
4. Unsatisfactory Results Display reports.

The following paragraphs describe the purpose and basic characteristics of these four report types. Specific examples of reports, their use by the OTC personnel, and the benefits to overall TNDS operations are discussed.

#### 4.3.1 Performance indicators

The Performance Indicator Reports (PIRs) are intended for weekly troubleshooting of TNDS operational problems and data abnormali-

ties. Those personnel most immediately responsible for data administration for the various TNDS systems access CSAR on a regular basis to detect these problems and isolate their causes. A PIR exists for each system in the TNDS scored under the TPMP. A PIR request retrieves weekly traffic unit data pertinent to one system in TNDS for any specified company organization. The performance measurements that appear on these weekly reports include those listed in Table I. The measurements typically quantify indications of missing data, validation failures, and record base inconsistency or errors. The report content is formatted as one line per traffic unit so the data for many traffic units can be scanned quickly. All performance measurement values that fall into the Unsatisfactory band are highlighted on the report.

Options are available with the PIR that make it even more powerful as a troubleshooting tool. These include "history" and "exception" options, which can be included in the initial PIR request, and the "detail" option, which can be selected after the PIR has been printed and perused. The history option allows inclusion of up to 15 previous weeks of information on the PIR. The exception option causes CSAR to report information on only those units for which one or more of the week's performance measures exceed the band U threshold. The history and exception options may both be exercised on the same PIR request to highlight offices consistently performing unsatisfactorily.

The detail option is somewhat open-ended. After the PIR is printed for a given week, organization, and TNDS system, the user can request details for any Traffic Unit (TU) in the report. After the detailed report for that unit is produced for the given week and TNDS system, the user is asked if details are desired for the same unit and week for another system in the TNDS. Thus, where applicable, the user can look upstream in the TNDS to see where problems first appeared in the TNDS process or downstream to see their ultimate effect. This may be very helpful in tracking down subtle problems.

There are too many combinations of PIRs and options to describe here, but a typical example of analyzing results and problems using a few reports should be indicative of the assistance provided by CSAR. Figure 2 shows a PIR for the No. 5 Crossbar Central Office Equipment Reports System (5XB COER)[1] for one district. An administrator responsible for monitoring 5XB COER results can obtain this report through the dialogue shown on the top two lines of the figure. In the dialogue, the questions from CSAR are in the large type to the left of "=:" followed by the user inputs in the smaller type. No options were selected in this example.

The report header identifies the report type, organization, and study date. As we can see in Fig. 2, this is the 5XB COER Performance

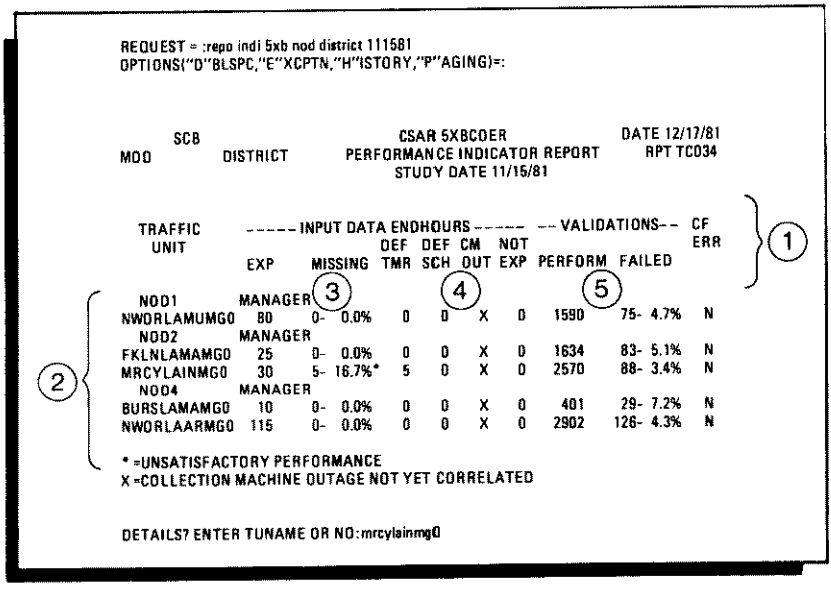



Fig. 2—Performance indicator report for the 5XB COER for one district.

Indicator Report for the New Orleans District of South Central Bell for the traffic study week of 11/15/81. A data header (feature ① of the figure) identifies the performance measurement data below. Under TRAFFIC UNIT, feature ②, five 11-character TU Common Language codes are shown. These are grouped together by the various organizations directly subordinate to the district being reported. Three subdistricts (NOD1, NOD2 and NOD4) are shown. Thus, any performance pattern associated with the subordinate organizational level would be obvious to the district level.

The next two columns, feature ③, pertain to the 5XB COER performance measure, MISSING DATA. The first of these columns shows the number of one-hour intervals expected for each TU by the 5XB COER system. These expectations are a part of the 5XB COER record base. The next column indicates missing traffic data, both the expected percentage and the number of one-hour intervals. The asterisk for MRCYLAINMGO indicates the percent of missing data for this week exceeds the band U threshold. Quick response to correct the deficiency will prevent it from persisting long enough to be band U for the official reporting period of one month.

The next four columns, feature ④, provide supplementary information resulting from CSAR automatically attempting to indicate the source of any missing data. In this case, the five one-hour missing intervals for MRCYLAINMGO are accounted for in the first column

by a deficient Traffic Measurement Request (TMR). (TMRs are used to request that intervals be passed from TDAS to the downstream systems such as 5XB COER, if perhaps an interval was missing from the TMR.) The next column of this feature contains supplementary information indicating that there were no deficiencies in the scheduling of this data from the collection machine to TDAS. The "X" in the following column, Collection Machine Outage (CM OUT), indicates that there were disturbances in the collection phase of TNDS sometime during the intervals when data were gathered for 5XB COER. But since all data have been accounted for, these must have been minor. The last of these four columns indicates that no unexpected data (NOT EXP) was received for any TU. That is, TDAS did not forward to 5XB COER any data that were not expected (excess TMR). Thus, all 5XB COER data missing for this district are accounted for by the DEF TMR, which is the difference between what COER was told to expect and what TDAS was told to forward by their respective record bases.

The next two columns of the PIR, feature ⑤, give the number of validation tests performed by COER and the number and percent that failed. Validation failures are performance measurements, but the absence of asterisks indicates that none of the TU's are in band U for that measure. If the administrator, however, sees the failures exceeding reasonable rates for any TU, detailed information can be obtained as described later.

The last column of the PIR indicates where the COER internal record base validation tests have detected a record base inconsistency (control file error). None has been detected in this district.

Thus, perusal of this week's PIR indicates that the only severe problem is a data loss for one TU caused by a scheduling mismatch between the TDAS and 5XB COER record bases.

After the PIR and its accompanying footnotes are printed, CSAR asks the user to enter a TU name if further details are desired. In this typical example, the last line of Fig. 2 shows that the user responded with MRCYLAINMGO, the TU having the missing data. A 5XB COER Traffic Unit Details Report, Fig. 3, is generated from that request. The top section of this report, feature ①, is a repeat of the PIR format for the one TU. This redundant information is included in the details report so that the report is complete for that TU without requiring reference to the PIR. A few lines below, feature ② shows the status of the data input to 5XB COER. The interpretation of this information, guided by the appropriate on-line lesson, is that six one-hour intervals (ending at 10:00, 10:30, 11:00, 11:30, 16:00, and 16:30) are expected by COER on all of the weekdays. However, the data expected for the one hour ending at 11:30 (between 11 and 12 on the

```
SCB                          CSAR 5XBCOER                    DATE 12/17/81
MRCYLAINMG0 TU               TRAFFIC UNIT DETAILS REPORT     RPT TC036
                             STUDY DATE 11/15/81

 TRAFFIC      ------ INPUT DATA ENDHOURS ------    -- VALIDATIONS --      CF
  UNIT                          DEF DEF CM  NOT                          ERR
            EXP   MISSING       TMR SCH OUT EXP PERFORM      FAILED
MRCYLAINMG0  30   5- 16.7%*      5   0   X   0     2570       88- 3.4%     N

INPUT DATA SUMMARY:        # RECORDS   EARLIEST - LATEST
        NEW                   7475     11/16/81 - 11/20/81
        SAVE                     0
        PURGED                   0

INPUT DATA STATUS CODES ON AN ENDHOUR BASIS:
                                    1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
11/15/81 SUN
         MON                        ...D          ..
         TUE                        ...D          ..
         WED                        ...D          ..
         THU                        ...D          ..
         FRI                        ...D          ..
         SAT

  KEY TO THE INPUT DATA STATUS CODES USED ABOVE:
  . =AVAILABLE, >90% DATA RECVD      M =MISSING, <=90% DATA RECVD
  D =DEFICIENT TMR                   I =INVALID IN TDAS
  E =EXCESS TMR                      U =UNAVAILABLE TO TDAS
  (^) BELOW ENDHR WITH CM OUTAGE     BLANK =DATA NOT REQUIRED

DATA VALIDATION TEST RESULTS ON AN EQUIPMENT COMPONENT BASIS:

  COMPONENT   # EXPECTED   # PERFORMED   # FAILED   % FAILED
      AM          100           75          0        0.0%
      CM          250          248          1        0.4%
      CS            0            0          0        0.0%
      CX            0            0          0        0.0%
      DM          231          214         11        5.1%
      DT           75           75          0        0.0%
      ET            0            0          0        0.0%
      IR          525          437         19        4.3%
      LL          438          383          4        1.0%
      MC           75           75          0        0.0%
      OR          177          165          0        0.0%
      OS          541          428         53       12.4%
      TK          500          470          0        0.0%
* =UNSATISFACTORY PERFORMANCE
X =COLLECTION MACHINE OUTAGE NOT YET CORRELATED

ENTER OTHER SYSTEM OR NO:tdas
ENTER DETAIL TYPE(S) <OCU,SCH,TMR,VD,ALL,NONE>: :tmr
```

Fig. 3—A 5XB COER traffic unit details report.

figure) were not provided to 5XB COER because they were not scheduled through TDAS on a TMR.

Thus, the 5XB COER details report can serve as a comprehensive trouble indication to organization(s) responsible for scheduling 5XB traffic data for this TU. In this case, the analysis and trouble referral would be complete after a brief on-line computer session.

The bottom of Fig. 3, feature ③, gives details on the data validation tests. When the TU validation failures are higher than an experienced

administrator thinks is normal, they could use these details to determine which component tests have abnormal failure rates.

The last two lines of Fig. 3 show the dialogue for optionally obtaining access to further information from other systems in the TNDS for this same TU. In this example the user continues the analysis by requesting complete TDAS details on all TMRs for this TU week.

#### *4.3.2 Performance summaries*

The Performance Summary Reports (PSRs), derived from weekly results in the database, evaluate performance in the TNDS operation for management at all levels of the telephone company organization. The summaries are available for one or all systems in the TNDS. The summarization technique determines individual traffic unit results for each performance measurement and places each traffic unit measurement into the appropriate band: Objective, Lower than Objective, or Unsatisfactory. As we discussed earlier, this banding technique avoids averaging the results across entities and masking specific cases of poor performance. As shown in Fig. 4, the summarized results reported are the number of traffic units and the percentage of traffic units in each performance band for each measurement. Totals are included on a system basis in the TNDS, along with grand totals. The on-line summarization is computed for the time period specified by the user (one or more study weeks) and for a requested organization (one or more traffic units).

As part of the regular CSAR processing, an official performance summary is run for monthly results of the total company. These results for the Service Observing Month (SOM) are then retained as part of the CSAR database. The grand total results for each company constitute the official input to TPMP.

As an aid to isolating and correcting problems CSAR also generates an optional list of all traffic units that appear in the unsatisfactory category (Fig. 5). This list can serve as a trouble list, and appropriate remarks regarding disposition and/or remedy can be made in the comments column.

#### *4.3.3 Performance monitors*

In addition to traffic unit banding and summary aggregation, CSAR also monitors other aspects of TNDS operations that are not within the scope of TPMP. Performance Monitor reports serve this important purpose and include the following areas of interest:

1. Common Update transaction statistics
2. Collection machine downtime
3. TNDS processing timeliness
4. TNDS software abnormal terminations and run times

REQUEST = :repo offsum all miss area 1081
OPTIONS ("B"RIEF)=:b

SCB
MISS AREA — CSAR ALL — DATE 12/17/81
OFFICIAL PERFORMANCE SUMMARY REPORT — RPT TC900
STUDY PERIOD 09/27/81-10/24/81

| MEASUREMENT CATEGORY | BAND "L" RANGE | TU'S TEST | NO. OF OFC-WK | NUMBER AND % TESTED IN BAND "O" | | "L" | | "U" | |
|---|---|---|---|---|---|---|---|---|---|
| DCU-INT MISSING | 7%-15% | 95 | 291 | 86 | 91% | 7 | 7% | 2 | 2% |
| DCU-INT ANY VER FAIL | 10%-25% | 95 | 291 | 85 | 89% | 8 | 8% | 2 | 2% |
| DET TEST FAILURES | 3%- 5% | 0 | 0 | 0 | 0% | 0 | 0% | 0 | 0% |
| INVAL DET TEST SCAN | 20%-30% | 24 | 54 | 24 | 100% | 0 | 0% | 0 | 0% |
| TOTALS-TDAS | | 214 | | 195 | 91% | 15 | 7% | 4 | 2% |
| ICAN/CU COUNT MSMTCH | 3%- 5% | 25 | 96 | 25 | 100% | 0 | 0% | 0 | 0% |
| ICAN/CU OTHER ERROR | .5%- 1% | 25 | 96 | 25 | 100% | 0 | 0% | 0 | 0% |
| SUSPECT SCHV ASSIGN | 1%- 2% | 25 | 96 | 23 | 92% | 1 | 4% | 1 | 4% |
| UA/UE SCHV USAGE | 3%- 5% | 25 | 96 | 24 | 96% | 1 | 4% | 0 | 0% |
| TOTALS-ICAN | | 100 | | 97 | 97% | 2 | 2% | 1 | 1% |
| CGMT INSUFF DATA | 5%-10% | 93 | 146 | 86 | 91% | 5 | 5% | 3 | 3% |
| CGMT OTHER INVALID | 15%-25% | 93 | 146 | 83 | 89% | 3 | 3% | 7 | 8% |
| CGMT UNASSOCIATED | 5%-10% | 93 | 146 | 90 | 97% | 1 | 1% | 2 | 2% |
| TOTALS-TSS | | 279 | | 258 | 92% | 9 | 3% | 12 | 4% |
| LU INVALID FOR LBI | 10%-25% | 69 | 143 | 66 | 96% | 3 | 4% | 0 | 0% |
| LU MEAS INVALID | 5%-15% | 69 | 143 | 66 | 96% | 3 | 4% | 0 | 0% |
| TOTALS-LBS | | 138 | | 132 | 96% | 6 | 4% | 0 | 0% |
| 5XCOER CNTRL FILE ER | 25%-75% | 35 | 138 | 34 | 97% | 1 | 3% | 0 | 0% |
| 5XCOER ENDHR MISSING | 10%-15% | 35 | 138 | 33 | 94% | 0 | 0% | 2 | 6% |
| 5XCOER VAL FAIL/PERF | 5%-10% | 35 | 138 | 35 | 100% | 0 | 0% | 0 | 0% |
| TOTALS-5X8C | | 105 | | 102 | 97% | 1 | 1% | 2 | 2% |
| TOTALS | | 836 | | 784 | 94% | 33 | 4% | 19 | 2% |

Fig. 4—Performance summary report.

5. TSS data by servicer responsibility.

These reports present an additional view of the effectiveness of data collection equipment and various company organizations.

### *4.3.4 Unsatisfactory results display*

The management users of CSAR often prefer results that more clearly reflect general performance levels, trends over time, and direct comparison between organizations or between the component systems of the TNDS. The Unsatisfactory Results Display reports plot the poor performance (band U) in a series of horizontal bar graphs. The graphs are simple in appearance and offer a wide variety of options that reveal performance trends at a glance. The results can be plotted with subtotals by any system in the TNDS, or by the organizations that are directly subordinate to the one requested. For example, as we

REQUEST = :repo summ tss ma district 102581-111581
OPTIONS ("B"RIEF OR "L"IST)=:1

SCB
MA DISTRICT CSAR TSS DATE 12/17/81
PERFORMANCE SUMMARY REPORT RPT TC601
STUDY PERIOD 10/25/81-11/21/81

| MEASUREMENT CATEGORY | BAND "L" RANGE | TU'S TEST | NO. OF OFC-WK | NUMBER AND % TESTED IN BAND "O" | | "L" | | "U" | |
|---|---|---|---|---|---|---|---|---|---|
| CGMT INSUFF DATA | 5%-10% | 18 | 67 | 17 | 94% | 0 | 0% | 1 | 6% |
| CGMT OTHER INVALID | 15%-25% | 18 | 67 | 18 | 100% | 0 | 0% | 0 | 0% |
| CGMT UNASSOCIATED | 5%-10% | 18 | 67 | 17 | 94% | 0 | 0% | 1 | 6% |
| TOTALS | | 54 | | 52 | 96% | 0 | 0% | 2 | 4% |

SCB
MA DISTRICT CSAR TSS DATE 12/17/81
BAND U LIST
STUDY PERIOD 10/25/81-11/21/82

| TRAFFIC UNITS | CATEGORIES | VALUE | COMMENTS |
|---|---|---|---|
| BLZNMSMA247 | CGMT UNASSOCIATED | ( 22.6% ) | |
| CLEVMSMA84A | CGMT INSUFF DATA | ( 33.3% ) | |
| | 2 BAND U TRAFFIC UNITS OF 18 | | |

Fig. 5—Performance summary report showing unsatisfactory traffic units.

see in Fig. 6, an area manager can request a five-week performance graph containing an area total and subtotals for each of the divisions in that area. The official summary results are also available as graphs.

#### *4.3.5 Bell System summaries*

Analysis of TNDS performance can extend to the level of the entire Bell System when the performance results for all OTCs are aggregated together. Here, there is an opportunity to assess the need to modify TNDS, systems or operating methods in the CSAR itself. There are four types of reports available at this level. One gives Bell System total results in the form of the "Brief" version of the Performance Summary report (no band U list). The "Company" option, when exercised, produces a report of the number of performance measurements falling into each of the bands O, L, and U for each company. The "graph" option generates a graph of the band L and band U results by company. The remaining option is especially valuable for TNDS and CSAR project managers at AT&T and Bell Laboratories. This "Measure" option generates a list of each company's percentages and the Bell System totals in bands O, L, and U for each of the performance measures. Thus, problems common to a large cross sec-

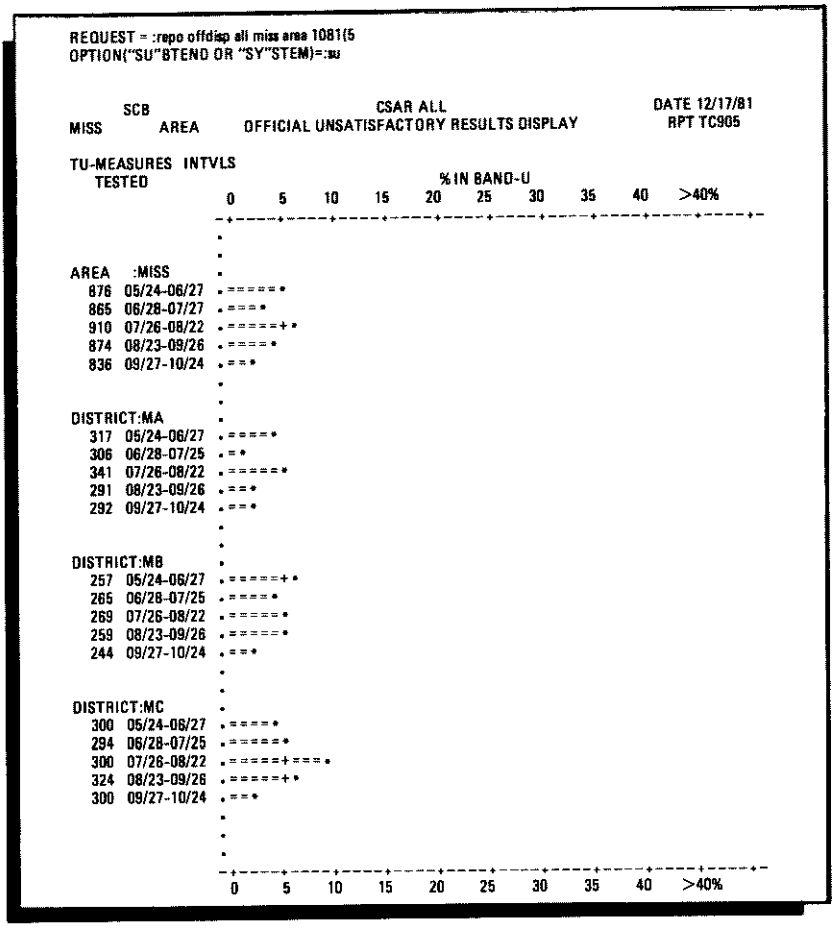



Fig. 6—Unsatisfactory results display.

tion of users can be examined to determine what system design or training efforts might be appropriate.

### *4.4 Database controls*

CSAR creates and maintains separate central databases for each OTC using the system. On-line database controls permit the CSAR coordinator or Headquarters staff personnel to decide on the size of the database, the retention periods of three classes of performance information, the timing of certain automatic CSAR processing, and the organizational structure to be used for reporting purposes. Each of these controls may be exercised at any time to adapt to changes in data needs, company reorganization, storage costs, and other considerations.

#### 4.4.1 Data retention

While the current week or current Service Observing Month (SOM) performance results are of primary interest, past results can also be examined for evidence of trends or persistence. To provide this historical perspective, CSAR databases contain fundamental performance data organized on a TNDS subsystem and traffic unit basis for distinct study weeks. In addition, summarized performance results are stored for each SOM. Because different classes of performance data serve distinct purposes and make significantly different demands on the physical database storage capacity, three data retention parameters are available to the user:

1. Short-term Retention (STR) (from 6 to 16 weeks)
2. Long-term retention [from (STR+5) to 52 weeks]
3. Official results retention (from 12 to 30 months).

The most voluminous data, deleted after the short-term retention period, are the supporting details from which the weekly results are derived. These details reflect TNDS processing at the half-hour, or hourly, level, and include measurements in terms of TDAS Data Collection Units (DCUs), switching office equipment components, Trunk Servicer responsibility codes, etc. This fine level of detail helps the CSAR user to track and isolate specific problems. The data analysis algorithms also require these data to correlate cross-system effects, and accurately calculate the weekly traffic unit measures.

Weekly performance results are kept for a period of several months (long-term retention), and SOM results are retained for more than a year (official results retention) so that performance can be compared under similar seasonal conditions.

During the CSAR Batch processing the data retention restrictions are applied, all outdated database information is automatically deleted, and the storage space is freed for future use.

#### 4.4.2 Organizational map

The user-defined CSAR organization map provides a direct and dynamic control over database retrievals. The CSAR user is presented with a hierarchical logical view of the performance database for on-line retrieval purposes. Unlike most hierarchical database management systems, the hierarchy is not a static structure defined at database generation time; instead CSAR provides a flexible structure that can be modified easily on-line without database reorganization.

The basic reporting unit for most CSAR information is the traffic unit. Data tracking and problem diagnosis for the TNDS is most successfully accomplished by analyzing the scheduling, collection, processing, and validation activities at a fundamental traffic unit level. Data reported at this low level reveal problems that can be isolated to

specific physical devices, user transactions, record base items, or operational procedures. For performance measurement of complete systems, functions, or company organizations, and for more effective management use of the information, aggregate views of traffic unit data are more appropriate. Responding to these needs, CSAR provides flexible multi-level views that permit data access at any organizational level, from the elementary traffic unit up to the entire company.

The user defines a hierarchical structure (up to nine levels) that establishes a company organization desired for CSAR reporting purposes. The first or top level is reserved for the company name, and the lowest level is designated as the traffic unit level. The remaining seven hierarchical levels can be named to represent the company structure and levels of management responsibility (i.e., district level, division level, area level, etc.). Not all seven intermediate levels have to be defined. At each of the levels two through eight, one or more entities can be named and placed subordinate to an entity named in the level above it. Figure 7 shows an organizational map with a total of five levels.

The CSAR map logically associates each of the many company traffic units to the desired organizations identified in this hierarchical structure. The physical database storage of traffic unit performance data is completely independent of this logical hierarchy. The CSAR dialogue then allows groups of traffic units to be collectively requested by any designated organization at any level.

The map's structure and the association of traffic units to higher-level organizations is defined using special data-entry features of the CSAR dialogue. A complete set of update commands allow the user to add, delete, change, rename, and move individual traffic units, complete organizations, or levels. The changes are performed on-line with the resulting organization in effect immediately for all subsequent report requests. CSAR features also include the ability to list the map in a variety of ways to determine or verify the current company organizational structure.

The direct control over a retrieval hierarchy and the ability to modify it easily make the CSAR organizational map a most versatile and innovative system feature.

### *4.5 Processing controls*

CSAR on-line features can override certain automatic batch processing functions. These controls are enacted through the normal CSAR dialogue, but only by the privileged headquarter's users. The overrides exist to meet two types of needs: (1) a management demand for an early assessment of official monthly results; and (2) an administrative need to respond quickly and conveniently to batch processing problems.

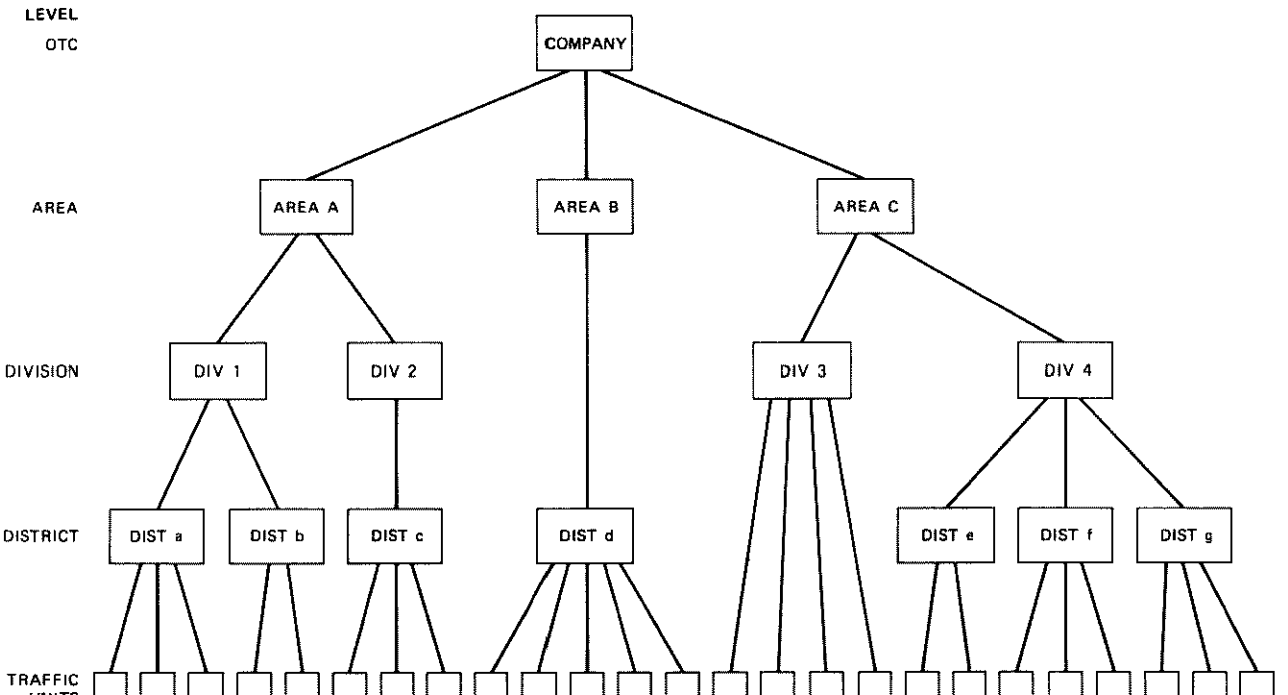



Fig. 7—Organizational map structure.

#### 4.5.1 Generating official results

A process delay parameter, entered as a company option, defines the longest expected delay between the collection of traffic data and the delivery of the TNDS performance data to CSAR. The database loading process closely monitors actual data delay, particularly to determine late data impact on official TPMP monthly results. The official performance summary results are generated automatically during the CSAR batch process after the appropriate period of time has elapsed. In cases where management personnel require official results prior to the expiration of the process delay period, CSAR permits on-line interactive generation of the official performance summary. This on-line feature enables the administrator to satisfy immediate needs or to correct past results by regenerating summaries in the event of unexpected, very late data.

#### 4.5.2 Batch job control

Process control features also exist to simplify overseeing the data transmission and CSAR batch database loading process, described in Sections 3.1.2 and 3.1.3. Each OTC's administrator responds to abnormal completion of CSAR batch jobs. Interactive features include Rerun and Remove commands allowing direct control over the disposition of individual data transmissions that were received but not successfully processed. These controls are exercised after the administrator determines the nature of the problem using diagnostic and error recovery information supplied by the system.

### 4.6 Administrative reporting

The telephone company administration of the CSAR software system requires activities such as: initial implementation, organizational map definitions and maintenance, selection of company options, coordination of data transmissions, and monitoring of the ongoing data processing. Each OTC has a designated CSAR administrator. Some of CSAR's features simplify the administrator's job. The data transmission and database entry operation have been automated to the fullest extent possible. The CSAR database provides on-line interactive access to status information and various reports that assist in system administration. The administrator relies on several sources of information to monitor the overall operation of CSAR:

1. Local and global login messages
2. News items
3. Tape processing status
4. Batch data processing summary
5. Map Data Discrepancy report

6. Interactive feature usage statistics
7. Operational cost information.

When an administrative user logs on, the Messages, News Items, and Tape Processing Status display important happenings without the need for database query. Based on any abnormal terminations indicated on the tape status, the corresponding batch job is investigated by accessing the Batch Data Processing Summary. The summary serves as a log of all processing activities and contains any error diagnostics identified by the CSAR software. Many conditions can be effectively identified and corrected by the administrator. Other problems may require investigation by the central development organization, and the error message would instruct the user to initiate the trouble referral.

The entry of new data into the CSAR database may involve traffic units that had not been anticipated, or for some reason not properly defined in the company organization map. The performance data are retained for these traffic units and are entered into the database. In addition, each previously unknown traffic unit is temporarily added to the map directly subordinate to the company level, where it remains until the administrator uses the CSAR dialogue to move the traffic unit to the appropriate position in the organization hierarchy. CSAR alerts the users of any such unexpected or unusual conditions via a Map Data Discrepancy report, and by Input Data Anomaly sections of the Batch Data Processing Summary. The software system design incorporates similar error detection and supporting diagnostic information throughout to simplify system administration and maintenance.

## V. CSAR SOFTWARE DESIGN

In the design of a software system several conflicting factors are addressed and balanced to provide an efficient, flexible system that is responsive to the user needs. Three aspects of CSAR software design deserve specific mention and are covered in the following paragraphs.

### *5.1 On-line response time*

A primary concern in the design of CSAR was flexibility in selecting the set of traffic units, TNDS systems, and time periods over which performance can be summarized. Depending on the particular request, tens of thousands of data items would have to be summarized on line and reported to the user. To accomplish this summarization with good on-line response time required special attention in the CSAR design. The physical database organization chosen reflects the high-level summarization options of "system" and "time period." Unique files

exist for each system/week that contain the key performance information for all reporting traffic units. CMS files are organized and accessed with a physical block size of 800 bytes. The key performance indicators for many traffic units are stored together in logical records whose lengths are an integral multiple of 800 (one TU requires much less than 800 bytes) to make efficient use of the operating system's file structure. File pointer tables exist to identify the CMS record (block) and location within the block for each active TU. The internal traffic unit identification scheme utilizes a hashing function together with the file pointer table to efficiently access the data for a particular TU within a file. The logical structure of the map efficiently converts the OTC organizational level to a set of reportable TU hash indices. Finally, the TU access order is optimized based upon the file pointers to minimize the amount of I/O necessary to read the raw performance data. These techniques used together result in worst-case summarization response times typically less than six seconds in an average size OTC.

### *5.2 Flexibility in reportable measures*

The measurement plan consists of 19 individual performance indicators. These basic indicators have been refined several times since the measurement plan was first introduced to provide equitable reporting across companies and to encourage continued performance improvement.

The design of the software to support these changing requirements had to be robust and easy to modify. The identification of distinct measures, the start and stop effective dates, the banding thresholds, and other key information are all specified in a single file external to the software programs. During system execution this file is accessed to load tables that control the summarization process. This table-driven summarization approach is also beneficial in that the effects of proposed changes to the mesurement plan can be observed by editing a development copy of this file and using this modified version when accessing the OTC data. Many such modifications can be studied and later implemented without changing or recompiling any software.

### *5.3 Overlay structure*

User chargeback generated by using this AT&T VM/CMS facility is highly sensitive to the core size of the virtual machine required for the application program. To minimize this aspect of chargeback, and to improve efficiency by reducing virtual storage paging, the system was designed with an overlay structure consisting of over 120 separate modules. The design of these modules reduced the total amount of

code generated by combining approximately 560 functional routines as building blocks.

The overlay structure maintains common routines and information required throughout the on-line session in a root area that is never overlayed. Specific software required to respond to a particular user request is overlayed in multiple levels below the root. In the most complex request, eleven individual overlays are used, reducing the amount of incore storage required from 500 to 130 kilobytes.

## VI. CONCLUSION

The goal of bringing performance measurements and operations analysis information to the TNDS has been achieved by implementing the TPMP through CSAR. TNDS managers and administrators now receive reports that are comprehensive, easy to obtain and use and on time. CSAR can be readily adapted and documented as TNDS changes. In fact, method changes are currently under way to improve the effectiveness of CSAR in isolating TNDS data collection and provisioning problems. Recent changes and future plans, not reflected in this article, include the revision or elimination of certain performance measurements. These changes will shift the emphasis from strictly end-system analysis to measurements that detect and quantify problems of data availability earlier in the flow of data through TNDS. As data responsibilities have become better focused organizationally, TPMP has been reexamined to ensure the plan meets the changing needs of OTC managers. This article reflects the CSAR software system and TPMP methods as of the middle of 1982.

CSAR became available to the OTCs in the last half of 1979. No official TPMP reporting was required by AT&T, however, until July of 1980. During the introductory interval, the OTCs began to use CSAR reports as effective tools for traffic data administration and management. At the start of the official reporting, the Bell System average for band U measurements was about 10 percent. (This is thought to be about one-half of what it was at the beginning of the introductory period.) The current Bell System average of band U measurements is about 5 percent. Thus, CSAR is providing effective in improving the delivery of valid traffic data so that the Bell System network can respond efficiently to customer needs.

## REFERENCES


1. N. D. Fulton, J. J. Galiardi, E. J. Pasternak, S. A. Schulman, and H. E. Voigt, "Total Network Data System: Equipment Systems," B.S.T.J., this issue.
2. R. F. Grantges, V. L. Fahrmann, T. A. Gibson, and L. M. Brown, "Total Network Data System: Central Office Equipment Reports for Stored Program Control Systems," B.S.T.J., this issue.

3. P. V. Bezdek and J. P. Collins, "Total Network Data System: Trunking Systems," B.S.T.J., this issue.
4. C. J. Byrne, D. J. Gagne, J. A. Grandle, Jr., and G. H. Wedemeyer, "Total Network Data System: Data Acquisition and Near-Real-Time Surveillance," B.S.T.J., this issue.
5. D. H. Barnes and J. J. O'Connor, "Total Network Data System: Small Office Network Data System," B.S.T.J., this issue.

## AUTHORS

**Dennis R. Anderson,** B.S.C.S., 1976, Michigan State University; M.S.C.S., 1978, Purdue University; Bell Laboratories, 1977—. Mr. Anderson initially worked on software development for the No. 5 Crossbar Central Office Equipment Reports system. He later worked on performance measurement of the traffic data collection and processing functions performed by the Total Network Data System. He defined new measurement methods and designed interactive software for efficient data analysis, database access, and performance summarization. In addition to development responsibilities, he wrote feature requirements and participated in project planning. He is currently working on the Digital Ordering and Planning System for No. 5 *ESS*® switching equipment. Member, ACM, Tau Beta Pi.

**Melvin J. Evans,** B.S.E.E., 1957, University of Utah; M.E.E., 1959, New York University; Bell Laboratories, 1957—. After joining Bell Laboratories, Mr. Evans worked on guidance equations used by the Bell Laboratories Command Guidance System for rocket launch vehicles. He has supervised guidance equation work, lunar mission studies, and military system effectiveness studies. He has also supervised work on operations support systems that assist Bell System network and data administrators. Starting in 1976, his group took part in a study of Total Network Data System performance, and then developed the Centralized System for Analysis and Reporting (CSAR). Currently, he supervises the Planning Mechanization group, which provides computerized support for operations planning.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

## *Total Network Data System:*

# Operating Company Perspective

By J. PFEIFFER, Jr.*

(Manuscript received June 21, 1982)

Implementation and management of the many individual systems comprising the Total Network Data System (TNDS) is a monumental undertaking, with far-reaching impact on a telephone company's organization and methods. This paper describes how one such company—Southern Bell—planned for and converted to TNDS, extending over a number of years. It also discusses the current operations and benefits of TNDS.

## I. PRE-TNDS IN SOUTHERN BELL

Before examining TNDS as it exists now within an operating telephone company, it should be helpful to review data-collection procedures before the introduction of TNDS.

### *1.1 Manual switch counts*

Starting with the introduction of dial Step-by-Step (SXS) switching machines in the 1920s, traffic studies were made by manually counting the number of individual switches that were in use, over fixed intervals of time. A large group of clerks would descend upon an SXS central office to make these studies, typically once or twice a year. Counts were made on a sample basis, generally every three minutes, of all

* Southern Bell.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

busy switches in specific groups, such as trunking groups or a switching stage. A supervisor would control the timing of each count with a whistle, and record the number of busy switches for subsequent summarization and calculation. (See Fig. 1.)

The deficiencies inherent in these studies were severe and obvious. They lacked accuracy; there was significant data "skew" due to inherent difficulties in starting and stopping these counts with precision; they were labor intensive; and since they could only be made infrequently, there was a high probability of missing the busy hour or busy season for a given office. Although these traffic studies were the only basis for engineering and administering the telephone network, there was a general lack of confidence in their accuracy. A judgment factor was frequently added to the study results in order to protect service. For this reason, a great many offices tended to be over-engineered in at least some of their switching stages. No one really knows how much capital was wasted in providing unnecessary equipment, but it had to be considerable.

### *1.2 Introduction of the traffic usage recorder*

A major improvement was made in the mid-1950s with the introduction of the Traffic Usage Recorder (TUR). The TUR permitted individual switches to be wired into a "grouped" register, each register

Fig. 1—A manual count being recorded of number of switches engaged with calls. The count was made to determine usage in SXS dial offices.

representing a specific trunking group or switching stage. Also, the TUR made it possible to study traffic in crossbar offices, which had not been practical with the manual count method. (See Fig. 2.)

The TUR had one distinct advantage. It needed to be read only hourly, rather than every three minutes. This sharply reduced the clerical load, and it also improved the data-skew problem. It then became practical to schedule traffic studies more frequently, and confidence in data quality increased. As office sizes grew, however, the traffic-register readings and data-summarization chores grew as well.

#### *1.2.1 Camera added to TUR*

A further improvement came in the late 1950s. A camera and flash unit were positioned over groups of up to 160 traffic registers. Under the control of a pre-set TUR timer, the camera took pictures of these banks of registers for as many hourly periods as study needs dictated. The advantage gained was to virtually eliminate data skew, and further reduce the clerical effort in gathering the data and manually summarizing it. Computer programs were created in Southern Bell to process this raw data and produce a family of user-oriented reports downstream.

However, other problems were created. The network administrator who was responsible for doing the traffic study now had to get involved in the manual camera and film-processing procedures. A substantial investment in personnel and equipment became necessary in order to view and keypunch data from the film strips. A key verify procedure was also introduced that virtually eliminated keypunch errors, but at the cost of doubling the key-entry costs. Finally, the work load on the network-administration staff was highly variable since traffic studies were not evenly distributed over time. Since studies are scheduled to capture peak loads, they tend to be concentrated in seasonal and monthly peaks. As a result of the uneven loads, some delays occurred in the manual processing activities.

### *1.3 OCR experiments*

Southern Bell, and one or two other Bell System operating companies, experimented with methods to mechanize this key-entry function during the early 1970s with Optical Character Recognition (OCR) techniques. While our trial was a technical success, it did not prove to be economical and was abandoned after two years.

### *1.4 Pre-TNDS problem areas*

Despite the progress made over the years, our data-collection situation just prior to the introduction of TNDS had four serious problem areas.

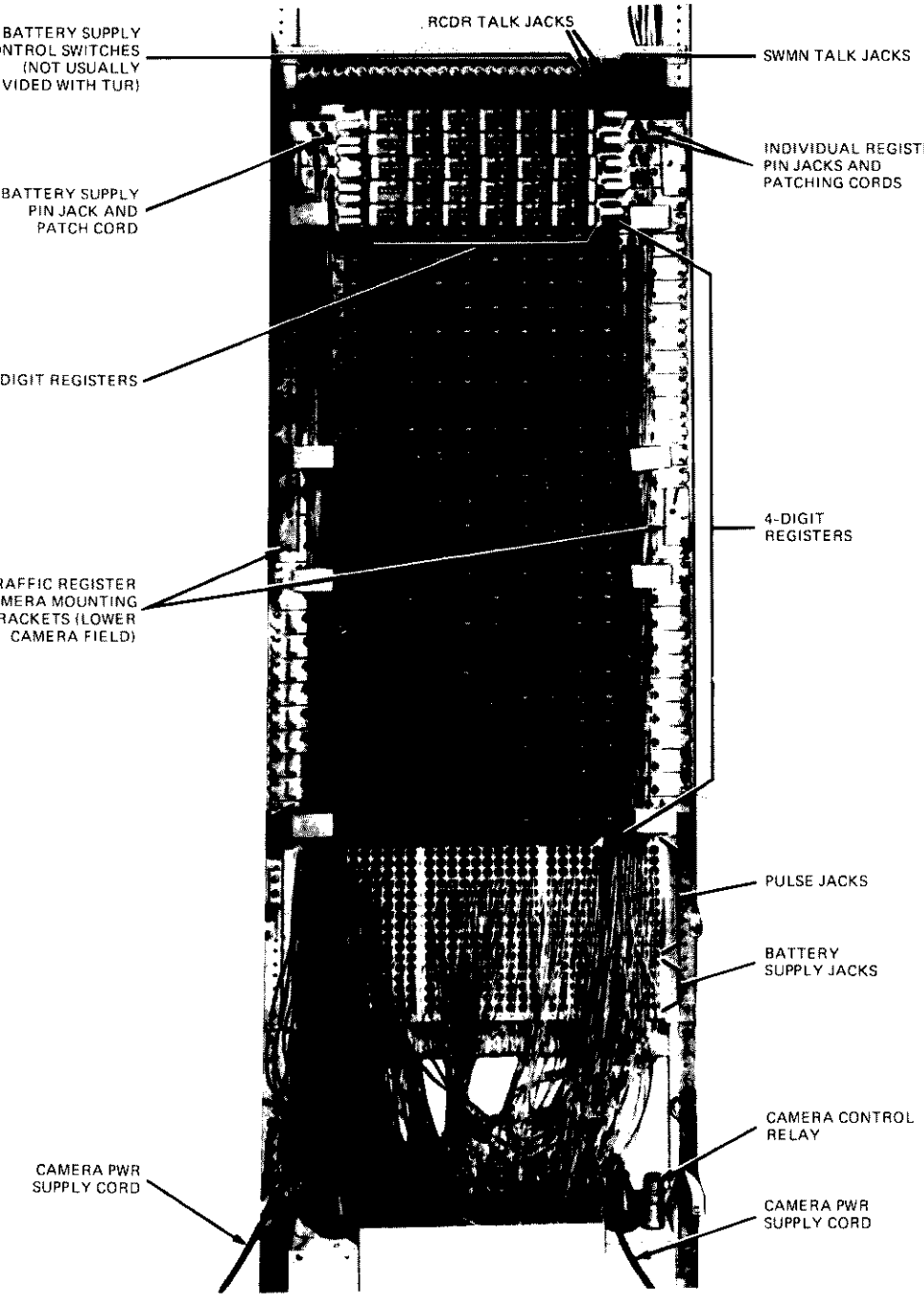



Fig. 2—TUR traffic register cabinet. Camera and hoods are not shown.

First, it was difficult to administer because it required extensive and timely coordination between several departments, each having different priorities.

Second, the data quality was lower than desired. This was due to inherent problems with the cameras, lighting, and film processing.

Third, the completed studies took too long to reach the ultimate user. This was due to the complex sequence of steps necessary, as well as the uneven study schedules. Frequently, the studies had lost much of their value by the time they were received.

Finally, each operating area within Southern Bell had evolved its own set of procedures for data collection. There was a lack of central direction and standardization. The stage was set for TNDS in Southern Bell.

## II. TNDS IMPLEMENTATION STRATEGY

### *2.1 Criteria for TNDS*

TNDS was approved for implementation in Southern Bell as a corporate-wide project in November 1973. The following criteria were basic to the implementation plan:

1. To realize the maximum benefits, all TNDS modules would be implemented (both available and planned) unless there were compelling reasons for doing otherwise.
2. The "downstream" data processing modules would be implemented on a centralized basis.
3. The "upstream" data-collection (EADAS) modules would be implemented on a decentralized basis. There would be one Engineering and Administrative Data Acquisition System (EADAS) installation in each administrative area, and administrative boundaries would not be crossed.
4. There would be a small, strong centralized staff to provide support to the areas for planning, implementation, ongoing technical assistance, and methods.
5. There would be no downstream processing of key-entry, filmed (non-EADAS) data in Southern Bell, even though TNDS could accommodate it. This would avoid slowing the downstream weekly-coordinated process to the slower pace of key-entry data.
6. The administrative areas would be cut over to TNDS on a phased basis, with each area's schedule determined by need and capital availability.
7. All central offices would interface with TNDS unless they were below a designated minimum size or were to be replaced within about three years.
8. Commercial courier service would be used to transmit EADAS

data-collection tapes to the single centralized processing location, and output reports back to the areas. Figure 3 illustrates the deployment.

### *2.2 Deployment schedule*

The "downstream" software of TNDS was installed shortly before the first EADAS installation to allow for a proper system test. The "upstream" EADAS systems, and their associated administrative staff, were installed according to the schedule shown in Table I.

This sequence was dictated principally by the rapid growth in Southern Florida, which emphasized the need for valid and prompt traffic data. The last three areas were more rural in nature, with slower growth and therefore less critical data needs, at least initially.

Depending upon the number of eligible central offices, one to two years were required to fully implement TNDS within each Southern Bell area. As was expected, the first few offices in each area required a greater amount of time to cut over, until the learning curve was mastered. Each area was given strong support from the centralized TNDS staff, particularly in the planning and early implementation stages.

TNDS was considered fully implemented in Southern Bell by 1980, with the addition of only growth central offices thereafter.

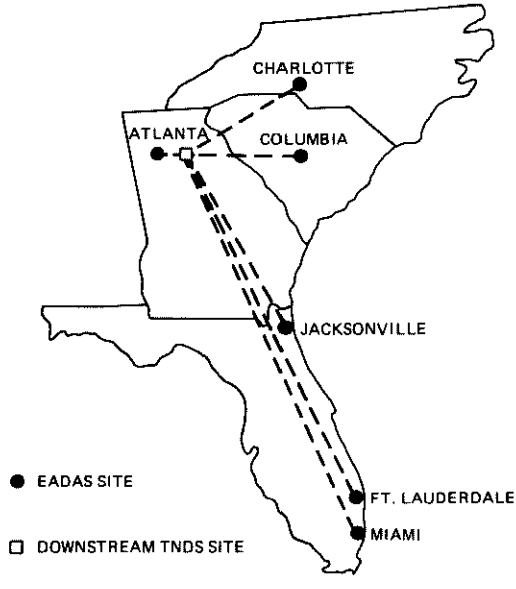



Fig. 3—Routes by which EADAS data-collection tapes and outputs are transmitted between centralized processing location and administrative areas.

Table I—EADAS installation schedule

| Area | City | Date |
|---|---|---|
| South Florida | Miami | October 1974 |
| Southeast Florida | Ft. Lauderdale | October 1974 |
| Georgia | Atlanta | June 1975 |
| North Carolina | Charlotte | July 1976 |
| North Florida | Jacksonville | June 1977 |
| South Carolina | Columbia | June 1977 |

## III. ORGANIZATION FOR TNDS

### *3.1 Work centers involved*

Southern Bell follows the guidelines of the Total Network Operations Plan (TNOP) quite closely. TNOP organizes all work functions within the Network Segment into clearly defined centers. The centers involved in our TNDS data-collection and report-distribution process are the:

1. Circuit Administration Center (CAC)
2. Network Administration Center (NAC)
3. Network Data Collection Center (NDCC)
4. Minicomputer Maintenance and Operations Center (MMOC)
5. TNDS Coordination Center (TCC)
6. Corporate Data Center (CDC).

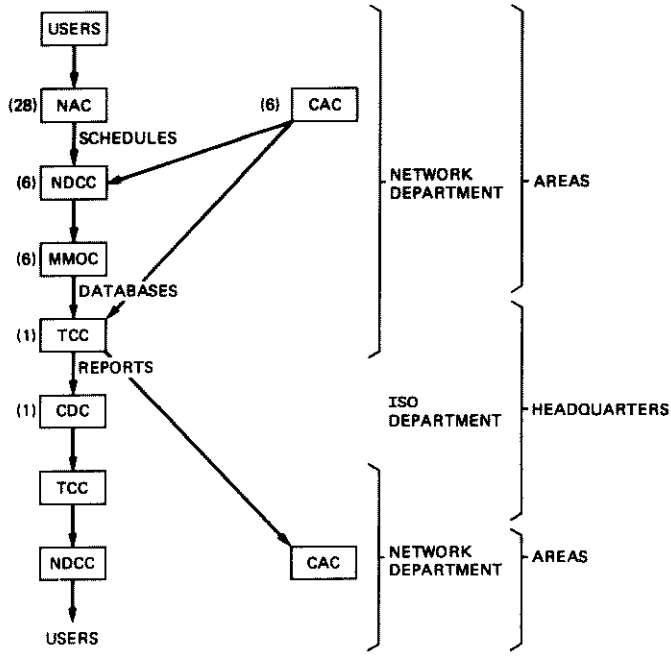
### *3.2 TNDS data flow*

The technical description of the data flowing through TNDS is described in detail in other articles of this issue. Figure 4 illustrates how the data flow is managed by these centers in Southern Bell. It should be recognized, however, that there are significant variations in this process within the Bell Operating Companies (BOCs) due to differences in geography, demography, and corporate structure.

1. There are twenty-eight Network Administration Centers (NACs) in Southern Bell, ranging from three to seven per area. Each NAC serves a geographic subset of an area. All requests for TNDS-equipment studies originate in, or are transmitted through, the appropriate NAC to the Network Data Collection Center (NDCC).

2. There are six Circuit Administration Centers (CAC) in Southern Bell, one per area. All requests for TNDS-Trunking studies are originated by the appropriate CAC, and are transmitted directly to the NDCC serving that area. The databases for trunking studies (TSS and TFS) are maintained by the CAC.

3. There are also six Network Data Collection Centers (NDCCs) in Southern Bell, one per area. The NDCCs are responsible for monitor-




NOTE: NUMBERS IN PARENTHESES INDICATE THE NUMBER OF CENTERS OF EACH TYPE IN SOUTHERN BELL.

CAC – CIRCUIT ADMINISTRATION CENTER
CDC – CORPORATE DATA CENTER
MMOC – MINICOMPUTER MAINTENANCE AND OPERATIONS CENTER
NAC – NETWORK ADMINISTRATION CENTER
NDCC – NETWORK DATA COLLECTION CENTER
TCC – TNDS COORDINATION CENTER

Fig. 4—TNDS data flow among administrative centers in Southern Bell.

ing the operation and maintenance of the EADAS Central Control Units (CCUs) in their area, monitoring the operation of the central office data-collection apparatus and associated data links, maintaining accurate record bases, and coordinating as required with the interfacing work centers to resolve problems. In Southern Bell, the NDCC is the principal focal point for data-collection activities in an area, and has no responsibilities other than successful data collection for TNDS.

4. The Minicomputer Maintenance and Operations Centers (MMOCs) are responsible for the operation and maintenance of the clustered minicomputers in their respective areas. EADAS is one of these clustered minicomputers. The MMOC performs preventive maintenance for EADAS on a schedule mutually agreed upon with the NDCC. It also is responsible for transmitting the magnetic tapes generated by EADAS to the TNDS Coordination Center (TCC) on a timely basis, via commercial courier service.

5. The TCC is a single, company-level work center. It serves as an interface between the various network users and the Corporate Data Center (CDC) of the Information Services Organization (ISO), which operates our downstream programs as part of a utility service for all corporate data-processing requirements. The TCC receives magnetic tapes with study data from all area MMOCs, plus all off-line database updates originated by both the NDCCs and CACs. These are formed into data-processing jobs and submitted to ISO by the TCC. The TCC is responsible for tracking the successful completion of the TNDS job through the CDC. A measurement plan has been instituted to determine the timeliness of CDC production for TNDS output.

6. The CDC is responsible for producing the downstream TNDS reports for Southern Bell. Three types of output media are used, depending upon the wishes of the individual users. These are impact print, laser print, and microfiche. Since TNDS processing is organized into stages, the output is spread over the week to level the load on all centers and resources. After each processing stage, the TNDS output is delivered to the TCC promptly.

7. The TCC then reorganizes the TNDS reports into individual user "batches" destined for each area, and transmits the reports back to the six NDCCs and six CACs via commercial courier service. The EADAS tapes are also returned to the MMOCs for reuse on a monthly cycle.

8. The CACs are end users, and upon receipt of their output reports, can immediately put them to use. The NDCCs are the end users for administrative reports that relate to the accuracy and health of the data-collection process; these are used directly by the NDCC. The other user reports are transmitted to the ultimate users in that area by the NDCC.

## IV. MANAGEMENT OF TNDS

### *4.1 Centers involved in TNDS management*

Although each center involved in the TNDS data-collection and distribution process is essential for overall success, the "up-front" centers—NAC, NDCC, and CAC—have an especially important role. These three centers are assigned primary responsibilities in the data-flow process through TNDS, including record-base, scheduling, surveillance, and distribution functions. The NAC is assigned the basic central office assignment functions for equipment measurement needs; the CAC maintains trunking databases; and the NDCC provides a centralized center with the technical expertise to interface directly with the TNDS subsystems. The NDCC performs the surveillance of the EADAS CCUs and is an area distribution point for TNDS output.

### 4.2 Use of TPMP and CSAR

The TNDS Performance Measurement Plan (TPMP) was introduced in Southern Bell in 1979 to improve the management of TNDS. This plan, along with the Centralized System for Analysis and Reporting (CSAR), has added two greatly needed features. As an index plan, TPMP has increased emphasis on the performance and the resolution of problems in the area of traffic-data collection. Additionally, CSAR, as a reporting tool, gives management at all levels specific indications of the success or failure of the TNDS process.

The pre-TPMP environment often found TNDS problems competing for attention with other indexed items. The TNDS problems often were given much lower priorities, and lengthy time intervals could occur before problems would be resolved. Since TNDS results are now being reported along with other indices in Area, Southern Bell, and AT&T results books, TNDS problems are given a higher priority. The weekly reporting and monitoring capability of CSAR also maintains emphasis on any problems until they are resolved.

#### 4.2.1 CSAR reports

The CSAR program itself has proven a valuable reporting tool for TNDS management. By providing reports tailored for any organization or level from an individual traffic unit up to a total company report, CSAR provides management with increased insight into the problems that may need attention. The level of detail available in CSAR makes it useful for first-level managers in the NDCC, NAC, or CAC to monitor and resolve weekly problems in TNDS processing for their specific area of responsibility. It also provides district-level summaries for local management and provides area and company summaries for reporting results. All of these functions are accomplished without user effort other than requesting reports via a dial-up terminal. All CSAR processes are fully mechanized.

#### 4.2.2 Center responsibility

Based on the functional responsibility of the three centers, the nineteen TPMP measurement categories have each been assigned to a specific center for initiating corrective action. Six are the responsibility of the NAC manager, three of the CAC manager, and the remainder belong to the NDCC. Although coordination among these and other centers is frequently required, the specific centers assigned each category are responsible for initiating the necessary action to correct any problems reported by CSAR. In addition, Southern Bell has established an objective level of performance for TPMP results, which is that 90 percent of all traffic units perform in the objective range.

Steady progress has been made in Southern Bell toward this objective since TPMP became an official Bell System measurement plan in July 1980. Both goals of TNDS performance measurement and performance improvement have been achieved.

## V. BENEFITS OF TNDS

There are a variety of benefits of TNDS from the operating telephone company perspective, particularly when compared to the pre-TNDS environment.

### *5.1 Timeliness*

TNDS produces reports to the user far more rapidly than the old camera/register arrangements. Surveillance reports are generated by EADAS/NORGEN in near-real time. Further, they are provided directly to the appropriate Network Administration Center, or Circuit Administration Center, only for those offices that these centers manage. In this way, the Company personnel can detect and resolve problems in switching or trunking very early—often before customers are aware of any troubles.

The bulk, downstream reports of TNDS (COER,* load balance, trunk servicing) are also made available in a relatively short time. In general, these are delivered to the user a week after the end of any study period. Prior to TNDS, it typically required four to eight weeks for any meaningful network data to be made available to the user.

### *5.2 Quality*

In the pre-TNDS environment, errors were commonplace. These were caused by keypunching mistakes, camera problems, and database synchronization problems. The quality of TNDS data is far superior, owing to mechanization of database validation procedures and the availability of a standard measurements plant. The shorter turnaround time of TNDS also facilitates prompt detection and resolution of database or hardware problems. Those data errors that do occur are not institutionalized over long periods.

### *5.3 Quantity*

The pre-TNDS environment was labor intensive. This essentially limited the number, scheduling, and quality of studies. The quantity of studies produced by TNDS today would have been impossible just a decade ago. TNDS has also given us studies that are far more sophisticated and detailed than had been possible previously.

---

* Central Office Equipment Reports.

### *5.4 Network investment*

The family of TNDS reports permits us to engineer and provision the network more efficiently. This improves the existing network utilization and helps us to better forecast growth. Maintenance of the network is also enhanced, allowing further improvement in use of existing facilities. TNDS, therefore, allows fuller use of our present investment and better planning for new capital expenditures.

### *5.5 Credibility*

TNDS lends a high degree of credibility to network data. The improved timeliness, quality, and quantity of data have eliminated the uncertainties of yesterday that resulted in overprovision of facilities in an attempt to protect service.

## VI. FUTURE TNDS DEVELOPMENT NEEDS

A formal enhancement procedure is available to the operating telephone companies by which future needs can be submitted, evaluated on a system basis, and, when approved, designed and developed. Enhancements are grouped into annual releases that are accompanied by training and documentation updates.

TNDS is a relatively mature system. Most enhancements now consist of refinements in report content, or in data-processing capabilities.

Southern Bell is a large, geographically dispersed company, and has needs that differ somewhat from those of smaller, more concentrated companies. Our TNDS is logistically dispersed. We see a real need for a system-supported data-transmission capability that would be quick, accurate, yet relatively economical.

There are two "loops" to this need. The first would be a capability of transmitting magnetic-tape data from the deployed EADAS sites to the central CDC for processing. This would avoid the expense of shipping these tapes by courier, would reduce the turnaround time by at least one full day, and would eliminate occasional tape problems caused by the uncertain environment of courier service.

The second "loop" would transmit the TNDS output to each NDCC (or CAC), or alternatively to the individual user. This would also eliminate substantial handling and shipping costs and reduce the turnaround time by at least another full day.

TNDS, although relatively mature, is dynamic. New data needs and improved technology will guarantee that we will always have opportunities for enhancement as long as there is a need for TNDS.

## VII. ACKNOWLEDGMENT

The author wishes to acknowledge the contributions of J. S. Rain-

water, Staff Manager, Southern Bell. Mr. Rainwater provided invaluable assistance by reviewing this article and making valuable comments.

## AUTHOR

**John Pfeiffer, Jr.**, B.A., 1951, Wesleyan University; M.B.A., 1968, Georgia State University; Southern Bell, 1954—. Mr. Pfeiffer was initially assigned to the Commercial Department, involved in market forecasting and business office operations. Subsequently, he transferred to the Traffic Department and was responsible for Traffic Planning in the Georgia Area. He is currently responsible for Total Network Data System (TNDS) support as well as producing the Total Network Operations Plan (TNOP) and Six-Year Plan in Southern Bell.

## ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| 5XB COER | No. 5 Crossbar central office equipment reports |
| 10HD | ten high day |
| ABS | average busy season |
| ACP | action point |
| ACU | automatic calling unit |
| AER | apparatus exception reports |
| AHT | average holding time |
| AMPS | average measured peak service |
| ANI | automatic number identification |
| AOOC | automatic out-of-chain |
| AORB | all originating registers busy |
| ASH | abnormally short holding time |
| ATB | all trunks busy |
| ATEMIS | Alston Traffic, Engineering and Management Information System |
| AUTRAX | Automatic Traffic Recording and Analysis Complex |
| BIS | Business Information Systems |
| BDAM | basic direct access method |
| BDE | batch data entry |
| BHD | busy hour determination |
| BOC | Bell Operating Company |
| bpi | bits per inch |
| C | capacitor |
| CAC | circuit administration center |
| CCIS | common channel interoffice signaling |
| CCS | hundred call seconds |
| CCSA | common control switching arrangement |
| CCU | central control unit |
| CDC | Corporate Data Center |
| CDO | community dial office |
| CDR | customer digit receiver |
| CGA | circuit group analysis |
| CGM | circuit grouping map |
| CGMT | circuit group measurement |
| CLCI | common language circuit identification |
| CLLI | common language location identifier |
| CM OUT | collection machine outage |
| CMP | completions |
| CO | central office |
| COER | central office equipment reports |
| COSMOS | computer system for mainframe operations |

| | |
|---|---|
| CPU | central processing unit |
| CRT | cathode ray tube |
| CSAR | Centralized Systems for Analysis and Reporting |
| CU | central unit |
| CU/EQ | common update/equipment |
| CU/TK | common update/trunking |
| CV | coefficient of variance |
| DAS | Data Access System |
| DCD | data collection device |
| DCU | data collection unit |
| DDD | direct distance dialing |
| DDM | daily data manager |
| DGU | detector group usage |
| DIS | Data Inquiry System |
| DIXC | data interchange |
| DLI | dial line index |
| DMA | direct memory access |
| DNHR | dynamic nonhierachical routing |
| DOC | dynamic overload control |
| DOC-N | district operations center – network |
| DP | dial pulse |
| DRE | directional trunk reservation |
| DRP | division of revenues process |
| DTD | dial tone delay |
| DTM | dial tone marker |
| DTP | data transfer point |
| E2AP | E2A process |
| EADAS | Engineering and Administrative Data Acquisition System |
| EADAS/NM | Engineering and Administrative Data Acquisition System/Network Management |
| ECCS | economic hundred call seconds |
| EDP | electronic data processing |
| EM | electromechanical |
| EMC | equipment measurement code |
| EMS | electromechanical system |
| EQ | equipment |
| ER | exception reports |
| ETDC | electronic traffic data collection |
| EVE | extreme value engineering |
| FEPS | Facility and Equipment Planning System |
| FIPS | file interface process |
| FIT/RAT | fitted ratio |
| FM | failure to match |

| | |
|---|---|
| GDG | generation data group |
| GFA | growth factor analysis |
| GP | graphics process |
| GTF | general trunk forecast |
| HD | high day |
| HT | holding time |
| HTR | hard-to-reach |
| ICAN | individual circuit analysis |
| ICUR | individual circuit usage recording |
| ID | identification |
| IMA | ineffective machine attempts |
| IMAP | Interactive Map Assembly Program |
| INA | ineffective network attempts |
| INU | incoming usage |
| INWATS | Inward Wide-Area Telephone Service |
| I/O | input/output |
| IPC | incoming peg count |
| IPC | interprocessor communication |
| IR | incoming register |
| ISO | Information Services Organization |
| KDP | keyboard, display, printer |
| LBI | load balance index |
| LBS | Load Balance System |
| LEX | lexical |
| LFG | line finder group |
| LFN | logical file number |
| LFS | Logical File System |
| LL | Long Lines |
| LTB | last trunks busy |
| LTU | last trunk usage |
| LU | load unit |
| MBC | maintenance busy count |
| MLSM | MLSS data manager |
| MLSS | Machine Load Service Summary |
| MMAR | monthly machine administrative report |
| MMOC | Minicomputer Maintenance and Operations Center |
| MON | Monitor System |
| MPRT | monitor print process |
| MPS | Message Processing System |
| MRCS | Modification Request Control System |
| MRS | Management Reporting System |
| MS | main station |
| MTO | message trunk order |
| MTS | message telecommunications service |

| | |
|---|---|
| MTU | maintenance usage |
| MUX | multiplexor |
| MVS | multiple virtual storage |
| NAC | network administration center |
| NCP | network control point |
| NDA | NORGEN data analyzer |
| NDCC | network data collection center |
| NM | network management |
| NMC | network management center |
| NOC | network operations center |
| NOCS | Network Operations Center System |
| NORGEN | network operation report generation |
| NPA | numbering plan area |
| NSEC | network switching engineering center |
| NSPMP | Network Switching Performance Measurement Plan |
| O | overflow |
| OCR | optical character recognition |
| ODF | office description file |
| OR | originating register |
| ORRS | On-line Records and Reporting System |
| OS | Operations System |
| OS | outgoing sender |
| OTC | operating telephone company |
| OTSS | Off the Shelf System |
| OVF | overflow |
| PA | periodic average |
| PATROL | Program for Administrative Traffic Reports On Line |
| PBC | peripheral bus computer |
| PBS | Playback System |
| PBX | private branch exchange |
| PC | peg count |
| PECC | product engineering control center |
| PGTF | preliminary general trunk forecast |
| PIR | performance indicator report |
| PSF | pending status flag |
| PSR | performance summary report |
| PWB | Programmer's Workbench |
| QCL | quality control limits |
| RA/BS | ratio to busy season |
| RCMAC | recent change memory administration center |
| RDUS | Reference Data Uptake System |
| ROC | regional operations center |
| ROP | receive-only printer |
| RSL | report specification language |

| | |
|---|---|
| RSS | Remote Switching Systems |
| RTS | reroute to seizure |
| SCC | switching control center |
| SCCS | Source Code Control System |
| SDOC | selective dynamic overload control |
| SLC | subscriber line concentrators |
| SLU | subscriber line usage |
| SMA | service month-to-date accumulation |
| SOE | standard operating environment |
| SOM | service observing month |
| SONDS | Small Office Network Data System |
| SPA | sequential projection algorithm |
| SPC | stored program control |
| SPCS | Stored Program Control System |
| SPCS COER | Stored Program Control System Central Office Equipment Reports |
| STP | signal transfer point |
| STR | short-term retention |
| STR | selective trunk reservation |
| SXS | step by step |
| TAMP | Trunk Administration Measurement Plan |
| TCBH | time-consistent busy hour |
| TCC | TNDS coordination center |
| TCT | telemetry computer translator |
| TDA | traffic data administration |
| TDAS | Traffic Data Administration System |
| TDF | tape data formatter |
| TDRS | Traffic Data Recording System |
| TFS | Trunk Forecasting System |
| TGSN | trunk group serial number |
| TIP | Trunk Implementation Plan |
| TIRKS | Trunks Integrated Record Keeping System |
| TMR | traffic measurement request |
| TNDS | Total Network Data System |
| TNDS/EQ | Total Network Data System/Equipment |
| TNDS/TK | Total Network Data System/Trunking |
| TNOP | Total Network Operations Plan |
| TNPC | traffic network planning center |
| TOPC | total office originating peg count |
| TPMP | TNDS Performance Measurement Plan |
| TRK | trunk |
| T/S | total to sample |
| TSO | time-sharing option |
| TSS | Trunk Servicing System |

| | |
|---|---|
| TTL | transistor-transistor logic |
| TTY | teletypewriter |
| TU | traffic unit |
| TUR | traffic usage recorder |
| U | usage |
| UA | unassigned |
| UE | unequipped |
| UPCO | usage, peg count, and overflow |
| USG | usage |
| USITA | United States Independent Telephone Association |
| VM/CMS | Virtual Machine/Conversational Monitor System |
| VSAM | virtual sequential access method |
| VSS | Voice Storage System |
| WBP | wallboard process |
| WCPC | wire center planning center |
| WMAR | weekly machine administrative report |
| XPRT | exception print process |
| YACC | yet another compiler compiler |

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 October 1983 Number 8, Part 1

# High-Power Lasers and Optical Waveguides for Robotic Material-Processing Applications

By CHINLON LIN,* G. BENI,* S. HACKWOOD,* and T. J. BRIDGES*

(Manuscript received April 4, 1983)

For various material-processing applications with robots we propose the use of high-power continuous wave and pulsed lasers ($Nd^{3+}$:YAG, Argon ion, $CO_2$, excimer, etc.) and optical waveguides for delivering high powers in the ultraviolet (UV), the visible, and the infrared (IR) regions. We discuss the use of low-loss silica glass fiber waveguides for delivering high-power laser beam in the UV to near-IR spectral region (0.3 to 2 $\mu$m), and the use of a waveguiding articulating arm for delivering high-power laser beam in the long IR (2 to 10 $\mu$m). We also describe a design for fitting a $CO_2$ laser waveguiding arm to the robotic arm, as well as the advantages of using optical waveguides for high-power laser delivery to robots for material processing.

## I. INTRODUCTION

Optical waveguides are known to be useful for optical signal transmission in which low-power, modulated semiconductor injection laser

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

light is used for lightwave communication applications.[1] The advent of low-loss optical fiber waveguides, for example, has made possible long-distance, high-bandwidth lightwave communication systems for transmitting audio, data, and video signals. This paper discusses the use of optical waveguides for a different application: *high-power laser transmission* for robotic material-processing applications. Using high-power continuous wave (cw) and pulsed lasers and appropriate optical waveguides for the ultraviolet (UV), the visible, the near infrared (IR), and the longer IR, a robot can manipulate the output beam of a variety of high-power lasers for various processing functions. In Section II we discuss the available flexible waveguides for high-power laser transmission. In Section III we describe a manually operated $CO_2$ laser waveguiding articulating arm, and in Section IV a design for fitting the waveguiding articulating $CO_2$ laser arm to a robot arm.

At present, automation of material processing using high-power lasers requires costly, dedicated, large-size equipment. We believe that an inexpensive, small-size robot controlling a high-power laser beam with the help of optical waveguides will make possible many new applications in material processing.

## II. HIGH-POWER LASER TRANSMISSION IN OPTICAL FIBER WAVEGUIDES

While the use of high-power lasers for material processing is well known,[2,3] the use of low-loss optical waveguides for high-optical-power transmission is not widely practiced.[4] For robotic applications (applications requiring the unique dexterity and versatility of robots), it is essential that the combination of high-power laser technology and robotics does not reduce the dexterity or flexibility of the robots. The essential element here for providing the flexible link between the high-power lasers (usually heavy and bulky) and the robots is the optical waveguide.

Figure 1 illustrates the basic system schematic for using high-power lasers and optical waveguides in robotic material-processing applications. Depending on the type of high-power lasers, different optical waveguides can be used. For example, in the near-infrared region of 1 to 2 $\mu$m, e.g., for high-power Nd:YAG lasers at 1.06 $\mu$m, silica glass fibers have excellent transmission characteristics (see Fig. 2). As a result of advances in lightwave communications technology, the loss in a silica fiber waveguide can be very low ($\sim$1 dB/km, or 0.01 dB/10 m at 1.06 $\mu$m). In this case the loss due to coupling into and out of the fiber waveguide is much larger than the transmission loss for even a 1-km-long optical fiber. Losses in silica glass fibers also can be low enough (for 10 to 100m lengths) for guiding blue-green and red lasers; thus such silica fibers are useful for transmission of high-power Argon

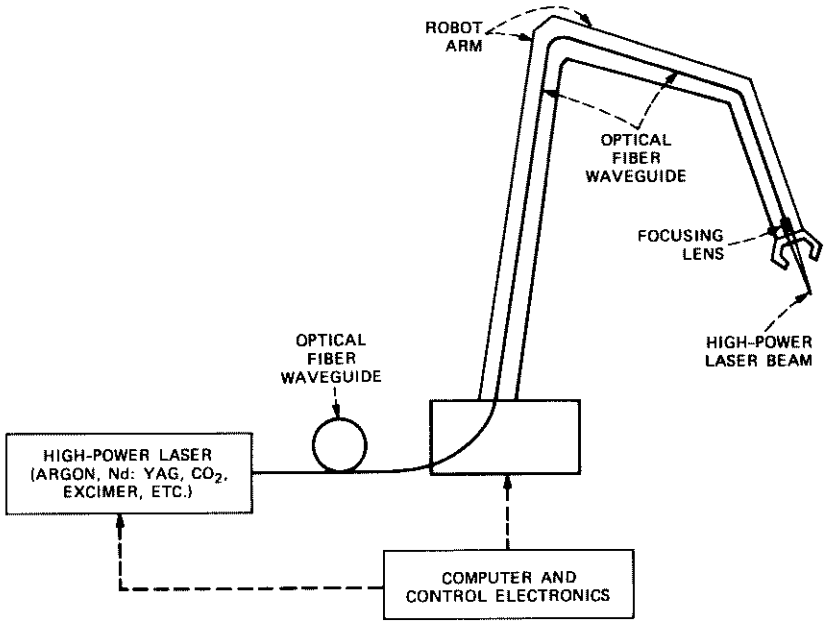



Fig. 1—Schematic of a system using flexible optical fiber waveguides for delivering high-power laser radiation to the robotic arm/hand for various material-processing applications.

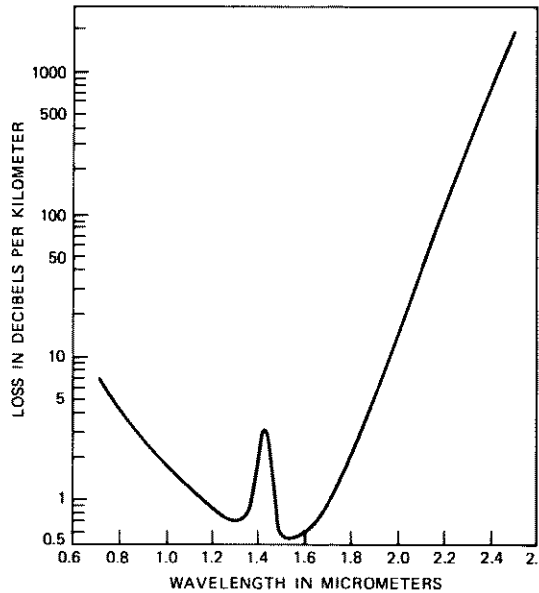



Fig. 2—Loss spectra of a typical low-loss silica glass fiber waveguide. For practical robotic material-processing applications, fiber loss of 1 dB/10 m (or 100 dB/km) could be considered low loss.

ion and Krypton ion lasers, as well as high-power ruby and alexandrite lasers. Recently available special UV silica glass fibers* may also be used for the ultraviolet wavelength region (0.3 to 0.4 $\mu$m). The loss is about 1 to 2 dB for every 10m, which is still low. Such fibers are useful for transmitting UV lasers (e.g., He-Cd lasers and excimer lasers) in, for example, photochemical applications.

Thus we have available appropriate optical fiber waveguides for transmitting high-power laser radiation in the spectral region from UV to near IR through at least 5 $\sim$ 10 meters. This allows the bulky high-power laser head and its high-energy power supply (and cooling system, if any) to be separated from the robot, while allowing the powerful laser beam to be delivered to the robot arm or fingertip. As we saw in Fig. 1 a high-power Nd:YAG laser and silica glass fiberguide could be used for guiding the laser radiation to the robot hand (gripper). The silica glass fiber can be routed inside the robotic arm assembly, or mounted externally but attached to the side of the arm, depending on the situation or work requirement. The output fiber end can have a microlens (such as a half-pitch graded-index-rod lens) or a small conventional lens attached for output beam focusing.

For transmitting Nd:YAG lasers, ruby lasers, Argon ion lasers (in the visible and the near-infrared spectral region), the silica glass fibers are typically very small in dimension: outer diameters are on the order of a few hundred micrometers to a few millimeters, including the protecting jacket or cable. For high-laser-power output with well-defined spatial distribution (e.g., for maximum brightness, or best focusing), single-mode fibers with appropriate refractive index difference $\Delta n$ and core diameter 2a can be designed (with normalized frequency $V \leqslant 2.4$ at the laser wavelength) for use in these different wavelength regions. If maximum overall energy transmission without concern for the spatial quality of the laser beam output is desired, a large-core, high numerical aperture (N.A.) silica glass fiber can be used for high-energy delivery to the robot. For the propagation properties and design considerations in single-mode and multimode silica glass fibers, appropriate references[1] should be consulted.

For transmitting high-power, longer infrared (2 to 10 $\mu$m) lasers such as $CO_2$ lasers, a configuration similar to that shown in Fig. 1 can be used, if a truly flexible $CO_2$ laser fiberguide is available. Presently, various glass and crystal fibers are being developed for this spectral region.[5] Notably among them are the polycrystalline KRS-5 fibers[6] and the single-crystal AgBr fibers[7] for $CO_2$ lasers transmission. However, presently available long infrared fibers tend to be very lossy and

* UV fibers with losses in the 150 dB/km range for $\lambda \sim 310$ nm range have been reported by, for example, Quartz and Silice, France.

fragile. Thus the mechanical and optical properties are not yet truly satisfactory. Therefore, at present, bulky conventional articulating arms (consisting of aligned mirrors) are used for most applications requiring some flexibility in $CO_2$ laser delivery. To improve the flexibility and stability, Bridges and Strnad have developed a novel "waveguiding" articulating arm for transmitting high-power $CO_2$ laser radiation.[8] The arm, shown in Fig. 3, has been designed for manual control. It is compact and relatively articulate. In the future, truly

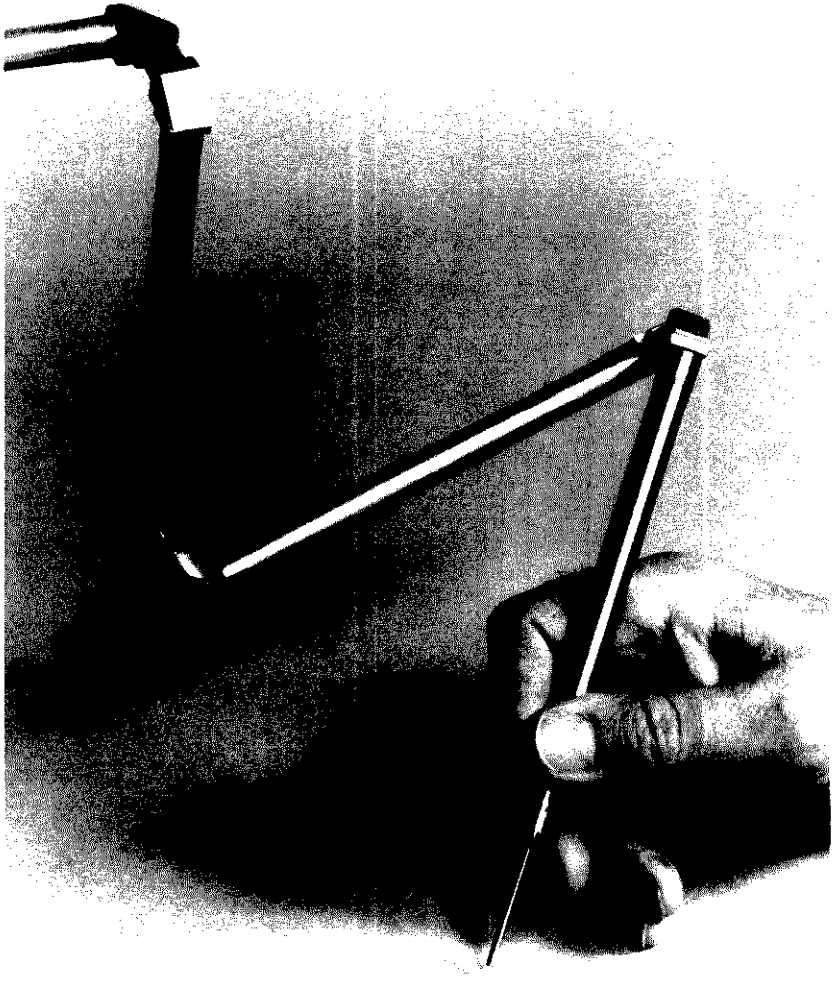


Fig. 3—The Bridges/Strnad waveguiding articulating arm for high-power $CO_2$ laser delivery.

flexible long-wavelength fibers are expected to have lower loss and higher strength than those presently available. Until then, the Bridges/Strnad waveguiding articulating arm would be the choice for $CO_2$ laser delivery to the robot. In Section III we discuss in more detail the design of this $CO_2$ laser arm; in Section IV we describe designing this laser arm to fit onto a robot arm for material processing.

## III. BRIDGES/STRNAD WAVEGUIDING ARTICULATING ARM FOR LONG-IR LASER RADIATION

Figure 4 shows the design details of the Bridges/Strnad arm used for manual operation (see Fig. 3). This articulating arm uses the principles of waveguiding in hollow dielectric tubes. This new arm has

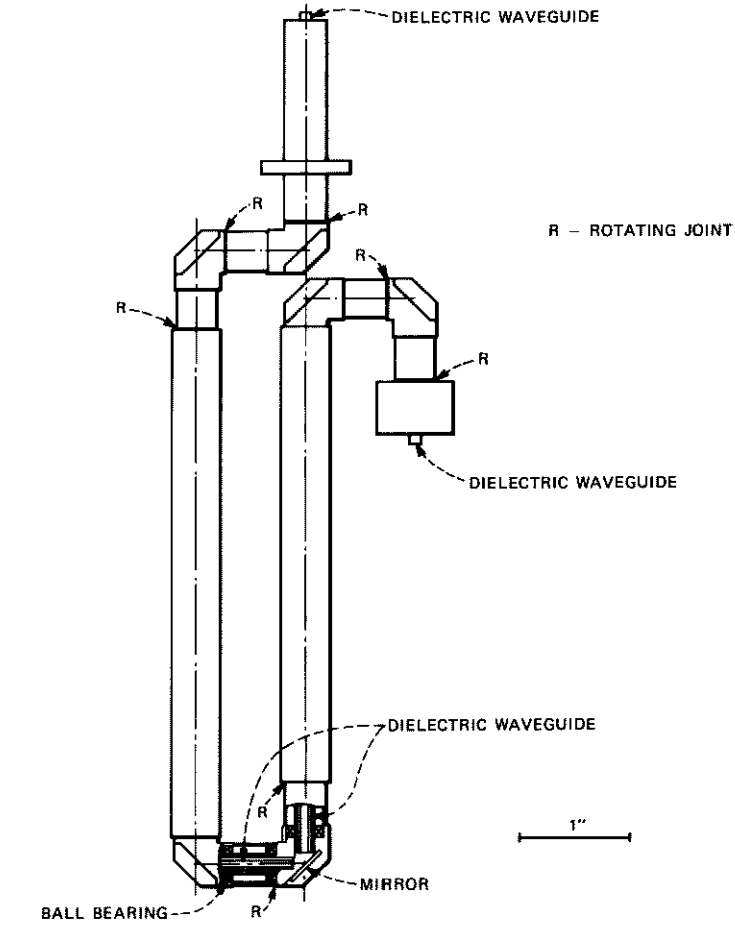



Fig. 4—The design of the Bridges/Strnad waveguiding articulating laser arm for flexible delivery of high-power, long-IR (e.g., $CO_2$ laser at 10.6 $\mu$m) optical radiation.

a number of advantages over previous articulating arms, including compactness and better pointing accuracy when compared with conventional articulating arms.[9] Flexible waveguides such as metal waveguides[10] and presently available infrared fibers are problematic because of the multimode nature of the guide. Single-mode radiation from the laser is rapidly degraded into a multiple-mode pattern that changes in form as the guide is moved. The degradation reduces considerably the maximum intensity that can be obtained by focusing the output radiation. In the case of articulating arms of conventional design the single mode is preserved, but unless the input beam is launched precisely on axis and the mechanism of the arm is precisely correct, the output beam will wander in a complicated manner as the arm is manipulated. Such arms are also characteristically large and cumbersome. The Bridges/Strnad arm design avoids this problem by propagating the radiation in straight, hollow, dielectric waveguides of the Marcatili-Schmeltzer type.[11] A single mode can be maintained in the guide, while the pointing accuracy is far less affected by initial launch conditions and accuracy of construction. (Pointing accuracy is determined by how closely the direction of the output beam conforms to the mechanical axis of the arm.) A further advantage is the compact design resulting from the elimination of diffraction spreading of the beam, by the guiding action of the waveguide.

The Marcatili-Schmeltzer waveguide carries radiation in the hollow circular bore of a dielectric tube. The dielectric need not be transparent to the radiation being guided. The mechanism of guiding can be thought of as a continual-glancing-angle Fresnel reflection from the dielectric walls. This reflection is not total, but close to 100 percent for very shallow incident angles to the walls. The modes of propagation have been calculated by Marcatili and Schmeltzer,[11] and they find that the lowest loss mode is the $EH_{11}$ mode. An appropriate waveguide size is 50 to 200 wavelengths in diameter. This size is large enough to give low loss, but still retain adequate guiding so that straightness of the tube is not an important factor, although curvature of the tube axis introduces extra loss by an amount that increases with tube diameter.

Since the dielectric need not be transparent to the radiation, glass or quartz tubing which is readily obtainable in precision bore form can be used to transport 10.6-$\mu$m radiation. Single-mode laser radiation is conveniently launched into the waveguide by means of a lens (Fig. 5). The focal length of the lens is chosen to closely match the input Gaussian beam to the guided beam with small loss.[12] Short gaps in the tube can be tolerated with small loss so that mirrors which turn the beam through a 90-degree angle and are basic to the operation of the infrared articulating arm can be used in a simple arrangement (see Fig. 4).

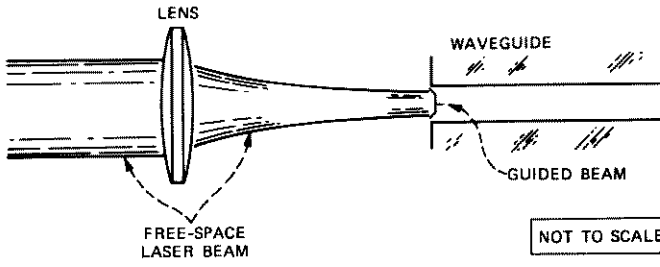



Fig. 5—Launching a free-space Gaussian beam into waveguide by means of a lens.

As a secondary feature, the glass or other visibly transparent waveguide tube can act as a light pipe to carry visible light through the arm. This light can be used for illuminating the work area, or for aiming the output beam. For this application the mirrors should be highly reflective in the visible as well as the infrared range. A suitable material is evaporated silver.

With the above concept in mind, various components of a possible waveguide articulating arm were tested in the laboratory, using a 10-$\mu$m $CO_2$ laser as a source. A 13.9-cm length of fused quartz tubing with 1.55-mm bore was tested. When a 30-cm focal length lens was used to focus the radiation into the guide, a transmission of 93 percent was found. To test the effect of small misalignments, the tube was pivoted off axis around the input point by one-half of one degree, and the transmission dropped by only 2 percent. Finally, a mock-up of a corner elbow (see Fig. 4) was made on the bench and a transmission of 95 percent was measured. This information demonstrated the feasibility of the idea and a complete arm was designed and fabricated (see Fig. 3). The arm contains three sections of waveguide that are 13 cm long and three more that are 2 cm long. The six corner mirrors used were commercially obtained. They were made from silicon 1 mm thick and were coated with silver and a transparent protective layer. The corners swivel on precision ball bearings. With a total length of 40 cm the arm can access any point in a 80-cm-diameter sphere. The completed arm was tested and found to have a transmission of 80 percent. Power up to 5W cw was transmitted with no damage. The 1.55-mm diameter beam from the output tube was substantially single mode and could be focused to a near-diffraction limited spot. As we expected, there was no wander of the output beam relative to the output tube as the arm was moved. The small size and light weight made it very easy to manipulate the arm and to place the output beam in any desired position.

## IV. A ROBOTIC ARM FITTED WITH THE WAVEGUIDING ARTICULATING $CO_2$ LASER ARM

The simplistic approach to using the Bridges/Strnad waveguiding $CO_2$ laser arm is to make the robot gripper hold and maneuver the tip of the articulating laser waveguide. This approach, however, has a major drawback. Reorientation of the laser-beam output requires, in general, rotations of all five revolute joints of the articulated waveguide. In many cases, this complex reconfiguration of the articulated waveguide prevents a continuous rotation of the laser-output tip and requires the robot to follow a complicated path.

In addition, a force and torque sensor on the gripper would be essential to ensure that the articulated waveguide is not damaged by the robot in the attempt of imposing a particular five-link configuration. This is still beyond state of the art robotics, since even the turning of a simple two-link crank by a robot arm is a complex compliance problem not yet satisfactorily solved.

A second approach is to fit the waveguide within or beside the robot. Because of the required 90-degree revolute joint articulations, this is not a trivial task. In fact, many existing robots have prismatic joints and/or unsuitable dimensions.

We now propose a new robot system consisting of two arms: master and slave. The master arm is positioned by motors, whereas the slave arm only carries the waveguide. The slave arm is the Bridges-Strnad-type five-link waveguiding laser arm with 90-degree rotational joints. Unlike the original Bridges-Strnad arm shown in Fig. 2, it now has nine (rather than six) mirrors and a different link geometry as described below. The master arm is, for example, a Microbot Alpha* whose hand gripper and side casing have been removed. The robot has a repeatability of $\sim 250$ $\mu$m and a positioning speed of 50 cm/s. The two arms are connected "in parallel" as follows.

Figure 6 shows schematically the connection between the master and the slave arms. The mirrors of the slave arm are labeled 'b' to 'j'. Laser input and output are at 'a' and 'k', respectively. Five mirrors are rigid and four ('b', 'e', 'g', and 'h') are movable. The axes of rotation of the master arm are indicated by rotation angles $\theta_1$ to $\theta_5$. Except for axis 5, the axes of rotation of the slave arm coincide with the corresponding axes of rotation of the master arm. For example, a rotation $\theta_3$ of the master arm corresponds to an equal angle rotation of the slave arm about the direction 'f'–'g'. The slave arm direction 'i'–'j' does not coincide with, but is parallel to, $\theta_5$ of the master arm. The connection between these two axes is through a pair of identical gears, as shown in Fig. 7. The connection between the two arms at the other

* New industrial-quality product of Microbot, Inc.

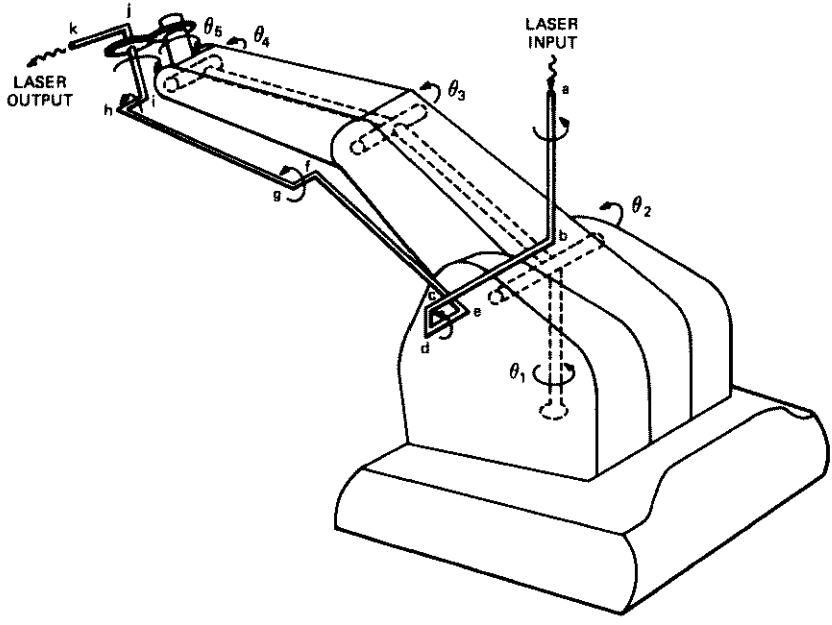



Fig. 6—Schematic diagram showing the master slave configuration of the $CO_2$ laser beam positioning robot.

four axes of rotation is via rigid mounts (not shown in Fig. 6) except for axis 3, where a moderately compliant plastic mount is used for attaching the two arms. This connection compensates for possible slight misalignments between the first four pairs of axes and thus prevents damage to the slave arm.

A precise description of the two-arm assembly is conveniently done using Denavit-Hartenberg[13] notation, which is standard for robots. The five links of the slave arm are defined as follows. The origin is at the intersection between axes $\theta_1$ and $\theta_2$. Link 1 is segment 'bcde'; Link 2 is segment 'defg'; Link 3 is segment 'fghi'; Link 4 is segment 'hij'; and Link 5 is segment 'ijk'. The exact geometry is given in Table I, where $\alpha_i$ is the twist angle, $a_i$ is the $i$th link length, and $d_i$ is the (i-1) to $i$th link distance. These definitions correspond to the conditions: 1) $a_2 = a_3 =$ '*hg*' = '*ef*'; 2) '*hi*' = '*gf*' = '*de*' = $2d_4$; 3) '*ab*' '*cd*', '*ij*', '*jk*' have arbitrary length.

The transmitted power efficiency should remain high. Extrapolating from the 6-mirror configuration, approximately 70-percent efficiency is expected. The resolution is determined by the master arm. In our case it is approximately 250 $\mu$m. Note that the slave arm is detachable so that the robot can be used for other tasks.

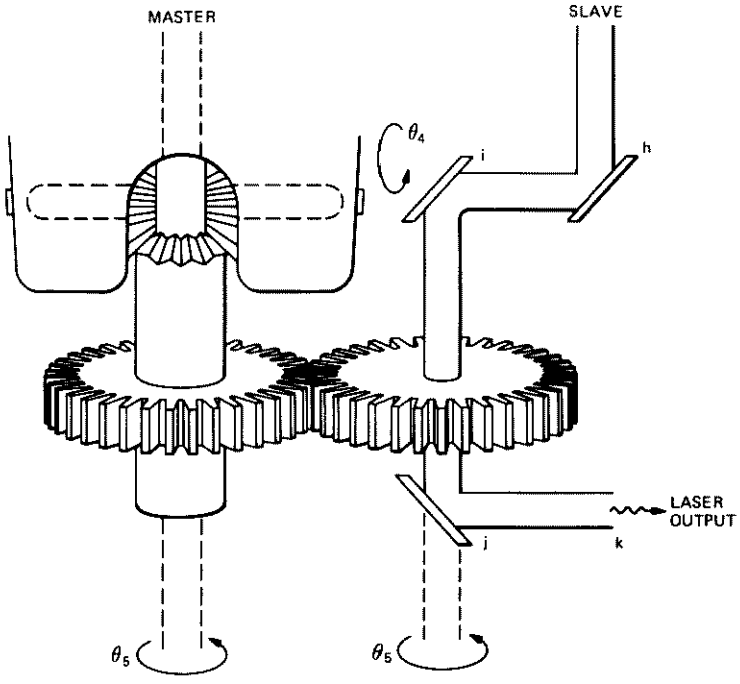



Fig. 7—Detail of the gear-connection of the 4th and 5th revolute axes of the master arm with the 'h-i' and 'i-j' revolute axes of the slave arm.

Table I—Denavit-Hartenberg manipulator parameters for the two-arm robot system

| Link | Master | | | Slave | | |
|---|---|---|---|---|---|---|
| | Twist Angle, $\alpha_i$, in degrees | Link Length, $a_i$, in cm | Link Distance, $d_i$, in cm | Twist Angle, $\alpha_i$, in degrees | Link Length, $a_i$, in cm | Link Distance, $d_i$, in cm |
| 1 | −90 | 0 | 0 | −90 | 0 | 10.0 |
| 2 | 0 | 17.78 | 0 | 0 | 17.78 | 0 |
| 3 | 0 | 17.78 | 0 | 0 | 17.78 | 0 |
| 4 | +90 | 0 | 0 | +90 | 0 | 1.27 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |

## V. ROBOTIC MATERIAL PROCESSING

The use of high-power Nd:YAG lasers and $CO_2$ lasers for material processing such as welding, cutting, drilling, scribing, trimming, heat treating, annealing, etc., are well-documented.[2] Thermally and photochemically induced reactions are also well known. The advantages of robot-laser-processing of materials are dexterity in robotic-laser-beam maneuvering, processing of complex-shaped materials, and versatility in adapting the changing environments and changing material-

processing functions. Existing nonrobotic, dedicated laser material processing apparatus[2,3] is much more restricted and expensive to modify should work requirement change. The combination of robots and high-power lasers is a natural technological direction to pursue for more versatile material handling and processing. The various forms of optical waveguides we describe here provide the important, and maybe indispensible, flexible links between robots and high-power lasers.

The use of these flexible, lightweight optical fiber waveguides (assuming they will also be available at long IR in the near future) for delivering high-power laser radiation to robots for material processing has several distinct advantages:

1. The bulky, heavy laser system could be remotely located so that any high electromagnetic interference (noise interfering with computer signal control of the robot and data transmission) could be eliminated.
2. The use of lightweight flexible optical fiber waveguides on the robot arm (or body) allows laser-material processing in mobile robots without undue constraints on their mobility.
3. Since the work space is not crowded by the use of high-power lasers, multiple robots can work together simultaneously in a complicated laser-material-processing task.
4. The use of several different kinds of high-power lasers at different wavelengths in a single robot can be achieved easily by routing multiple waveguides of different types through the robot, with appropriate shutters to control the switching of laser beams.

Our preliminary experimental results show that we can transmit (deliver) 5W of cw Nd:YAG laser power to the silica glass fiber output suitable for laser soldering. With high-power Nd:YAG lasers and more effort in fiber design and coupling, we expect to be able to deliver more than 10W (cw) through single-mode fibers and more than 25W (cw) through multimode, large-core silica glass fibers. Since the damage threshold for silica glass fibers is in the $GW/cm^2$ range, pulsed laser of high peak power also can be transmitted. With such lightweight optical fibers giving out such high-output laser power at the fingertip (gripper) of a robot, even a small inexpensive robot can perform many complicated material-processing or microprocessing functions.

With long IR lasers such as $CO_2$ lasers and the Bridges/Strnad type waveguiding articulating arm, 5W of power has been transmitted. Much higher-power (20 to 100W) transmission is expected before mirror damage occurs. With future advances in low-loss long IR fibers, truly flexible $CO_2$ laser transmission at 20- to 40W levels[6] can be expected.

The positioning resolution and repeatability in robotic laser material processing depend on the specific robot design. High positioning

precision (10-$\mu$m repeatability) and high positioning speed (1.6-m/s) robots[14] are fast becoming available. The use of lightweight flexible waveguides for high-power laser delivery to robot arm ensures that such high positioning accuracy and speed will not be compromised. This is another distinct, significant advantage.

In summary, the use of appropriate optical waveguides for transmitting and delivering high-power laser radiation to a robot arm will make possible complex robotic-laser-processing of materials. Medical and biological applications of robotic microprocessing with fiber-guided lasers can also be envisioned. These could be considered a special case of robotic material processing. The combination of technology of high-power lasers, optical waveguides, and robotics will certainly open up a new era of laser material processing.

## REFERENCES


1. See, for example, the review papers in Proc. IEEE (October 1980), special issue on Lightwave Communication Technology, and their references.
2. S. S. Charschan, *Lasers in Industry*, Western Electric Series, New York: Van Nostrand Reinhold, 1972.
3. J. F. Ready, *Industrial Applications of Lasers*, New York: John Wiley, 1980.
4. G. Nath, "Hand-held laser welding of metals using fiber optics," Optics and Laser Technology, *6*, No. 5 (October 1974), pp. 233–5.
5. T. Miyashita and T. Manabe, "Infrared optical fibers," IEEE Trans. Microwave Theory and Tech., *MTT-30*, No. 10 (October 1982), pp. 1420–38.
6. S. Sakuragi, "Polycrystalline KRS-5 infrared fibers for power transmission," in Advances in IR Fibers, Tech. Digest, SPIE, Los Angeles, CA., January 26–28, 1982, paper 320-02.
7. T. J. Bridges, J. S. Hasiak, and A. R. Strnad, "Single crystal AgBr infrared optical fibers," Opt. Lett., *5*, No. 3 (March 1980), pp. 85–6.
8. T. J. Bridges and A. R. Strnad, "Waveguiding Articulating Arm for Laser Radiation," Tech. Digest, 1st International Congress on Applications of Lasers and Electro-Optics, Boston, Massachusetts, September 20–23, 1982, paper 1C-3, p. 25.
9. D. R. Herriott, "Application of Laser Light," Scientific American, *219* (September 1968), pp. 141–56.
10. E. Garmire et al., "Low loss propagation and polarization rotation in twisted infrared metal waveguides," Appl. Phys. Letters, *34*, No. 1 (January 1979), pp. 35–7.
11. E. A. J. Marcatili and R. A. Schmeltzer, "Hollow Metallic and Dielectric Waveguides for Long Distance Optical Transmission and Lasers," B.S.T.J. *43* (July 1964), pp. 1783–809. U. S. Patent #3,386,043, "Dielectric waveguide, maser amplifier and oscillator," E. A. J. Marcatili and R. A. Schmeltzer; assigners to Bell Telephone Laboratories.
12. P. W. Smith, "A waveguide gas laser," Appl. Phys. Letters, *19*, No. 5 (September 1971), pp. 132–4.
13. See, for example, R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*, Boston: MIT Press, 1981.
14. See, for example, Robotics Today, *5*, No. 1 (February 1983).


## AUTHORS


**Chinlon Lin**, B.S.E.E., 1967, National Taiwan University; M.S., 1970, University of Illinois; Ph.D., 1973, University of California at Berkeley; Bell Laboratories, 1974—. Mr. Lin has worked on tunable dye lasers, short-pulse generation, nonlinear optics in fibers for frequency conversion, single-mode fiber dispersion and bandwidth studies, picosecond-injection-laser pulse generation, and high-speed optoelectronics. At Berkeley, he received an IBM

Fellowship and a Lankersheim Scholarship. He is currently in the Laser Science Research Department. Mr. Lin received an IEE (London) Electronics Letters Premium Paper Award in 1980 for a paper on zero-dispersion-wavelength tailoring in single-mode fibers. Senior Member, IEEE, Topical Advisor for Fiber and Integrated Optics for the Optical Society of America since 1982.

**Gerado Beni**, Laurea (Physics), 1970, University of Florence (Italy); Ph.D. (Physics), 1974, University of California, Los Angeles; Bell Laboratories, 1974—. Mr. Beni has been involved in research with Bell Labs, first at Murray Hill as a Resident Physicist in the Scattering & Low Energy Physics Department (1974–76), and as MTS in the Theoretical Physics Department (1976); at Holmdel he was a member of the Solid State Device Physics Department (1977–82), and currently is in the Robotics Systems Research Department. His expertise includes four major fields: theoretics, electrochemistry, displays, and robotics. In these fields he has contributed approximately 80 papers and 8 patents. Fellow, American Physics Society; Member, New York Academy of Science, the Electrochemical Society, SID, and Robotics International.

**Susan Hackwood**, Ph.D., 1979 (Solid State Electrochemistry), Leicester Polytechnic, United Kingdom; Bell Laboratories, 1980—. Ms. Hackwood has conducted research in catalysis, display technology, physical electrochemistry, sensors, and device robotics. Currently in the Electronics Research laboratory, she is involved with device robotics research for semiconductor fabrication. She has contributed approximately 40 technical papers and holds eight patents in the fields of physical electrochemistry, displays, and robotics. She is on the steering committee for Display Technology (ECS), which now includes robot sensors, and is the robotics representative of the Electronic Materials Committee of the Metallurgical Society of AIME. Member, American Physical Society, The Electrochemical Society, SID, and Robotics International.

**Thomas J. Bridges**, B.Sc. (Engineering), 1943, London University; Royal Naval Scientific Service (U. K.), 1944–1956, 1959–1963; Bell Laboratories, 1956–1958, 1963—; Massachusetts Institute of Technology, 1967–1968. At MIT Mr. Bridges was a Visiting Professor of Electrical Engineering. At Bell Laboratories he has worked on millimeter waves, microwaves, infrared lasers, and optically pumped far-infrared lasers. He currently supervises a group working on vapor phase epitaxy of semiconductor compounds.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Estimates of Path Loss and Radiated Power for UHF Mobile-Satellite Systems

By D. O. REUDINK*

(Manuscript received March 29, 1982)

This paper examines the satellite power requirements for land-mobile satellite systems, taking into account both shadow and multipath fading. Depending upon reliability objectives, present-day satellite capabilities will permit from 20 to 200 "(Advanced Mobile Phone Service) AMPS-like" circuits. Since satellite systems cost hundreds of millions of dollars, mobile-satellite telephony in a conventional sense would be expensive. Techniques are examined that in the far-term may permit a factor of 10 increase in capacity while still using moderate-size satellites.

## I. INTRODUCTION

For the next decade or more the new 900-MHz land-mobile systems using cellular concepts will be introduced only in large cities. Any kind of nationwide service with full coverage will take many years. However, there are many nonurban applications where telephone service would be highly desirable, including service to vehicles along the nation's interstate highways, to rural residences currently without means of obtaining wire-line service, and to aircraft.[1-3] Communication from a satellite to small, portable terminals has been achieved, primarily demonstrating technical feasibility.[4,5] Other studies examined system costs assuming satellite payloads much larger than current capability, and paying little attention to propagation effects.[6-8] Although the monetary costs looked rather favorable, no such satellites or launch

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

capabilities are expected in the near future. Here, the capacity-performance trade-offs are examined on realistic assumptions of near-term satellite capabilities, considering propagation conditions in some detail.

The issue of cost is not addressed except to mention that a modern, one-of-a-kind* satellite, in orbit, might be expected to cost well over 100 million dollars. At the same time the circuit capacity compared to a satellite system with large, fixed ground antennas will be smaller.

The procedure for the remainder of this paper is first to calculate an optimistic link budget based upon the assumption that there is a line-of-sight path between the satellite and the mobile system. Following this, estimates are made for the additional losses on the path due to obstructions near the mobile system; then an attempt is made to determine whether it is feasible to obtain sufficient radiated power in the satellite to make up these losses; finally, more spectrum- and power-efficient options are examined.

## II. LINE-OF-SIGHT LINK BUDGET

The radiated power requirements on the satellite resources will be determined by working backwards from the mobile system. Assume for now frequency-division multiple access with a minimum carrier-to-noise ratio (CNR) of 10 dB in a 20-kHz bandwidth. These numbers are roughly consistent with the FM threshold and bandwidth requirements for present-day cellular mobile systems and would correspond to something that might be feasible in terms of a digital system in the future. These numbers imply 10 log C/kT = 53 dBW/K/Hz.

Since the mobile system can be driven in any direction, it is necessary that the antenna be omnidirectional in the azimuthal plane; however, since the satellite is never directly overhead, it is possible to form a conical beam in the elevation plane, and obtain a gain of about 6 dB. Assume a system noise temperature of 400 degrees for the mobile system. Although lower-noise receivers certainly could be built, limitations of man-made noise will prevent the effective use of lower-noise receivers; furthermore, the additional cost of ultra-low-noise receivers in mobile systems may be prohibitive. These two assumptions imply a G/T = −20. With these numbers the required illumination on the earth is −134 dBW/$m^2$. The path loss from the satellite to the mobile system calculates to 184 dB. As shown in Table I, the required Effective Isotropic Radiated Power (EIRP) per channel is 28 dBW at the satellite.

* Multiple-satellite operation with frequency reuse would be practically impossible to achieve because the mobile antennas radiate essentially in an omnidirectional pattern, so there is no way to discriminate one satellite from another.

Table I—Link budget for a single mobile-satellite channel

| | |
|---|---|
| Minimum CNR at mobile | 10 dB |
| Noise bandwidth | 20 kHz |
| Mobile receiver noise temperature | 400°K |
| Antenna gain | 6 dB |
| G/T | −20 |
| Path loss | 184 dB |
| Required satellite EIRP | 28 dBW |

It is possible to have an antenna with nearly a 15-foot diameter within the Space Shuttle. At 900 MHz such an antenna would provide an on-axis gain of 31 dB, while the gain at the edge of the country would be about 29 dB. We assume on the average that 30 dB of antenna gain is available, which implies a required radiated power per channel of −2 dBW for Continental United States (CONUS) coverage. The average RF power consistent with moderate-size satellites in the 1980s time frame is about 200W (+23 dBW), which implies 316 satellite-mobile channels.

## III. ADDITIONAL PATH LOSSES

Excess path losses on land-mobile paths have been measured at numerous frequencies over a variety of paths worldwide. A great amount of data exists in the 800- to 900-MHz frequency band for land-mobile paths, but little data have been published on the losses over satellite-to-mobile paths. To estimate what losses might be expected, Fig. 1 shows a plot of the median path loss in excess of the free space path loss as measured for various base-station antenna heights,[9] plotted in terms of the elevation angle between the mobile and the base station. For distances of both 1 and 2 km from the base station, the points lie nearly on a straight line on semilog paper.

A recently published paper indicates that satellite path losses in the Denver area (elevation angle 32 degrees) range from 3 to 20 dB over line-of-sight.[10] The author's statistical description for excess path loss, corresponding to 50-percent large-scale coverage, estimates 9.8-dB excess path loss for a suburban environment with an elevation angle of 30 degrees. This value is in good agreement with the curve depicting a suburban environment plotted in Fig. 1. For more rural locations, satellite measurements would predict 5.3-dB excess path loss,[3] which tends to agree with the straight-line projection of the two data points taken 10 km distant from the high-elevation base stations. In all cases, due to the more favorable elevation angles, excess path losses are less severe on satellite-mobile paths compared to typical land-mobile paths, as long as the satellite is located at a longitude such that the slant range is not excessive.

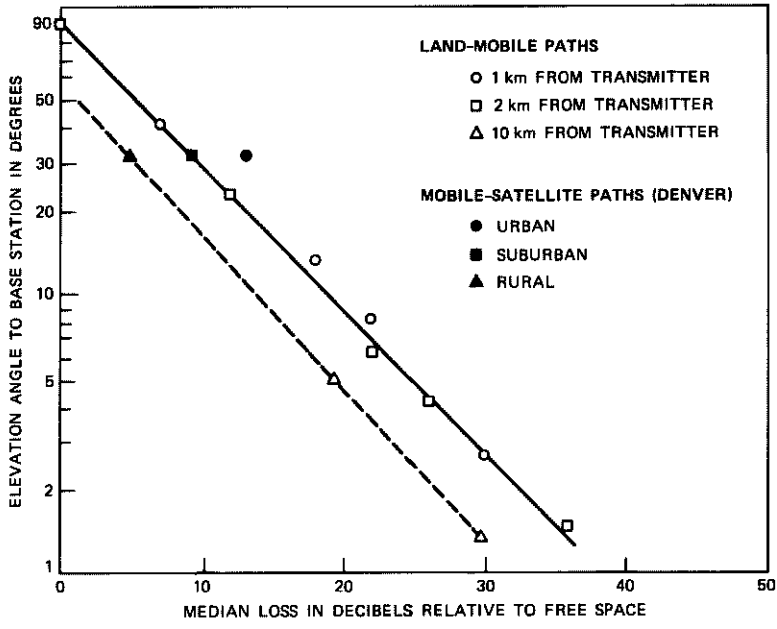



Fig. 1—Excess attenuation vs elevation angle at 900 MHz.

For a favorable satellite longitude and for most locations in the United States, the elevation angle to the satellite generally will be greater than 30 degrees and less than 60 degrees, indicating that it is reasonable to expect the median losses over free space to range from 3 to 10 dB, except possibly in urban areas where greater losses would be expected.

Data reported on land-mobile paths indicate that the distribution of the signal about the median is log normal with standard deviations ranging from 5 to 10 dB.[9] The log-normal distribution found in the land-mobile case arises from large-scale obstructions such as tall buildings and hills, which shadow the line-of-sight path. Intuitively, one might expect the variation of signal strength to be less severe on a satellite-mobile path, since typical elevation angles for a satellite are 30 degrees or more, where usual land-mobile paths have elevation angles more on the order of a degree or so. Based on the satellite data of Ref. 10, the variance of the log normal appears to be somewhat less for satellite paths than for land-mobile paths, but not dramatically so.

Figure 2 is a plot of what might be called the expected range of additional losses relative to free space based on available measured data.[10] It can be seen that high margins are required to provide service to approximately 99 percent of the regions of the country. Even assuming most regions of service interest are rural, margins in excess

of 10 dB are often required. Furthermore, this would assume the propagation characteristics of the entire country are similar to those in the Denver area. A far safer approach (and perhaps more accurate) is to employ the surburban model to represent the small cities and towns where the bulk of the demand may be expected. In this case, to cover 90 percent of locations would require a signal approximately 16 dB above the line-of-sight value, and the value would increase to 21 dB for 99-percent coverage.

## IV. MULTIPATH FADING

Another factor crucial to system performance is the effect of multipath fading. A line-of-sight component can be expected frequently on the satellite path. This component plus signal components, which scatter into the mobile antenna from nearby objects, produce a Rician signal distribution with significantly less fading than occurs with a Rayleigh distributed signal. Data on satellite paths confirm Rician-like signal statistics.[10] For example, level crossing rates can be an

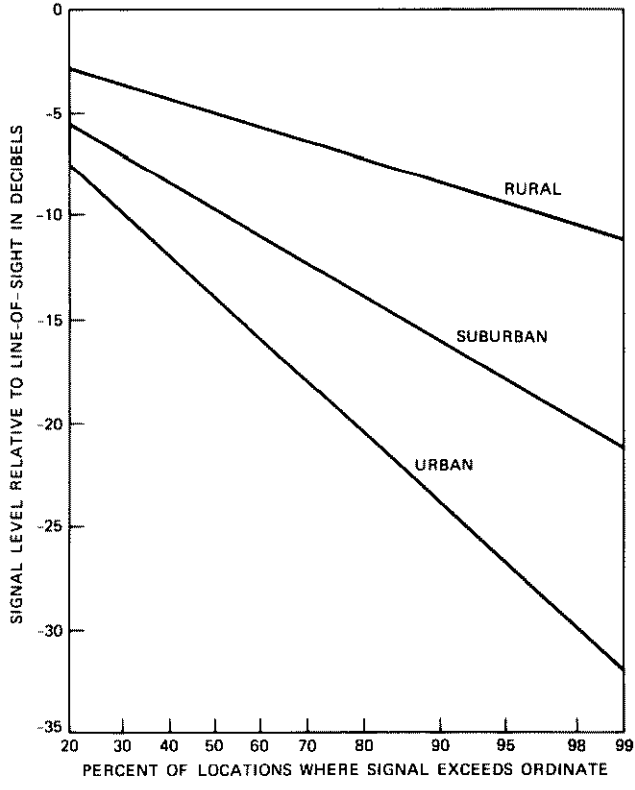



Fig. 2—Estimates of signal strength distribution on satellite-mobile paths.

order of magnitude or more lower compared to Rayleigh fading. Unfortunately, fading will be worst just at the wrong time—when the line-of-sight component is obscured and the received signal strength is low. Under these conditions the received signal can be assumed Rayleigh, and expected performance has been calculated.[11] In white Gaussian noise, a $10^{-3}$ Bit Error Rate (BER) is attained with an average energy per bit ($E_b$) 6.9 dB above the noise density ($N_0$) for biphase coherent phase shift keyed signals, while in the presence of Rayleigh fading an average $E_b/N_0$ of 24 dB is required. Spatially separated antennas whose signals are combined in phase can significantly reduce bit error rates. Figure 3 is a plot of BER vs $E_b/N_0$ for various numbers of diversity elements. For a $10^{-3}$ BER with three elements, $E_b/N_0$ per element drops to 7 dB, and with eight diversity elements it drops to 0 dB $E_b/N_0$. This somewhat surprising result is readily understood if one considers the eight elements as an antenna array, whose effective gain is 9 dB higher than a single element.

Space diversity works well on mobile systems, even with closely spaced elements (<1λ) that have highly correlated (0.5) signals.[9] However, space diversity cannot be achieved at the satellite because the arriving signal is essentially a plane wave, and extremely large separation of the satellite antennas would be required. A technique

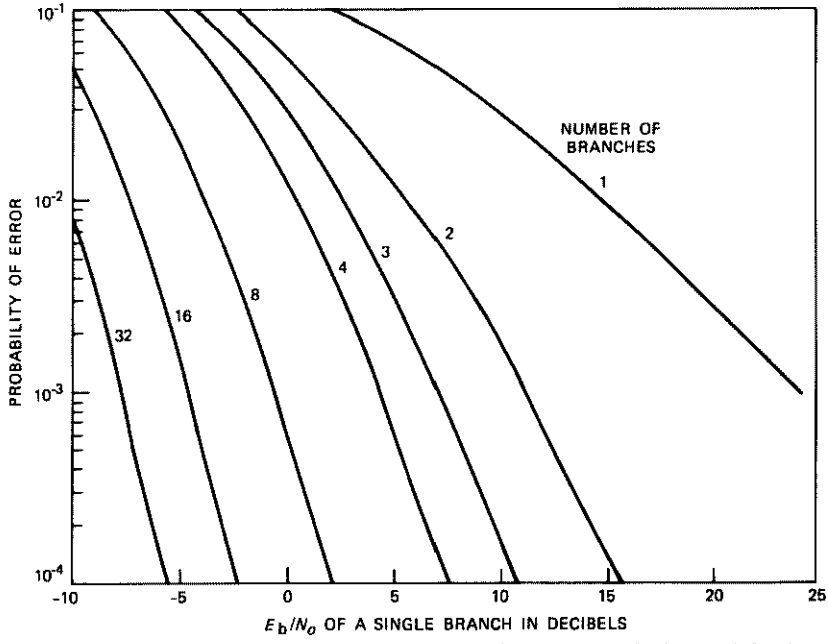



Fig. 3—Error probability of Two Coherent Phase Shift Keying (CPSK) with Rayleigh fading and diversity.

called retransmission diversity has been suggested for both analog[9] and digital transmission.[12] The idea is to transmit the conjugate phase of the received carrier on the same or a closely spaced carrier. By doing so, the signals from all mobile diversity branches will automatically arrive in phase at the satellite. In the original thinking for land-mobile system use, retransmissions would occur at the base stations, thereby simplifying the mobile system. In this instance, the only workable method is to place the retransmission apparatus at the mobile system. The analog scheme mentioned above would be very difficult to build, since the transmit and receive frequencies are closely spaced (<100 kHz), and the two signals differ in power by many tens of decibels. The digital techniques employ packet transmissions, and any hope for compatibility with cellular systems would be lost.

If the system were not constrained in bandwidth, then similar results could be obtained by employing frequency diversity channels. Beyond its obvious inefficiency, another drawback with frequency diversity is that the satellite must transmit the same signal on multiple channels, which requires more equipment and power. Another approach that achieves diversity advantage is to employ frequency hopping or spread-spectrum techniques.[13] This would eliminate the retransmission problem of space diversity but with possible reduction in capacity. An interesting possibility would be to combine space diversity with spread spectrum, using space-diversity mobile reception (because satellite power is at a premium) and spread-spectrum mobile transmission to combat Rayleigh fading.

Table II summarizes the per-channel power margins required for the various conditions of shadow and multipath fading, as discussed previously. We assume that a threshold of $10^{-3}$ BER is achieved with a calculated signal-to-noise ratio (s/n) = 10 dB. This allows 3.1 dB of implementation margin for filter and transmission line losses, antenna pointing errors, and nonideal detection equipment. The first observation from this table is that, without diversity, satellite-mobile com-

Table II—Power margin in excess of free space propagation, required to overcome shadow fading, and attain BER $\leq 10^{-3}$ in Rayleigh fading

| | Margin Above Line-of-Sight (dB) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Urban Environment | | | Suburban Environment | | | Rural Environment | | |
| Percent Coverage | No Diversity | 2 Branch Diversity | 8 Branch Diversity | No Diversity | 2 Branch Diversity | 8 Branch Diversity | No Diversity | 2 Branch Diversity | 8 Branch Diversity |
| 50% | 31 | 18 | 13 | 27 | 14 | 9 | 22 | 9 | 4 |
| 90% | 41 | 28 | 23 | 33 | 20 | 15 | 25.5 | 12.5 | 7.5 |
| 99% | 49 | 36 | 31 | 38 | 25 | 20 | 28 | 15 | 10 |

munications in dense urban areas is unachievable for the majority of locations. However, it is expected that cellular mobile systems will handle this traffic anyway. A margin of 15 dB with two-branch diversity would provide service to 99 percent of the rural locations, 60 percent of suburban locations, but only 25 percent of the urban locations.

Using Figs. 2 and 3, together with the link budget calculated in Table I, we can estimate quality of service for a given satellite EIRP. Table III gives the percentage of rural or suburban locations where performance exceeds the given BER for the indicated per-channel satellite transmitter power and for mobile systems with three diversity branches. Having only two diversity branches would increase power requirements by 3 to 5 dB, while having four diversity branches would lower the transmitter requirement by 2 to 3 dB, depending on the chosen threshold.

It is clear that typical satellite configurations are severely power limited when it comes to providing mobile services. Previously we saw that 316 circuits were available when all mobiles are line-of-sight. Permitting 15 dB of margin on each circuit reduces the capacity to only 10 circuits. Thus, ways of obtaining more EIRP must be found.

## V. ADDITIONAL SATELLITE EIRP

It appears obvious that an approximate 10-dB signal-strength margin will be required for any reasonable satellite-mobile system. In Section II, calculations showed that 200W of RF power were required for 316 channels based on line-of-sight propagation. If only 32 channels were used, then an additional 10 dB of radiated power per channel would be available. However, cost estimates in this paper's introduction indicate that this would be almost certainly a cost-ineffective approach. On the other hand, the state-of-the-art cannot provide 2 kW of RF power in a satellite today; thus, we look to other means of effecting higher EIRP or its equivalent.

The question of efficient multiple-channel satellite transmission is

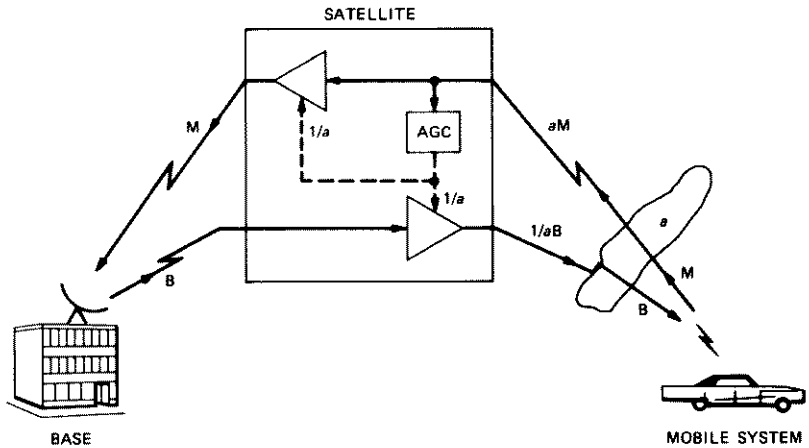
Table III—System performance for mobiles with three diversity branches

| Percentage of Locations With BER Less Than Value | Satellite EIRP/Channel | | | | | |
|---|---|---|---|---|---|---|
| | 33 dBW | | 38 dBW | | 43 dBW | |
| | Rural | Suburban | Rural | Suburban | Rural | Suburban |
| $10^{-1}$ | >99.9 | 85 | >99.9 | 92 | >99.9 | 98 |
| $10^{-2}$ | 96 | 45 | >99.9 | 82 | >99.9 | 91 |
| $10^{-3}$ | 50 | 18 | 97 | 50 | >99.9 | 78 |
| $10^{-4}$ | 12 | 6 | 82 | 35 | 99.7 | 65 |

very complex. Given a total payload mass for power amplifiers and solar cells and batteries, what techniques best satisfy the system requirements? When there are only a few channels, a single-amplifier-per-channel operation is usually the simplest. For a large number of channels, the hardware complexity of a multitude of amplifiers and of multiplexing these RF signals onto an antenna feed necessitates another approach. Multicarrier operation (for analog or digital signals) simplifies the satellite tremendously but at a cost of power efficiency and potential intermodulation distortion. Transmitting digitally multiplexed signals from the satellite eliminates intermodulation and allows efficient high-power class C amplification, but requires that all signals be of the same power and that mobiles have high-speed Time-Division Multiple Access (TDMA) receivers. These issues as well as other transmission-efficient techniques are addressed in the following subsections.

### *5.1 Power control*

We note that it is wasteful to provide all vehicles with the 10 dB or so margin to ensure that most of the vehicles have a signal above threshold. After all, some vehicles will be line-of-sight to the satellite and require substantially less power than those behind a mountain. Ideally, just enough power should be made available to ensure that each vehicle has a signal above threshold. This can be accomplished by using the technique illustrated in Fig. 4. The automatic gain-control signal is applied to a second transmitter carrying the message from the ground to the satellite. For illustration, the circuit is shown in the




AGC – AUTOMATIC GAIN CONTROL

Fig. 4—A technique to reduce shadow fading on mobile-satellite paths.

satellite, but it could be located at the earth station just as well if the satellite amplifier is linear. Because of the path delays, it could take a half-second or so to make a power adjustment. Therefore, the technique can be applied against slowly varying shadowing such has hills or large terrain features but not against multipath fading.

Recent work of Yeh and Schwarz allows us to calculate the total power expected from the sum of any number of log-normally distributed carriers.[14] Figure 5 contains plots that show the mean decibel value of a log-normal distribution that is derived from the sum of a number of log normals with the same mean (0 dB) and standard deviations ($\sigma$ = 2.5, 5, 7.5 and 10 dB); these $\sigma$'s correspond roughly to rural, suburban, urban, and dense urban environments. For example, the resultant of summing 100 carriers whose standard deviations are 5 dB is (approximately) a log normal whose mean is 22.5 dB. This means that providing a total peak RF power that is 22.5 dB above that of a single carrier would satisfy the power demand 50 percent of

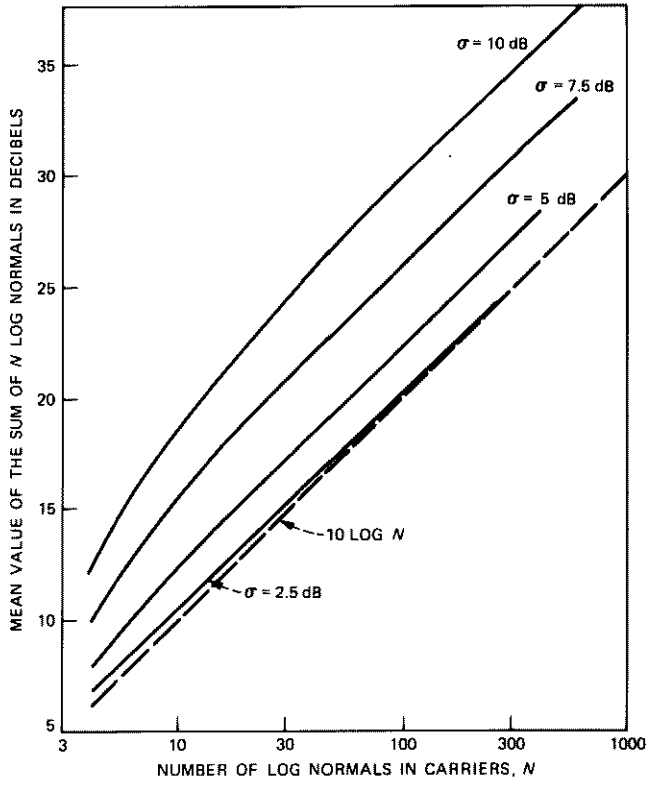



Fig. 5—Mean decibel values of log-normal distribution derived from log normals with same mean and standard deviations.

the time when 100 log-normal carriers are individually transmitted. Figure 6, similar to Fig. 5, is a plot of the mean plus twice the standard deviation of the resultant log normals when a number of identically distributed carriers are summed. Permitting an average power of the value shown in Fig. 6 reduces to 4.3 percent the time fraction that power is not available to meet demand. Compared with the previous example of 100 individually transmitted carriers with $\sigma = 5$ dB, the peak power must now be 24.5 dB above that for a single carrier, about a 2-dB increase.

To ensure 95.7-percent coverage assuming a standard deviation of 5 dB for a single log-normal carrier, requires a margin 10 dB ($2\sigma$) above its mean. Thus, it appears that transmitting each carrier with power just sufficient to overcome the path attenuation results in a power savings of about 5.5 dB compared to transmitting all signals with power 10 dB above the mean.

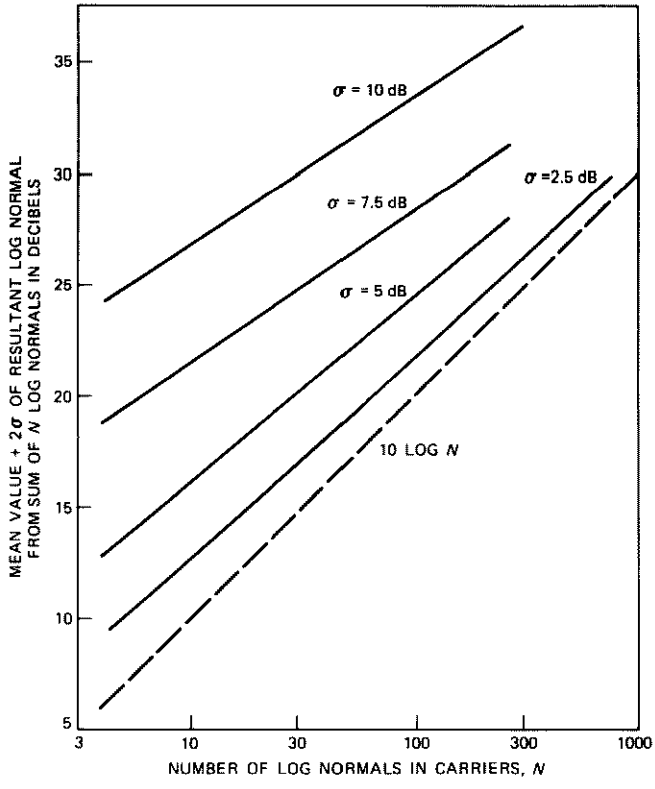



Fig. 6—Plot of mean plus twice the standard deviation of resultant log normals when identically distributed carriers are summed.

### 5.2 Multicarrier considerations

The calculations above imply that each carrier is transmitted separately. Usually it is far more efficient from the point-of-view of spacecraft hardware and complexity to combine signals and transmit them from a single high-power amplifier rather than employ many (hundreds) of low-power amplifiers. Since the signals add in voltage, the peak power requirement will be considerably above the average; and since amplifiers are not ideal, intermodulation results. Calculations of intermodulation distortion for the case of equal-amplitude but randomly phased signals for both ideal and typical power amplifiers have been made by Saleh.[15] Typical results are shown in Fig. 7. To achieve the minimum acceptable $C/(I + N)$ of 10 dB requires that the average power of an ideal amplifier be "backed off" 3 dB from its peak power and that any realistic amplifier be backed off 3.5 dB from its peak power point. For more typical operating conditions, $C/(I + N)>$ 15 dB, back-offs would be 5 to 7 dB, depending on the amplifier and compensation used. This implies that, even for a power amplifier which is 50 percent efficient at saturation when operated in the linear region, the dc-to-RF efficiency can only be about 12 percent. Still there can be an overall weight savings in the satellite compared to using individual amplifiers, but quantitative calculations are beyond the scope of these considerations.

The power back-off numbers cited are calculated for equal-amplitude carriers with random phases. For the mobile-satellite case with power control, the carriers have independent randomly distributed phases, but the amplitudes are log-normally distributed. Greenstein has reasoned that similar results should be expected for this case; however, the calculations remain to be done.*

### 5.3 Resource sharing and coding

Resource sharing has been suggested for TDMA systems as a means to increase a link margin by nearly 10 dB. The idea is to assign the user in a shadow fade a longer time slot, and encode the signal. For digital mobile-satellite service, this technique could be considered, but for analog modulation its implementation is less obvious. For resource sharing to be effective, it is necessary that the majority of mobiles not require the use of the additional resource of a coded signal. For a given system bandwidth, the cost of resource sharing is a reduction in

* Simulations of L. Greenstein and A. Saleh have shown that for sample cases with as few as 100 log-normal carriers with random phases, the resultant envelope tends to be Rayleigh distributed. Thus, at least over time periods where the amplitudes of the carriers, and thus the average power, can be considered fixed, the calculated results based on equal amplitude carriers should be usable.

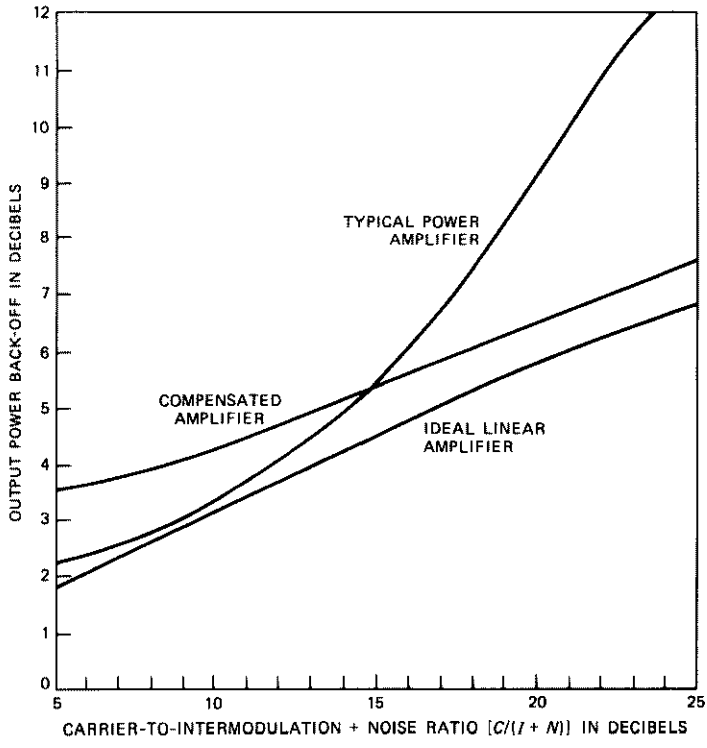



Fig. 7—Back-off required to achieve a given $C/(I + N)$.

transmission rate, but at an increase in margin. For 12/14 GHz fixed-satellite systems, the intent is to apply resource sharing only to those stations experiencing rain fading. Even using a rate 1/3 code (i.e., three transmitted bits per information bit) for these users, it was estimated that the loss in throughput is only a few percent, since very few users need resource sharing simultaneously.[16] As shown in Fig. 8, the throughput drops dramatically when the fraction of simultaneous users of resource sharing becomes significant. The curve in Fig. 8 assumes a fixed total system bandwidth and a rate 1/3 code for users of resource sharing. For example, when 10 percent of the users need resource sharing, the total number of users decreases by 17 percent while, if 50 percent of the users need resource sharing, the system capacity drops to 50 percent of the original value. To ensure that at any one time only a small percentage of the mobiles require resource sharing implies that the system normally operates with a margin somewhat above the median excess path loss. From Fig. 2 we see that (depending upon the degree of optimism), providing somewhere between 6 and 12 dB extra power over free space propagation would

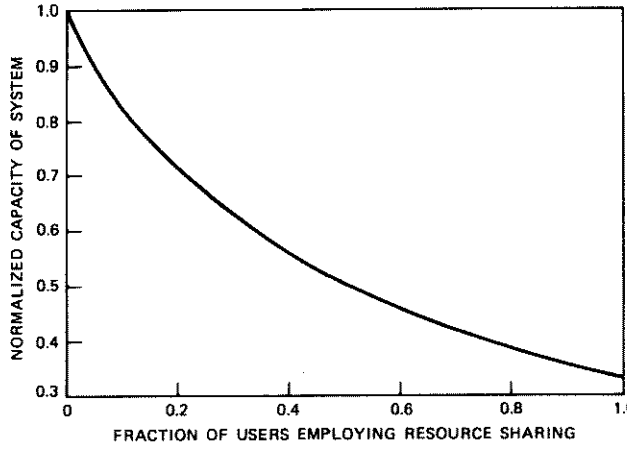



Fig. 8—Capacity loss with resource sharing using rate one-third code.

ensure that fewer than one third of the mobiles would require resource sharing at any given time. The additional margin obtainable from resource sharing then can extend the range of coverage from approximately two thirds of all locations to more than 90 percent. As in the case of power control, resource sharing cannot be applied instantaneously because of the time delays involved from the time measurement to the application of coding. Thus, the application is only for cases of slowly varying changes in signal strength.

The original resource sharing concept assumed that all users shared a single wideband channel. Implementation in a frequency-channelized system is less straightforward. However, if the system is not bandwidth limited, then all channels could use, say, a rate 1/3 code, thereby gaining an advantage of roughly 4 dB, plus or minus a decibel, depending upon the constraint length of the code and the particular implementation of the decoder.[17] Since the channel bandwidth is now three times wider, the mobile receiver is degraded by 5 dB, thus the large apparent gains of resource sharing cannot be realized.

### *5.4 Trade-offs between radiated power and antenna gain*

There are two ways to increase effective radiated power from a satellite. The first is simply to increase the transmitter power. The second is to increase the antenna gain. Although the techniques are equivalent in terms of the radiation to a point on earth, each implementation has different ramifications. Increasing radiated power is relatively straightforward in that the coverage area remains the same, and the burden on the satellite is to obtain more dc power through the use of more solar cells and to provide more battery power for eclipse

operation. Note that the eclipses only occur at night when, presumably, usage and power requirements would be greatly reduced.

Normally the satellite-mobile service would not be a broadcast mode, that is, it would not be necessary to talk to more than one mobile system over a single channel; therefore, the information to that mobile system can be confined to a small antenna beam. Thus, having a large number of high-gain beams covering the entire country becomes power efficient, since the power is only radiated in the beam intended for that mobile.[1,2,5] An additional benefit of spot beams is the possibility that the channels can be reused between areas that are spatially separated by a few beamwidths. In the case of conventional point-to-point telephony, demand is spatially nonuniform and highly peaked. If such is the case here, the advantages of reuse cannot be fully realized.

In a recent note it is proved for idealized constraints that maximum EIRP is obtained when the communications payload of the satellite is divided equally between the RF power subsystem and the antenna subsystem.[18] Sample calculations indicate that at frequencies above 1 GHz, multibeam antennas are more effective in increasing EIRP compared to using United States coverage antennas and high-power amplifiers. Earlier it was stated that a satellite providing 53-dBW EIRP (30-dB antenna gain, 200W RF power) could be achieved fairly conveniently. With battery backup either reduced or eliminated, 55 dBW should be attainable in an advanced state-of-the-art satellite using a United States coverage antenna.

Plotted in Fig. 9 are two curves that show EIRP as a function of payload mass. The lower curve is for a satellite with a United States coverage antenna where EIRP is increased only through increased transmitter power. The upper curve is the case where EIRP is maximized by dividing the payload equally between the antenna and the transmitter subsystems. At the point where a satellite with a United States coverage antenna provides 55 dBW, the maximum EIRP available is 58.2 dBW, assuming gain is achieved at 50 times isotropic per kg, and RF power is produced at 2.5 W/kg.* Under these conditions the antenna would weigh 73 kg and have a gain of 35.5 dB, implying three or four zone beam coverage of the United States, and a total RF power of approximately 180W would be transmitted.

Use of spot beams for EIRPs near 55 dB can generate an EIRP increase of about 3 dB. At 900 MHz the antenna diameter is already

* The units here are somewhat unusual, but for a given frequency, antenna gain is proportional to the antenna surface area that weighs so many kg/m$^2$. Likewise, power is derived from solar cells producing so many watts/kg. Thus, both antenna gain and RF power can be expressed as functions of mass.

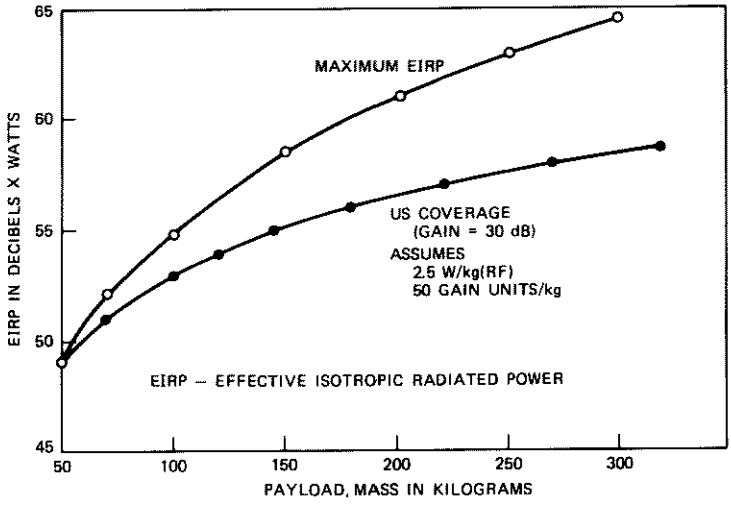



Fig. 9—Effective radiated power as a function of payload mass.

around 8m to provide a gain of 35.5 dB. For large payload satellites with EIRPs of 65 dBW, over 3 kW of RF power would have to be transmitted if United States coverage antennas were used, while use of a spot beam antenna with maximized EIRP permits 65 dBW of radiated power with one quarter the payload mass. Still, the required payload mass is 300 kg, twice the size of most current-day satellites.

### *5.5 Source coding and narrowband modulation*

Although obvious, it should be mentioned that employing narrower band channels improves the predetection CNR. Although modulations such as companded Single-Sideband (SSB) have been demonstrated, their performance in multipath environments will be degraded. Likewise, low-bit-rate voice coders (10 kb/s) may provide reasonable voice quality, but such coders are complex, and whether acceptable performance can be achieved with channel errors is not known. However, since the prospects for compatibility of satellite transmission with present-day land-mobile radio look dismal, the possibility of using other techniques to gain perhaps as much as 6 dB in link budget compared to analog FM need to be investigated further.

### *5.6 Summary of possible improvements in link budget*

Since it is envisioned that much more than a thousand channels are necessary for any practical system, the single-amplifier-per-channel approach is not feasible. The next most straightforward technique is to use multiple carriers on a single (or few) wideband channel. This

does not rule out compatibility with FM cellular systems, but major system problems remain.

Multicarrier operation should result in increased payload, and thus higher potential EIRP, but actual increases are difficult to calculate. Back-off can be eliminated if the downlink signals are digital and multiplexed onto a single carrier; however, downlink power control cannot be achieved under this condition. Mobiles would require TDMA receivers, and for bandwidths greater than 1 MHz, reception may be impaired by multipath propagation.

Table IV lists techniques that can potentially increase or decrease effective radiated power. For cellular-like service using a present-day satellite, techniques D, E, F, and G can be applied, yielding possible increases in EIRP from 0 to 8 dB. With SSB an additional 6 dB advantage may be possible (Item C). With digital modulation Item A is added and B replaces C, for a link-budget gain of 3 to 15 dB. With digital multiplexing, D and E are eliminated and H can replace A, for possible link-budget increases of 11 to 19 dB. Finally, using more diversity elements (Item I) helps the link budget significantly, especially at low BER.

## VI. CAPACITY CALCULATIONS

Satellite design is very complex, and no claim is made that actual satellites can be designed with the calculated capacities. Rather, the purpose here is to determine the effect in a general sense to some of the many options available. To that end, we make the simplifying assumption that changes in capacity are proportional to changes in effective radiated power. Thus when bandwidth is not a constraining factor, the number of circuits is determined by the simple relationship,

$$C = C_\beta 10^{\left(\frac{G_L - M}{10}\right)},$$

where $C_\beta$ is the baseline capacity calculated in on a line-of-sight basis

Table IV—Possible improvements in link budget

| Technique | dB Increase |
|---|---|
| A Channel coding (Sec. 5.3) | 2 to 4 |
| B Source coding (Sec 5.5) | 1 to 3 |
| C Modulation (receiver bandwidth) (Sec 5.5) | 3 to 6 |
| D Power control on each channel (Sec. 5.1) | 3 to 5 |
| E Back-off loss (Sec. 5.2) | −7 to −4 |
| F Maximize EIRP (Sec. 5.4) | 2 to 4 |
| G Reduced battery for eclipse (Sec. 5.4) | 2 to 3 |
| H Resource sharing (Sec. 5.3) | 6 to 9 |
| I Diversity elements (3 to 8) (compared to 2) (Sec. 4.0) | 5 to 8 |
| J Double-size payload (Sec. 5.4) | 3 to 6 |
| K Four-times payload (Sec. 5.4) | 6 to 12 |

in Section II, $G_L$ are the gains in the link budget discussed in Section V, and $M$ is the margin determined in Section III for a given grade of service and terrain type.

For example, assume a rather low grade of service with only a 7-dB margin. Combine this with the most optimistic link-budget gain of 19 dB. Then the possible number of circuits is $316 \times 10^{1.2} = 5008$, and a 300-MHz spectrum allocation would be required. More realistically, a 10-dB improvement in the link budget might be obtained for a specially designed satellite system not compatible with current cellular systems. On the other hand a 10-dB margin is almost essential for good service. Taken together the 10-dB improvement is offset by a 10-dB margin and the capacity calculates to the baseline value of about 300 circuits.

## VII. DISCUSSION AND CONCLUSIONS

It should be noted that use of diversity and tolerance of more bit errors can lower margin requirements significantly. For example, from Fig. 3 we see that, at $10^{-3}$ BER, going from two to four diversity elements reduces the margin requirement by 7 dB. Also, using two-branch diversity, but setting the system threshold at $10^{-2}$ BER instead of $10^{-3}$ BER, reduces the system margin by over 5 dB. The combination of the four-branch diversity at $10^{-2}$ BER threshold permits a 9-dB reduction in satellite EIRP compared to two-branch diversity and a $10^{-3}$ BER threshold. If this power savings could be directly traded for capacity, then eight times the number of circuits could be achieved. Downsizing the baseline calculations for low-bandwidth applications such as paging or emergency telephony is also possible.

As noted in Table IV, there is the potential of economy of scale. Satellite costs tend to run nearly linearly with weight[19], but EIRP can increase with the square of satellite mass;[18] and provided there are no bandwidth constraints, channel capacity can increase in direct proportion to EIRP. Thus, for satellites of twice the size (and at least twice the cost) four times as many circuits are obtained. In-orbit satellite mass as high as 5,000 kg are envisioned using the shuttle/Centaur. This represents a factor of 5 to 10 compared to present-day technology and suggests that future land-mobile service via satellite could become attractive. Trading power for capacity comes at the expense of bandwidth, a very precious commodity. On the other hand, terrestrial cellular systems will reuse frequencies hundreds of times nationwide. Making satellite-mobile systems spectrally efficient through the use of multibeam satellite antennas that can reuse frequencies is a tremendous technical challenge. Mile-diameter antennas are needed to get cell sizes comparable to terrestrial radio systems.

For service to aircraft, a 6-dB antenna gain for an aircraft in level flight seems reasonable, thus, line-of-sight capacity numbers apply

directly. For service to residences, an antenna gain of 16 dB is readily obtainable using a 1-m dish. Assuming there is a line-of-sight path and that there is sufficient bandwidth available, about 3000 circuits should be obtainable with present-day satellite capabilities, before power limitations become constraining.

Finally, it is safe to conclude that (1) cellular-compatible satellite-mobile systems are highly unlikely to be developed in the near future, (2) systems with modest-coverage objectives using enhanced-capability satellites and high-performance mobile sets look marginally attractive, and (3) very large satellites offer a possibility for mobile systems, in the long-term.

## VIII. ACKNOWLEDGMENT

The author wishes to thank Y. S. Yeh and S. Schwarz for providing guidance on log-normal distributions; to A. Saleh for many helpful discussions and information on power amplifier performance; and to L. J. Greenstein for critically reading this manuscript and providing his helpful insights on this problem.

## REFERENCES

1. G. H. Knouse and P. A. Castruccio, "The Concept of an Integrated Terrestrial/Land Mobile Satellite System," ICC 1980 Int. Conf. Commun. (June 1980), pp. 35.1.1–.5.
2. G. H. Knouse, "Terrestrial/Land Mobile Satellite Considerations, NASA Plans, and Critical Issues," IEEE Trans. Veh. Technol., *VT-29*, No. 4 (November 1980), pp. 370–4.
3. M. W. Cardullo and C. Dorian, "Mobile Communications via Satellite," Telecommun. J. *36*, (September 1969), pp. 426–36.
4. I. J. Jones et al., "A Study and Experiment Plan for Digital Mobile Communication Via Satellite," NASA Contract NASZ-9936, Report Number N81-1964.
5. R. E. Anderson et al., "Satellite-Aided Mobile Communications, Experiments, Applications and Prospects," IEEE Trans. Veh. Technol., *VT-30*, No. 2 (May 1981), pp. 54–61.
6. T. Iida, "Satellite Systems for Land Mobile Communications," J. Radio Res. Lab., *27*, Nos. 122/123, March-July 1980, pp. 97–115.
7. D. L. Wright and J. D. Kiesling, "Communication Satellite Services for Special Purpose Users," Proc. First Int. Telecommun. Exposition, Atlanta, *1*, (October 1977), pp. 200–8.
8. J. A. Weiss, "Low Cost Satellite Land Mobile Service for Nationwide Applications," 28th IEEE-VT Conf., Denver, CO, March 22–24, 1978, pp. 428–37.
9. W. C. Jakes, Jr., Ed., *Microwave Mobile Communications*, N. Y.: John Wiley, 1974.
10. G. C. Hess, "Land-Mobile Satellite Excess Path Loss Measurements," IEEE Trans. on VT, *VT-29*, No. 2 (May 1980), pp. 290–7.
11. M. Schwartz, W. A. Bennett, and S. Stein, "Communication Systems and Techniques," New York: McGraw Hill, 1966, Chapter 7.
12. B. Glance and P. S. Henry, "A New Approach to High-Capacity Digital Mobile Radio," B.S.T.J., *60*, No. 8 (Oct. 1981), pp. 1891–904.
13. D. J. Goodman, P. S. Henry, and V. K. Prabhu, "Frequency Hopped-Multilevel FSK for Mobile Radio," B.S.T.J., *59*, No. 7 (September 1980), pp. 1257–75.
14. Y. S. Yeh and S. Schwarz, "On the Distribution Function and Moments of Power Sums with Log-Normal Components," B.S.T.J., *61*, No. 7 (September 1982) 1441–62.
15. A. A. M. Saleh, "Intermodulation Analysis of FDMA Satellite Systems Employing Compensated and Uncompensated TWT's," IEEE Trans. Commun. *COM-30*, No. 5 (May 1982), pp. 1233–42.

16. A. S. Acampora, "Rain Margin Improvement Using Resource Sharing in 12 GHz Satellite Downlink," B.S.T.J. *60*, No. 2 (February 1981), pp. 167–92.
17. I. M. Jacobs, "Practical Applications of Coding," IEEE Trans. on I.T., *IT-20*, No. 3 (May 1974), pp. 305–10.
18. D. O. Reudink, "Maximum Power Transmission in Mass-Limited Radio Systems," IEEE COM SOC, February 1982.
19. D. H. Staelin and R. L. Harvey, "Future Large Broadband Switched Satellite Communication Networks," NASA Report NAS-5-25091, December 1979.

## AUTHOR

**Douglas O. Reudink,** B. A. (Physics), 1961, Linfield College; Ph.D. (Mathematics), 1965, Oregon State University; Bell Laboratories, 1965—. Mr. Reudink was initially involved in studies related to mobile telephony. He has headed the Satellite Systems Research Department and holds nearly two dozen patents in satellite and mobile communications, and mathematics. Mr. Reudink is currently Director of Radio Research at Crawford Hill Laboratory, Holmdel, New Jersey. Member, American Institute of Aeronautics and Astronautics (AIAA), Commission C of the U. S. National Committee for the International Union of Radio Science (URSI), USIA/Voice of America Radio Engineering Advisory Committee. Fellow, Institute for Electronic and Electrical Engineers (IEEE).

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Coding of Two-Level Pictures by Pattern Matching and Substitution

By O. JOHNSEN,* J. SEGEN,* and G. L. CASH*

(Manuscript received September 28, 1982)

A pattern matching approach is proposed for coding of two-level pictures. Patterns, which are either symbols such as characters, or fractions of black regions, such as line segments, are extracted from the facsimile. They are compared and matched to already transmitted patterns, called library patterns. If a correct match is detected, only the position of the pattern and the identification of the matching library pattern are transmitted. If a pattern does not match any library pattern, it is added to the library and its binary description is transmitted. Compared to conventional two-dimensional codes, the compression is often doubled and is sometimes 4.5 times higher. Compared to a symbol-matching coding technique,[2] the compression has increased by 20 to 80 percent, depending upon the document.

## I. INTRODUCTION

Conventional two-level picture coding techniques are based on the statistical dependence between neighboring picture elements (pels).[1] The calculation of entropies, according to a local source model, gives the maximum achievable bit rates. Run length or predictive coding techniques or a combination of them takes advantage of the statistical dependence between neighboring pels and leads to bit rates close to the entropy. Each exploits what can be called the microscopic (pel) properties of a facsimile.

Pattern-recognition coding techniques exploit macroscopic proper-

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

ties of the facsimiles. The image source is a source of patterns such as characters, lines, and black spaces. We can code the facsimile more efficiently, since the description is closer to the perceptual level. We can consider two kinds of pattern-recognition coding techniques. The first technique is pattern (or image) understanding. It recognizes a certain pattern, for example a letter, that possibly includes some font information. The second technique is pattern matching. Here, a pattern is not recognized, but is simply matched with already transmitted patterns, and if a correct match is detected, it is replaced by the matching pattern. It does not use the image-understanding level. The image-understanding approach has the potential advantage of a very high compression, but the often important aesthetic details of the documents can be lost, and there is a risk of errors at the present level of such techniques. The matching approach yields lower compression, but keeps more of the original pictorial information. There are also lower risks of errors, since matching allows only slight modifications in the pattern shapes. Naturally, neither of the pattern-recognition techniques is lossless, since they modify the picture content.

Ascher and Nagy[3] and Pratt et al.[2] have already proposed facsimile coding techniques using matching techniques. In the system presented here, not only the symbols, as in Pratt's case, but also graphical elements such as line segments and black regions are matched. The patterns are efficiently coded and updated, leading to significantly higher compressions.

## II. SYSTEM DESCRIPTION

Figure 1 shows the block diagram of the system. The pattern locator examines the facsimile line by line. When it locates a black pel, the pattern isolator picks up a pattern. The pattern is either a symbol (defined as a set of black pels completely surrounded by white pels) or, when no symbol can be extracted, a fraction of the black region. Therefore, contrary to Ref. 2, there is no residue to be coded, since all black pels belong to a pattern.

The matcher makes a template matching of the incoming pattern, with existing library patterns to determine whether the incoming pattern is similar to an already transmitted pattern. The system screens the library patterns to reduce the time-consuming template matching. Thus, we consider only the patterns that might match the incoming pattern. We screen by comparing features of the library patterns with those of the incoming pattern. We apply a very efficient and simple two-pass screening. If a correct match is detected, the matcher sends the information about the position of the pattern and its library identification to the coder. If no match has occurred, the incoming pattern is added to the pattern library. The pattern library

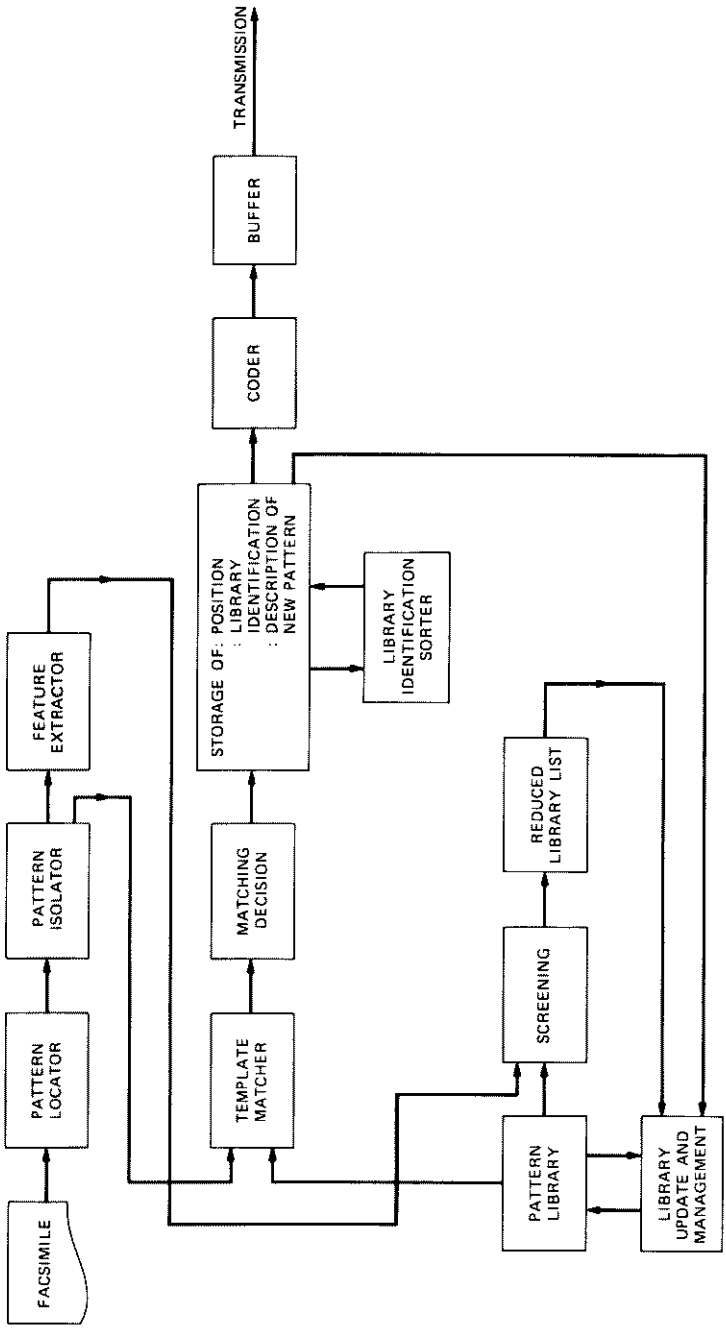



Fig. 1—Block diagram of facsimile coder by pattern matching.

is empty at the beginning of the coding and is gradually built up by the incoming library patterns. The matcher then also sends the information about the position and description of the new library patterns to the coder.

A library update and management unit takes care of the addition and deletion of library patterns and organizes them for the quickest possible match and most efficient coding. All the patterns isolated along one line are stored in the coder. When the end of the line is reached, we sort the patterns, which allows a more efficient coding.

## III. LOCATION AND ISOLATION OF PATTERNS

Patterns, in the present context, are the primitive elements of the coding process. They are isolated, and sent to the matching block sequentially, in a raster order. We distinguish two classes of patterns, relative to a square window of a predetermined size, $W$.

1. A symbol is defined as a connected region consisting of black pels and completely surrounded by white pels, such that it can completely fit into the window.

2. A nonsymbol is defined as a windowed portion of a black connected region that is larger than the window.

Usually, characters and small graphics elements can be represented as symbols, while lines and larger figures can be decomposed into nonsymbols. The decomposed figures can be later reconstructed by taking the union of the nonsymbols. The nonsymbols do not have to be disjoint, and a better compression may sometimes result from a decomposition into overlapping symbols.

Decomposing large figures into nonsymbols allows us to use matching techniques to compress graphical information, as well as text. A figure can be decomposed in many ways, and the compression that results from grouping similar nonsymbols usually depends on the decomposition. The final compression, or the number of different classes of nonsymbols, can be used as a measure of quality of the decomposition, and one may try to find the best decomposition in respect to such measures. Finding the optimal decomposition, however, may be computationally quite complex (we do not know of any related study) and it would certainly require many passes through a figure. At present, we use a one-pass isolation procedure, which allows us to keep the computation within reasonable bounds.

The isolation procedure repeatedly isolates and removes the upper-left portion of a black region, up to a maximum size allowed by the window. If the isolated pattern has no black pel extensions, then it is a symbol; otherwise it is a nonsymbol.

The isolation algorithm operates on a two-dimensional one-bit array containing the original picture. The picture memory is scanned line

by line from the upper-left element. When a black pel is found, the procedure attempts to trace the boundary of a black region, clockwise. The tracing algorithm is a standard one; however, we describe it here for further reference. Let us call the first black pel $(x_1,y_1)$. The neighbors (adjacent pels in eight directions of $(x_1,y_1)$ are being examined, beginning at $(x_1+1,y_1)$ and searching clockwise around $(x_1,y_1)$ up to $(x_1-1,y_1+1)$. If a black pel is found, it becomes the second pel of the contour — $(x_2,y_2)$; otherwise $(x_1,y_1)$ is erased from the picture memory (single pels are neglected) and the scan continues. Each subsequent pel of the contour is found by searching around the current pel $(x_i,y_i)$, beginning two steps clockwise from the previous pel $(x_{i-1},y_{i-1})$ (Fig. 2). The contour trace ends when it returns to the first pel in such a way that the next pel would by $(x_2,y_2)$. The tracing algorithm checks for the limits of the picture array and it maintains a window. Pels beyond the limits of the picture array and those outside of the current window are always treated as white (0 valued). The purpose of the window is to restrict the maximal size of isolated pattern to $W \times W$. The window is initially set to a size $2W \times W$, and positioned in such a way that $(x_1,y_1)$ is in the center of its upper edge. When the traced part of the boundary reaches a width of $W$, the window is reset to a size $W \times W$, and it is placed over the boundary part that has been traced, such that $(x_1y_1)$ is still at the upper edge of the window (Fig. 3).

The tracing of the boundary is recorded in a two-dimensional one-bit array S in the following way. When the search around the current boundary pel $(x_i,y_i)$ goes past the pel $(x_i+1,y_i)$, a 1 is put in $S(x_i+1,y_i)$. If the search goes past the element $(x_i-1,y_i)$ then a 1 is put in $S(x_i,y_i)$. All the elements of S are initially set to 0. The information in S (Fig. 4), after the trace termination, completely represents the boundary (it is a form of run-length code). The pattern now can be isolated by copying and erasing the portion of the picture that is enclosed by the boundary (including the boundary). This is accomplished using the

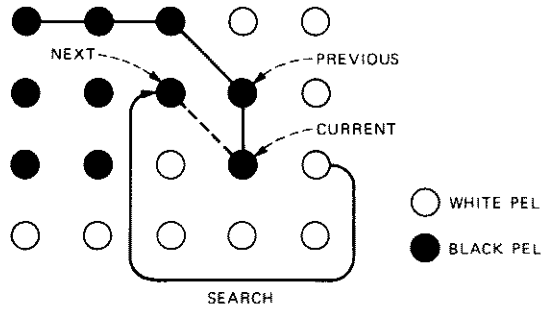



Fig. 2—Contour tracing.

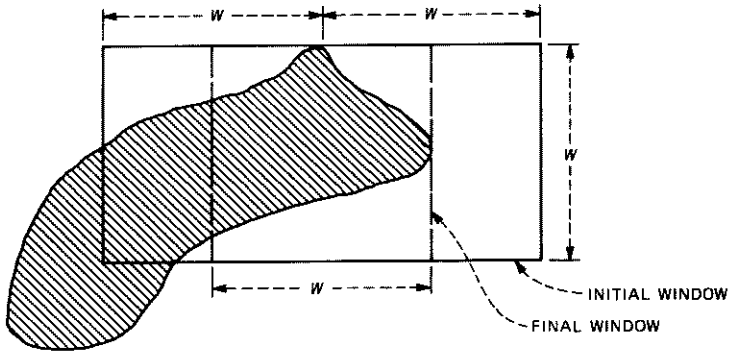



Fig. 3—Window positioning for isolation.

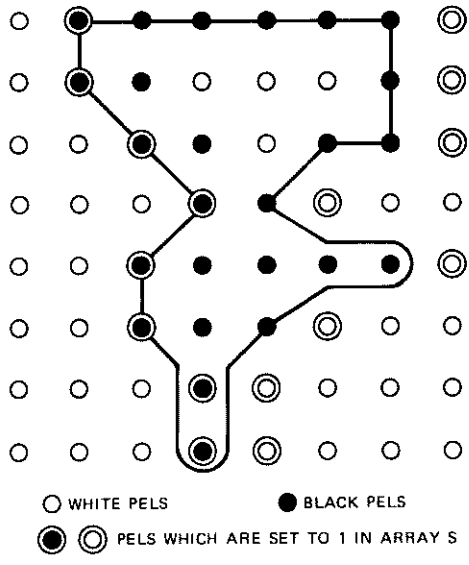



Fig. 4—Contour encoding in array S.

information in the array S. For any row of S, let $S_1$, $S_2 \cdots S_n$ be the position (x-coordinates) of 1-valued elements in a row. The number $n$ is always even, which is a property of the boundary encoding that we use. For every row of S, the pixels of a corresponding row of the picture memory between $S_1$ and $S_2$, $S_3$ and $S_4$, etc., are copied to another array, and set to 0 in the picture memory, including $S_1, S_3 \cdots$ and excluding $S_2, S_4, \cdots$. The pattern is now isolated and erased from the picture memory. While the isolation algorithm described above always works correctly, i.e., it isolates symbols and completely decomposes large

figures into nonsymbols, it does not attempt in any way to optimize the decomposition, so the results are not always pleasing.

To improve the decomposition in cases commonly occurring in graphics, we have added two extensions to the basic isolation scheme:

1. L-pattern suppression

L-pattern suppression improves the segmentation of large blobs that otherwise may generate many dissimilar nonsymbols (Fig. 5). This extension is implemented in the tracing phase of the isolation algorithm as follows: If the beginning part of the traced boundary goes straight down from either first or second pel over more than $k$ (currently $k = 10$) pels, then an attempt to turn immediately to the right resets the lower edge of the window to the last pel before the right turn, so the boundary is forced to turn left (see Fig. 6).

2. Cross decomposition

If the isolated pattern can be represented as an intersection of a horizontal and a vertical line segment (a cross), then each segment becomes a separate pattern. This is implemented by comparing each isolated pattern (with the matching technique described in Section 4) to a cross formed by secting this pattern with vertical and horizontal lines one pel from the edges of the final window (Fig. 7). If a sufficiently close match is found, then one of the line segments from the cross is

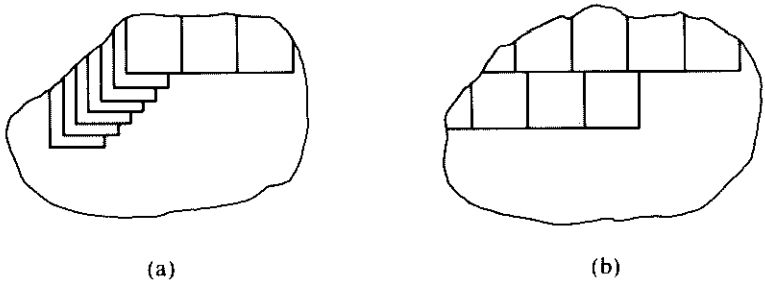



Fig. 5—Improvement in segmentation due to L-pattern suppression. (a) Before suppression. (b) After suppression.

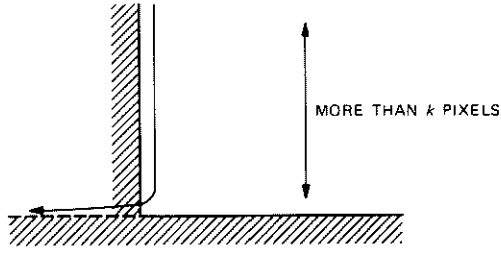



Fig. 6—L-pattern suppression. When tracing reaches the corner, it is forced to follow the dashed line.

returned to the picture memory, while the other replaces the isolated pattern. This extension reduces the number of patterns generated by line crossings in grids and tables (Fig. 8).

The basic isolation algorithm is similar to the region extraction method of Dudani[4], but in contrast to the latter it does not need to store and process a list of boundary points, and it extracts regions containing holes in one pass. This algorithm can be shown to work correctly in every case and it is well suitable for a hardware implementation. The extensions of the basic algorithm are heuristic in nature, but they improve considerably the decomposition of large regions. Examples of such improvements are shown in Fig. 5 and 8. Additional improvements may be possible at some increase of the computational cost.

## IV. MATCHING

The matching includes all the processes necessary to know whether an incoming pattern matches any of the library patterns. In this system, we divide the matching into three parts.

1. The screening unit makes a selection of the library patterns, and

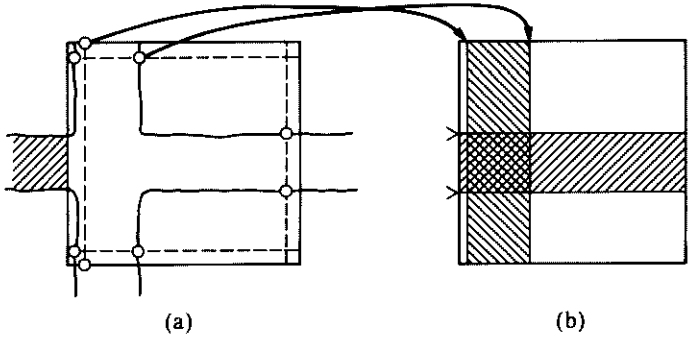



Fig. 7—Forming an intersection pattern.

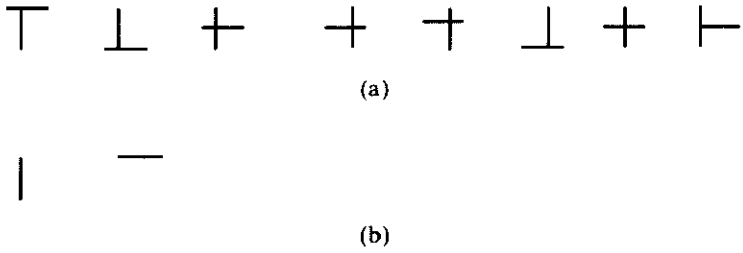



Fig. 8—Patterns resulting from grid segmentation. (a) Before cross decomposition. (b) After cross decomposition.

directs for template matching only those library patterns that might match.

2. The template matcher creates a new binary picture called error picture, containing black pels or 1's in the locations where the two template-matched patterns are dissimilar.

3. The matching decision process uses the error pictures and other information to decide whether a correct match has occurred.

#### *4.1 Screening*

The purpose of the screening is to reduce the time-consuming task of the matcher. It should direct to the template matcher only the library patterns that might match the incoming (unknown) pattern. The screening is obtained by measuring some characteristics of the patterns, called features, and comparing them. The features must be easy to compute and compare, and also must form an easily classifiable space. The digitization of a facsimile adds much noise to a pattern. To get an efficient screening, the features must also be relatively noise independent. Four features were chosen for the screening. Two of them are obvious: the pattern length and the pattern height. The two others are the number of horizontal and the number of vertical white runs enclosed in the pattern. They are characteristics of the inside of a pattern, separating, for example, c from e or o. The chosen features are shown in Fig. 9. The straightforward feature "number of black pels" was found to be of little use because of its high variability and dependency upon the other features.

The screening process also must decide in which order to send the library patterns to the matcher. The most probable match should be sent first, to reduce the number of matches. The probability of a match between patterns depends not only on the similarity of their features, but also on the probability of occurrence of a library pattern. For example, an incoming pattern having the same feature distance to an O and a Q is much more likely to match the O than the Q since O is much more frequent than Q. The screening takes into account both the feature similarity and the probability of occurrence of a library pattern. We consider the probability of occurrence by sorting the library patterns according to the number of times they have matched (see Section 5.2.1). We take the feature distance into account by allowing for each feature only a fixed margin between the two patterns. The margin must be wide enough not to preclude any correct match and tight enough to reduce the number of template matches. A two-pass screening was found very efficient. In the first screening, only library patterns with features very similar to those of the incoming patterns are sent to the template matcher. A second, much looser, screening is applied only in the few cases where no match occurred.

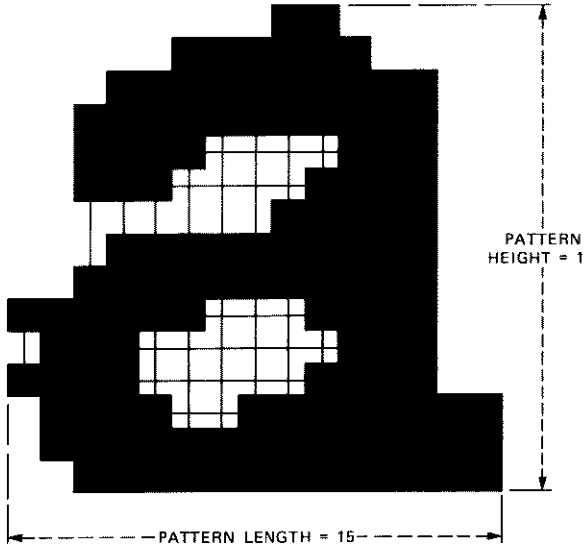



Fig. 9—Features chosen for screening. Horizontal lines indicate which runs are included in count of horizontal runs. Vertical lines indicate which runs are included in count of vertical runs. Horizontal run count is six and vertical run count 15.

The screening and the sorting are very efficient in reducing the number of matches. For example, for a typewritten document, the average number of matches per incoming pattern is reduced to 2.5, compared to 25 without screening and sorting.

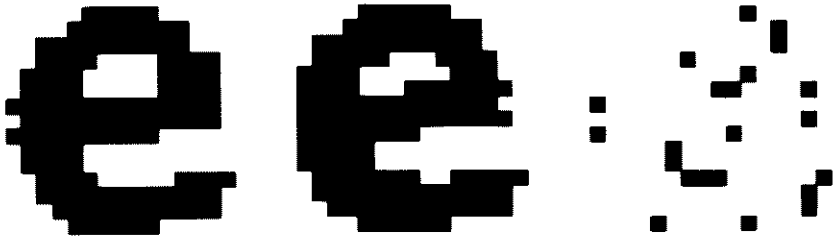
### *4.2 Template matching*

The template matcher creates a new picture called error picture, which contains 1's in the locations where the two patterns are different. The error picture is obtained simply by superimposing the two patterns and making "exclusive or" of the corresponding pels. Figure 10 is an example of matching two patterns of the same character, while Fig. 11 shows the matching of two unlike patterns. Two patterns are always matched nine times, allowing the displacement of one pattern compared to the other by $\pm 1$ in both the horizontal and vertical directions.

### *4.3 Matching decision*

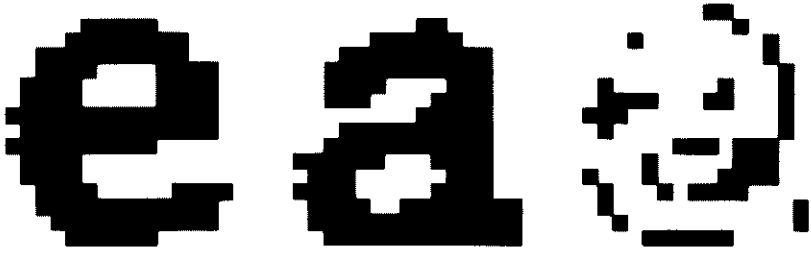
The matching decision unit must process the error picture to detect whether there is a correct match, and to decide which relative position of the library pattern gives the best match.

The straightforward approach is to count the number of errors (or 1's) in the error picture and to threshold it to make the decision. Such



(a) (b) (c)

Fig. 10—Template matching of two similar patterns, with (a) and (b) original patterns and (c) error picture.

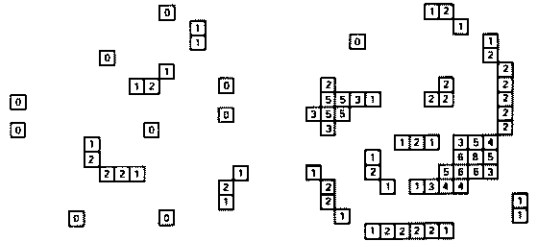


(a) (b) (c)

Fig. 11—Template matching of two different patterns, with (a) and (b) original pattern and (c) error picture.

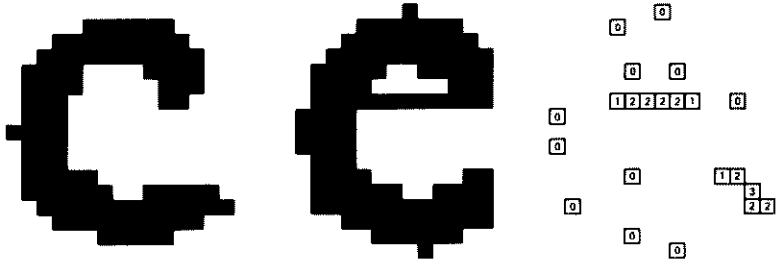
a technique would lead to many mismatches or many undetected matches, since, as shown in Ref. 2, the error count for two patterns corresponding to the same character is sometimes higher than the count for two patterns corresponding to different characters. This is caused by the digitization noise. Figure 10 shows that the template matching of two patterns of the same character gives relatively randomly distributed errors. Figure 11 shows that in the case of patterns of different characters, a cluster of errors appears where there are morphological differences between patterns.

As Ref. 2 shows, we could apply a weighted error count where the weight of an error is equal to the number of error pels among its eight neighbors. Single errors are erased and the maximum weight is eight. Figure 12 gives the weighted error pictures from the error pictures of Figs. 10 and 11. The weighted error count is not sufficient for the matching decision, as shown by Fig. 13. We must look at local error patterns to make the decision. The reason is that it is the local characteristics of the pattern that indicate whether two patterns are the same. Therefore, any decision made upon a count or integration



(a) (b)

Fig. 12—Weighted error pictures. (a) Weighted error count is 18 in Fig. 10 and (b) 144 in Fig. 11.



(a) (b) (c)

Fig. 13—A weighted error count matching criterion lead to a mismatch, with (a) and (b) original patterns, and (c) weighted error picture.

may be incorrect. The matching decision described below uses only local measures and is also made locally with the simple rule that the match is considered correct if no local rejections are detected during a template matching.

The following rule of decision is made. A match is rejected if:

Condition 1: An error pel has a weight of 4 or more, or

Condition 2: (a) an error pel has a weight of 2 or more, (b) at least two of its neighboring error pels are not connected, and (c) one of the two pels from the patterns used to obtain the error pel has a weight of 0 or 8 (corresponding to 0 or 8 surrounding black pels).

Most mismatches are detected by Condition 1, but Condition 2 is necessary in order to reject, for example, the possible match of an e and a c shown in Fig. 13. It is easy to see that Condition 2a is not necessary since it is included in 2b, but Condition 2a reduces the computation.

With these matching criterion, no visible mismatches have been detected, except slight distortion in line drawings. It is important to notice that a rejection can often be detected after processing a small

fraction of the error picture. A matching decision made at the same time as the template matching would lead to an early abortion of template matchings and thus reduce the computation.

When a correct match is detected, several relative positions sometimes give a correct match. The chosen relative position will be the one with the lowest error count. The best relative position will decide where the library pattern will be put to replace the incoming pattern.

## V. CODING

Contrary to many conventional facsimile coding techniques, we must code several different kinds of events and design several separate code books. The code for a pattern includes the position and the description of the pattern. The description is usually its library identification, or in the case of a new pattern, its complete description. The coding procedure is described here for the size of the International Telegraph and Telephone Consulative Committee (CCITT) test facsimiles having 1728 pels per line and 2376 lines, but it can easily be modified for other cases.

### *5.1 Coding of the position of the pattern*

To obtain a good-quality reproduction with pattern matching, we must position the patterns accurately. Considering the CCITT test documents, 23 bits are necessary for an absolute fixed length coding (11 bits horizontally, 12 bits vertically). We choose to transmit the horizontal position uncoded (11 bits) because variable-length run-length coding would lead only to slightly smaller coding length (typically 1 to 1.5 less bits/pattern) since the horizontal distance between patterns is large. Also since the absolute horizontal position is coded, the patterns can be transmitted in a nonsequential order, which, as shown later, leads to a significant decrease in the average coding length for the library identification code words. It should be noted that with 1728 pels/line and an 11-bit code word, the code words starting with 111 are not used and therefore can be used as special code words.

We code the vertical position of the patterns in the following way:

1. A mode bit is sent at the beginning of each line to indicate whether there are any patterns starting on that line.
2. If there are no patterns on the line, operation 1 is repeated on next line.
3. If there are patterns on a line, they are all coded. The special horizontal code word 111 indicates that there are no more patterns on the line and that the next line can be considered.
4. When a pattern is replaced by a library pattern, the position of the library pattern might be moved up or down by one line. Therefore,

after the library identification has been coded, the code words 10 and 11 are used to position the library pattern up or down, while code word 0 is used to indicate no vertical displacement. No vertical displacement code word is sent with a new library pattern, since there are no changes in vertical position.

Figure 14 shows examples of the message format for the pattern positioning.

### *5.2 Coding of the pattern identification*

The coder must send a pattern identification word with each pattern. We can transmit the pattern number uncoded. It requires, for example, seven bits in the case of a library size of 128 and nine bits in the case of a library size of 512. The coding procedure used here will lead to an average coding length of the pattern identification of fewer than five bits/pattern. It will be obtained by a continuous library updating and by variable-length coding.

#### *5.2.1 Library updating and management*

The library management and updating is done for the following purposes:

1. Accept new library patterns, and if necessary, delete a seldom used library pattern to make room for the new one.
2. Organize the library for the fastest match, taking into account the screening and matching procedures.
3. Organize the library for minimum average library identification coding length.

All three require the same processing: to keep track of the number of times each library pattern is used. By ordering the library pattern in order of decreasing usage, the correct match will be obtained rapidly,

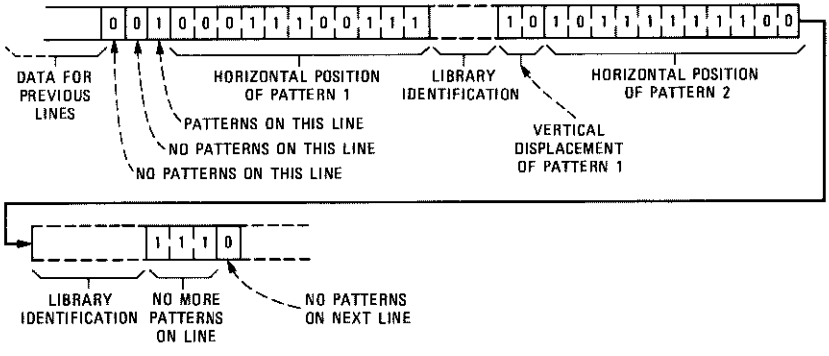



Fig. 14—Coding of positions of patterns. Two lines have no patterns, then a line has three patterns; the first on position 231 is replaced by a library pattern, the second on position 1532 is a new library pattern, There are no patterns on next line.

since the most used library patterns will be accessed first. An efficient coding of the patterns' identification is obtained by giving short code words to the first patterns in the list. The last pattern in the list, which is one of the least used patterns, can be deleted to make room for a new one.

The updating must be deterministic and use no future information, since the receiver must make the same updating to decode correctly.

The updating rule of the patterns in the library is as follows:

1. When a pattern matches a library pattern number $K$, that library pattern is moved to number $K/2$ and all the pattern numbers from $K/2$ to $K$-1 are increased by 1.

2. When a new pattern is added to the library, it gets number $N/2$ where $N$ is the total number of library patterns. The patterns with numbers from $N/2$ to $N$ will be increased by 1, and if $N$ is equal to the maximum number of library patterns $M$, the library pattern with number $N + 1$ is dropped.

This updating procedure was found to efficiently give low identification numbers to often used patterns and high numbers to seldom used patterns. If $M$ is the maximum number of library patterns, it guarantees that a new library pattern will stay in library for at least $M/2$ matches, but generally for many more.

#### *5.2.2 Pattern identification coding table*

The pattern identification coding table includes two special code words: "new pattern" and "same pattern." They are added to increase the coding efficiency. The "new pattern" code word is chosen because it is not necessary for a new library pattern to send an identification number, since the decoder uses the same rule as the coder to assign the identification number to the new pattern. The "same pattern" code word indicates that the transmitted pattern is the same as the previously transmitted pattern. It is useful particularly for typewritten text where the line-by-line search for a pattern often detects the same pattern (character).

The coding table for the pattern identification is given in Table I for a pattern library with a maximum of 512 patterns.

This code leads to an average library identification length of fewer than seven, compared to nine with a fixed-length code. The next section shows a more efficient coding procedure.

#### *5.2.3 Pattern identification coding by sorting*

Since an absolute code gives the horizontal position of a pattern, it is possible to transmit the patterns detected along a line in any order. The only condition is that the library updating be done at the end of the line. The average coding length of the library identification is

Table I—Coding table for identification of library patterns

| Symbol | Code Word | Code Word Length |
|---|---|---|
| Same pattern | 000 | 3 |
| Library pattern 1–16 | 1XXXX | 5 |
| New pattern | 00100 | 5 |
| Library pattern 17–32 | 010XXXX | 7 |
| Library pattern 33–64 | 0011XXXXX | 9 |
| Library pattern 65–128 | 00101XXXXXX | 11 |
| Library pattern 129–512 | 011XXXXXXXXX | 12 |

reduced to fewer than five bits, by sorting the patterns on a line according to their library number. That is because:

1. Many of the patterns are the same.

2. The library pattern identification number is run-length coded (only the increase compared to the previous identification number is coded).

3. The new library patterns are sent at the end of the line; therefore, the new pattern code word is sent only once, since any more patterns are automatically new patterns.

This can be illustrated by an example. Let a line have the following pattern: pattern 23, new pattern; pattern 28, same; pattern 23, new pattern. By looking at Table I, the coding length is $7 + 5 + 7 + 3 + 7 + 5 = 34$ bits. With sorting, the patterns become: pattern 23, same; pattern 28, same; new pattern; new pattern. The coding length is $7 + 3 + 5 + 3 + 5 + 0 = 23$ bits. It should be noted in this example that pattern 28 is coded as pattern 5 since only the increase in identification number compared to the previous pattern is coded.

The library updating is done at the end of each line. This creates problems when accepting new library patterns. They must be added immediately to the top of the library, since the position of the other patterns should not be changed. It is also not possible to delete patterns to make room for the new ones. For that reason, before scanning a line, enough library patterns should be deleted to avoid an overflow of the pattern library.

### *5.3 Coding of the library pattern description*

The size of a pattern is limited to $32 \times 32$ bits. The description starts with a 5-bit word, which indicates the height, $H$, of a pattern in binary. The length of a pattern is extended to 32 pels by filling the right end with 0's. Therefore, there are $32 \times H$ pels to code. For coding efficiency, one white pel (0) is added at the beginning. A coding line is made of the $32 \times H + 1$ pels considered in the raster scan order. The reference line is similar to the coded line except that all the pels are shifted to the right by 32 pels (one line). Therefore, a line is coded

using the previous line as the reference. The line is then coded by the CCITT two-dimensional code,[5] with the only modification that the first code word, which is always the horizontal mode code word, is deleted, since it doesn't give any information. For coding efficiency, it is chosen to allow switching between two modes for the coding of the library pattern description. The first mode is as described above and called "horizontal coding." The other is called "vertical coding" and is the same as above except that the pattern is coded column after column from top to bottom. Therefore, in the vertical mode the description starts with a 5-bit word indicating the length of a pattern. A header bit indicates which mode is chosen, with a 0 for horizontal mode and a 1 for vertical mode. We could also code the pattern description using a code better matched to the source. This would reduce the coding length, but at the expense of requiring a specific code in place of a standard code.

### *5.4 Coding summary*

The coding procedure can be summarized in the following way:

1. All the patterns isolated along a scan line are matched.
2. At the end of the line, the matched patterns are sorted in order of increasing pattern identification number. The new library patterns are added at the end in sequential order.
3. The patterns are coded and transmitted with the information sent in the following order:
   a. Horizontal position of pattern.
   b. Pattern identification. If it is a new pattern, the identification is sent only for the first new pattern on the line.
   c. A 1- or 2-bit code word to specify the vertical shift of a pattern, except if it is a new library pattern.
   d. For a new library pattern the following bits are sent: (1) a header bit indicating whether the horizontal or vertical coding mode is chosen, (2) a 5-bit word indicating the number of lines of the pattern to be coded, and (3) the CCITT two-dimensional coding of the pattern (see 5.2).
   e. After all patterns on a line have been sent, the special horizontal code word 111 indicates the end of the line.
   f. The library update is made according to 5.2.3. The patterns are updated in order to increasing identification number. After updating, all patterns with a number greater than 480 are deleted, thus allowing for at least 32 new library patterns to be added on the next line.

Figure 15 is an example of message transmission. The different code words are summarized in Table II.

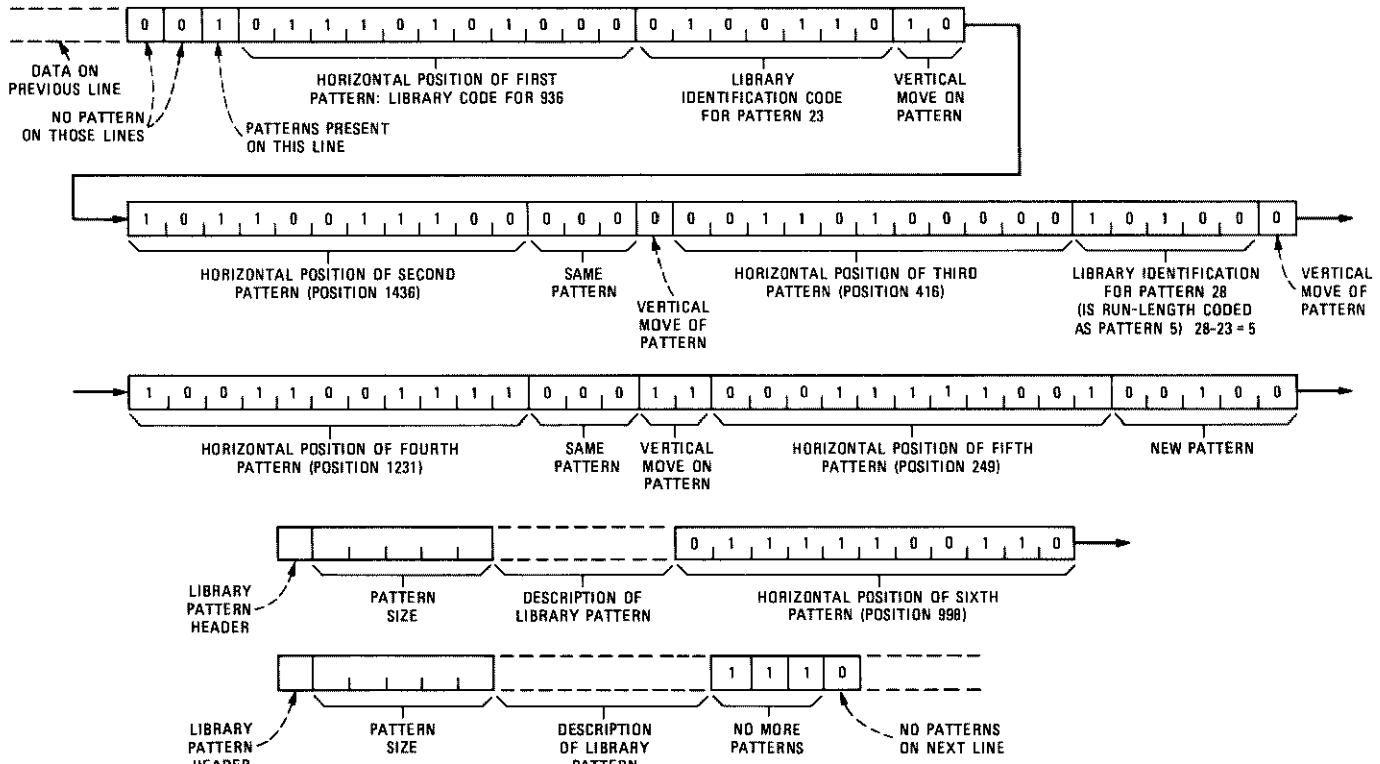



Fig. 15—Message transmission. There are two lines without patterns, next pattern 23 is in position 936, same pattern is in position 1436, pattern 28 is in position 416, same pattern is in position 1231; two new patterns are in position 249 and 998, and there are no patterns on next line.

Table II—Description of the code words for pattern matching coding

| Code Definition | Word Size | Description |
|---|---|---|
| Mode bit | 1 | Indicates whether there are any patterns on the line. |
| Horizontal position | 11 | Gives in binary the absolute position of a pattern. |
| No more pattern | 3 | Indicates that there are no more patterns on the line (this code word: 111 is a special horizontal position code word). |
| Vertical move of pattern | 1 or 2 | Indicates whether the pattern must be moved up or down by one line or is not moved. |
| Library identification code | Variable | Defines which library pattern is transmitted. |
| Library pattern description header | 1 | Indicates whether the library pattern is coded in horizontal or vertical mode. |
| Library pattern size | 5 | |
| Library pattern description | Variable | Slightly modified CCITT two-dimensional code. |

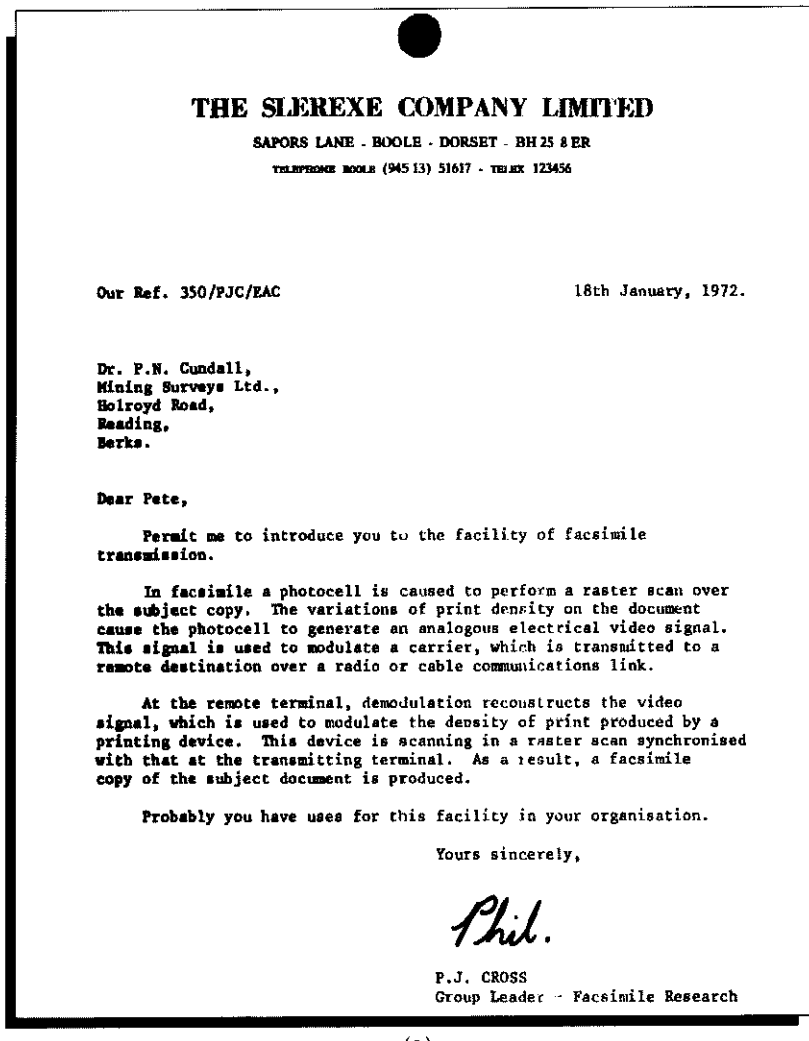
## VI. SIMULATION RESULTS

The important criteria are the compression and the quality of the received documents. For that purpose, the set of eight CCITT facsimile documents are used. Their resolution is 7.7 pels/mm (200 pels/in.) in both the horizontal and vertical directions. They have 1728 pels/line and 2876 lines. Documents one, two, four and five are shown in Fig. 16. All eight documents are shown in Ref. 5. For accurate comparison with the matching technique by Pratt et al.,[2] the simulations were also made with an older nonofficial version of the CCITT documents, which is similar except each document has 1728 pels/line and 2128 lines.

### *6.1 Facsimile quality*

In order to improve the quality of the decoded picture, a local filtering using a $3 \times 3$ window is applied. In addition, large library patterns are slightly expanded on their borders. This operation erases artifacts in large black regions.

The encoding scheme modifies the binary picture. We must therefore verify that the alterations are not visible or at least not annoying. We can consider three picture alterations: wrong matches, matches with a slightly distorted pattern, and wrong positioning. In the case of a wrong match, a pattern is replaced by a different pattern. The only detected wrong matches are such as between 0 and O, dot and comma, I and 1, which even people cannot recognize correctly without using the context. Therefore, it can be considered that the system has practically no wrong matches. A match with a slightly distorted pattern can occur with characters. A character might match a same character of a different font. Or a character might match a same but thinned or



THE SLEREXE COMPANY LIMITED

SAPORS LANE - BOOLE - DORSET - BH25 8ER

TELEPHONE BOOLE (945 13) 51617 - TELEX 123456

Our Ref. 350/PJC/EAC 18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

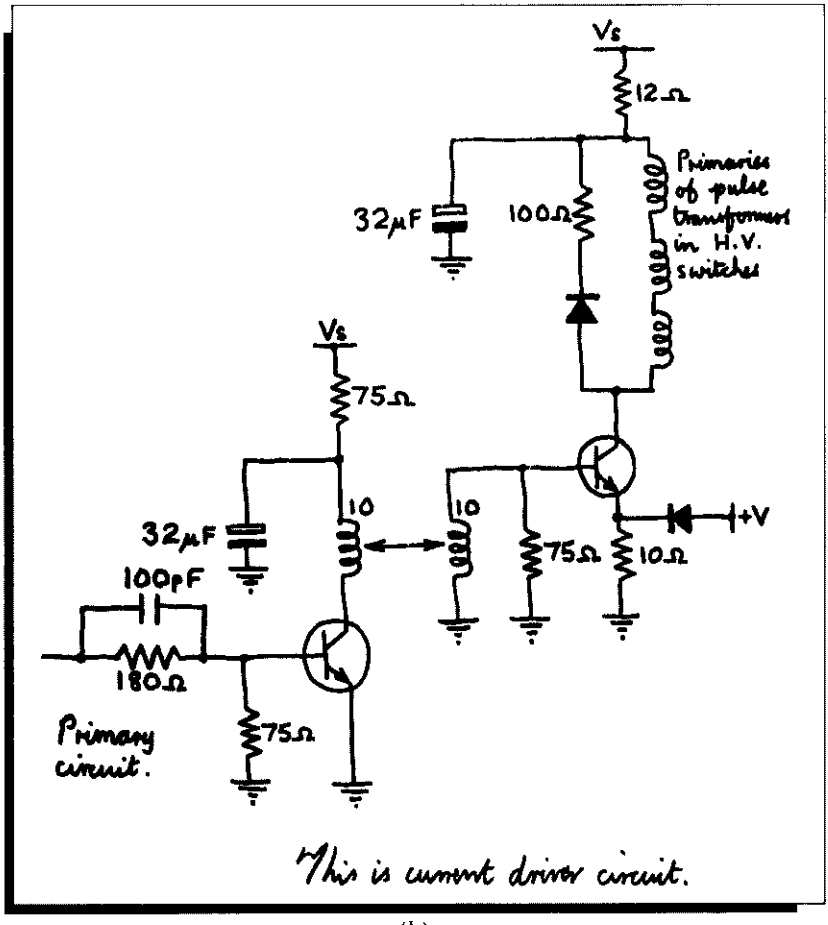
Phil.

P.J. CROSS
Group Leader - Facsimile Research

(a)

Fig. 16(a)—Original CCITT document one (first 2000 lines).

thickened character. Such matches, contrary to wrong matches, are tolerable if they don't appear too often. Such distorted matches appear when two slightly different fonts are used on a same page or when characters of a page come from a low-quality typewriter or scanner. The wrong positioning of a pattern decreases the quality of the received facsimile. No noticeable wrong positioning for patterns such as characters or other symbols is observed. Some visible wrong positionings are observed for nonsymbol patterns such as line segments, where the



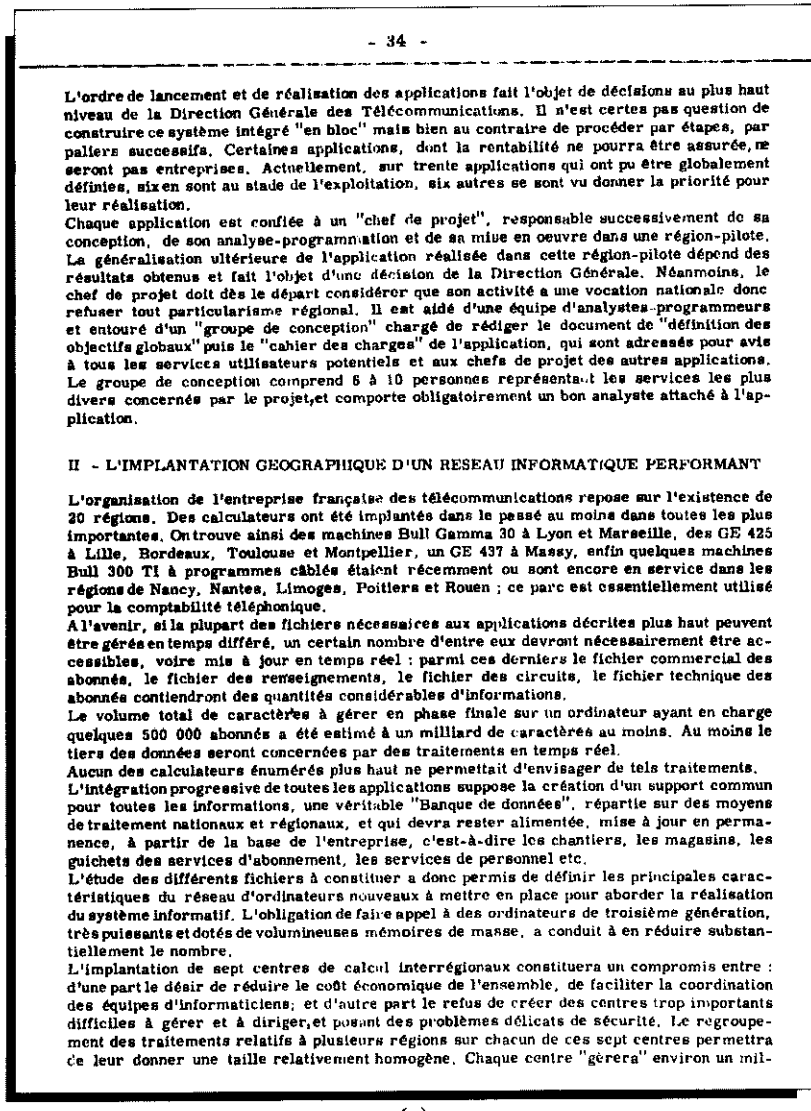

(b)

Fig. 16(b)—Original CCITT document two (first 2000 lines).

successive patterns make the lines slightly jagged. Figure 17 shows the same CCITT facsimiles as Fig. 16, but after transmission by pattern matching. It can be seen that there are no significant degradations. There are some slight irregularities in line drawings, as for example in Fig. 17d. A few distorted matches appear on CCITT document one (Fig. 17a).

### *6.2 Compression*

To make an accurate comparison with both the symbol matching and two-dimensional coding techniques, the coding simulations have been made with both the official set of CCITT facsimile documents



- 34 -

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.
Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet,et comporte obligatoirement un bon analyste attaché à l'application.

II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.
A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.
Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.
Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.
L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.
L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.
L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger,et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gérera" environ un mil-

(c)

Fig. 16(c)—Original CCITT document four (first 2000 lines).

and a former nonofficial version often used for facsimile compression comparisons. Table III gives the coding lengths for the CCITT documents for the official and nonofficial set of CCITT documents, respectively. They include the code length for the different codes necessary for the pattern matching coding. Table IV gives the compression ratio for the same CCITT documents and compares them with the

Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de $f$ pour les valeurs numériques indiquées page précédente.

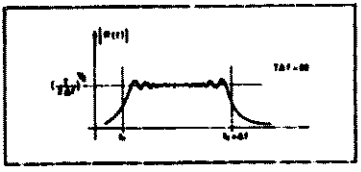


FIG. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 \leqslant f \leqslant f_0+\Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0+\Delta f$, filtre ne modifiant pas la phase des composants le traversant ;

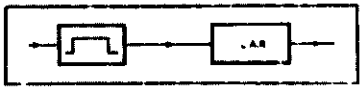
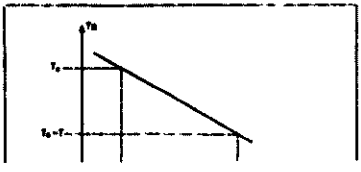


FIG. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe $T_g$ décroissant linéairement avec la fréquence $f$ suivant l'expression :

$$T_g = T_0+(f_0-f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).



telle ligne à retard est donnée par :

$$\varphi = -2\pi \int_0^f T_g \, df$$

$$\varphi = -2\pi \left[T_0+\frac{f_0 T}{\Delta f}\right] f + \pi \frac{T}{\Delta f} f^2$$

Et cette phase est bien l'opposé de $|\phi(f)|$,

à un déphasage constant près (sans importance) et à un retard $T_0$ près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard $T_0$ près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre $f_0$ et $f_0+\Delta f$, et nulle de part et d'autre de $f_0$ et de $f_0+\Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0+\Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\frac{T}{1/\Delta f} = T\Delta f$

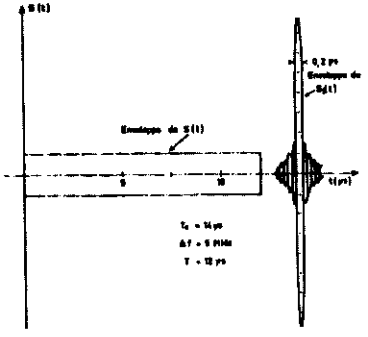


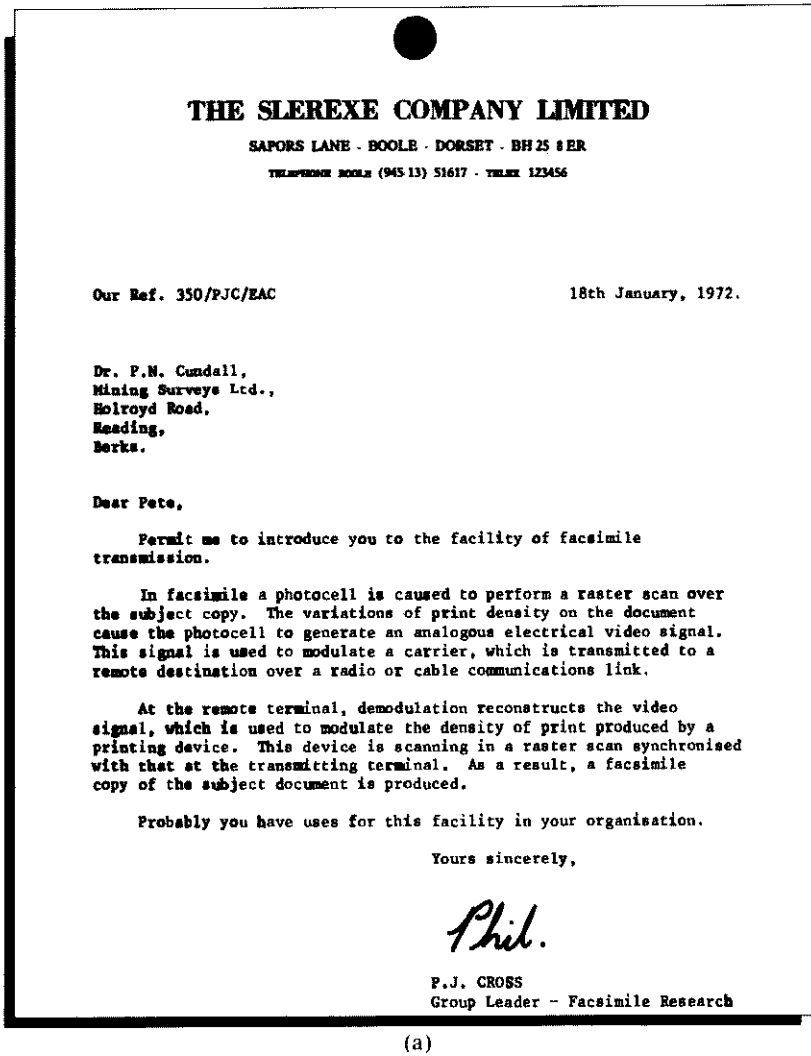
FIG. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse $f_0$.

(d)

Fig. 16(d)—Original CCITT document five (first 2000 lines).

symbol-matching technique of Pratt et al.[2] and the two-dimensional CCITT code. The results are without any synchronization or stuffing bits, which is natural since pattern matching coding would be intended for future facsimile networks such as group four facsimile machines with fewer overhead bits. Therefore, the compressions of the two-dimensional CCITT code and symbol matching have been corrected by deleting the synchronization and stuffing bits and are different from their values given in Refs. 2 and 5.



THE SLEREXE COMPANY LIMITED

SAPORS LANE · BOOLE · DORSET · BH25 8ER

TELEPHONE BOOLE (945 13) 51617 · TELEX 123456

Our Ref. 350/PJC/EAC

18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

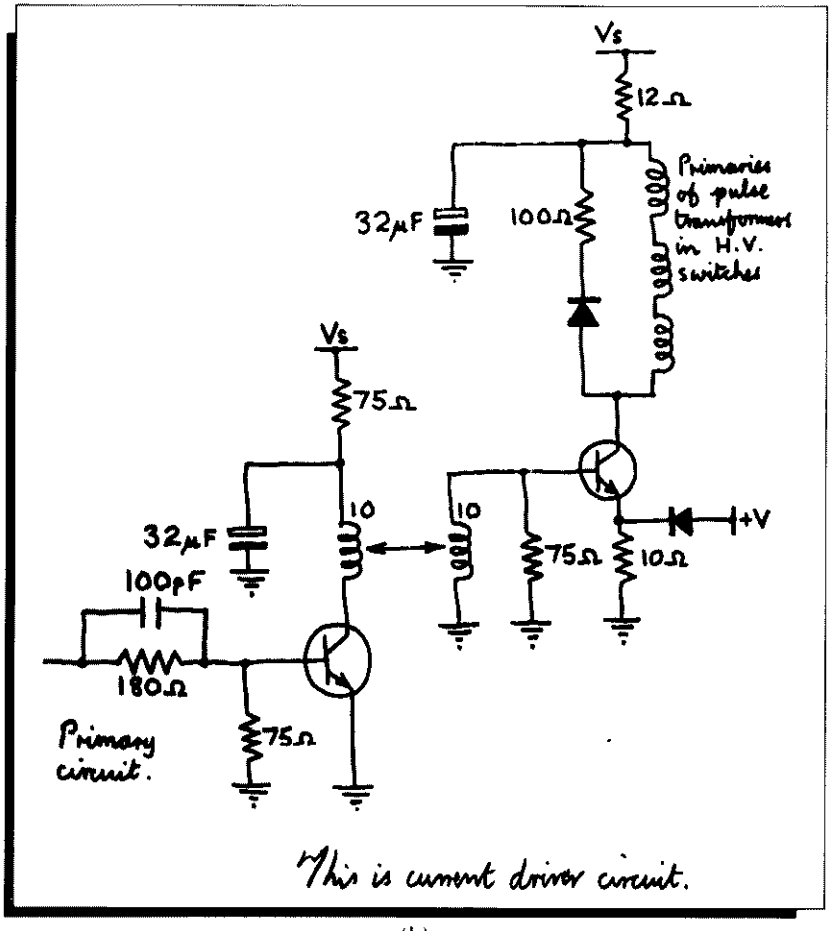
P.J. CROSS
Group Leader - Facsimile Research

(a)

Fig. 17(a)—Document on (first 2000 lines) after pattern matching (first 2000 lines).

Very high compressions are obtained—up to 80. The compression has often doubled compared to that of the two-dimensional CCITT code and is sometimes 4.8 times higher. The compression is, depending upon the documents, 20 to 80 percent higher than the compression derived from the symbol matching technique by Pratt et al.[2] More detailed comparisons and observations are useful when considering the performances of facsimile coding by pattern matching:

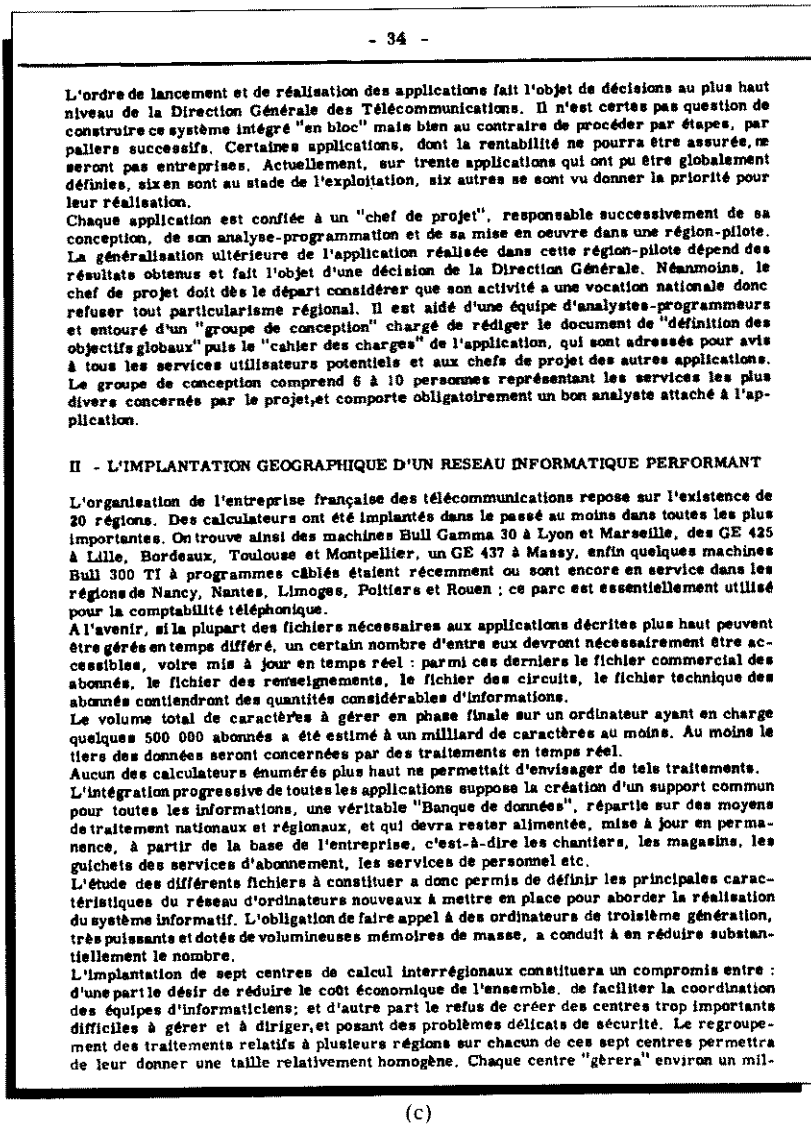
1. An astonishing fact is the difference in compression observed




(b)

Fig. 17(b)—Document two (first 2000 lines) after pattern matching.

between the old and the official version of the CCITT documents. For documents three and five the compressions are nearly twice as high for the old version than for the official version. Significant discrepancies are also observed for documents one and eight. This is in spite of the fact that old and official documents are the same except that they were scanned differently. It can also be noted that for the two-dimensional CCITT code, the difference in compression is smaller than five percent except for document eight, where the difference is about 20 percent. It must therefore be concluded that the performances of the pattern matching coding techniques are much more dependent upon the scanning and binary thresholding. Observing both versions of documents three and five, the main difference is that in the official



- 34 -

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.
Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet, et comporte obligatoirement un bon analyste attaché à l'application.

II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.
A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.
Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.
Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.
L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.
L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.
L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger, et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gèrera" environ un mil-

(c)

Fig. 17(c)—Document four (first 2000 lines) after pattern matching.

version, characters are often clustered together, which leads to incorrectly (or rather "nonconveniently") isolated characters (as shown in Fig. 18a), while for the old version, the characters are rarely clustered together. However, sometimes a character in the old version is isolated into several patterns because not all its pels are connected (as shown in Fig. 18b). The old version of documents three and five should have

Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de $f$ pour les valeurs numériques indiquées page précédente.

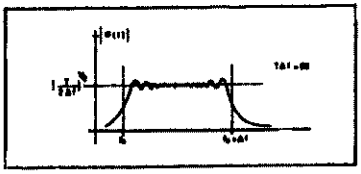


Fig. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 \leqslant f \leqslant f_0+\Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0+\Delta f$, filtre ne modifiant pas la phase des composants le traversant ;

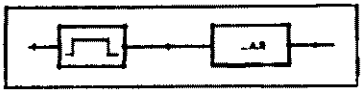
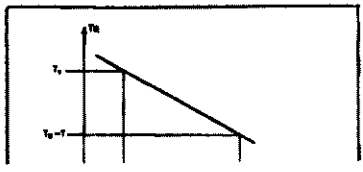


Fig. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe $T_R$ décroissant linéairement avec la fréquence $f$ suivant l'expression :

$$T_R = T_0+(f_0-f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).



telle ligne à retard est donnée par :

$$\varphi = -2\pi\int_0^f T_R\,df$$

$$\varphi = -2\pi\left[T_0+\frac{f_0 T}{\Delta f}\right]f+\pi\frac{T}{\Delta f}f^2$$

Et cette phase est bien l'opposé de $/\phi(f)$,

à un déphasage constant près (sans importance)

et à un retard $T_0$ près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard $T_0$ près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre $f_0$ et $f_0+\Delta f$, et nulle de part et d'autre de $f_0$ et de $f_0+\Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0+\Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\dfrac{T}{1/\Delta f} = T\Delta f$

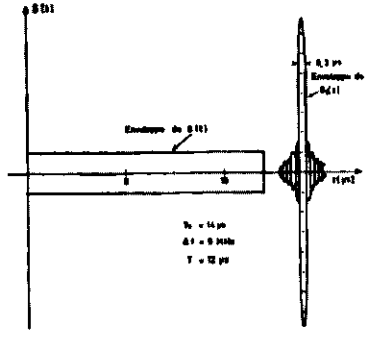


Fig. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse $f_0$.

(d)

Fig. 17(d)—Document five (first 2000 lines) after pattern matching.

more patterns than the official but fewer library patterns. In fact, old CCITT document three has 2199 patterns and 225 library patterns, while the official version has 1945 patterns and 551 library patterns. The coding length of a library pattern is much greater (by a factor of about 10) than that of a nonlibrary pattern, which explains the difference in compression ratios. It can be concluded that pattern matching is much more dependent on the scanning quality and the thresholding than two-dimensional facsimile codes.

Table III—Coding length in bits with pattern matching coding

(a) CCITT documents (2376 lines, 1728 pels/line)

| | CCITT1 | CCITT2 | CCITT3 | CCITT4 | CCITT5 | CCITT6 | CCITT7 | CCITT8 |
|---|---|---|---|---|---|---|---|---|
| Mode bit | 2376 | 2376 | 2376 | 2376 | 2376 | 2376 | 2376 | 2376 |
| Horizontal position | 11847 | 8096 | 21395 | 46464 | 23551 | 15356 | 36828 | 31240 |
| No more pattern | 888 | 1587 | 2193 | 1854 | 2235 | 1929 | 3165 | 3231 |
| Vertical move of pattern | 1114 | 411 | 1921 | 5724 | 2242 | 1347 | 3427 | 3642 |
| Library identification code | 4895 | 3749 | 9458 | 18584 | 10848 | 6426 | 19289 | 13085 |
| Library pattern description | 43983 | 71427 | 120749 | 41138 | 94923 | 71959 | 204113 | 115415 |
| Total | 65103 | 87646 | 158092 | 116140 | 136175 | 99393 | 269198 | 168989 |

(b) Older CCITT documents (same as documents used in Ref. 1) (2128 lines, 1728 pels/line)

| | CCITT1 | CCITT2 | CCITT3 | CCITT4 | CCITT5 | CCITT6 | CCITT7 | CCITT8 |
|---|---|---|---|---|---|---|---|---|
| Mode bit | 2128 | 2128 | 2128 | 2128 | 2128 | 2128 | 2128 | 2128 |
| Horizontal position | 12342 | 7733 | 24189 | 49610 | 27511 | 16049 | 38566 | 30635 |
| No more pattern | 846 | 1533 | 2103 | 2574 | 2091 | 2007 | 3942 | 3168 |
| Vertical move of pattern | 1155 | 463 | 2448 | 5726 | 3028 | 1411 | 3812 | 2441 |
| Library identification code | 4958 | 3599 | 11078 | 21385 | 11940 | 6927 | 21842 | 12773 |
| Library pattern description | 23254 | 62825 | 29368 | 26700 | 22441 | 55077 | 155681 | 122986 |
| Total | 44683 | 78281 | 71314 | 108123 | 69139 | 83599 | 225971 | 174131 |

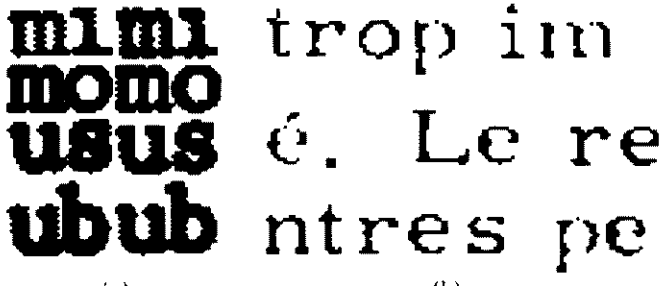
Table IV—Comparison of compression ratios

(a) Official CCITT documents (2376 × 1728 pels)

| Picture | Pattern Matching | Two-Dimensional CCITT Code | Increase Versus Two-Dimensional CCITT Code |
|---|---|---|---|
| CCITT1 | 63.1 | 28.3 | 122% |
| CCITT2 | 46.8 | 47.5 | −1% |
| CCITT3 | 26.0 | 17.9 | 46% |
| CCITT4 | 35.4 | 7.4 | 378% |
| CCITT5 | 30.2 | 15.9 | 90% |
| CCITT6 | 41.3 | 30.8 | 34% |
| CCITT7 | 15.3 | 7.4 | 106% |
| CCITT8 | 24.3 | 26.9 | −8% |

(b) Nonofficial CCITT document (2128 × 1728 pels)

| Picture | Pattern Matching | Symbol Matching | Two-Dimensional CCITT Code | Increase Versus Symbol Matching | Increase Versus Two-Dimensional CCITT Code |
|---|---|---|---|---|---|
| CCITT1 | 82.3 | 63.1 | 28.6 | 30% | 188% |
| CCITT2 | 47.0 | 38.1 | 45.4 | 23% | 3% |
| CCITT3 | 51.6 | 32.4 | 18.5 | 59% | 179% |
| CCITT4 | 34.0 | 25.6 | 7.5 | 33% | 353% |
| CCITT5 | 53.2 | 33.6 | 16.5 | 58% | 222% |
| CCITT6 | 44.0 | 29.5 | 30.2 | 49% | 46% |
| CCITT7 | 16.3 | 9.0 | 7.1 | 81% | 130% |
| CCITT8 | 21.1 | 17.8 | 22.1 | 19% | −5% |




(a) (b)

Fig. 18—Characters isolated in an unwanted way. (a) Characters clustered together. (b) Text containing characters isolated into several symbols.

2. The increase in compression ratio compared to the two-dimensional run-length code is quite variable. For documents containing mostly handwritten drawings and text, such as documents two and eight, there is sometimes a slight decrease in the compression ratio. That is because there are few matching patterns. For example, for document two, there are 736 patterns, but 448 of them are library patterns. For documents containing mostly text, such as document four, the compression ratio increases by a factor of about 4.5. For documents containing a mixture of text and drawings, the increase varies between 35 and 220 percent, depending on the content and

thresholding. It should be noted that for document seven, which contains printed ideograms, the increase in compression ratio is smaller than for regular printed text because there are more ideograms than letters, but the compression ratio still doubled.

3. The increase in compression ratio by pattern matching is 20 to 80 percent compared to the symbol matching of Ref. 2. The increase has been obtained by a combination of several factors. The most important are (a) isolation of nonsymbols (lead to significant improvement for documents three, four, five, and six, but has a slight negative effect on documents two and eight), (b) better matching, leading to fewer library patterns, and (c) improved coding efficiency obtained by sorting the patterns and by other coding modifications.

By looking at the coding length necessary for the different kinds of code words, in Table III, it is clear that the predominant part of the code is used for the description of the library patterns, accounting generally for more than 60 percent of the total coding length. Therefore, improving it can bring the highest reward. The improvement can be obtained by reducing the number of library patterns or by coding the pattern description more efficiently. The next most bit-consuming part is the coding of the horizontal position; it uses about 20 percent of the total coding length.

### *6.3 Complexity*

The pattern-matching coding has the disadvantage of being complex and time-consuming—the price to pay for an efficient coding. The most time-consuming parts are: the isolation, the template matching, and the matching decision. The isolation is both complex and time-consuming, and therefore the most difficult part, but by using fast logic, it is possible to isolate all the patterns in about one second. The template matching is a simple operation, but it takes a long time. It is therefore less of a challenge, since it is easily done in parallel and with simple hardware. The most time-consuming part of the matching decision uses local operators on, for example, $3 \times 3$ windows and can therefore also be realized without much complication. Most of the high-level operations are much slower and can be done by microprocessors. This system should not be more complicated than in Ref. 2.

An important factor is that the decoding is much easier and faster than the coding, since there is no isolation or matching. Such a technique is therefore particularly suited for transmission with one sender and several receivers.

An experimental pattern matcher has been built to show that the same kind of compression can be obtained when scanning real documents. By using a mixture of custom logic and programmed logic, transmission has reached speeds at rates up to 64 kb/s. A document is then usually sent in one to two seconds.

## VII. EXTENSIONS

Several improvements or different applications of a pattern matcher can be considered. Some of them will be described here.

### *7.1 Multipage document and prestored libraries*

When transmitting several pages of a document, the library from one page can be used for next page, thus reducing the number of library patterns for each page. In such cases, compressions up to eight times higher than with conventional coding techniques can be achieved. If a few fonts are prestored in the coder and decoder, the compression can be increased significantly.

### *7.2 Very high-quality transmission*

It is possible to use a tighter matching algorithm when even slight distortions are not tolerated. Such a mode can easily be implemented. It reduces the compression by an average of 15 percent. In that case, most of the postprocessing can be deleted.

### *7.3 Standardization*

The CCITT is looking into standardizing facsimile coding techniques for future facsimile machines communicating over digital links (Group four facsimile apparatus). The modified READ code (also called two-dimensional CCITT code) has been standardized. The pattern matching coding technique has been proposed by AT&T to the CCITT as an optional coding technique yielding much higher compression. The only difference in the proposal compared to this paper is that no cross decomposition is applied. The compression is therefore slightly lower.

### *7.4 High-resolution graphics*

Future scanners and coders will probably include resolutions higher than 200 pels/in. They will probably use 300 and 400 pels/in. The pattern matching technique can easily be modified for such resolution. The maximum size of the patterns should be increased to keep the coding efficient. In addition, the codes for the positioning of patterns must be slightly changed. The matching algorithm would stay unchanged. Compared to conventional techniques, the improvement in the compression will be as high and often even higher at such resolutions.

## VIII. CONCLUSION

A system for coding of facsimiles using pattern matching has been described. It allows an important increase in the compression ratio compared with a symbol matching system[2] and gives a compression

ratio that is up to 4.8 times that of conventional facsimile coding techniques. The improvement is naturally greater for printed text than for handwritten text. It is felt that further significant inprovements are possible by better matching and coding. An important observation is that pattern matching coding is very dependent on the digitization and thresholding. Therefore, the combination of the thresholding and the isolation could lead to significant improvements in compressions. Another consequence is that if a bad quality scanner is used, the pattern matching will hardly lead to higher compressions than conventional facsimile codes. With modern electronics components, a pattern matcher can be realized by hardware and would lead to an important reduction in the transmission costs of high-volume facsimiles.

## IX. ACKNOWLEDGMENT

The authors would like to thank A. N. Netravali for the numerous and fruitful discussions, E. F. Brown and E. S. Joseph for their contribution to the experimental unit, and S. M. Rubin for providing useful software.

## REFERENCES

1. R. B. Arps, "Binary image compression," in *Image Transmission Techniques*, W. K. Pratt, ed., New York:Academic Press, 1979.
2. W. K. Pratt, et al., "Combined Symbol Matching Data Compression System," Proc. IEEE, *68*, No. 7 (July 1980), pp. 786–96.
3. R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized text," IEEE Trans. Comput., *C-23* (November 1974) pp. 1174–9.
4. S. A. Dudani, "Region extraction using boundary following," in *Pattern Recognition and Artifical Intelligence*, C. H. Chen, ed., New York: Academic Press, pp. 216–32.
5. R. Hunger and A. H. Robinson, "International Digital Facsimile Coding Standards," Proc. IEEE, *68*, No. 7 (July 1980), pp. 854–67.

## AUTHORS

**Glenn L. Cash,** B.S.E.E., 1981, Monmouth College; Bell Laboratories, 1981—. Mr. Cash has worked on the design and implementation of a real-time pattern matching facsimile coder. His research interests include image processing, artificial intelligence, and computer system design. Member, Eta Kappa Nu, Sigma Pi Sigma.

**Ottar Johnsen,** E.E., 1974, Ph.D., 1979, Ecole Polytechnique Federale de Lausanne, Switzerland; Ecole Polytechnique Federale de Lausanne, 1974–1979; Bell Laboratories, 1979—. Mr. Johnsen was assistant at the Signal Processing Laboratory of the Ecole Polytechnique Federale de Lausanne, working in the field of picture coding. At Bell Laboratories, as a member of the Visual Communications Research Department, he is involved in research in picture processing and coding, and in communication systems.

**Jakub Segen,** M.S.E.E., 1971, Technical University of Wroclaw, Poland; Ph.D., 1980, Carnegie-Mellon University; Bell Laboratories, 1980—. Mr. Segen worked as a computer engineer at ZETO, Woclaw, and New York University. Recently, he has been involved with signal and image analysis, and his main research interests are machine perception and learning. He is a member of Robotics Systems Research Department, working on aspects of computer vision.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Data-Transport Performance Analysis of Fasnet

By D. P. HEYMAN*

(Manuscript received February 25, 1983)

This paper presents a queueing model to assess the performance of Fasnet, a recently invented local-area network. Fasnet is intended for high-speed lines capable of carrying a wide mix of traffic. We confine our attention to data traffic only. An approximate formula for the expected delay of a packet is obtained; the approximate formula compares favorably to simulations of Fasnet.

## I. INTRODUCTION

Fasnet is an implicit token-passing local-area network.[1] It is intended for high-speed lines capable of carrying a wide mix of traffic (data, voice, video, and facsimile). In this paper, we present a queueing model to assess the performance of Fasnet with data traffic only. An approximation for the expected delay of a packet is obtained; the approximate solution compares favorably with measurements taken from a simulation of Fasnet. Our numerical results yield a mean delay that is less than 1 ms for a 1-kb packet when the line speed is 100 Mb/s and the occupancy of the line is 0.9.

Section II consists of a brief description of Fasnet. Section III describes our model and its approximate solution and Section IV presents comparisons with simulations. The effects of bursty traffic are given in Section V and our conclusions are stated in Section VI.

## II. A BRIEF DESCRIPTION OF FASNET

We will now give a description of Fasnet that will enable the reader to appreciate the model in Section III. A complete description is given in Ref. 1.

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

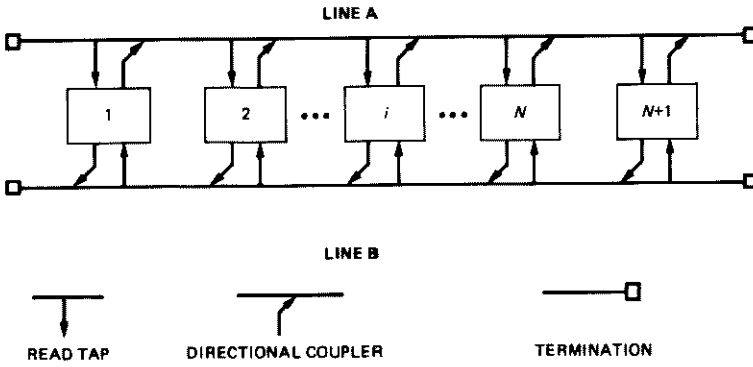



Fig. 1—Physical configuration of a Fasnet link.

The basic link as shown in Fig. 1 consists of two lines. One line carries traffic in one direction and the other line carries traffic in the reverse direction. For line A (which carries traffic from station 1 towards station $N$), station 1 is called the *head* station and station $N + 1$ is called the *end* station. For line B, the roles are reversed. Each station makes two connections to each line. A read tap precedes a passive directional coupler used for writing. The signal read from the read tap will be unaffected by the signal being written simultaneously on the line via the directional coupler. Except for specific fields of the header, the protocol ensures that only one station at a time writes on the line. Thus, once a message is written on a line, it is not removed or changed by any station.

The access control is similar for lines A and B. For line A, the head station (station 1) will initiate a *cycle*, which operates in the following way. One time each cycle, each station with packets destined toward the end station is allowed to access the line for a single time interval, during which at most $p_{\max}$ packets can be sent. A station knows when to place its packets on the line by reading a particular bit (called the busy bit) in the access control field. This bit is added to the message packet by the network layer of the protocol. Thus, in each cycle, station 1 has the first opportunity to send packets, station 2 has the second opportunity, and station $N$ has the $N$th and last* opportunity. Each station has exactly one opportunity per cycle to send packets. When station $N + 1$ receives a packet in which the busy bit indicates that the packet has not been used, it sends a message to station 1 (using line B) to start a new cycle. There may be synchronization delays at each end of the transmission of this message. The operation on line B is identical to the operation of line A, with station $N + 1$ as the head station and all flows reversed accordingly.

* We assume that station $N + 1$ will not send messages to itself.

## III. THE MODEL

Fasnet behaves as an implicit token-passing protocol because control passes from station to station as if a token were sent from station 1 to station 2, $\cdots$, to station $N + 1$ to station 1, and so forth. This suggests that a queueing model of queues served in cyclic order would be appropriate. In this type of model, there is a single server that visits $N + 1$ different queues in the cyclic order described above. For Fasnet, the server is conceptual: it is the opportunity to place packets on the line. The service time is the length of time to write a packet. The queues correspond to the buffers at each station.

Several papers have been written about queues served in cyclic order. In most of these papers, it is assumed that there is *exhaustive service* at each station. This means that the server processes all customers waiting at the station at the epoch that the server reaches the station. The most general model of exhaustive service is in Eisenberg,[2] where each queue is of the M/G/1 type and the times to travel between adjacent stations may depend on the pair of stations involved. The solution of this model is in terms of transforms that are not given in closed form; however, the equations can be solved numerically. A special case of the model in Ref. 2 is treated in Konheim and Meister.[3] Here, all service times are the constant $\Delta$, and all travel times between adjacent stations have the same distribution, which is concentrated on $\Delta$, $2\Delta$, $\cdots$. Konheim and Meister are able to obtain closed-form solutions for steady-state performance measures in this case. Their results will be used in our analysis. The only paper where service is not exhaustive is Ref. 4 by Eisenberg. That paper contains two M/M/1 type queues, and the server can process at most one customer during a visit to a server. The solution to this model requires extensive calculations, and the restriction to two stations is unrealistic for Fasnet.

We have chosen to seek approximate solutions where each station is of the M/D/1 type and service is either exhaustive or one-at-a-time (as in Ref. 4). These correspond to $p_{\max} = \infty$ and $p_{\max} = 1$, respectively. When the system is not heavily loaded, each station will have a small load so $p_{\max} = \infty$ should not behave much differently from $p_{\max} = 1$. This behavior is exhibited by our approximate solution and by simulations.

We will analyze our models by embedding them in a server-vacation model. In an M/G/1/FIFO queue, assume that at the end of each busy period the server takes a vacation. The vacations are iid random variables generically denoted by $T$. The expected delay of a customer, in the steady state, is given by

$$\mathrm{E}(D^*) = \mathrm{E}(D) + \frac{\mathrm{E}(T^2)}{2\mathrm{E}(T)}. \tag{1}$$

Here $E(D)$ is the Pollaczek-Khintchine formula for the expected delay in an M/G/1 queue: $E(D) = ab_2/2(1 - ab_1)$, where $a$ is the arrival rate and $b_1$ and $b_2$ are the first and second moments of the service times. A derivation of eq. (1) can be found in Levy and Yechiali.[5] Our first step is to describe the vacations.

### 3.1 Assumptions and notation

The notation used in this paper and some self-explanatory symbols are given below.

$N$ = number of potential transmitting stations.
$\lambda_i$ = packet arrival rate at station $i$.
$\Delta$ = constant service time/packet.
$\rho_i = \lambda_i\Delta$ = load due to station $i$.
$R = \sum_1^N \rho_i$ = load on the line.
$\Lambda = \sum_1^N \lambda_i$ = total arrival rate.
$\tau$ = one-way propagation delay.
$\gamma = 2\tau + \Delta$ = average overhead/cycle.

Lines A and B shown in Fig. 1 are the same except for direction, so it is sufficient to model only line A. Notice that station $N + 1$ does not send messages on line A and that station 1 does not send messages on line B. We assume that packets to be transmitted appear at station $i$ according to a homogeneous Poisson process with rate $\lambda_i$, $i = 1, 2, \cdots, N$. The arrivals at station $i$ and $j$ are independent when $i \neq j$. We assume that all the packets contain the same number of bits. The amount of time that a line spends taking a packet from a station is the time required to read the bits of the packet. Therefore, the service time of each packet is a constant, $\Delta$, say, where $\Delta$ is the number of bits/packet divided by the line speed in b/s. The time for a bit to travel between adjacent read taps is called the *walk time*; the average walk time between stations is denoted by $w$.

Let $\tau$ be the time to send a bit from station 1 to station $N + 1$ (and from $N + 1$ to 1). Then $2\tau$ is expended in each cycle to send a message indicating that the busy bit must be reset. An average of one-half of a packet processing time is lost in synchronization at each end station, so $\gamma = 2\tau + \Delta$ is the average overhead per cycle. Notice that $\tau$ is the time to walk from station 1 to station $N + 1$.

### 3.2 Analysis of the exhaustive service model

In this subsection, we assume that the packet arrival rates are the same at each station. We let $\lambda$ denote the common arrival rate and $\rho$ denote the common load on a station. The load on the line is $R = N\rho$. We also assume that stations $1, 2, \cdots, N + 1$ are evenly spaced along the line.

Let $T$ represent the amount of time between a departure from

station 1 and the beginning of the next visit to station 1. Then $T$ is the length of a vacation from station 1. By symmetry, $T$ is the length of a vacation between successive visits to any station. Thus, when a station gains access to the line, an average of $\lambda E(T)$ packets are present. With exhaustive service, the expected time to clear a station is $\lambda E(T)\Delta/(1-\rho)$. Since a vacation from station 1 consists of reading packets at stations 2 through $N$ and overhead,

$$E(T) = \gamma + (N-1)\frac{\lambda E(T)\Delta}{1-\rho},$$

so

$$E(T) = \frac{\gamma(1-\rho)}{1-R}, \qquad R < 1. \tag{2}$$

Equation (2) is a special case of eq. (54) in Ref. 2.

Let $X$ denote the number of packets at station 1 at the end of a vacation. Then

$$E(X) = \lambda E(T), \tag{3a}$$

and

$$\mathrm{Var}(X) = \lambda^2 \mathrm{Var}(T). \tag{3b}$$

Theorem 4.7 in Ref. 3 asserts that

$$\mathrm{Var}(X) = \frac{\lambda\rho^2[1-(N+1)\rho+(2N-1)\rho^2]}{(1-R)^2}. \tag{4}$$

From eqs. (2), (3), and (4) we obtain

$$\frac{E(T^2)}{E(T)} = \frac{\rho(N-1)\Delta + \gamma(1-\rho)^2}{(1-\rho)(1-R)}. \tag{5}$$

Equation (5) is exact when the walk times between any pair of stations are concentrated on $\Delta, 2\Delta, \cdots$. In Fasnet, we expect that the walk times are much less than $\Delta$ because the packets travel between stations at the speed of light and the read times are controlled by the speed of the line. Thus, we should regard eq. (5) as an approximation.

From eqs. (1) and (5),

$$\begin{aligned} E(D^*) &= \frac{\rho\Delta}{2(1-\rho)} + \frac{\rho(N-1)\Delta}{2(1-\rho)(1-R)} + \frac{\gamma(1-\rho)}{2(1-R)} \\ &= \frac{R\Delta}{2(1-R)} + \frac{\gamma(1-\rho)}{2(1-R)}, \qquad R < 1. \end{aligned} \tag{6}$$

By symmetry, eq. (6) gives the expected delay of a packet at any station. The expected number of packets in a buffer, $E(Q)$, say, is obtained from eq. (6) and Little's theorem:

$$E(Q) = \lambda E(D^*) = \frac{R\rho}{2(1-R)} + \frac{\gamma\lambda(1-\rho)}{2(1-R)}. \quad (7)$$

Consequently, the expected number of customers in buffers 1 through $N$ is

$$NE(Q) = \frac{R^2}{2(1-R)} + \frac{\gamma\Lambda(1-\rho)}{2(1-R)}. \quad (8)$$

Equation (7) is compared to simulation experiments in Section IV.

An intuitive understanding of eq. (6) may be gained by considering what happens for light loads. When $\rho$ is small, the vacations are almost of constant length because (mostly) no customers are served during a vacation. Then $E(T^2)$ is about $[E(T)]^2$. Now merge all the customers to obtain an M/D/1 queue with mean delay $R\Delta/[2(1-R)]$. Then eqs. (1) and (2) yield

$$E(D^*) = \frac{R\Delta}{2(1-R)} + \frac{\gamma(1-\rho)}{2(1-R)},$$

which is eq. (6). It is surprising that this heuristic light traffic argument produces the exact (modulo our other approximations) result for any $R < 1$.

### *3.3 Analysis of the one-at-a-time service model*

In this subsection we will obtain an approximate solution to a model where $p_{max} = 1$. The approximation is based on an idea used in Lehoczky, Sha, and Jensen[6] for a similar model. We do not assume that the arrival rates are the same at each station (as we did in Section 3.2).

A central notion in the approximation is the *completion time* of a station. The completion time of a station is the duration of the interval that starts when that station begins processing a packet, and ends at the first epoch that another packet may begin (does begin, provided it is present) its processing at the station. The purpose of the approximate solution is to estimate the mean and variance of the completion time, use these moments in the Pollaczek-Khintchine formula, and apply eq. (1) to an appropriate server-vacation model.

Let $V_i$ denote a generic completion time at station $i$, in the steady state. Then

$V_i$ = cycle overhead + $\Delta$(1 + number of other stations sending a packet in this cycle).

Let $p_i$ be the asymptotic proportion of time that packets are present at station $i$. For $T$ very large, the number of packets served at station $i$ by time $T$ is

$$\frac{p_i T}{\mathrm{E}(V_i)} + o(T),$$

where $o(T)$ is some function such that $o(T)/T \to 0$ as $T \to \infty$. The arrival rate at station $i$ is $\lambda_i$. Equating the arrival and departure rates yields

$$p_i = \lambda_i \mathrm{E}(V_i). \tag{9}$$

Now we make our first approximation.

*Approximation* 1: In each cycle, the probability that station $i$ transmits a packet is $b_i \triangleq \lambda_i \mathrm{E}(V_i)$.

The effect of approximation 1 is to use the long-run proportion $p_i$ as a probability for each cycle. Thus, the expected number of other stations sending a packet during a completion time of station $i$ is $\sum_{j \neq i} b_j$, so

$$\mathrm{E}(V_i) = \Delta \left(1 + \sum_{j \neq i} b_j\right) + \gamma. \tag{10}$$

Hence,

$$b_i \triangleq \lambda_i \mathrm{E}(V_i) = \rho_i \left(1 + \sum_{j \neq i} b_j\right) + \lambda_i \gamma, \qquad i = 1, 2, \cdots, N. \tag{11}$$

The solution of eq. (11) is

$$b_i = \frac{\rho_i}{1 + \rho_i} \frac{1 + \gamma/\Delta}{1 - \alpha}, \qquad \alpha = \sum_1^N \frac{\rho_i}{1 + \rho_i}, \qquad i = 1, 2, \cdots, N, \tag{12}$$

which can be verified by substitution into eq. (11). When $\rho_i \equiv \rho$ and $\gamma = 0$, eq. (12) becomes

$$b_i = \frac{\rho}{1 - (N - 1)\rho} = \frac{R/N}{1 - \dfrac{N - 1}{N} R} \qquad \text{for all } i,$$

which is the approximation given in Lehoczky et al.[6]

Since $b_i$ must be no larger than one (because it is a probability), eq. (12) constrains the feasible values of $\{\rho_i\}$. When all the arrival rates have the common value $\lambda$, $\alpha = N\rho/(1 + \rho) = R/(1 + \rho)$, so

$$b_i < 1 \Leftrightarrow N > \frac{R\gamma/\Delta}{1 - R}. \tag{13}$$

Equation (13) shows that for a given total load, $R$, stability is achieved only when the load is shared by a sufficiently large number of stations. For Fasnet, $\gamma/\Delta = 3$ is a typical value. Then eq. (13) asserts that for $R = 0.9$, $N > 27$ is required for stability; for $R = 0.8$,

$N > 12$ is required for stability; and for $R = 0.5$, $N \geq 8$ is required for stability.

Here is why $N$ cannot be too small. When $N$ is small, the overhead per cycle ($\gamma$) will be spread over a few customers, which has the effect of decreasing the capacity of the line. To see this more precisely, let $c$ be the expected number of packets processed in a cycle. Then use eq. (11) to obtain

$$c = \sum_1^N b_i = \frac{1 + \gamma/\Delta}{1 - \alpha} \sum_1^N \frac{\rho_i}{1 + \rho_i} = \frac{(1 + \gamma/\Delta)\alpha}{1 - \alpha}. \tag{14}$$

When all the arrival rates have the common value $\lambda$, eq. (14) yields

$$c = \frac{R(1 + \gamma/\Delta)}{1 + \rho - R}, \tag{15}$$

where $\rho = \lambda\Delta$. From eq. (15) we compute the average amount of overhead expended per packet per cycle* $\gamma/c$:

$$\frac{\gamma}{c} = \frac{1 - R + R/N}{R(1 + \gamma/\Delta)}\gamma = \frac{\gamma}{1 + \gamma/\Delta}\left(\frac{1 - R}{R} + \frac{1}{N}\right). \tag{16}$$

Eq. (16) shows that the average overhead per packet is a decreasing function of $N$, so if $N$ were small, the overhead per packet might cause the line to be overloaded.

It is interesting to note that this consideration does not arise in Section 3.2. When $p_{\max} < \infty$, from time to time a station will stop transmitting packets because its quota for the cycle has been filled. This is the effect that is shown in eq. (16). We conjecture that when all other parameters are fixed, $\gamma/c$ is a decreasing function of $p_{\max}$.

Now we turn to an approximation for the mean delay. This will be done by proposing a suitable server-vacation model and applying eq. (1). The service-time moments in the Pollaczek-Khintchine formula correspond to completion-time moments here. When a packet arrives at station $i$, and no other packets are waiting to be transmitted at station $j$, that packet cannot be transmitted until station $i$ gains access to the line. The length of time that station $i$ does not have access to the line is the length of time to process the other stations in a cycle plus the overhead time, which is the completion time less one service time. Thus, $T_i = V_i - \Delta$, so

$$\mathrm{E}(T_i) = \mathrm{E}(V_i) - \Delta, \tag{17}$$

---

* The fact that $\gamma/c$ is the right quantity to compute may not be obvious. A rigorous proof could use ergodic theory for regenerative processes (see, e.g., Section 6-4 of Heyman and Sobel,[7] particularly Theorem 6-8).

and

$$\mathrm{Var}(T_i) = \mathrm{Var}(V_i). \tag{18}$$

From eqs. (10), (12), and (17) we can obtain $E(T_i)$. From eqs. (1) and (18) we see that only $\mathrm{Var}(V_i)$ remains to be obtained. To get it, we make our second approximation.

*Approximation* 2: In each cycle, the event that station $i$ transmits a packet is independent of the event that station $j$ transmits a packet for every $j \neq i$.

The effect of approximation 2 is that the variance of the number of packets served at station $i$ is $b_i(1 - b_i)$ and the variance of the number of packets served in a cycle is $\sum_{i=1}^{N} b_i(1 - b_i)$. This yields the approximation

$$\mathrm{Var}(V_i) = \Delta^2 \sum_{j \neq i} b_j(1 - b_j). \tag{19}$$

Using eqs. (10), (12), (17), (18), and (19) in eq. (1) produces our approximation for the expected delay at station $i$, $i = 1, 2, \cdots, N$. The resulting formula does not appear to provide any insight and is omitted.

As a partial check on the efficacy of our approximation for the mean delay, we consider the limiting case of no overhead and identical stations. In this situation, the total content of the output buffers at stations 1, 2, $\cdots$ $N$ fluctuates as the queue length in an M/D/1 queue with arrival rate $\lambda$ and service time $\Delta$. From the Pollaczek-Khintchine formula, the expected queue length in the steady state, $\mathrm{E}(Q_0)$, say, is

$$\mathrm{E}(Q_0) = \frac{R^2}{2(1 - R)}, \qquad R < 1.$$

Our approximations produce (after some algebra)

$$\mathrm{E}(\hat{Q}_0) = \frac{R^2}{2(1 - R + \rho)} \frac{\rho[R + (1 - R)(R - \rho)] + 1 - R}{1 - R}, \qquad R < 1.$$

Now let $N \to \infty$ and $\rho \downarrow 0$ with $R = N\rho$ held fixed. This represents a system with many lightly loaded stations. Then

$$\mathrm{E}(\hat{Q}_0) \to \frac{R}{2(1 - R)} \qquad \text{as} \quad \rho \downarrow 0.$$

In this limiting case, $\mathrm{E}(\hat{Q}_0)$ overestimates $\mathrm{E}(Q_0)$ by $R/2$ and, the relative error is $(1 - R)/R$. Thus, the absolute error increases with $R$ and is less than one-half, and the relative error decreases as the absolute error increases.

## IV. COMPARISONS WITH SIMULATIONS

A computer simulation of Fasnet has been constructed. There are $N = 50$ sending stations equally spaced along the line. The propagation time ($\tau$) equals the packet processing time at each station ($\Delta$). We have chosen to consider 1000-bit packets and a line speed of 100 Mb/s, which is representative of the operating region. Then $\Delta = 10$ $\mu$s, and $\lambda = (R/50) \times 10^5$ packets/s.

The measure of performance is the average queue length at the stations. Specifically, the simulation estimates the steady-state distribution of the queue length at station $i$ and then computes the mean, $E(Q_i)$, say. The average queue length is $\sum_1^{50} E(Q_i)/50 \triangleq E(Q)$. The corresponding quantity from our formulas is called $E(\hat{Q})$. From the queueing formula $E(Q) = \lambda E(D)$ we use $E(\hat{Q})$ to estimate the average delay of a packet, $E(\hat{D})$.

Table I shows the results.

The analytic approximation adequately replicates the simulation results. Table I and Fig. 2 demonstrate that for $R$ as large as 0.8, $p_{max} = 1$ and $p_{max} = \infty$ produce nearly the same average queue size. This means that the efficiency (in terms of not incurring too much overhead) of $p_{max} = \infty$ and the protection against a few stations dominating the line of $p_{max} = 1$ can be simultaneously obtained by setting $1 < p_{max} < \infty$. Table I shows that $p_{max} = 3$ is almost as efficient as $p_{max} = \infty$.

The approximation for the mean delay of a packet is less than 1/2 ms even when $R = 0.9$ and $p_{max} = 1$.

## V. THE EFFECTS OF BURSTY TRAFFIC

In this section we return to the exhaustive service model and replace the assumption that packets arrive according to a Poisson process with the assumption that packets arrive according to a compound Poisson process. Fuchs and Jackson give statistical analyses of arrival times for terminal-to-computer calls.[8] Two of their conclusions are as follows:

1. The exponential distribution is a reasonably good approximation of the times between bursts.

Table I—Comparison of simulations and analytic approximations

| | $p_{max}$ | E(Q) Simulation | $E(\hat{Q})$ Analysis | $E(\hat{D})$ |
|---|---|---|---|---|
| $R = 0.8$ | 1 | 0.216 | 0.209 | 130 $\mu s$ |
| | 3 | 0.184 | — | — |
| | $\infty$ | 0.184 | 0.150 | 94 $\mu s$ |
| $R = 0.9$ | 1 | 0.684 | 0.782 | 430 $\mu s$ |
| | 3 | 0.440 | — | |
| | $\infty$ | 0.398 | 0.346 | 192 $\mu s$ |

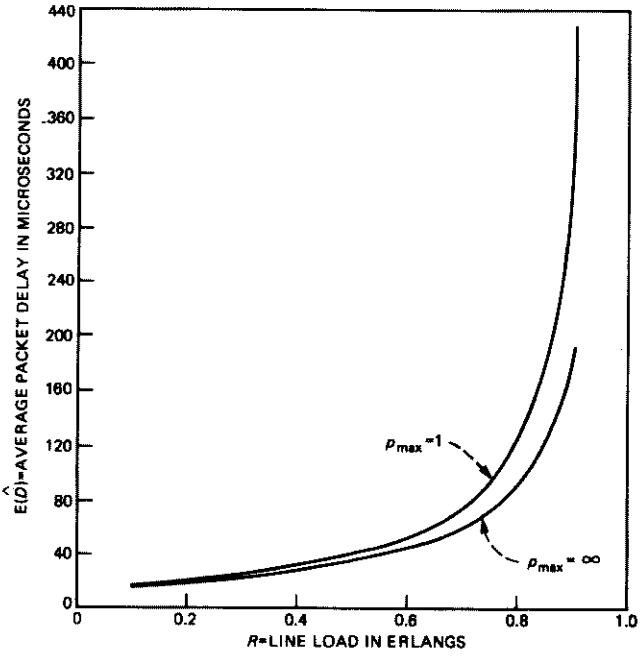



Fig. 2—Expected delay vs $R$ for 1-kb packets, 50 stations, and 100-Mb/s line speed.

2. The size of a burst (measured in various ways) has a geometric distribution.

Recent analyses by Morgan of host-to-host file traffic indicate that the assumption of Poisson arrivals may not be justified.[9]

The purpose of this section is to find out how sensitive the average delay is to the assumption of Poisson arrivals. We will see that in the exhaustive service model, the average delay can be significantly greater with bursty arrivals than with Poisson arrivals with the same rate.

Specifically, we assume that the bursts arrive according to a Poisson process with rate $\lambda_b$ and the burst sizes $B_1, B_2, \cdots$ are iid with

$$P\{B_1 = i\} = (1 - \xi)\xi^{i-1}, \qquad i = 1, 2, \cdots.$$

This arrival process can be interpreted as one where messages arrive according to a Poisson process with rate $\lambda_b$, and the $j$th message consists of a random number of packets with a geometric distribution. One reason for choosing a geometric distribution is that it equates the average delay of a packet and the average delay of a message (see Halfin[10]). (The delay of a message is the delay of its last packet.) Another reason is that it uses only one parameter, and so we can

specify the mean ($Mt$, say) and variance ($Vt$, say) of the number of arrivals in an interval of length $t$ and then solve for $\lambda_b$ and $\xi$. Doing so yields

$$\lambda_b = \frac{2M^2}{M + V} \quad \text{and} \quad \xi = \frac{V - M}{V + M}.$$

Letting $z = V/M \geq 1$ yields

$$\lambda_b = \frac{2M}{1 + z} \quad \text{and} \quad \frac{z - 1}{z + 1}. \tag{20}$$

Equation (20) relates the parameters we might obtain from measurements, $M$ and $z$, to the parameters of the model, $\lambda_b$ and $\xi$.

We will now obtain the delay of an arbitrary packet in the steady state. The analysis is similar to the analysis in subsection 3.2; as before we assume that the stations have statistically identical arrival processes.

The analog of the Pollaczek-Khintchine formula for compound Poisson arrivals is given in Burke.[11] In our notation, the formula is

$$\mathrm{E}(D) = \frac{\lambda_b \Delta^2 (1 + \xi)}{2(1 - \rho)} + \frac{\xi \Delta}{1 - \xi}, \tag{21}$$

where $\rho = \lambda_b \Delta/(1 - \xi) = M\Delta$.

To obtain the mean and variance of the vacation times, let $X$ denote the number of packets at station 1 at the end of a vacation. From theorem 4.7 in Ref. 3, for $R < 1$

$$\mathrm{E}(X) = \frac{\gamma M(1 - \rho)}{1 - R}, \tag{22a}$$

and

$$\mathrm{Var}(X) = \frac{XV}{(1 - R)^2}[1 - (N + 1)\rho + (2N - 1)\rho^2]. \tag{22b}$$

Since $X$ is the number of packets that arrive in an interval of length $T$,

$$\mathrm{E}(X) = M\mathrm{E}(T), \tag{23a}$$

and

$$\begin{aligned} \mathrm{Var}(X) &= \mathrm{E}[\mathrm{Var}(X \mid T)] + \mathrm{Var}[\mathrm{E}(X \mid T)] \\ &= \mathrm{E}[VT] + \mathrm{Var}[MT] \\ &= V\mathrm{E}(T) + M^2\mathrm{Var}(T). \end{aligned} \tag{23b}$$

From eqs. (22) and (23) we obtain

$$\frac{E(T^2)}{E(T)} = \frac{\mathrm{Var}(X)}{ME(X)} - \frac{V}{M^2} + E(T)$$

$$= \frac{(N-1)V\Delta^2 + \gamma(1-\rho)^2}{(1-\rho)(1-R)}. \tag{24}$$

Substituting eqs. (21) and (24) into eq. (1) yields

$$E(D^*) = \frac{\lambda_b \Delta^2(1+\xi)}{2(1-\rho)} + \frac{\xi\Delta}{1-\xi} + \frac{(N-1)V\Delta^2 + \gamma(1-\rho)^2}{2(1-\rho)(1-R)}.$$

Using eq. (20) yields

$$E(D^*) = \frac{R\Delta z}{2(1-R)} + \frac{\gamma(1-\rho)}{2(1-R)} + \frac{(z-1)\Delta}{2} - \frac{\rho\Delta z(z-1)}{2(1-\rho)(1+z)}. \tag{25}$$

To compare eqs. (6) and (25), let a subscript $z$ denote batch arrivals with variance to mean ratio $z$. Then

$$E(D_z^*) - E(D^*) = \frac{\Delta(z-1)}{2(1-R)} - \frac{\rho\Delta z(z-1)}{2(1-\rho)(1+z)}.$$

When $N$ is large, so that $\rho$ is much smaller than $R$, the first term dominates, especially in heavy traffic.

Table II shows the values of $E(D_z^*)$. The case $E(D_1^*)$ represents Poisson arrivals. The data are the same as in Section IV: $\Delta = \tau = 10$ $\mu$s, and $N = 50$.

Table II shows that bursty traffic can have much larger expected delays than Poisson traffic with the same arrival rate. A crude approximation of the increase is $E(D_z^*) = E(D_1^*)\sqrt{z-1}$ for $2 \le z \le 10$. Even when $z = 10$ and $R = 0.9$, the mean delay is less than 1 ms.

## VI. CONCLUSIONS

We have three conclusions. The first is that the approximations presented in Section III are sufficiently accurate for data transport performance studies of Fasnet. The second is that $p_{\max} = 3$ appears to be a good choice if the offered traffic is reasonably smooth (Poisson) and approximately equal to all stations. The third is that Fasnet should be able to provide 1-ms average-delay performance for bursty traffic.

Table II—$E(D_z^*)$ in $\mu$s for several values of $z$

| | $E(D_1^*)$ | $E(D_2^*)$ | $E(D_5^*)$ | $E(D_{10}^*)$ |
|---|---|---|---|---|
| $R = 0.8$ | 94 | 119 | 193 | 318 |
| $R = 0.9$ | 192 | 242 | 392 | 642 |

## VII. ACKNOWLEDGMENTS

I would like to thank Lois E. Flamm for running the simulations, and John Limb and Jon Mark for their comments on a draft of this paper.

## REFERENCES

1. J. O. Limb and C. Flores, "Description of Fasnet—A Unidirectional Local-Area Communications Network," B.S.T.J., *61,* No. 7, Part 1 (September 1982), pp. 1413–40.
2. M. Eisenberg, "Queues With Periodic Service and Changeover Times," Oper. Res., *20,* No. 2 (March–April 1972), pp. 440–51.
3. A. G. Konheim and B. Meister, "Waiting Lines and Times in a System with Polling," J. Ass. Comput. Mach., *21,* No. 3 (July 1974), pp. 470–90.
4. M. Eisenberg, "Two Queues with Alternating Service," SIAM J. Appl. Math., *36,* No. 2 (April 1979), pp. 287–303.
5. Y. Levy and U. Yechiali, "Utilization of Idle Time in an $M/G/1$ Queueing System," Manage. Sci., *22,* No. 2 (October 1975), pp. 202–11.
6. J. P. Lehoczky, L. Sha, and E. D. Jensen, "A Comparative Study of Variable TDM Schemes and CSMA Schemes," Technical Report No. 27, Department of Statistics, Carnegie-Mellon University (August 1981).
7. D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research,* Vol. I, New York: McGraw-Hill, 1982.
8. E. Fuchs and P. E. Jackson, "Estimates of Distributions of Random Variables for Certain Communications Traffic Models," Commun. ACM, *13,* No. 12 (December 1970), pp. 752–7.
9. S. P. Morgan, unpublished work.
10. S. Halfin, "Batch Delays Versus Customer Delays," B.S.T.J., *62,* No. 7, Part 1 (September 1983), pp. 2011–15.
11. P. J. Burke, "Delays in Single-Server Queues with Batch Input," Oper. Res., *23,* No. 4 (July–August 1975), pp. 830–3.

## AUTHOR

**Daniel P. Heyman,** B.Mgt.E. (Management Engineering), 1960, Rensselaer Polytechnic Institute; M.I.E. (Industrial Engineering), 1962, Syracuse University; Ph.D. (Operations Research), 1966, University of California, Berkeley; U.S. Air Force, 1960–1963; Bell Laboratories, 1966—. Mr. Heyman has worked on a variety of operations research problems at Bell Laboratories, including studies of queues, inventories, and financial markets. He was a visiting faculty member at Yale University for the 1976–1977 academic year. Member, Sigma Xi, Alpha Pi, Mu, Operations Research Society of America.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Forecasting With Adaptive Gradient Exponential Smoothing

By A. FEUER*

(Manuscript received December 20, 1982)

Exponential Smoothing (ES) as a forecasting technique has been extensively used since its introduction in the 1960s. It is simple, hence easy to implement, and in many cases performs surprisingly well. However, many phenomena require a more sophisticated forecasting technique. In this paper we introduce a new forecasting technique, Adaptive Gradient Exponential Smoothing (AGES). This technique extends the classical ES as used on simple data or on data with linear trend. For data with both linear trend and seasonal effects this extension results in a new and more general form of ES, which is presented in this paper. The new forecasting technique is tested on simulated data and some real data of the types mentioned above, and its performance in all these tests is clearly superior to ES. It is shown by analysis and by the experimentations that for certain types of data it does in fact converge to the optimal (in the mean square error sense) forecasts.

## I. INTRODUCTION

The need for quick and reliable forecasts of various time series is often encountered in economic and business situations. In the Bell System, forecasting is used to help plan trunk and facilities for the telephone network,[1-3] as well as to project computer workload, to determine staffing levels for operators or service observers, and more.

Many forecasting techniques exist and different time series may require different techniques. In general, there is a clear trade-off between simplicity (resulting in cheaper implementation) and per-

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

formance of the forecasting technique. One of the simplest forecasting techniques, Exponential Smoothing (ES), has surprisingly good performance. This technique was presented originally by Winters[4] and Brown[5] and is described briefly in Section II. In Ref. 6 the optimality properties of ES are studied and we expand on these studies and use the conclusions as the basis for a new technique we introduce here.

In fact, these studies revealed a relationship between the ES and the Autoregressive-Integrated Moving Average (ARIMA) model-fitting-based forecasting suggested by Box and Jenkins.[7] This is further discussed in Section III.

The extensive use of ES clearly indicated that for time series with nonstationary discontinuities or changes in the generating parameters, ES performance is not satisfactory. This prompted a number of researchers to develop the Adaptive Exponential Smoothing (AES) idea. In these techniques the algorithm is supposedly evaluating its own performance and correcting its parameters to obtain improved performance. Recently, the existing AESs (see, for example, Refs. 8 through 11) were reviewed critically by Ekern.[12] One of the points raised in Ref. 12 was that none of the existing AESs is supported by analysis or general performance claims (e.g., optimality). In addition, it should be pointed out that only Roberts' and Reed's AES[11] can be used on data with both linear trend and seasonal effects, while the other AESs are limited to simpler data and have no natural generalization.

In this paper we present a new AES algorithm, which we call Adaptive Gradient Exponential Smoothing (AGES). This technique naturally generalizes to data with both linear trend and seasonal effect. In addition, analysis of AGES for simple data and extensive simulations, using simple as well as more general data, strongly suggests that this technique converges to optimal performance in the mean square error (MSE) sense.

Section II presents ES as commonly used. A new, more general form is developed with a discussion of its optimal properties. The new technique, AGES, is derived and presented in Section III, while the results of experiments with this technique on both real and simulated data are presented in Section IV.

## II. EXPONENTIAL SMOOTHING AND ITS OPTIMAL PROPERTIES

First we consider ES as Winters[4] did for three types of data: simple* (S), with linear trend (LT), and with both linear trend and multiplicative seasonal effects (LSM). Common to all the configurations is

*Simple data are of the form $a + n(t)$, where $a$ is a fixed value and $n(t)$ is noise with zero mean.

the following: a time series $\{x(t)\}$ is measured every time interval $T$ (e.g., hour, day, or week), and $t$ is an integer representing the time $tT$. Then, one is interested in forecasting the value $x(t+1)$* based on the data available up to and including $t$, namely $x(0), x(1), \cdots, x(t)$.

If $\hat{x}(t+1)$ denotes the forecast, carried out at time $t$ for $x(t+1)$, from Ref. 4 we have (using our own notation for consistency with the discussions in the sequel), for S data:

$$\hat{x}(t+1) = \alpha x(t) + (1-\alpha)\hat{x}(t) \tag{1a}$$

$$0 \le \alpha \le 1; \tag{1b}$$

for LT data:

$$\hat{x}(t+1) = \hat{a}(t) + \hat{b}(t) \tag{2a}$$

$$\hat{a}(t) = \alpha x(t) + (1-\alpha)[\hat{a}(t-1) + \hat{b}(t-1)] \tag{2b}$$

$$\hat{b}(t) = \beta[\hat{a}(t) - \hat{a}(t-1)] + (1-\beta)\hat{b}(t-1) \tag{2c}$$

$$0 \le \alpha, \beta \le 1; \tag{2d}$$

and for LSM data:

$$\hat{x}(t+1) = (\hat{a}(t) + \hat{b}(t))\hat{c}(t-L+1) \tag{3a}$$

$$\hat{a}(t) = \alpha \frac{x(t)}{\hat{c}(t-L)} + (1-\alpha)[\hat{a}(t-1) + \hat{b}(t-1)] \tag{3b}$$

$$\hat{b}(t) = \beta[\hat{a}(t) - \hat{a}(t-1)] + (1-\beta)\hat{b}(t-1) \tag{3c}$$

$$\hat{c}(t) = \gamma \frac{x(t)}{\hat{a}(t)} + (1-\gamma)\hat{c}(t-L) \tag{3d}$$

$$0 \le \alpha, \beta, \gamma \le 1, \tag{3e}$$

where $L$ is the known periodicity of the season.

In all the equations above, the parameters $\alpha$, $\beta$, and $\gamma$ are called the "smoothing coefficients".

Our first step is to rewrite eq. (1) and, more importantly, eq. (2). This provides the basis for a new form of ES for LSM data, more general than (3). The new form, which is a natural extension of (1) and (2), suggests types of data for which the ES algorithm can result in optimal (in the MSE sense) performance.

Equation (1) can be readily rewritten as

$$\hat{x}(t+1) = \hat{\theta}_1 \hat{x}(t) + (1-\hat{\theta}_1)x(t), \tag{4a}$$

*Note that we restrict our discussions to one-interval-ahead forecasting with the understanding that it can be generalized to more time intervals ahead.

where clearly

$$\hat{\theta}_1 = 1 - \alpha. \tag{4b}$$

With some algebra one can show that eq. (2) is equivalent to

$$\hat{x}(t+1) = \hat{\theta}_1 \hat{x}(t) + \hat{\theta}_2 \hat{x}(t-1) + (2 - \hat{\theta}_1)x(t) - (1 + \hat{\theta}_2)x(t-1), \tag{5a}$$

where

$$\hat{\theta}_1 = 2 - \alpha(1 + \beta) \tag{5b}$$

$$\hat{\theta}_2 = \alpha - 1. \tag{5c}$$

The basic difference between (2) and (5) is that (5) reflects the assumption that the noise-free part of the data $x(t)$ is generated by the difference equation

$$y(t) - 2y(t-1) + y(t-2) = 0, \tag{6}$$

while (2) reflects the assumption that the solution of (6) is

$$y(t) = a + bt. \tag{7}$$

[Note that in (2) $\hat{a}(t)$ is the current estimate of '$a + bt$' and $\hat{b}(t)$ is the current estimate of '$b$.']

The ES as given in (3) for LSM data is clearly based on the assumption that the noise-free part of the data has the form

$$y(t) = (a + bt)c(t), \tag{8a}$$

where

$$c(t + L) = c(t). \tag{8b}$$

The difference equation satisfied by (8) is

$$y(t) - 2y(t - L) + y(t - 2L) = 0, \tag{9}$$

and the corresponding ES

$$\hat{x}(t+1) = \sum_{j=1}^{M} \hat{\theta}_J \hat{x}(t-j+1) - \sum_{j=1}^{M} \hat{\theta}_j x(t-j+1) + 2x(t-L+1) - x(t-2L+1). \tag{10}$$

The parameters $\hat{\theta}_j$, $j = 1, \cdots, M$ and the constraints they have to satisfy are discussed later. Also, the claimed correspondence between (9) and (10) will become more apparent in later discussion.

At this point, however, we emphasize that while (7) is the general solution of (6), and thus (2) and (5) are equivalent, (8) is only one of

many possible solutions of (9). Hence (10) represents an ES form that is more general than (3).

Similarly, data with linear trend and additive seasonal effects* (LSA) have the underlying difference equation

$$y(t) - y(t-1) - y(t-L) + y(t-L-1) = 0 \tag{11}$$

and the corresponding ES is

$$\hat{x}(t+1) = \sum_{j=1}^{M} \hat{\theta}_j \hat{x}(t-j+1) - \sum_{j=1}^{M} \hat{\theta}_j x(t-j+1) + x(t) + x(t-L+1) - x(t-L). \tag{12}$$

To unify and simplify the discussions ahead we introduce the following notation. Let $D$ be a unit delay operator, namely $Dx(t) = x(t-1)$, and let $A(D)$ be a polynomial in $D$ such that

$$A(D) = \begin{cases} 1 & \text{for S data} \\ 2 - D & \text{for LT data} \\ 2D^{L-1} - D^{2L-1} & \text{for LSM data} \\ 1 + D^{L-1} - D^{L} & \text{for LSA data.} \end{cases} \tag{13}$$

With these definitions (4), (5), (10), and (12) can be unified as

$$\hat{x}(t+1) = \sum_{j=1}^{M} \hat{\theta}_j D^{j-1}(\hat{x}(t) - x(t)) + A(D)x(t), \tag{14}$$

where $M = 1$ will result in (4) and $M = 2$ in (5).

It should also be pointed out that the ES as given by eq(s). (1) [(2) or (3)] has an implicit assumption in it. The assumption is that one (two or three) coefficient(s) can, in fact, smoothen the data. In other words, $M$ in (14) is equal to one (two or three). However, its general form, (14), allows for a larger number of coefficients to get better approximations.

To observe the optimal properties of the ES forecasts we define the forecast error as

$$e(t) = x(t) - \hat{x}(t) \tag{15}$$

and use as our criteria for the forecast quality the mean square error (MSE), i.e., $E\{e^2(t)\}$. With this in mind, it is clear that optimal performance is achieved if the $e(t)$ becomes a white noise sequence (i.e., independent and identically distributed with zero mean). Namely, the ES technique, while assuming knowledge of the generating process for the noise-free component of the data, attempts to "whiten" the

*This type of data was not addressed in Ref. 4 and, as far as we know, no form of ES applicable to it was proposed before the one here.

noise component. This attempt implies an underlying assumption that the data are generated through, or at least approximated by, the process

$$[1 - DA(D)]x(t) = \left[1 - \sum_{j=1}^{M} \theta_j D^j\right] \epsilon(t), \tag{16}$$

where $\epsilon(t)$ is a white noise with variance $\sigma^2$.

Substituting (14) and (16) into (15) results in

$$\left[1 - \sum_{j=1}^{M} \hat{\theta}_j D^j\right] e(t) = \left[1 - \sum_{j=1}^{M} \theta_j D^j\right] \epsilon(t). \tag{17}$$

This equation satisfied by $e(t)$ is the basis of our claims for correspondence between eqs. (9) and (10), and (11) and (12). Equation (17) immediately suggests the conditions for optimal forecasting. First, to get bounded MSE one must require:

*Condition 1*: All zeros of the polynomial $[1 - \sum_{j=1}^{M} \hat{\theta}_j \lambda^j]$ are outside the unit circle.

If, in addition, we also require:

*Condition 2*: $\hat{\theta}_j = \theta_j$, $j = 1, 2, \cdots, M$,

then, clearly, from eq. (17), $e(t)$ will converge to $\epsilon(t)$ and optimal forecasting (in the MSE sense) is achieved.

*Remark 1*: As we discussed here, the sufficiency of Conditions 1 and 2 is quite obvious; however, they are also necessary. This is argued in Appendix A.

*Remark 2*: In Ref. 4 $\alpha$ and $\beta$ for LT data are restricted to interval [0, 1], which corresponds to the set $S_2$ in Fig. 1. The actual constraints follow from applying Condition 1 to the $M = 2$ case. This results in the set $S_1$ in Fig. 1, which clearly contains $S_2$ and is considerably larger. Allowing for a larger constraint set for $\hat{\theta}_1$ and $\hat{\theta}_2$ (or, correspondingly, $\alpha$, $\beta$) will result in more cases for which ES could result in optimal performance.

## III. ADAPTIVE GRADIENT EXPONENTIAL SMOOTHING

In the previous section we argued that for data that can be approximated by (16), forecasting with ES of the form (14) can result in optimal performance in the MSE sense. To achieve this, Conditions 1 and 2 must be satisfied. However, while Condition 1 can be satisfied by proper choice of $\hat{\theta}_j$, Condition 2 is, in general, hard to satisfy since the values of $\theta_j$ in eq. (16) are not known. Basically, the ARIMA

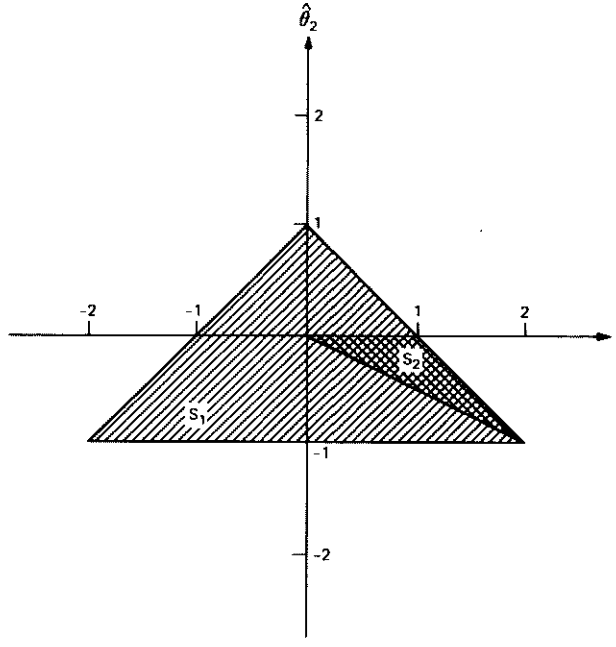



Fig. 1—The constraint sets:

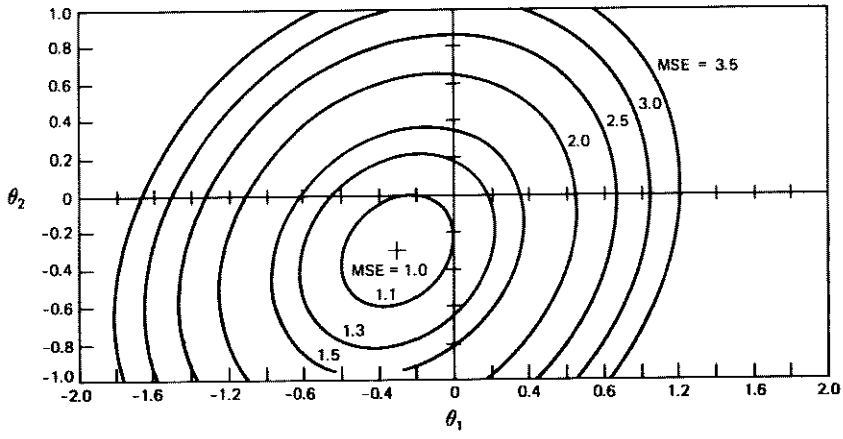
$$S_1 = \{(\hat{\theta}_1, \hat{\theta}_2): |\hat{\theta}_2| < 1, \hat{\theta}_2 + \hat{\theta}_1 < 1, \hat{\theta}_2 - \hat{\theta}_1 < 1\}$$

$$S_2 = \{(\hat{\theta}_1, \hat{\theta}_2): \hat{\theta}_1 = 2 - \alpha(1 + \beta), \hat{\theta}_2 = \alpha - 1, 0 < \alpha, \beta < 1\}.$$

model-fitting-based forecasting[7] deals with exactly this type of problem. The $\theta_j$'s of eq. (16) are estimated and these estimates are then used as the $\hat{\theta}_j$'s in eq. (14) in an attempt to satisfy Condition 2. In the ES algorithm no such attempt is made. In practice, the forecasters using ES choose some fixed values for the $\theta_j$, which satisfy Condition 1 [or even more restrictively, e.g., eq. (2d)]. These values are based on intuition, experience, and familiarity with the data they forecast.

However, considerable differences between the underlying $\theta_j$'s and the chosen $\hat{\theta}_j$'s can result in significant performance degradation. This is demonstrated in Fig. 2 for the case $M = 2$. The MSE for this case was computed in a closed form as a function of $\theta_1$ and $\theta_2$ for some fixed $\hat{\theta}_1$ and $\hat{\theta}_2$ and graphed in the figure. Together with phenomena like nonstationary discontinuity* and changes in the data-generating process (i.e., the $\theta_j$ change values), this resulted in unsatisfactory performance of the ES. The realization of what may cause this poor performance brought about the idea of using adaptive schemes where

*Step-like changes in the data.




MSE – MEAN SQUARED ERROR

Fig. 2—The mean squared error as a function of the data-generating parameters for $M = 2$. (The smoothing coefficients are fixed at $\hat{\theta}_1 = -0.3$, $\hat{\theta}_2 = -0.3$.)

the $\hat{\theta}_j$ are not fixed but are adjusted in an attempt to improve performance.

Compared to the existing Adaptive Exponential Smoothing (AES) techniques (see, e.g., Refs. 8 through 11), the new technique we introduce here is analytically more sound and there are strong indications that it converges to opptimal performance in the MSE sense for the data approximated by (16).

This new technique is based on the gradient search for the minimum of the MSE. If the MSE would have been available as a function of the $\theta_j$, then one could compute the gradient

$$\nabla = \frac{\partial E\{e^2(t)\}}{\partial \hat{\theta}}, \tag{18}$$

where $\hat{\theta} = [\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_M]^T$, and recursively update the $\hat{\theta}_j$ through

$$\hat{\theta}(t+1) = \hat{\theta}(t) - \mu\nabla, \tag{19}$$

where $\mu > 0$ is the adaptation constant. This is the gradient search technique, sometimes referred to as the steepest descent technique. In general, however, the MSE is not available as a function of the $\hat{\theta}_j$; hence, neither is the gradient. Instead, we use an instantaneous estimate of this gradient. To get this estimate we replace $E\{e^2(t)\}$ by $e^2(t)$ and the gradient by

$$\hat{\nabla} = \frac{\partial e^2(t)}{\partial \hat{\theta}} = 2e(t)\,\frac{\partial e(t)}{\partial \hat{\theta}}. \tag{20}$$

Let us denote

$$s(t) = \frac{\partial e(t)}{\partial \hat{\theta}}$$

as the "sensitivity vector," since it gives an indication of how sensitive the error $e(t)$ is to the values of $\hat{\theta}_j$.

While $s(t)$ is not available we can use eq. (17) to develop a means for generating it. Let us take partial derivatives of both sides of this equation with respect to $\hat{\theta}$. Since the right-hand side does not depend explicitly on $\hat{\theta}$ we get:

$$\left(1 - \sum_{i=1}^{M} \hat{\theta}_i D^{i-1}\right) s_j(t) = D^{j-1} e(t) \qquad j = 1, 2, \cdots, M. \tag{21}$$

At this point we are ready to introduce the Adaptive Gradient Exponential Smoothing (AGES) technique. Combining eqs. (14), (19), (20), and (21) we get:

the *forecast*:

$$\hat{x}(t+1) = A(D)x(t) - \sum_{i=1}^{M} \hat{\theta}_i(t) e(t - i + 1) \tag{22}$$

[see definition of $A(D)$ in eq. (13)],

the *sensitivity functions*:

$$s_j(t+1) = \sum_{i=1}^{M} \hat{\theta}_i(t) s_j(t - i + 1) + e(t - j + 1)$$

$$j = 1, 2, \cdots, M, \tag{23}$$

and the *coefficient adjustments*:

$$\hat{\theta}_j(t+1) = \hat{\theta}_j(t) - 2\mu e(t) s_j(t) \qquad j = 1, 2, \cdots, M. \tag{24}$$

Recall that the error $e(t) = x(t) - \hat{x}(t)$.

Both our simulations and our experiments (as described in the next section) strongly indicate that AGES converges to optimal performance through convergence of $\hat{\theta}_j(t)$ to $\theta_j$. Namely, the error $e(t)$ is adaptively whitened. Despite these indications, since the resulting equations are quite complex, a global proof of convergence of the AGES technique is beyond the scope of this paper. However, we conclude this section by treating the special case $M = 1$ and show local convergence properties for it.

Let $M = 1$; then eqs. (17), (23), and (24) become

$$e(t+1) = \hat{\theta}_1(t) e(t) + \epsilon(t+1) - \theta_1 \epsilon(t)$$

$$s_1(t+1) = \hat{\theta}_1(t) s_1(t) + e(t),$$

and

$$\hat{\theta}(t+1) = \hat{\theta}_1(t) - 2\mu e(t)s_1(t).$$

Assuming $\hat{\theta}_j(t)$ is independent of $e(t)$ and $s_1(t)$ (similar assumptions are common in convergence proofs of adaptive filters) and observing that $E\{\epsilon(t)\cdot e(t)\} = \sigma^2$, $E\{\epsilon(t)\cdot\epsilon(t+1)\} = 0$, $E\{s(t)\cdot\epsilon(t)\} = 0$ we get

$$\begin{aligned} E\{e^2(t+1)\} &= E\{\hat{\theta}_1^2(t)\}\cdot E\{e^2(t)\} \\ &\quad + \sigma^2[1 + \theta_1^2 - 2\theta_1 E\{\hat{\theta}_1(t)\}] \\ E\{s_1(t+1)\cdot e(t+1)\} &= E\{\hat{\theta}_1^2(t)\}\cdot E\{s_1(t)\cdot e(t)\} \\ &\quad + E\{\hat{\theta}_1(t)\}\cdot E\{e^2(t)\} - \theta_1 c^2\sigma^2 \\ E\{\hat{\theta}_1(t+1)\} &= E\{\hat{\theta}_1(t)\} - 2\mu E\{s_1(t)e(t)\}. \end{aligned} \tag{25}$$

If we assume in addition that $\hat{\theta}_1(t)$ has a small variance, namely $E\{\hat{\theta}_1^2(t)\} \approx [E\{\hat{\theta}_1(t)\}]^2$ (the simulation results tend to support this assumption), defining

$$\begin{aligned} \gamma_1(t) &= E\{e^2(t)\} - \sigma^2 \\ \gamma_2(t) &= E\{s_1(t)\cdot e(t)\} \\ \gamma_3(t) &= E\{\hat{\theta}_1(t)\} - \theta_1 \end{aligned} \tag{26}$$

and substituting in (25) results in

$$\begin{aligned} \gamma_1(t+1) &= [\gamma_3(t) + \theta_1]^2\gamma_1(t) + \sigma^2[\gamma_3(t)]^2 \\ \gamma_2(t+1) &= [\gamma_3(t) + \theta_1]^2\gamma_2(t) + [\gamma_3(t) + \theta_1]\gamma_1(t) + \sigma^2\gamma_3(t) \\ \gamma_3(t+1) &= \gamma_3(t) - 2\mu\gamma_2(t). \end{aligned} \tag{27}$$

Clearly, if we could prove that $\gamma_1(t)$, $\gamma_2(t)$, and $\gamma_3(t)$ converge to the origin globally (i.e., independent of the initial values), it would mean that [see eq. (26)] the MSE converges to the minimum $\sigma^2$ and $E\{\theta_1(t)\}$ converges to $\theta_1$. However, despite strong indications from our simulations that these variables do converge globally, we can prove only local convergence. In addition, the proof provides an indication as to how to choose the parameter $\mu$.

Let us linearize eq. (27) around the origin to get

$$\begin{aligned} \gamma_1(t+1) &= \theta_1^2\gamma_1(t) \\ \gamma_2(t+1) &= \theta_1^2\gamma_2(t) + \theta_1\gamma_1(t) + \sigma^2\gamma_3(t) \\ \gamma_3(t+1) &= \gamma_3(t) - 2\mu\gamma_2(t). \end{aligned} \tag{28}$$

The coefficients matrix is

$$A = \begin{bmatrix} \theta_1^2 & 0 & 0 \\ \theta_1 & \theta_1^2 & \sigma^2 \\ 0 & -2\mu & 1 \end{bmatrix}$$

and to ensure convergence all eigenvalues of $A$ must be within the unit circle. The eigenvalues of $A$ are

$$\lambda_1 = \theta_1^2$$

$$\lambda_{2,3} = 1/2\{1 + \theta_1^2 \pm [(1 - \theta_1^2)^2 - 8\mu\sigma^2]^{1/2}\},$$

and it can be verified that choosing

$$\mu < \frac{1 - \theta_1^2}{2\sigma^2} \tag{29}$$

will guarantee the convergence of eq. (28).

Condition (29) implies that if $|\theta_1|$ is close to one, $\mu$ must be chosen very small and the convergence will be slow. Again, our simulation experiments verified this observation.

## IV. SIMULATION RESULTS

We divide our experiments with AGES into two parts. In the first part we applied both ES and AGES on data generated by the computer

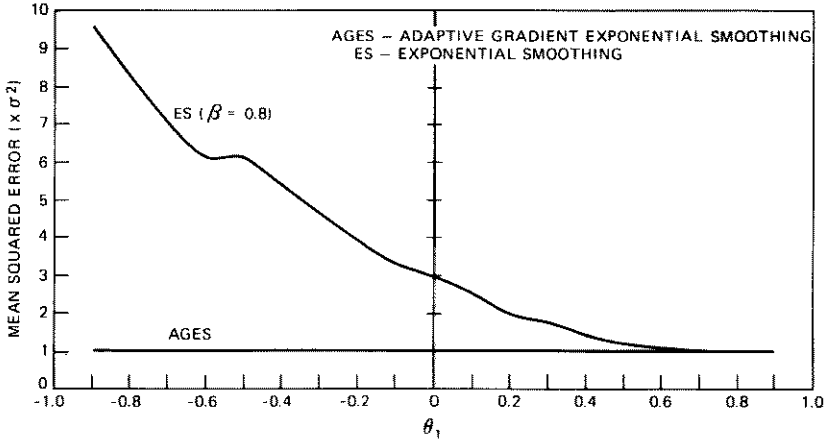



Fig. 3—Comparison of forecasting performance between ES ($\beta = 0.8$) and AGES.

and compared the results. In the second part we applied the AGES to real data that we took from Ref. 7.

Equation (16) was used to generate data of type S, LT, and LSM by the computer. The results of applying both ES and AGES on these data are presented in Figs. 3, 4, and 5 and in Table I. Each point on the curves of Fig. 3 corresponds to a complete run on a sequence of data generated with the particular choice of the $\theta_i$. The resulting MSE for the ES and the AGES forecasts are presented and the comparison clearly indicates the superiority of the AGES algorithm. In addition, we observe that the AGES results, in almost all the runs, in a MSE very close to the minimum, $\sigma^2$.

In Fig. 5, we followed the variation of the $\theta_i(t)$ with time in a number of runs. The results clearly show that the $\hat{\theta}_i(t)$ converge to the $\theta_i$ from a variety of initial values; this indicates global convergence properties. Similar results are observed in Table I for data with seasonal multiplicative effects and linear trend. The $\hat{\theta}_i(t)$ clearly converge to the $\theta_i$'s, and the MSE, when AGES is applied, is again very close to the optimal

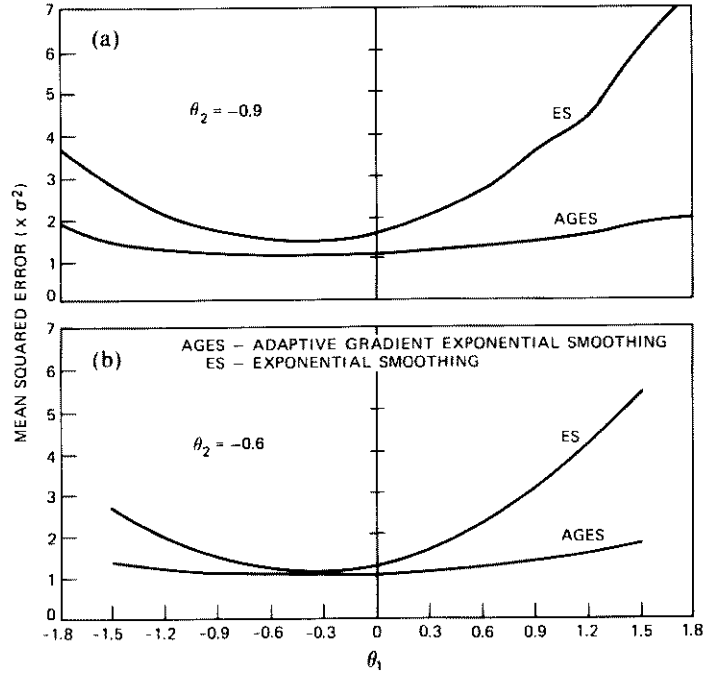



Fig. 4—Comparison of mean squared error in forecasting with ES ($\hat{\theta}_1 = \hat{\theta}_2 = -0.3$) and AGES as a function of the data-generating parameters $\theta_1$ and $\theta_2$. (a) $\theta_2 = -0.9$. (b) $\theta_2 = -0.6$.

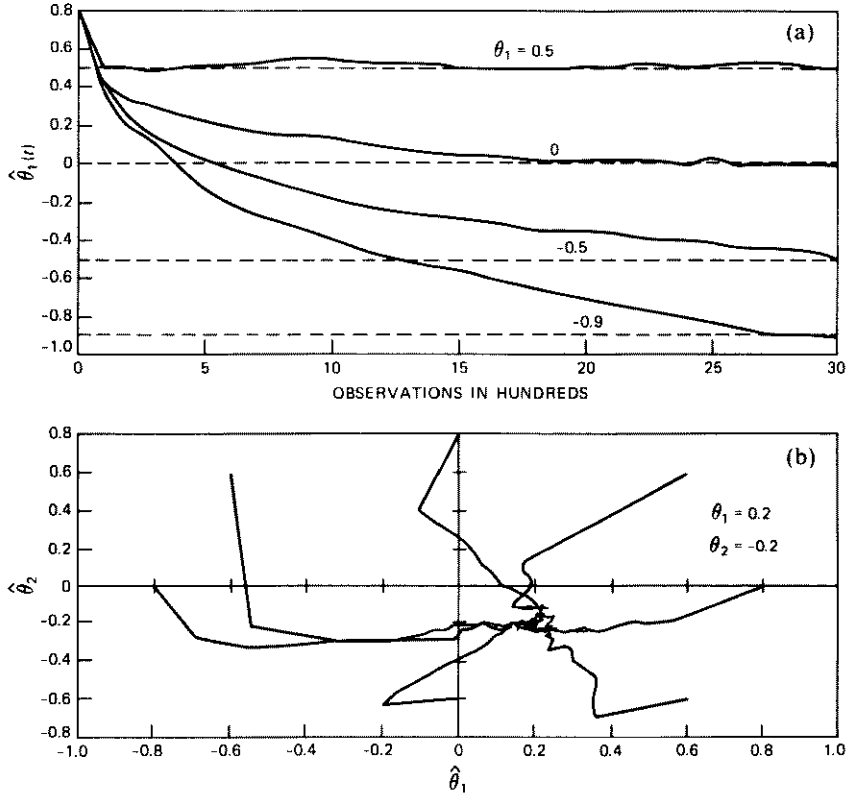



Fig. 5—Convergence of: (a) $\hat{\theta}_1(t)$ to the optimal value $\theta_1$ in the AGES method. (b) $\hat{\theta}_1(t)$ and $\hat{\theta}_2(t)$ from various initial conditions to $\theta_1$ and $\theta_2$, using AGES on data with linear trend.

value, $\sigma^2$. From Ref. 7 we took data of the simple kind (no linear trend or seasonal effects): The IBM common stock closing prices, daily, from May 17, 1961 through November 2, 1962. On the data we applied both ES and AGES and the results are presented in Fig. 6. Each point on the curves corresponds to a run on the same data, each time with a different coefficient (for the ES) and different initial condition (for the AGES). The further the coefficient used in the ES is from $\theta_1$ (which in this case is equal to $-0.1$, as indicated in Ref. 7), the better the performance is for AGES.

Further experiments were conducted on monthly international airline passengers data.[7] These data, as Fig. 7 indicates, are with linear trend and multiplicative seasonal effects. We applied the AGES algorithm (with $M = 3$) and the results are presented in Fig. 8. In Ref. 7 it is claimed that sometimes rather than work with the actual data it

Table I—Comparison of MSE in forecasting with ES ($\hat{\theta}_1 = -0.2$, $\hat{\theta}_2 = 0.5$, $\hat{\theta}_3 = 0.4$) and AGES

| Data-Generating Coefficients (and Adaptive Coefficients) | | | MSE ($\times\sigma^2$) | |
|---|---|---|---|---|
| $\theta_1$ ($\hat{\theta}_1$)* | $\theta_2$ ($\hat{\theta}_2$) | $\theta_3$ ($\hat{\theta}_3$) | AGES | ES |
| 1.4 (1.39) | −1.3 (−1.29) | 0.8 (0.75) | 1.1972 | 11.7155 |
| 2.1 (1.97) | −1.95 (−1.79) | 0.8 (0.68) | 1.3831 | 20.5821 |
| 0.75 (0.69) | −0.6 (−0.56) | 0.8 (0.78) | 1.0749 | 5.1431 |
| 0.6 (0.6) | −0.75 (−0.76) | 0.8 (0.77) | 1.0600 | 5.0256 |
| −0.75 (−0.73) | 0.6 (0.6) | 0.8 (0.79) | 1.0663 | 1.3577 |
| 0.0 (−0.04) | 0.0 (0.01) | 0.0 (0.03) | 1.0040 | 1.5737 |
| 1.0 (0.98) | −1.0 (−0.99) | 1.0 (0.94) | 1.2001 | 8.6414 |
| −0.2 (−0.16) | 0.5 (0.49) | 0.4 (0.4) | 1.0397 | 1.0137 |
| −0.1 (−0.09) | 0.25 (0.26) | 0.4 (0.41) | 0.9973 | 1.0839 |
| 1.2 (1.19) | −0.9 (−0.81) | 0.4 (0.36) | 1.1044 | 7.1069 |
| 1.8 (1.74) | −1.35 (−1.24) | 0.4 (0.34) | 1.2192 | 13.0215 |
| 0.3 (0.31) | −0.75 (−0.75) | 0.4 (0.38) | 1.0574 | 3.8630 |
| 1.0 (0.96) | −0.5 (−0.48) | 0.0 (−0.04) | 1.0605 | 4.2218 |
| 1.5 (1.42) | −0.75 (−0.66) | 0.0 (−0.05) | 1.1165 | 6.9455 |
| 0.75 (0.77) | 0.0 (0.03) | 0.0 (0.03) | 1.0212 | 2.4687 |
| 0.0 (−0.03) | −0.75 (−0.74) | 0.0 (0.02) | 1.0314 | 3.5907 |
| 0.2 (0.19) | 0.5 (0.53) | −0.4 (−0.43) | 1.0186 | 1.7692 |
| 1.2 (1.19) | −0.15 (−0.14) | −0.4 (−0.39) | 1.1115 | 4.0338 |
| −1.8 (−1.7) | −1.35 (−1.22) | −0.4 (−0.36) | 1.2077 | 13.8494 |
| −0.3 (−0.3) | −0.75 (−0.76) | −0.4 (−0.39) | 1.0438 | 4.6957 |
| −0.75 (−0.79) | −0.3 (−0.28) | −0.4 (−0.38) | 1.0062 | 3.9623 |
| 0.4 (0.43) | 0.5 (0.49) | −0.8 (−0.78) | 1.0461 | 2.6126 |
| −0.7 (−0.73) | −0.65 (−0.62) | −0.8 (−0.8) | 1.0670 | 6.1628 |
| −0.6 (−0.61) | −0.75 (−0.75) | −0.8 (−0.79) | 1.0815 | 6.8677 |
| −0.75 (−0.74) | −0.6 (−0.56) | −0.8 (−0.74) | 1.1065 | 7.0182 |
| −0.5 (−0.52) | −0.4 (−0.38) | −0.8 (−0.81) | 1.0759 | 5.1676 |

* The values to which $\hat{\theta}_i(t)$ converge are given in parentheses.

is more convenient to work with the logarithm of the data. As we argue in Appendix B, these logarithms, as data, have linear trend and *additive* seasonal effects (see Fig. 8). Hence, on the logarithms we applied AGES for linear trend and additive seasonal effects and the results are presented in Fig. 8a ($M = 3$). We used the same data (the logarithms) to see whether the performance improves with larger $M$. AGES was applied with $M = 13$ and the results, as presented in Fig. 8b, clearly indicate that for this data $M = 3$ was sufficient.

## V. CONCLUSIONS

In this paper we have introduced a new forecasting technique, Adaptive Gradient Exponential Smoothing (AGES), which is based on Exponential Smoothing (ES). We have elaborated on the optimality properties in the MSE sense of the ES. For certain types of data, the ES can result in optimal performance provided some coefficients are known. In general, these coefficients are unavailable, and the AGES shows strong indications of converging to these unknown coefficients and providing optimal performance.

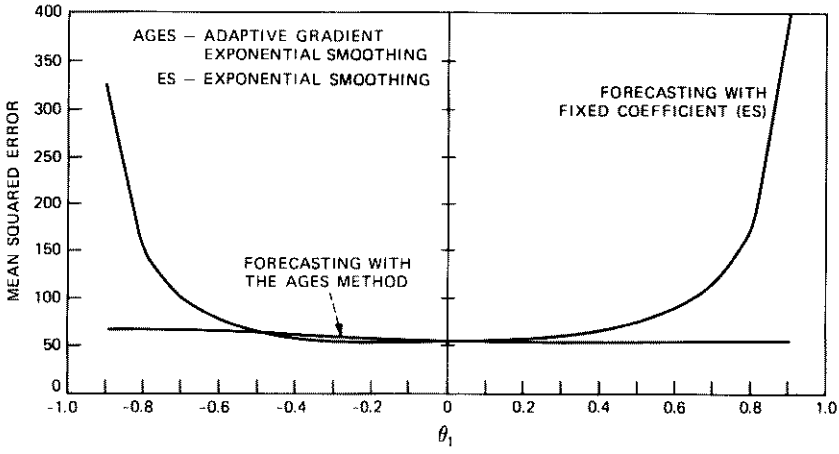



Fig. 6—Comparison of performance of forecasting with the ES (varying the coefficient in each run) and the AGES methods.

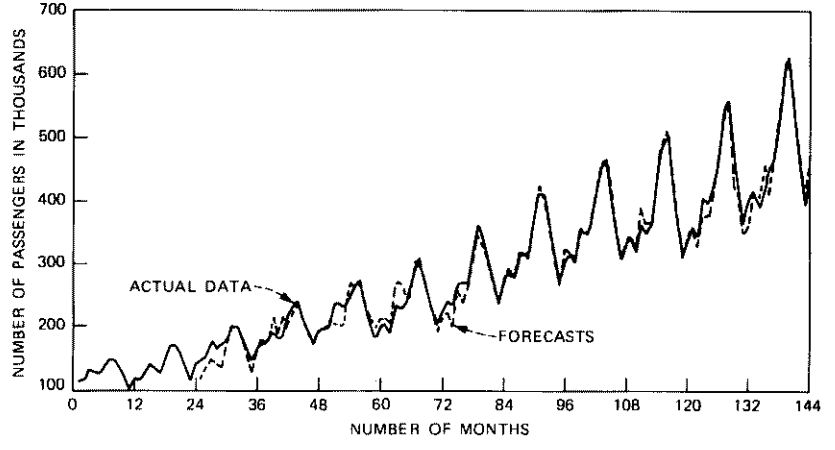



Fig. 7—Forecasting with AGES international airline passengers ($M = 3$). (Note that these data have linear trend and multiplicative seasonal effects.)

Clearly, more extensive experiments and practical use of the proposed forecasting technique, the AGES, are required. A user-friendly software package can be developed for implementation of this technique if sufficient interest is generated.

## VI. ACKNOWLEDGMENT

The author wishes to thank the reviewers for their thoroughness. Their comments were very helpful in the revision of this paper.

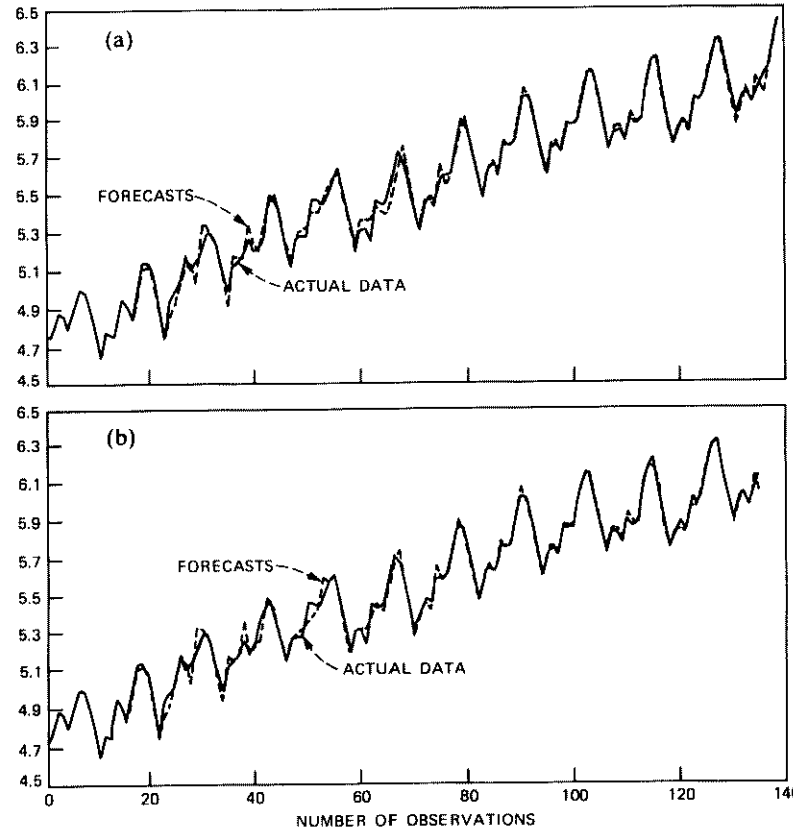



Fig. 8—Forecasting with AGES the logarithm of the data in Fig. 7 for: (a) $M = 3$. (b) $M = 13$. (Note that the logarithm of the data in Fig. 7 has the form of data with additive seasonal effects and linear trend.)

## REFERENCES


1. C. D. Pack and B. A. Whitaker, "Kalman Filter Models for Network Forecasting," B.S.T.J., *61*, No. 1 (January 1982), pp. 1–14.
2. J. P. Moreland, "A Robust Sequential Projection for Traffic Load Forecasting," B.S.T.J., *61*, No. 1 (January 1982), pp. 15–38.
3. C. R. Szelag, "A Short-Term Forecasting Algorithm for Trunk Demand Servicing," B.S.T.J., *61*, No. 1 (January 1982), pp. 67–96.
4. P. R. Winters, "Forecasting Sales, by Exponentially Weighted Moving Averages," Management Sciences, *6*, No. 3 (April 1960), pp. 324–42.
5. R. G. Brown, "Smoothing, Forecasting and Prediction of Discrete Time Series," Englewood Cliffs, NJ: Prentice-Hall, 1962.
6. J. F. Muth, "Optimal Properties of Exponentially Weighted Forecasts of Time Series With Permanent and Transitory Components," J. Am. Statis. Assn., *55* (June 1960), pp. 299–306.
7. G. P. Box and G. M. Jenkins, "Time Series Analysis, Forecasting and Control," San Francisco, CA: Holden-Day, 1970.
8. W. M. Chow, "Adaptive Control of the Exponential Smoothing Constant," J. Indust. Eng., *16*, No. 5 (October 1965), pp. 314–7.

9. D. W. Trigg and A. G. Leach, "Exponential Smoothing With an Adaptive Response Rate," Oper. Res. Quart., *18*, No. 1 (March 1967), pp. 53–60.
10. D. C. Whybark, "A Comparison of Adaptive Forecasting Techniques," Logist. Transport. Rev., *8* (1973), pp. 13–26.
11. S. D. Roberts and R. Reed, "The Development of a Self-Adaptive Forecasting Technique," AIIE Trans., *1* (December 1969), pp. 314–22.
12. S. Ekern, "Adaptive Exponential Smoothing Revisited," J. Oper. Res. Soc., *32*, No. 9 (September 1981), pp. 775–82.

## APPENDIX A

### *Necessity of Conditions 1 and 2 for the Convergence of* e(t) *to* $\epsilon$(t) *in Equation (17)*

Condition 1 is clearly necessary (as well as sufficient) for the convergence of $E\{e^2(t)\}$ to a finite value. We want to show that Condition 2 is necessary for $E\{e^2(t)\}$ to converge to $\sigma^2$.

Let

$$\gamma_{ee}(\tau) = E\{e(t) \cdot e(t - \tau)\} \tag{30}$$

and

$$\gamma_{e\epsilon}(\tau) = E\{e(t)\epsilon(t - \tau)\}. \tag{31}$$

Clearly,

$$\gamma_{ee}(\tau) = \gamma_{ee}(-\tau) \tag{32}$$

and, from eq. (17) and the definition of $\epsilon(t)$

$$\gamma_{e\epsilon}(-\tau) = 0. \tag{33}$$

With these definitions it follows from eq. (17), after transients die, that

$$\begin{aligned}
&\gamma_{e\epsilon}(0) = \sigma^2 \\
&\gamma_{e\epsilon}(1) - \hat{\theta}_1\gamma_{ee}(0) = -\theta_1\sigma^2 \\
&\gamma_{e\epsilon}(2) - \hat{\theta}_1\gamma_{e\epsilon}(1) - \hat{\theta}_2\gamma_{e\epsilon}(0) = -\theta_2\sigma^2 \\
&\quad\vdots \\
&\gamma_{e\epsilon}(M) - \hat{\theta}_1\gamma_{e\epsilon}(M-1) - \cdots - \hat{\theta}_M\gamma_{e\epsilon} = -\theta_M\sigma^2,
\end{aligned}$$

or in matrix form

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\theta_1 & 1 & 0 & \cdots & 0 \\ -\theta_2 & -\theta_1 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ -\theta_M & -\theta_{M-1} & \cdot & \cdots & 1 \end{bmatrix} \begin{bmatrix} \gamma_{e\epsilon}(0) \\ \gamma_{e\epsilon}(1) \\ \cdot \\ \cdot \\ \cdot \\ \gamma_{e\epsilon}(M) \end{bmatrix} = -\sigma^2 \begin{bmatrix} -1 \\ \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_M \end{bmatrix}. \tag{34}$$

Also,

$$\gamma_{ee}(0) - \hat{\theta}_1\gamma_{ee}(1) - \hat{\theta}_2\gamma_{ee}(2) - \cdots - \hat{\theta}_M\gamma_{ee}(M)$$
$$= \gamma_{e\epsilon}(0) - \theta_1\gamma_{e\epsilon}(1) - \cdots - \theta_M\gamma_{e\epsilon}(M)$$
$$\gamma_{ee}(1) - \hat{\theta}_1\gamma_{ee}(0) - \hat{\theta}_2\gamma_{ee}(1) - \cdots - \hat{\theta}_M\gamma_{ee}(M-1)$$
$$= -\theta_1\gamma_{e\epsilon}(0) - \cdots - \theta_M\gamma_{e\epsilon}(M-1)$$
$$\gamma_{ee}(2) - \hat{\theta}_1\gamma_{ee}(1) - \hat{\theta}_2\gamma_{ee}(0) - \cdots - \hat{\theta}_M\gamma_{ee}(M-2)$$
$$= -\theta_2\gamma_{ee}(0) - \cdots - \theta_M\gamma_{e\epsilon}(M-2)$$
$$\vdots$$
$$\gamma_{ee}(M) - \hat{\theta}_1\gamma_{ee}(M-1) - \hat{\theta}_2\gamma_{ee}(M-2) - \cdots - \hat{\theta}_M\gamma_{ee}(0)$$
$$= -\theta_M\gamma_{e\epsilon}(0)$$

or again in a matrix form

$$\left\{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\hat{\theta}_1 & 1 & 0 & & 0 \\ -\hat{\theta}_2 & -\hat{\theta}_1 & 1 & & 0 \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ -\hat{\theta}_M & -\hat{\theta}_{M-1} & \cdots & & 1 \end{bmatrix} - \begin{bmatrix} 0 & \hat{\theta}_1 & \hat{\theta}_2 & \cdots & \hat{\theta}_{M-2} & \hat{\theta}_{M-1} & \hat{\theta}_M \\ 0 & \hat{\theta}_2 & \hat{\theta}_3 & \cdots & \hat{\theta}_{M-1} & \hat{\theta}_M & 0 \\ 0 & \hat{\theta}_3 & \hat{\theta}_4 & \cdots & \hat{\theta}_M & 0 & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ 0 & \hat{\theta}_M & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}\right\} \begin{bmatrix} \gamma_{ee}(0) \\ \gamma_{ee}(1) \\ \gamma_{ee}(2) \\ \cdot \\ \cdot \\ \cdot \\ \gamma_{ee}(M) \\ \gamma_{ee}(M) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -\theta_1 & -\theta_2 & \cdots & -\theta_M \\ -\theta_1 & -\theta_2 & -\theta_3 & & 0 \\ -\theta_2 & -\theta_3 & -\theta_4 & & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \theta_M & 0 & 0 & & 0 \end{bmatrix} \begin{bmatrix} \gamma_{ee}(0) \\ \gamma_{ee}(1) \\ \gamma_{ee}(2) \\ \cdot \\ \cdot \\ \cdot \\ \gamma_{ee}(3) \end{bmatrix}. \quad (35)$$

Now, if we claim that $e(t)$ converges to $\epsilon(t)$, it means that

$$\gamma_{e\epsilon}(\tau) = \delta(\tau)\sigma^2 = \begin{cases} \sigma^2 & \text{for } \tau = 0 \\ 0 & \text{for } \tau \neq 0 \end{cases}$$

and

$$\gamma_{ee}(\tau) = \delta(\tau)\sigma^2.$$

Then, substituting this in (34) or (35) results in

$$\hat{\theta}_j = \theta_j \qquad \text{for } j = 1, 2, \cdots, M,$$

which is Condition 2. Hence this condition is necessary as claimed.

## APPENDIX B

### *Possible Transformation of Multiplicative Seasonal Effects Into Additive Seasonal Effects*

Suppose we are given data of the noise-free form

$$\left.\begin{aligned} y(t) &= (a + bt)c(t) \\ c(t + L) &= c(t) \end{aligned}\right\}, \tag{36}$$

which is with linear trend and multiplicative seasonal effects.

Let

$$z(t) = \text{Log}[y(t)]. \tag{37}$$

Then substitution of (36) gives (if we assume $bt \ll a$, which is true in most real data):

$$\begin{aligned} z(t) &= \log a + \log\left(1 + \frac{b}{a}t\right) + \log c(t) \\ &\approx \log a + \frac{b}{a}t + \log c(t) \\ &\approx \tilde{a} + \tilde{b}t + \tilde{c}(t), \end{aligned} \tag{38}$$

where

$$\begin{aligned} \tilde{a} &= \log a \\ \tilde{b} &= \frac{b}{a} \\ \tilde{c}(t) &= \log c(t). \end{aligned}$$

Hence $z(t)$ clearly has the form of data with linear trend and *additive* seasonal effects.

## AUTHOR

**Arie Feuer,** B.Sc. (Mechanical Engineering), 1967, M.Sc. (Mechanical Engineering), 1973, Technion, Israel; Ph.D. (Control Systems Engineering), 1978, Yale University; Bell Laboratories, 1978—. Since joining Bell Laboratories Mr. Feuer has been involved in telephone network measurement planning and implementations. He is actively involved in research in control system theory, signal processing, and adaptive systems, and currently looks into their possible applications for network measurements and operations.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Application of the Minimum-Weight Spanning-Tree Algorithm to Assignment of Communication Facilities

By N. A. STRAKHOV*

(Manuscript received March 2, 1982)

COSMOS—the Computer System for Main Frame Operations—is an operational support system that inventories and assigns central office facilities to serve customer circuits. As part of this assignment responsibility, COSMOS must provide the central office personnel who will physically connect the circuit not only with information about the facilities to be connected, but also the order in which they will be connected (i.e., connection sequence or "connectivity"). Also, COSMOS must determine the circuit connectivity to permit automatic assignment of tie pairs—inter- and intra-frame cables that permit the connection of facilities that are widely separated physically. A new algorithm has been added to COSMOS to permit the determination of connectivity. This algorithm is based on the algorithm that determines the minimum-weight spanning tree of a connected graph. However, the algorithm is specialized for COSMOS by taking into account such factors as minimizing the maximum number of connections at any node and restricting certain nodes to a maximum number of connections.

## I. INTRODUCTION

When a mechanized system assigns facilities to provide a telephone circuit (to fulfill a request for service, say), it must accomplish three things. It must

1. Determine which facility types are required to provide the service

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

2. Select a particular unit which is available for each such facility type

3. Determine the circuit topology of the assigned facilities.

Steps 1 and 2 can be reduced to an algorithm using straightforward procedures. Step 3, however, has proved to be difficult and has been left for manual determination in all but the simplest cases. In this paper an algorithm is reported that has been successful in determining the circuit topology for most of the circuits encountered in telephony.

## II. A PARTICULAR APPLICATION—COSMOS

COSMOS is the name of a minicomputer system (DEC PDP 11/70 and PDP 11/45) designed for use by telephone operating companies to assign and administer central office equipment.[1] A major impetus for its development and continued deployment is to increase the efficiency of central office personnel who must physically connect, rearrange, and disconnect facilities to provide service to customers. Accordingly, an important feature of COSMOS is its ability to produce a report (called the Frame Output Report or "FOR") for the central office personnel that clearly specifies what should be connected to what. Most circuits for which COSMOS must create a FOR are simple, i.e., only two facilities must be interconnected on the frame. Some circuits, however, can be quite complicated in that some facilities in the circuit must be interconnected in series while others must be connected in parallel. An example of such a case would be a circuit with a main line and an off-premises extension where the bridge point is in the central office. This example becomes more complex if signal conditioning equipment must be placed in series with each line.

Such an example is illustrated in Fig. 1. This circuit includes a main line (cable pair 4-980) and three off-premises extensions (cable pairs 4-981, 4-982, and 4-983). Each cable pair must be connected to the line equipment through a bridge lifter (BL 49, 50, 51, and 52). Since the bridge lifters are located on a different frame from the line equipment and the cable pairs, tie pairs (TP 107, 304, 305, 306, and 307) must be used to interconnect all the components of this circuit.

Since the FOR must unambiguously state how the connections are to be made, either the person establishing the order for service in COSMOS must provide the connection sequence ("connectivity"), or COSMOS itself has to be capable of determining the connectivity. All initial versions of COSMOS had to be connected manually. Starting about 1977 logic was added so COSMOS could automatically determine connectivity in certain situations. The current generic of COSMOS (generic 9.0) is being developed to incorporate connectivity determination in all cases but still allow the user to manually override the automatic connectivity logic if necessary. This paper presents the

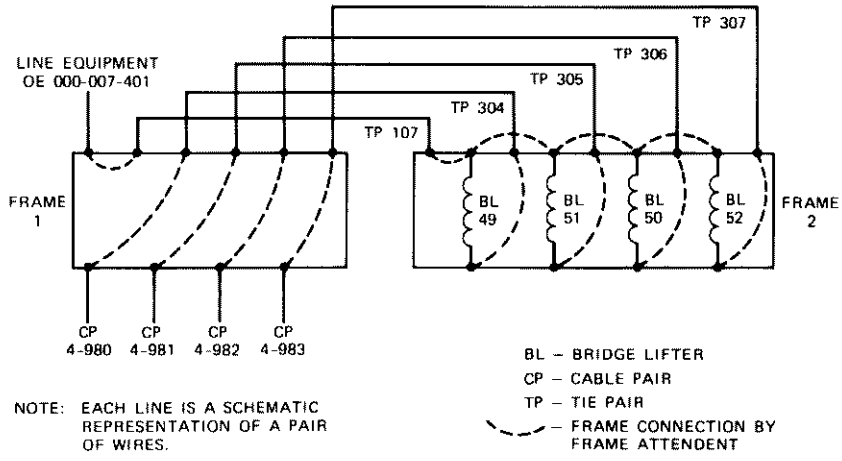



Fig. 1—Example of a main line with three off-premises extensions.

algorithm developed to achieve this capability and illustrates how it benefits the COSMOS user.

## III. REVIEW OF COSMOS CAPABILITIES

As we already mentioned, COSMOS accepts a service order as input and creates the FOR as output. The service order is input to COSMOS by a clerk in the Loop Assignment Center (LAC). Certain information must be entered by the clerk so the order can be processed by COSMOS, while other information is optional, depending on the particular order. The required information is the order number and the order due date. If a switching equipment connection is to be assigned to the customer, then the switching equipment features and the customer class of service must be specified also. Specific facilities to be assigned to the customer can either be specified when entered or automatically assigned by COSMOS. Actually, the automatic assignment takes place in two levels, depending on the facilities: 1) COSMOS determines the need for the facility and then selects a particular facility for the circuit, or 2) the LAC clerk specifies the need for a facility on input and COSMOS selects a particular facility for the circuit. Table I lists the facilities administered by COSMOS and how they are selected for a particular circuit by COSMOS—i.e., manual specification of the particular facility, manual specification of the need for the facility, or complete automatic selection by COSMOS. Table I also specifies that some facilities are terminated on a Main Distributing Frame (MDF), while others are not. The facilities that are not terminated on an MDF either have no physical termination (for

Table I—COSMOS administered facilities

| Facility | MDF Frame Termination | Assignment Mode | | |
|---|---|---|---|---|
| | | Manual | Need Specified | Full Auto |
| Telephone Number (TN) | No | Yes | Yes | No |
| Extra Number (XN) | No | Yes | No | No |
| Group (GP) | No | Yes | No | No |
| Terminal (TER) | No | Yes | No | No |
| Relay (RLY) | No | Yes | Yes | Yes |
| Message Register (MR) | No | Yes | Yes | Yes |
| Private Line Number (PL) | No | Yes | No | No |
| Special Equipment (SE) | No | Yes | No | No |
| Special Equipment (SE) | Yes | Yes | No | No |
| Cable Pair (CP) | Yes | Yes | No | No |
| Line Equipment (OE) | Yes | Yes | Yes | No |
| Concentrator (CON) | Yes | Yes | No | No |
| Tie Pair (TP) | Yes | Yes | Yes | Yes |
| Bridge Lifter (BL) | Yes | Yes | Yes | Yes |
| Trunk (TK) | Yes | Yes | No | No |

example, telephone numbers, groups, and terminals on an electronic switching system are software variables) while others are terminated on an intermediate distributing frame [such as relays and message registers on a No. 5 crossbar switching system (5XB)].

Appendix A describes an overview of the input language for the Service Order Establishment (SOE) transaction. An appreciation of the language is helpful in understanding the example presented in Appendix B, which shows the effect of the connectivity algorithm on the user input. The example in this appendix shows the service order input, as well as excerpts from the FOR to connect the circuit shown in Fig. 1. The detailed functioning of the connectivity algorithm for a particular example is described in Appendix C.

So far, only orders resulting from customer requests for service have been described as input to COSMOS. Another major source of input to COSMOS are work orders; i.e., orders initiated by the telephone company personnel to change out defective equipment or to rearrange circuits to accommodate growth. These transactions also use the connectivity algorithm.

## IV. THE CONNECTIVITY ALGORITHM

The connectivity algorithm, which was first proposed by H. L. York,[2] is based on the concept of a minimum-weight spanning tree of a connected graph. For each circuit whose connectivity is to be determined, a graph whose nodes correspond to each of the elements of the circuit is constructed. The edges of this graph are assigned weights such that the smaller the weight the more likely the two circuit elements (nodes) are to be connected directly to each other. The final connections between the circuit elements is determined by finding a

spanning tree whose edges have a total weight less than or equal to all other spanning trees for this graph. Well-known methods are available for finding the minimum-weight spanning tree of a connected graph. One very straightforward algorithm is given by E. Horowitz and S. Sahni[3] (see especially Section 6.2). An apparently more efficient algorithm plus other extensions of the spanning-tree concept is given by R. C. Prim.[4] As this paper describes later, none of these algorithms could be applied directly to the COSMOS problem because of additional side-constraints that had to be imposed on real circuits. These in turn led to a more efficient algorithm than can be obtained for the general case. The problem of determining circuit connectivity is now reduced to obtaining pair-wise connection weights for all facility combinations and to specifying the particular algorithm for calculating the minimum-weight spanning tree. These will each be discussed in turn.

### *4.1 Determination of connective weights*

In Table I, two types of special equipment (SE) are noted: those with a frame location and those without. Even among SE terminated on the frame there are subgroupings that must be treated differently by the connectivity algorithm. These will now be described.

The SE file in COSMOS contains "miscellaneous" equipment that is not explicitly recorded in any of the other COSMOS equipment files. The name of the SE is created during the order input and a record for the SE is allocated at that time. When an order to disconnect the circuit is established and completed, the record allocated to this SE is released to a list of free records. During input of the name of the SE, the frame location (if one exists) is input also. The SE receives special treatment by the connectivity algorithm, depending on the SE name and the presence or absence of a frame location. The various subcategories of SE are shown in Table II.

With these subdivisions of the equipment that can be represented in the SE field, plus the other facilities that have frame terminations as listed in Table I, the user can construct a complete list of facility

Table II—Subcategories of special equipment

| Input | Action |
|---|---|
| If no frame location is input | Ignore SE for connectivity determination |
| First two characters are RE | Treat as a REG (repeater with gain) |
| First two characters are DL | Treat as a DLL (dial Long Lines) unit |
| First two characters are VR | Treat as a VR (voice repeater)—an example would be an E6 repeater |
| First four characters are DPP- or character string begins with "." | Treat as a trunk |
| Anything else | Treat as an SE |

types that must be processed by the connectivity algorithm. The next step is to construct a matrix whose rows and columns represent each of these facilities and whose elements are the numerical weights associated with how likely the two facilities are to be connected directly to one another. This will be referred to as the generalized weight table. When the algorithm is presented an actual list of elements that must be connected together, the weights for the graph constructed for this circuit will be obtained from the generalized weight table.

The generalized weight table is constructed as follows:

1. All possible facility types are classified in four broad categories: switching equipment, conditioning equipment, metallic facilities, and tie pairs. In general, any circuit must be connected in the order: switching equipment–conditioning equipment–metallic facilities. Tie pairs are assigned as needed to facilitate these connections.

2. Table III shows this classification of facilities. When assigning weights, the user should note that some conditioning equipment is likely to be connected directly to another conditioning equipment of the same type while other types would be unlikely to be connected directly to each other. For example, if several bridge lifters (BLs) were in the same circuit, they would likely all be connected together. Facilities of this type are noted as "bunching" on Table III.

The range of weights is arbitrarily chosen to lie between zero and one hundred. With the considerations just described plus a review of many likely circuits, the generalized weight table shown in Table IV was developed.

As mentioned earlier, for a particular circuit a graph is established and the weights for the edges are taken from the generalized weight table. After that step the weights are further modified if any of the following additional information applies to the circuit.

1. If tie pairs are already present in the circuit (i.e., an existing circuit is being modified), then the two facilities connected by the tie pair are recorded in the tie pair record. The weight between these two facilities and the tie pair is reduced to a small value.

Table III—Grouping of facilities

| | |
|---|---|
| I Switching equipment | Line equipment |
| II Conditioning equipment | Bridge lifter (bunching)<br>Special equipment (RE)<br>Special equipment (DL)<br>Special equipment (VR)<br>Special equipment (bunching) |
| III Metallic facilities | Cable pair<br>Trunk<br>Concentrator<br>Special equipment (DPP–)<br>Special equipment (.) |

Table IV—Generalized weight table

| | OE | BL | SE: DL | SE: RE | SE: VR | SE | CP | TK | SE: . | SE: DPP- | CO | TP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OE | 90 | | | | | | | | | | | |
| BL | 35 | 5 | | | | | | | | | | |
| SE:DL | 40 | 50 | 90 | | | | | | | | | |
| SE:RE | 45 | 55 | 90 | 90 | | | | | | | | |
| SE:VR | 45 | 55 | 7 | 90 | 90 | | | | | | | |
| SE | 45 | 55 | 10 | 15 | 10 | 10 | | | | | | |
| CP | 65 | 60 | 30 | 30 | 6 | 30 | 70 | | | | | |
| TK | 65 | 60 | 25 | 25 | 5 | 25 | 75 | 70 | | | | |
| SE:. | 65 | 60 | 25 | 25 | 5 | 25 | 75 | 70 | 70 | | | |
| SE:DPP- | 65 | 60 | 25 | 25 | 5 | 25 | 75 | 70 | 70 | 70 | | |
| CO | 65 | 60 | 70 | 70 | 73 | 70 | 75 | 75 | 75 | 75 | 90 | |
| TP | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 |

2. If a party circuit is being processed, then the facilities associated with each party are identified by a party number. Those facilities that do not belong to the same party have their weights increased to a maximum value.

3. If a circuit with one or more off-premises extensions is being processed, then those facilities belonging to the same "leg" will have a Different Premise Address (DPA) value assigned to them. Consequently, the weights are increased to a maximum value for those edges connecting facilities in different "legs".

4. If the circuit contains tie pairs, then the weights between facilities terminated on different frames will be increased somewhat. This is done to avoid assigning tie pairs unnecessarily.

### *4.2 Fundamental considerations*

There are two special conditions that apply to determining circuits for central office facilities that do not apply to circuit connectivity in general. Not only do these conditions enable COSMOS to determine the correct configuration, but their use speeds up the algorithm as well since some nodes can be eliminated from consideration after a certain point in the algorithm has been reached. These special conditions are:

1. When a user is choosing among several Minimum-Weight Spanning Trees (MWSTs) (they need not be unique for a given graph), the tree with the minimum number of branches at the node with the most branches is preferred. An example of this is shown in Fig. 2. The algorithm does not actually calculate all possible MWSTs and then choose the one with the minimum number of branches at the node with the maximum number of branches. Instead, several strategies are employed, depending on the circuit being processed. If the circuit contains office equipment (OE), then most nodes have a maximum number of connections to them which is calculated before the MWST processing begins (as described in item 2 of this listing). The only

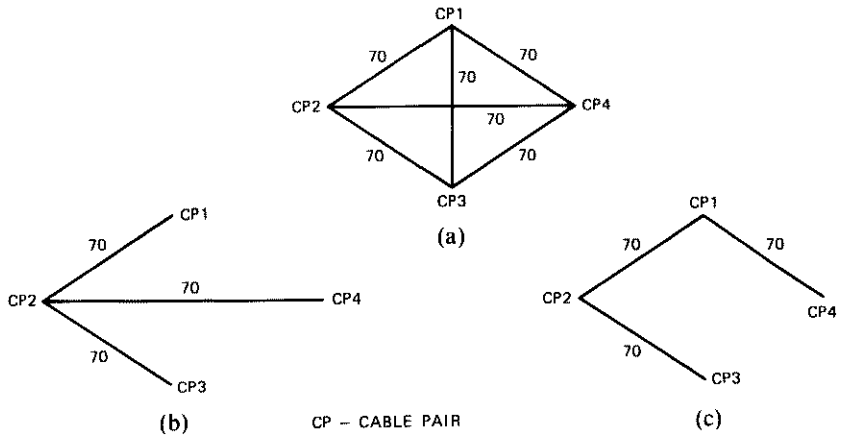



Fig. 2—The minimum-weight spanning tree showing (a) the graph; (b) a maximum of three branches; (c) a maximum of two branches. In this case (c) is the preferred tree.

exceptions are nodes that represent BLs. To prevent all BLs from connecting to one BL, the edge weights of all BL edges not yet selected for the MWST terminating on a BL just connected are incremented. For a circuit that contains no OE, the edge weights of all edges not yet selected for the MWST terminating on a node that has been selected for the MWST are incremented. These strategies direct the algorithm towards selecting the MWST with the required branch minimization. This requirement relates to how circuits are actually wired on the frame. If too many connections must be made at one terminal, the craftsperson may physically run out of room on the terminal and thus be unable to complete all the connections. Also, if an order is subsequently received to disconnect one of the legs of the circuit, proper "housekeeping" might require dismantling all connections at a terminal and then reconnecting the remaining legs. This process is much simplified if the number of connections at a terminal is minimized.

2. When an OE is present in the circuit, each facility is allowed a maximum number of "outward connections." An outward connection is defined as a connection away from the OE. When an OE is present, it will be the root of the tree and therefore a direction away from the OE (root) is always defined. The maximum number of outward connections is determined by the following rules:

(a) Metallic facilities (see Table III) have zero outward connections since they must always be at the outermost "tips" of the branches.

(b) Conditioning equipment (see Table III) is allowed one outward connection. In determining outward connections, a connection

between two BLs is not counted. This is because a BL will usually be at a branch point and therefore will have additional outward connections.

(c) If no BLs are present in the circuit, the number of outward connections from the OE equals the number of metallic facilities. If BLs are present, the number of outward connections from the OE equals the number of metallic facilities minus the number of BLs plus one. This formula reflects the actual way in which such circuits are wired: If BLs are present, they are the bridge point instead of the OE. In fact, BLs are often hard-wired in parallel in anticipation of their use as bridge points.

This set of rules (a through c) constitutes the principal reason for a speed-up of this algorithm over the general case, since once the maximum number of outward connections is achieved, a particular node no longer needs to be considered.

3. When an edge is chosen for the MWST, its weight is increased to the maximum value (100). This was chosen as the most efficient method to signal the algorithm not to consider this edge for the MWST again.

### *4.3 Detailed description*

This section describes determining the list of facilities to be connected, the actual algorithm, and how the output list of facilities in connectivity order is assembled from the internal tables populated by the connectivity algorithm. This breakdown parallels the construction of the actual software.

#### *4.3.1 The list of facilities to be connected*

Connectivity processing is initiated when another COSMOS module determines that connectivity must be established. If this is the case, the connectivity module is invoked and a list of facilities is presented to it. Before this list can be passed along to the connectivity algorithm, certain facilities must be "weeded out".

There are two types of facilities that must be excluded from connectivity considerations. The first type includes facilities that have no mainframe terminations. These facilities are telephone numbers (TNs), No. 1XB coded terminals (XNs), No. 5XB relays, electronic switching system groups and terminals (GP and TER), and special equipment (SEs) for which no frame termination has been entered.

The second type of facility that must be excluded is frame-terminated facilities that will not be in the circuit at the time that the order being processed will be worked. This situation can arise because COSMOS allows multiple orders to be established on the same circuit if they are logically consistent with one another. Thus order number

1 with due date $X$ may be removing facilities from an established circuit while order number 2 (the one being processed, say) with due date $Y$ is adding facilities to the same circuit. If due date $X$ precedes due date $Y$, then in processing order 2 the facilities being removed by order 1 should not be considered. However, if due date $X$ is later than due date $Y$, then all facilities must be considered in processing order 2.

With these two considerations, a list of facilities is prepared for processing by the connectivity algorithm. After connectivity is determined for these facilities, those facilities that were excluded are added to the end of the list of facilities that were placed in connectivity order.

#### *4.3.2 The algorithm itself*

The first step performed by the algorithm is to identify the equipment types that have been presented to it. It then proceeds to calculate the connection weights for all the edges of the graph describing the circuit using the considerations outlined in Section 4.1. These weights are stored in a weight table. Next the actual MWST processing begins. This is facilitated by updating a "working" table. Each row of the table contains the following information: facility, count of connections to the facility, available outward connections, lowest connection cost, and the facility connected by the "lowest connection cost" edge. Also, as the algorithm proceeds, a third table, the connection list, is created. The connection list table maintains a list of the edges selected for the MWST.

The "working" table is populated as follows: each input facility is placed into the table. Initially, the count of all connections to the facility is set to zero for each facility. The available outward connections for each facility are determined based on the considerations described in item 2 of Section 4.2. The lowest cost connection and the corresponding facility are determined by scanning the weight table for each facility. In case of a tie the first edge encountered in the weight table is chosen for inclusion in the working table.

Now the first facility to be placed in the circuit must be chosen. If there is an OE in the working table, it is chosen as the first facility; otherwise the first facility in the working table is chosen. The first facility and the facility it is connected to in the working table are placed in the connection list.

In the following, the first facility is taken as a facility appearing in the connection list. While there are still facilities that have not been connected to the circuit, the following instructions are repeated:

1. Choosing among the facilities already in the circuit (i.e., in the connection list), find the facility in the working table with the lowest cost connection. In case of a tie take the facility that appears first in

the working table. The facility connected by the lowest cost connection edge will be referred to as the "new facility"; the original facility will be called the "old facility".

2. If the number of connections to the new facility is not zero, this edge cannot be part of the MWST or else a cycle would be formed. Skip to instruction 8 below.

3. Add this connection to the connection list.

4. Increment the number of connections for the two facilities.

5. If the circuit contains an OE, decrement the number of available outward connections for the old facility unless both facilities are BLs. If both facilities are BLs, add one to the cost of all edges in the weight table that emanate from the old BL. This will reduce the maximum number of connections made at one bridge point, as explained in Section 4.2.

6. If the circuit contains an OE, and if either the old or the new facility (or both) have zero outward connections available, change the costs in the weight table for all edges emanating from such a node to a maximum value.

7. If the circuit does not contain an OE, and the old facility has two or more connections, add one to the cost of all edges in the weight table that emanate from the old facility node.

8. Change the cost of the edge in the weight table that connects the old and the new facility to a maximum value.

9. Reestablish the working table based on the new weight table costs.

We may now assume that all facilities have been placed in the connection list. (Note that if there are $N$ facilities to be connected, there will be $N$-1 entries in the connection list so that the end of the algorithm is readily detected.) Now the connection list must be converted to a linear list. A tree will be described by a linear list that enumerates each branch, one after another. The beginning of a new branch is detected by the repetition of a facility that already appears higher up on the list (the branch point).

The algorithm for creating the linear list makes use of the working table left over from the MWST algorithm and the connection list. The algorithm

1. Searches the working table (in reverse order) until a facility is found with only one connection. This facility is one end of a branch. It places the facility in the linear list.

2. Searches the connection list (in reverse order) for the facility just placed in the linear list. It places the facility connected to it in the linear list.

3. Decrements the connection count for both the old and the new facilities. It removes their connection from the connection list.

4. If the connection count for the new facility is greater than zero, it repeats Steps 2 and 3. If the connection count for the new facility is zero, the end of the current branch has been reached. It will then go to the next step.

5. For each facility already on the linear list, it determines the number of remaining connections in the working table. If all connections are zero, the linear list is complete. Otherwise, it selects the first facility encountered with a nonzero connection count.

6. Enters this facility in the linear list and proceeds to Step 2.

Note that the lists in Steps 1 and 2 are searched in reverse order so that the frame instructions are in a more "pleasing" sequence: line equipment first, then the first "leg", then the second "leg", etc.

The connectivity algorithm is now complete. The facilities that were excluded from consideration at the start of the algorithm can be added to the end of the list.

The steps just described are applied to a particular example in Appendix C.

## V. ACKNOWLEDGMENTS

As we mentioned earlier, the basic idea of applying the MWST algorithm to the problem of determining circuit connectivity in COSMOS is due to H. L. York, who also programmed the original version. The additional modifications for circuits containing an OE were conceived of and designed by J. B. Sharpless. The algorithm was rewritten by E. W. Merrill, who worked under Sharpless' direction.[5] The current "owner" of the program is D. P. Bates.

## REFERENCES

1. B. Bittner, "Computer System For Main Frame Operations (COSMOS)," Proc. IEEE Int. Conf. Commun., *1* (1976), pp 13–20.
2. H. L. York, unpublished work.
3. E. Horowitz and S. Sahni, "Fundamentals of Data Structures," Woodland Hills, CA: Computer Science Press, Inc., 1976.
4. R. C. Prim, "Shortest Connection Networks and Some Generalization," B.S.T.J., *36*, No. 6 (November 1957), pp. 1389–1401.
5. J. B. Sharpless and E. W. Merrill, unpublished work.

## APPENDIX A

### *COSMOS Service Order Language*

When COSMOS is ready to accept a command, it will print a prompt (%). Immediately preceding the prompt character two alphanumeric characters are printed. These two characters represent the wire center with whose facilities the user wishes to work. The wire center is identified by the user at log-in time.

After the prompt letters have been printed the user can enter the

transaction name. All service orders are initiated in COSMOS through the transaction SOE (Service Order Establishment). Particular inputs to SOE are established on separate lines, as many as are needed to specify the order.

The first character of the first input line must be an H (standing for header). The remainder of the line contains general data pertaining to the order. Typical data items that appear on this line are the order number (identified by the prefix ORD), the order type (OT), and the due date (DD). The prefix-data groupings are separated by a vergule (/). This applies to all line types, not just to the H line.

If all the data do not fit on the first H line, they may be continued on subsequent H lines. Once all the header data have been entered, facilities to be connected on the order are entered on a line (or lines) whose first character is I (standing for "in"). Facilities to be disconnected by the order are entered on a line (or lines) whose first character is O (standing for "out"). Typical data items that appear on I or O lines are Cable Pair (CP), Telephone Number (TN), Office Equipment (OE), Universal Service Order Code (US), features (FEA), Telephone Number Exchange code (NNX), and Resistance Zone (RZ).

In the case of facilities that are automatically assigned by COSMOS, the facility prefix may be followed by a question mark (?). This is a signal to COSMOS to assign the facility automatically. For example, if COSMOS is to select a telephone number somewhere on the I line the construction

I ····· /TN ?/ ····

should appear. However, some wire centers contain several different switching entities. To distinguish among them the user is instead required to specify the exchange code. In this case automatic telephone number selection is triggered by the input.

I ···· /NNX 351/ ·····

When all I and/or O lines have been input the user types a "." on a single line. At this point processing of the order commences. It should also be noted that as each line is entered, rudimentary checks are performed. When this processing is completed COSMOS prints an underscore (_) as a prompt to indicate that the next line can be processed.

## APPENDIX B

### *An Example of Automatic Connectivity Determination*

In this case COSMOS will be asked to process order number NAS0789. This is a new connect order (OT NC) and has a due date of August 1, 1981. The exchange code is 111 and COSMOS is to assign

the line equipment with a Universal Service Order Code (US) of 1FR and features (FEA) consisting of *Touch-Tone** service (T), nonsleeve lead (N), nonessential (N), and loop start (L). Four cable pairs are to be assigned to the order—pairs 980, 981, 982, and 983 in cable 4. The resistance zones of these pairs are 22, 11, 12, and 13, respectively. A parameter is maintained in the database to indicate whether bridge lifters are needed. If any one of these resistance zones exceeds this parameter, then all pairs will be assigned bridge lifters. In this case the parameter is set to 18, a value exceeded by the resistance zone of the first pair.

The input and the SOE response is as follows:

```
90% SOE
H ORD NAS0789/OT NC/DD 8-1-81
_I NNX 111/OE ?/US 1FR/FEA TNNL
_I CP 4-980/RZ 22
_I CP 4-981/RZ 11
_I CP 4-982/RZ 12
_I CP 4-982/RZ 13
_.
SO000122
ORD NAS0789
IN:  CP  4-0980
IN:  CP  4-0981
IN:  CP  4-0982
IN:  CP  4-0983
IN:  OE  000-007-401
IN:  TN  111-1096
IN:  BL  49
IN:  BL  51
IN:  BL  50
IN:  BL  52
IN:  TP  CM11-0107
IN:  TP  CM11-0304
IN:  TP  CM11-0305
IN:  TP  CM11-0306
IN:  TP  CM11-0307
**TRANSACTION COMPLETED
90%
```

The string "SO000122" immediately following the period is the record number in the service order file selected by COSMOS to hold information about the order. This record number is useful in the event

* Registered service mark of AT&T.

the order is not established properly and manual corrective action is required.

The rest of the SOEs output are COSMOS assignments. First the four cable pairs are echoed back. These are followed by eleven automatically assigned facilities: an office equipment, a telephone number, four bridge lifters, and five tie pairs. (The tie pairs are needed to interconnect the bridge lifters and the office equipment and cable pairs since the bridge lifters are terminated on a different frame.) The facilities are listed by SOE in the order in which they are assigned. This is not the connectivity order.

To show the connectivity order the frame output report must be executed. This is the report used by telephone company personnel to actually wire the circuit in the central office. The report itself is in a lengthy format for ease of reading. Instead of reproducing the entire report here, only excerpts that show connectivity are listed below:

```
        ⋮
LINE EQP IN     000-007-401
        ⋮
TIE PAIR IN     CM11-0107
        ⋮
MISC EQP IN     BL 49
        ⋮
TIE PAIR IN     CM11-0304
        ⋮
CABLE PR IN     4-0980
        ⋮
*MISC EQP IN    BL 49
        ⋮
MISC EQP IN     BL 51
        ⋮
TIE PAIR IN     CM11-0305
        ⋮
CABLE PR IN     4-0981
        ⋮
*MISC EQP IN    BL 51
        ⋮
MISC EQP IN     BL 50
        ⋮
TIE PAIR IN     CM11-0306
        ⋮
CABLE PR IN     4-0982
        ⋮
*MISC EQP IN    BL 50
        ⋮
```

```
MISC EQP IN      BL 52
                  ⋮
TIE PAIR IN      CM11-0307
                  ⋮
CABLE PR IN      4-0983
                  ⋮
```

Note the first leg of the circuit—extending from OE 000-007-401 to CP 4-0980. The beginning of the next leg is indicated by the asterisk (*) and the repetition of the facility BL 49. This is the first bridge point. This leg extends down to CP 4-0981. Now BL 51 is shown as the next bridge point. Notice that BL 49 is not the bridge point for all legs. This is the effect of the algorithm described in Section 4.2 to minimize the maximum number of legs emanating from a single bridge point. The remaining two legs extend from BL 51 to CP 4-0982 and BL 50 to CP 4-0983.

## APPENDIX C

### *An Example of the Algorithm's Execution*

The algorithm described in Section 4.3.2 will be followed in detail for a particular set of facilities: two bridge lifters (BL1 and BL2), two cable pairs (CP1 and CP2), and one line equipment (OE). The first step is to determine the connection weights for all the edges of the graph. These weights are determined from Table IV. Note that the diagonal terms are given a weight of 100, since a facility cannot be connected to itself.

Step 1—Weight table

| | BL1 | BL2 | CP1 | CP2 | OE |
|---|---|---|---|---|---|
| BL1 | 100 | 5 | 60 | 60 | 35 |
| BL2 | 5 | 100 | 60 | 60 | 35 |
| CP1 | 60 | 60 | 100 | 70 | 65 |
| CP2 | 60 | 60 | 70 | 100 | 65 |
| OE | 35 | 35 | 65 | 65 | 100 |

In this particular case CP1 and BL1 have been assigned a DPA value of " " (i.e., a blank) and CP2 and BL2 have been assigned a DPA value of "999" by a previously invoked load module of SOE. Thus those edges connecting facilities in different "legs" (i.e., BL1-CP2 and BL2-CP1) have their weights changed to a maximum value. (In the next and in all following tables entries that have changed from the previous table are enclosed in parentheses.)

Step 2—Weight table

| | BL1 | BL2 | CP1 | CP2 | OE |
|---|---|---|---|---|---|
| BL1 | 100 | 5 | 60 | (100) | 35 |
| BL2 | 5 | 100 | (100) | 60 | 35 |
| CP1 | 60 | (100) | 100 | 70 | 65 |
| CP2 | (100) | 60 | 70 | 100 | 65 |
| OE | 35 | 35 | 65 | 65 | 100 |

Now the working table is constructed. Each facility has an entry and the connection count is initially set to zero. The Available Outward Connections (AOC) equal zero for the two metallic facilities (CP1 and CP2), equals one for the two conditioning facilities (BL1 and BL2), and equals one for the OE based on the formula:

$$\text{AOC} = \#\text{ metallic facilities} - \#\text{ BL's} + 1$$

$$= 2 - 2 + 1 = 1$$

The lowest connection cost and corresponding facility are obtained from the weight table.

Step 3—Working table

| Facility | Connection Count | Available Outward Connections | Lowest Connection Cost | Corresponding Facility |
|---|---|---|---|---|
| BL1 | 0 | 1 | 5 | BL2 |
| BL2 | 0 | 1 | 5 | BL1 |
| CP1 | 0 | 0 | 60 | BL1 |
| CP2 | 0 | 0 | 60 | BL2 |
| OE | 0 | 1 | 35 | BL1 |

Since the circuit contains an OE, this facility is chosen first and placed on the connection list.

Step 4—Connection list

OE-BL1

The number of connections to the OE and BL1 are incremented (Step 6, working table). The number of AOC to the old facility (the OE) is decremented (Step 6, working table). The old facility now has zero AOC so the weight of all edges emanating from it is changed to a maximum value (Step 5, weight table). Finally, the working table is modified due to changes in the weight table.

Step 5—Weight table

| | BL1 | BL2 | CP1 | CP2 | OE |
|---|---|---|---|---|---|
| BL1 | 100 | 5 | 60 | 100 | (100) |
| BL2 | 5 | 100 | 100 | 60 | (100) |
| CP1 | 60 | 100 | 100 | 70 | (100) |
| CP2 | 100 | 60 | 70 | 100 | (100) |
| OE | (100) | (100) | (100) | (100) | 100 |

Step 6—Working table

| Facility | Connection Count | Available Outward Connections | Lowest Connection Cost | Corresponding Facility |
|---|---|---|---|---|
| BL1 | (1) | 1 | 5 | BL2 |
| BL2 | 0 | 1 | 5 | BL1 |
| CP1 | 0 | 0 | 60 | BL1 |
| CP2 | 0 | 0 | 60 | BL2 |
| OE | (1) | (0) | (100) | BL1 |

If we choose among the facilities already in the connection list (OE and BL1), the one with the lowest connection cost in the weight table is the first entry. The edge BL1-BL2 is added to the connection list.

Step 7—Connection list

OE-BL1
BL1-BL2

Since the connection count to BL2 is zero, this is an acceptable choice. The number of connections to BL1 and BL2 are incremented (Step 9, working table). However, the number of AOC to the old facility (BL1) is not decremented, since both facilities are BLs. Instead, one is added to the cost of all edges that emanate from BL1 (Step 8, weight table). Since neither the old nor the new facility has zero AOC, the edges emanating from these nodes do not have their weights set to 100. However, the BL1-BL2 weights are set to the maximum value (Step 8, weight table). Finally, the working table is modified according to changes in the weight table.

Step 8—Weight table

| | BL1 | BL2 | CP1 | CP2 | OE |
|---|---|---|---|---|---|
| BL1 | (101) | (100) | (61) | (101) | (101) |
| BL2 | (100) | 100 | 100 | 60 | 100 |
| CP1 | (61) | 100 | 100 | 70 | 100 |
| CP2 | (101) | 60 | 70 | 100 | 100 |
| OE | (101) | 100 | 100 | 100 | 100 |

Step 9—Working table

| Facility | Connection Count | Available Outward Connections | Lowest Connection Cost | Corresponding Facility |
|---|---|---|---|---|
| BL1 | (2) | 1 | (61) | (CP1) |
| BL2 | (1) | 1 | (60) | (CP2) |
| CP1 | 0 | 0 | (61) | BL1 |
| CP2 | 0 | 0 | 60 | BL2 |
| OE | 1 | 0 | 100 | (BL2) |

If we choose among the facilities already on the connection list (OE, BL1, and BL2), the one with the lowest connection cost in the weight table is the BL2 entry. Therefore, BL2-CP2 is added to the connection list.

Step 10—Connection list

OE-BL1
BL1-BL2
BL2-CP2

Since the connection count to CP2 is zero, this is an acceptable choice. The number of connections to BL2 and CP2 are incremented (Step 12, working table). The number of AOC to the old facility (BL2) is decremented (Step 12, working table). Since both BL2 and CP2 now have zero AOC, the weights for all edges emanating from BL2 and CP2 are set to the maximum value (Step 11, weight table). Finally, the working table is modified according to changes in the weight table.

Step 11—Weight table

| | BL1 | BL2 | CP1 | CP2 | OE |
|---|---|---|---|---|---|
| BL1 | 101 | 100 | 61 | 101 | 101 |
| BL2 | 100 | 100 | 100 | (100) | 100 |
| CP1 | 61 | 100 | 100 | (100) | 100 |
| CP2 | 101 | (100) | (100) | 100 | 100 |
| OE | 101 | 100 | 100 | 100 | 100 |

Step 12—Working table

| Facility | Connection Count | Available Outward Connections | Lowest Connection Cost | Corresponding Facility |
|---|---|---|---|---|
| BL1 | 2 | 1 | 61 | CP1 |
| BL2 | (2) | (0) | (100) | (BL1) |
| CP1 | 0 | 0 | 61 | BL1 |
| CP2 | (1) | 0 | (100) | BL2 |
| OE | 1 | 0 | 100 | BL2 |

If we choose among the facilities already on the connection list (OE, BL1, BL2, CP2), the one with the lowest connection cost in the weight table is the BL1 entry. Therefore, BL1-CP1 is added to the connection list.

Step 13—Connection list

OE-BL1
BL1-BL2
BL2-CP2
BL1-CP1

Since the connection count to CP1 is zero, this is an acceptable choice. The number of connections to BL1 and CP1 are incremented (Step 15, working table). The number of AOC to the old facility (BL1) is decremented (Step 15, working table). Since both BL1 and CP1 now have zero AOC, the weight of all edges emanating from BL1 and CP1 are set to the maximum value (Step 14, weight table). Finally, the working table is modified according to changes in the weight table.

Step 14—Weight table

| | BL1 | BL2 | CP1 | CP2 | OE |
|---|---|---|---|---|---|
| BL1 | 101 | 100 | (100) | 101 | 101 |
| BL2 | 100 | 100 | 100 | 100 | 100 |
| CP1 | (100) | 100 | 100 | 100 | 100 |
| CP2 | 101 | 100 | 100 | 100 | 100 |
| OE | 101 | 100 | 100 | 100 | 100 |

Step 15–Working table

| Facility | Connection Count | Available Outward Connections | Lowest Connection Cost | Corresponding Facility |
|---|---|---|---|---|
| BL1 | (3) | (0) | (100) | (BL2) |
| BL2 | 2 | 0 | 100 | BL1 |
| CP1 | (1) | 0 | (100) | BL1 |
| CP2 | 1 | 0 | 100 | BL2 |
| OE | 1 | 0 | 100 | BL2 |

The algorithm is completed when the connection list contains $N$-1 entries, where $N$ equals the number of facilities. In this case $N$-1 = 4 and so all connections have been obtained. The remainder of the algorithm transforms the connection list to a linear list.

Initially, the connection count for each facility, the connection list, and the linear list are as shown in Step 16. Search the connection count (from the bottom) to find a facility with a connection count of one. In this case the facility found is the OE. Next, search the

connection list (from the bottom) to find a corresponding facility. In this case the facility is BL1. Place these two facilities on the linear list and decrement the connection count for each.

Step 16—Linear list

| Facility | Connection Count | Connection List | Linear List |
|---|---|---|---|
| BL1 | 3 | OE-BL1 | |
| BL2 | 2 | BL1-BL2 | |
| CP1 | 1 | BL2-CP2 | |
| CP2 | 1 | BL1-CP1 | |
| OE | 1 | | |

Applying this algorithm results in the table shown in Step 17. Since the connection count for BL1 is greater than zero, search the connection list (from the bottom) to find another entry for BL1. The connection BL1-CP1 is found, so CP1 is added to the linear list, and the connection count for both BL1 and CP1 are decremented.

Step 17—Linear list

| Facility | Connection Count | Connection List | Linear List |
|---|---|---|---|
| BL1 | 2 | ------ | OE |
| BL2 | 2 | BL1-BL2 | BL1 |
| CP1 | 1 | BL2-CP2 | |
| CP2 | 1 | BL1-CP1 | |
| OE | 0 | | |

The table now changes to what is shown in Step 18. Since the connection count for CP1 is zero, the connection count list is again searched (from the bottom) but only for facilities on the linear list (i.e., OE, BL1, CP1) for an entry with a nonzero connection count. The entry found is BL1. Searching the connection list for a corresponding facility results in the addition of the BL1-BL2 connection to the linear list. The connection count of each of these facilities is therefore decremented.

Step 18—Linear list

| Facility | Connection Count | Connection List | Linear List |
|---|---|---|---|
| BL1 | 1 | ------ | OE |
| BL2 | 2 | BL1-BL2 | BL1 |
| CP1 | 0 | BL2-CP2 | CP1 |
| CP2 | 1 | ------ | |
| OE | 0 | | |

The table now changes to what is shown in Step 19. Since the connection count for BL2 is greater than zero, search the connection list to find another entry for BL2. The connection BL2-CP2 is found so CP2 is added to the linear list and the connection count for both BL2 and CP2 are decremented.

Step 19—Linear list

| Facility | Connection Count | Connection List | Linear List |
|---|---|---|---|
| BL1 | 0 | ------ | OE |
| BL2 | 1 | ------ | BL1 |
| CP1 | 0 | BL2-CP2 | CP1 |
| CP2 | 1 | ------ | BL1 |
| OE | 0 | | BL2 |

The table now changes as shown in Step 20. Since all connection counts are zero, the algorithm terminates.

Step 20—Linear list

| Facility | Connection Count | Connection List | Linear List |
|---|---|---|---|
| BL1 | 0 | ------ | OE |
| BL2 | 0 | ------ | BL1 |
| CP1 | 0 | ------ | CP1 |
| CP2 | 0 | ------ | BL1 |
| OE | 0 | ------ | BL2 |
| | | | CP2 |

## AUTHOR

**Nicholas A. Strakhov,** BSME, 1959, Massachusetts Institute of Technology; MEE, 1961, New York University; Ph.D., 1967, New York University; Georgia Institute of Technology, 1972–1974; Bell Laboratories, 1959—. At the Georgia Institute of Technology Mr. Strakhov was a part-time lecturer. He presently supervises a group developing software for COSMOS—an Operational Support System used by the telephone companies to administer facilities assigned to customers.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Note on the Properties of a Vector Quantizer for LPC Coefficients

By L. R. RABINER,* M. M. SONDHI,* and S. E. LEVINSON*

(Manuscript received March 15, 1983)

Vector quantization has been used in coding applications for several years. Recently, quantization of linear predictive coding (LPC) vectors has been used for speech coding and recognition. In these latter applications, the only method that has been used for deriving the vector quantizer code book from a set of training vectors is the one described by Linde, Buzo, and Gray. In this paper, we compare this algorithm to several alternative algorithms and also study the properties of the resulting code books. Our conclusion is that the various algorithms that we tried gave essentially identical code books.

## I. INTRODUCTION

The technique of vector quantization for LPC voice coding has been in use for several years, and has been shown to be of great utility for LPC analysis/synthesis systems.[1-4] Recently, vector quantization of LPC vectors has been applied to speech-recognition systems both in direct applications[5,6] and in conjunction with work on the application of hidden Markov models (HMMs) to recognition.[7,8]

The main idea of vector quantization is summarized as follows: assume that a training set $\{T\}$ of $I$ LPC vectors is given. It is desired to find a code book of $M^*$ LPC vectors such that the average distance of a vector in $\{T\}$ from the closest code book entry is minimized. Thus we wish to find a set $\{R\}$ of reference vectors that minimizes the

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

average distance $\bar{D}_I(M^*)$ given by

$$\bar{D}_I(M^*) = \min_{\{R\}} \left[ \frac{1}{I} \sum_{i=1}^{I} \min_{1 \le m \le M^*} [d(T_i, R_m)] \right], \tag{1}$$

where $d(T_i, R_m)$ is the LPC distance between training vector $T_i$ and code book entry $R_m$.

The optimum code book is generated by a method similar to the $K$-means algorithm. Starting with an initial guess of $M^*$ entries, each vector of the training set is assigned to the closest entry. The centroids of the $M^*$ subsets (clusters) obtained in this manner are used as new trial entries in the code book, and the iteration is continued until some stopping criterion is satisfied.

For large $M^*$, the choice of initial guesses can be quite important, and it is unlikely that a randomly chosen initial guess is a good one. For this reason the splitting algorithm was devised in Ref. 1. In this algorithm a code book of $M = 2$ entries is optimized, as described above, starting with a random initial guess. Next, each optimum code book entry for $M = 2$ is split into 2 and used as an initial guess for a code book of size $2M$. This process is used until $M = M^*$. To distinguish this algorithm from others considered later, we call it the *binary*-split algorithm.

To the best of our knowledge, all speech-related applications of vector quantization so far have used this binary-split algorithm. However, a priori, the requirement that every code word be split appears to be too restrictive. For example, after optimizing an $M = 2$ code book, if one cluster contains almost all the training set and the other contains just a few elements, it might be argued that only the larger cluster should be split. Thus it is of interest to consider "single-split" algorithms in which a single cluster is split at each iteration.

For very large $M^*$ (e.g., 1024 or 2048) single-split algorithms might require prohibitive amounts of computation. However, $M^*$ on the order of 64 or 128 can be quite useful in certain applications.[8] In these cases a single-split algorithm is quite feasible. In any case, it is of interest to know whether or not a single-split algorithm yields a better code book than the binary-split algorithm.

There are at least three reasonable ways of implementing the splitting rule of a single-split algorithm for training the vector quantizer. To describe these three splitting rules we need some definitions. Let

$\{\hat{T}_M(m)\}$ = The set of training vectors represented by the $m$th code book entry (cluster) in a size $M$ vector quantizer

$C_M(m)$ = The number of training vectors in $\hat{T}_M(m)$

$d_M(m)$ = The average distance (distortion) of the $C_M(m)$ vectors from the $m$th code-book entry

$D_M(m)$ = The total distance (distortion) of the $C_M(m)$ vectors.

We then have the relationships

$$I = \sum_{m=1}^{M} C_M(m) \tag{2}$$

$$d_M(m) = \frac{1}{C_M(m)} \sum_{q=1}^{C_M(m)} d(\hat{T}_M(m)_q, R_m) \tag{3}$$

$$D_M(m) = C_M(m) \cdot d_M(m). \tag{4}$$

Using eqs. (2) through (4) we can write the average distortion of eq. (1) as

$$\bar{D}_I(m) = \min_{\{R\}} \left[ \frac{\sum_{m=1}^{M} D_M(m)}{\sum_{m=1}^{M} C_M(m)} \right] \tag{5a}$$

$$= \min_{\{R\}} \left[ \frac{\sum_{m=1}^{M} d_M(m) C_M(m)}{\sum_{m=1}^{M} C_M(m)} \right]. \tag{5b}$$

Based on the above definitions, the three splitting rules we have considered are:

Rule 1: Split the cluster, $m$, with the largest number of vectors, $C_M(m)$. We denote the resulting (vector quantizer) VQ code-word set as $R_c$.

Rule 2: Split the cluster, $m$, with the largest average distortion, $d_M(m)$. We denote the resulting VQ code-word set as $R_d$.

Rule 3: Split the cluster, $m$, with the largest total distortion, $D_M(m)$. We denote the resulting VQ code-word set as $R_D$.

The key issue is how do the different splitting rules affect the properties of the resulting vector quantizer—in particular the average distortion [eq. (1)] and the coverage of the LPC space.

We have run a series of experimental evaluations of the single-split and binary-split algorithms for training the VQ. We have found that each of the different splitting criteria leads to a different reference prototype set (VQ code book); however, all the VQ sets had essentially the *same average distortion*. We were also able to show that the coverage of the LPC space for all VQ sets was identical, and that the

average distance of any one VQ set from another VQ set was smaller than the average distortion of the training set. Hence, the different implementations of the training algorithm for the VQ lead to equivalent VQ reference sets. Thus for any practical application the simple binary-split algorithm is effective for deriving the VQ code book entries.

The outline of this paper is as follows. In Section II we review the Linde et al.[1] implementation of the binary-split VQ training algorithm and show how we modified it to handle the single-split case. In Section III we discuss the results of several experiments on testing the different implementations of the training algorithm. In Section IV we provide a discussion and summary of the results.

## II. IMPLEMENTATION OF THE VQ TRAINING ALGORITHM

The implementation of the VQ training algorithm is essentially the one proposed by Linde et al.[1] A flow diagram of this procedure for the binary-split case is given in Fig. 1a and for the single-split case in Fig. 1b. Given $M$ code words, each vector of the training set $T$ is assigned to the code word closest to it. The average distortion $\bar{D}_I(M)$ is computed for this assignment of the $I$ training vectors to $M$ clusters. $M$ new code words are obtained as centroids (i.e., averaged normalized autocorrelations) of each cluster, and the distortion $\bar{D}_I(M)$ computed again. This process is iterated until it converges, i.e., until the percent change in distortion is less than a preset value $\epsilon$ (chosen to be 1 percent in our simulations). Once convergence is achieved, $M$ is doubled by splitting each code word into two. The entire process is repeated until $M = M^*$. The iteration is initialized by choosing two arbitrary code words.

In our implementation, we made one modification to the VQ training algorithm of Fig. 1. We inserted a check after the classification of the training set vectors to see if any cluster is empty (i.e., contains none of the training set vectors). In such a case the "largest" cluster is split into two clusters, and the convergence test is bypassed (to ensure a reclassification in which each cluster is nonempty). However, for the data used in this experiment, an empty cluster never occurred. In subsequent tests with larger $M^*$ we did encounter such cases.

For the single-split algorithm (Fig. 1b), only one modification is required. After convergence, only the "largest cluster" is split. Here largest can refer to the cluster with the largest average distortion, total distortion, or count.

For a convergence criterion of $\epsilon = 1$ percent, typically it takes three to six iterations of the classification loop to obtain a convergent set of clusters and centroids. We also found that the algorithms of Fig. 1a and 1b work extremely reliably over a broad range of types of training

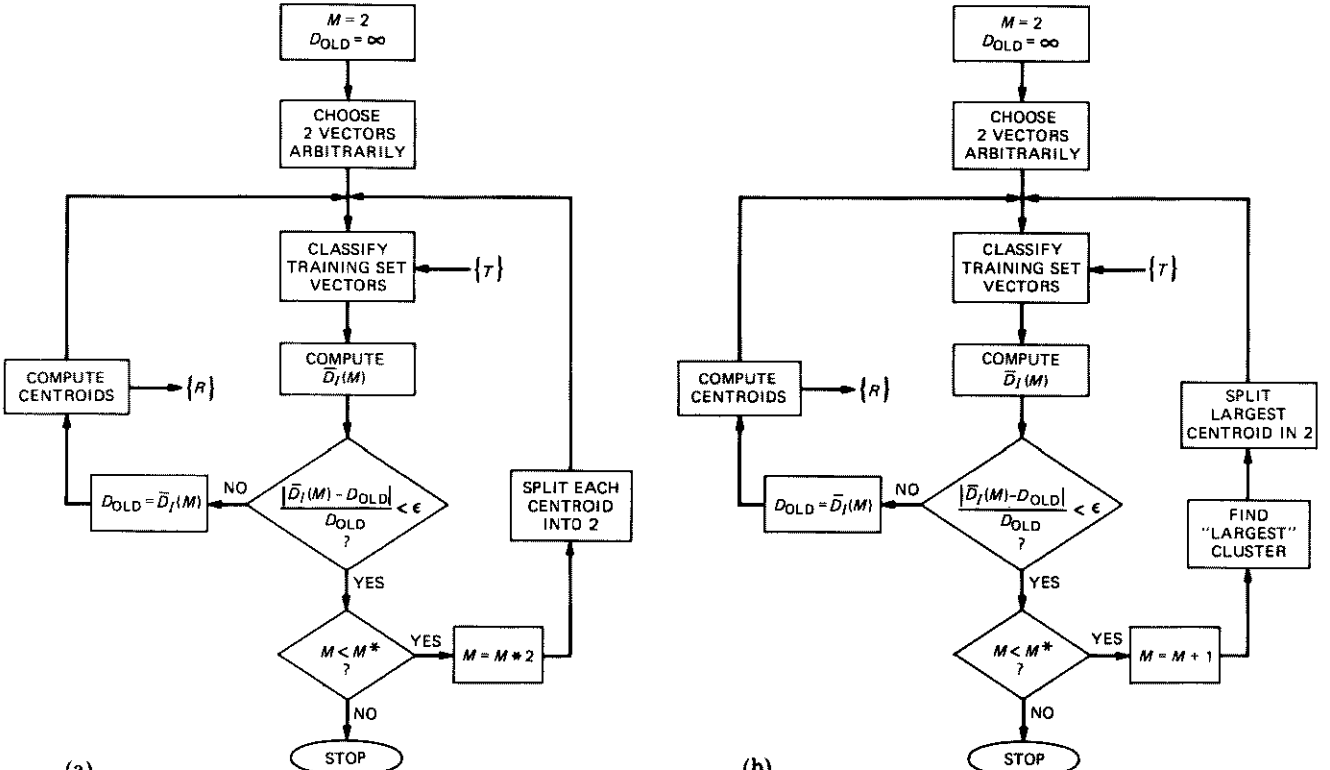



Fig. 1—Flow charts of the vector quantizer training algorithms. (a) The binary-split algorithm. (b) The single-split algorithm.

data (e.g., collected from a single talker, collected from many talkers, collected from a corpus of isolated words, collected from sentence-length material, etc.).

## III. COMPARISON OF THE BINARY- AND SINGLE-SPLIT ALGORITHMS

To compare the performances of the binary- and single-split VQ training algorithms of Fig. 1, several tests were run. The database consisted of a set of 39,708 LPC vectors. The LPC analysis used a 6.67-kHz sampling rate and an eigth-order analysis of 300 sample (45 ms) frames of speech. The sample frames had been preemphasized with a simple, first-order digital network (preemphasis factor of 0.95) and windowed by a 300-sample Hamming window. Frames were taken 100 samples apart across the duration of each word of a series of 1000 isolated words (digits) spoken by 100 talkers (50 male, 50 female). All recordings were made over dialed-up telephone lines through a local PBX connection. All silence outside the spoken words was eliminated by a word endpoint detector;[9] hence, all LPC training frames were from within word boundaries.

Several aspects of the binary- and single-split training algorithms were studied. The first question considered was whether the two training procedures yielded identical results (i.e., whether the resulting LPC code words and the clusters from which they were derived were identical). Figure 2 shows plots of the cluster splitting for an $M^* = 8$ solution for the binary-split algorithm (Fig. 2a) and the single-split algorithm based on average distance splitting (Fig. 2b). It can be seen that the resulting eight clusters in the single-split case come from very different splits than those for the binary-split case. For example, in

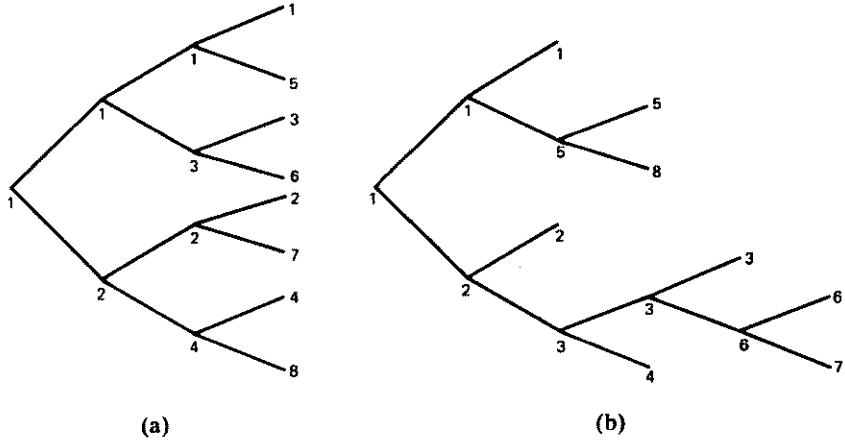



Fig. 2—Splitting charts for an $M^* = 8$ vector quantizer with splits based on average distortion. (a) The binary-split training algorithm. (b) The single-split algorithm.

the single-split case, final clusters 6 and 7 come from four splits of the original cluster 2, whereas final clusters 1 and 2 come from single splits of original clusters 1 and 2. In the binary-split case all final clusters come from two splits of original clusters 1 and 2. Similarly, the actual clusters were grossly different for the three different criteria for the single-split algorithm.

The next question we considered was how the different training procedures differed in performance. Figures 3 through 5 show a series of plots of statistics comparing some of the details of the individual training procedures. For each of these plots, Parts (a) through (d) show results for the binary-split case, the single-split case based on count, the single-split case based on average distortion, and the single-

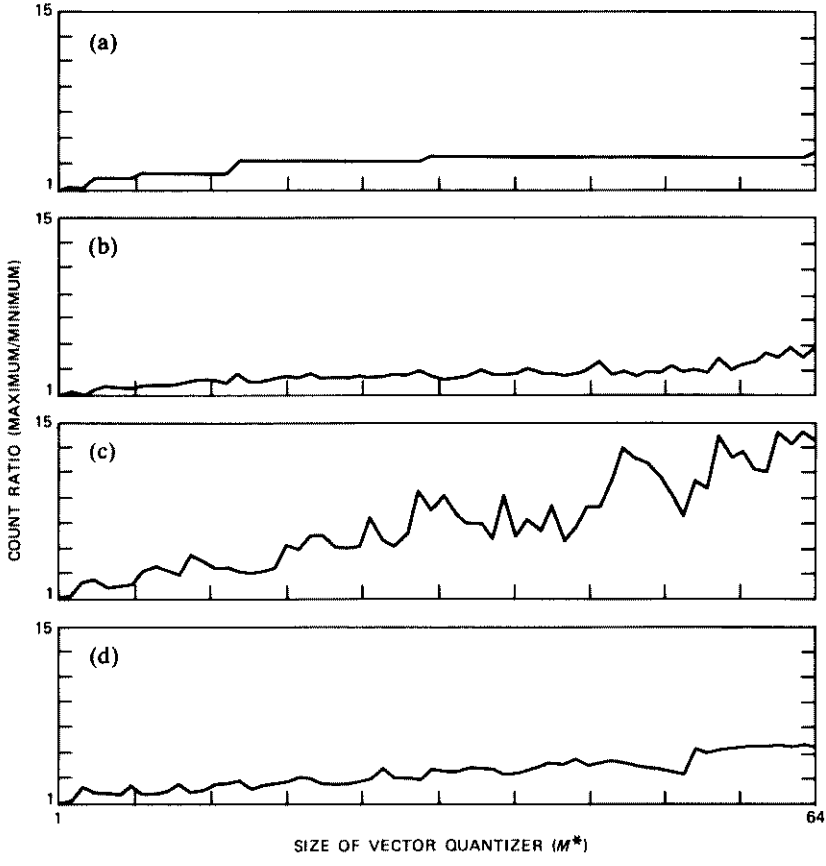



Fig. 3—Plots of count ratio (maximum cluster count divided by minimum cluster count) as a function of the size of the vector quantizer. (a) Binary-split training. (b) Single-split training based on count. (c) Single-split training based on average distortion. (d) Single-split training based on total distortion.

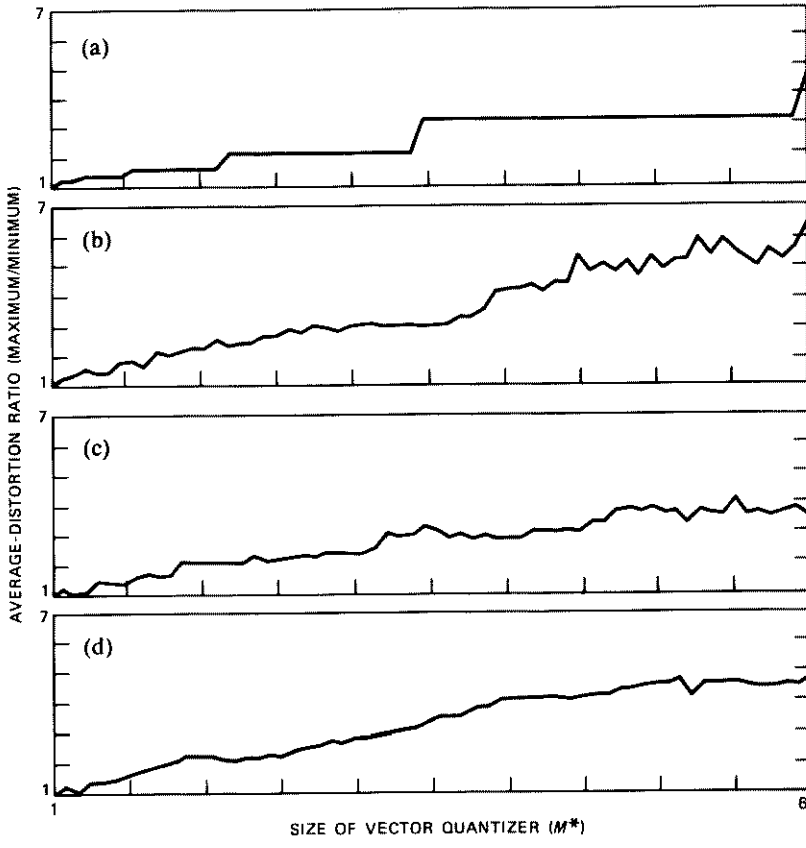



Fig. 4—Plots of average distortion ratio as a function of the size of the vector quantizer. (a) Binary-split training. (b) Single-split training based on count. (c) Single-split training based on average distortion. (d) Single-split training based on total distortion.

split case based on total distortion. The statistics plotted are ratio of maximum to minimum cluster count (Fig. 3), ratio of maximum to minimum average distortion (Fig. 4), and ratio of maximum to minimum total distortion (Fig. 5) versus size of the vector quantizer. These statistics were chosen because each of them should ideally approach 1.0 for clusters that are of equal size according to the corresponding splitting criterion. For example, we would expect the count ratio to approach 1.0 for the split on count criterion but not necessarily for the other splitting criteria.

Examination of Figs. 3 through 5 shows several interesting things. As seen in Fig. 3, the count ratio for the binary-split case for $M^* = 64$ (4.1) is actually smaller than the count ratio for the single split on count case for $M^* = 64$ (4.8). The count ratios for the other two split

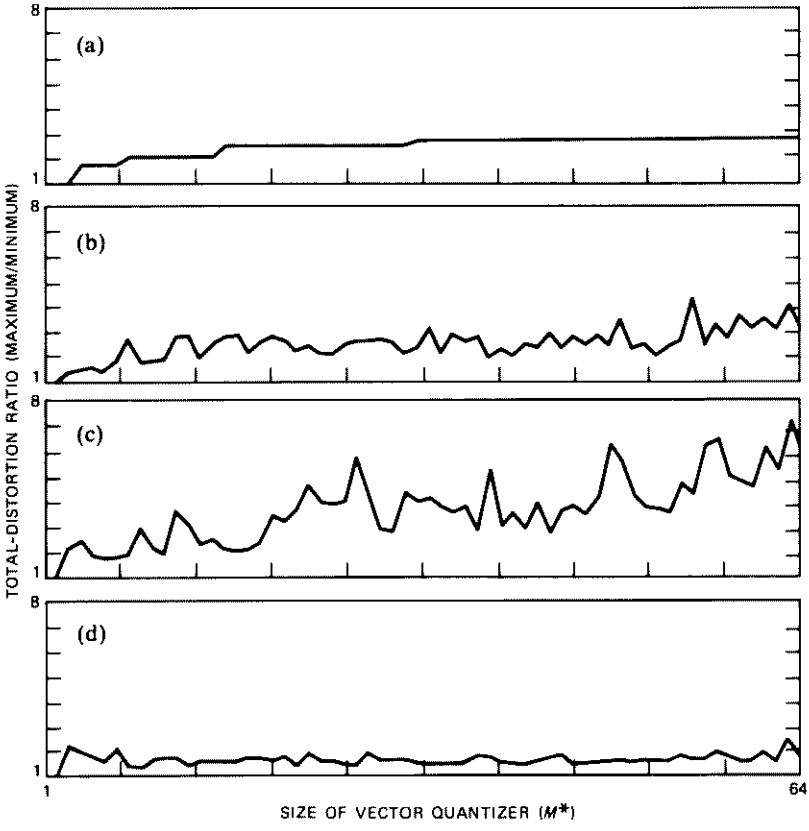



Fig. 5—Plots of total-distortion ratio as a function of the size of the vector quantizer. (a) Binary-split training. (b) Single-split training based on count. (c) Single-split training based on average distortion. (d) Single-split training based on total distortion.

criteria are indeed larger than for the split on count, as expected. Figure 4 shows that the average-distortion ratio is smallest (4.1) at $M^* = 64$ for the single split on average-distortion case; however, the distortion ratios for the binary case (4.4) and the single split on total-distortion (4.7) cases are only slightly larger. Finally, Fig. 5 shows a similar set of results on the total-distortion-ratio statistic in which the results for $M^* = 64$ for the binary-split case (2.7) are only slightly worse than for the single split on total-distortion case (2.6).

The results of Figs. 3 through 5 indicate that the binary-split case seems to yield cluster training statistics that are almost as good as the best statistics for any of the single-split cases in terms of count ratio, average-distortion ratio, and total-distortion ratio. Hence, from the point of view of cluster statistics, the binary-split cases appear to give the best overall performance.

Two gross performance checks were made on the training algorithms. In the first test, the average distance between vector quantizer sets obtained from the different training procedures was calculated as a function of $M^*$. The results of this test are given in Table I. It can

Table I—Average distance between code book entries of vector quantizers designed on the basis of count ($R_c$), average distortion ($R_d$), total distortion ($R_D$), and binary splitting ($R_B$)

| $M^*$ | $\bar{d}(R_c, R_d)$ | $\bar{d}(R_c, R_D)$ | $\bar{d}(R_c, R_B)$ | $\bar{d}(R_d, R_D)$ | $\bar{d}_I(M^*)$† |
|---|---|---|---|---|---|
| 4 | 0.384 | 0.019 | 0.047 | 0.270 | 0.707 |
| 8 | 0.125 | 0.138 | 0.157 | 0.101 | 0.426 |
| 16 | 0.148 | 0.143 | 0.160 | 0.065 | 0.326 |
| 32 | 0.191 | 0.108 | 0.175 | 0.132 | 0.255 |
| 64 | 0.216 | 0.131 | 0.148 | 0.131 | 0.203 |

† Average distance between the training vectors and the code words representing them.

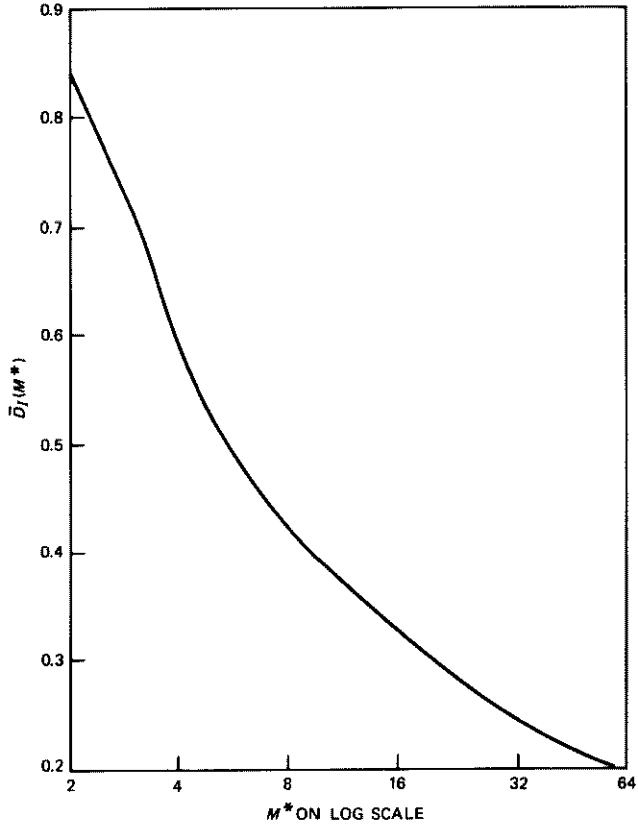



Fig. 6—Plot of average training set distortion $\bar{D}_I(M^*)$ as a function of the size of the vector quantizer.

be seen that the average distance between vector quantizer sets is as small or smaller than the average distance of the training vectors to the code book sets. Hence, the code book sets derived from the different training algorithms are, on average, quite close to each other.

The second test we performed was to measure the average distortion, $\bar{D}_I(M^*)$ versus $M^*$ for the different training algorithms for values of $M^*$ from 2 to 64. The results of this test are plotted in Fig. 6. On the

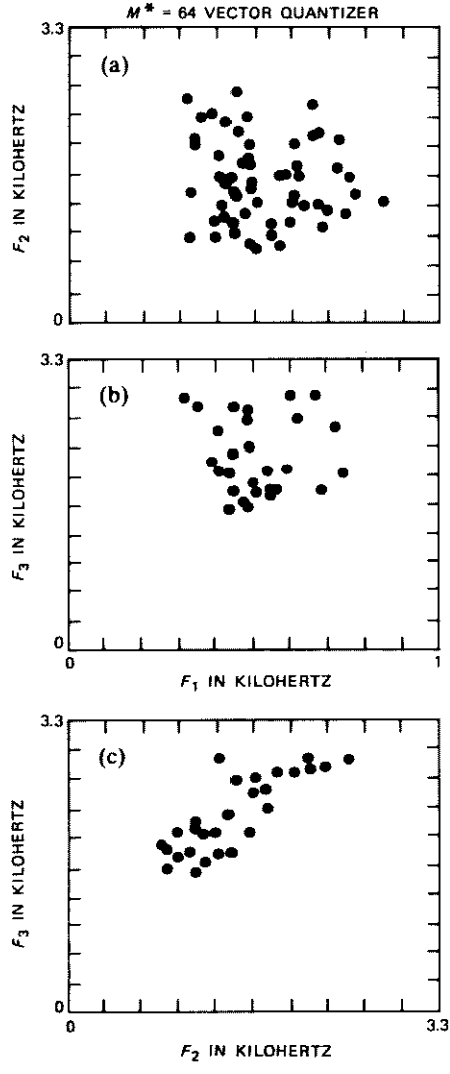



Fig. 7—Plots of code-word coverage in the $F_1$-$F_2$, $F_1$-$F_3$, and $F_2$-$F_3$ planes for an $M^*$ = 64 vector quantizer.

scale of this plot, the differences in average distortion are *indistinguishable* among the different vector quantizers.

The third and final question we considered concerns the coverage of the space of speech sounds by the optimum code books. A good way of displaying this coverage is to look at the code books in the space of formant frequencies. The formant frequencies (and bandwidths) for each entry of the code book are given by the zeroes of the trigonometric polynomial associated with it. Thus each code book may be displayed

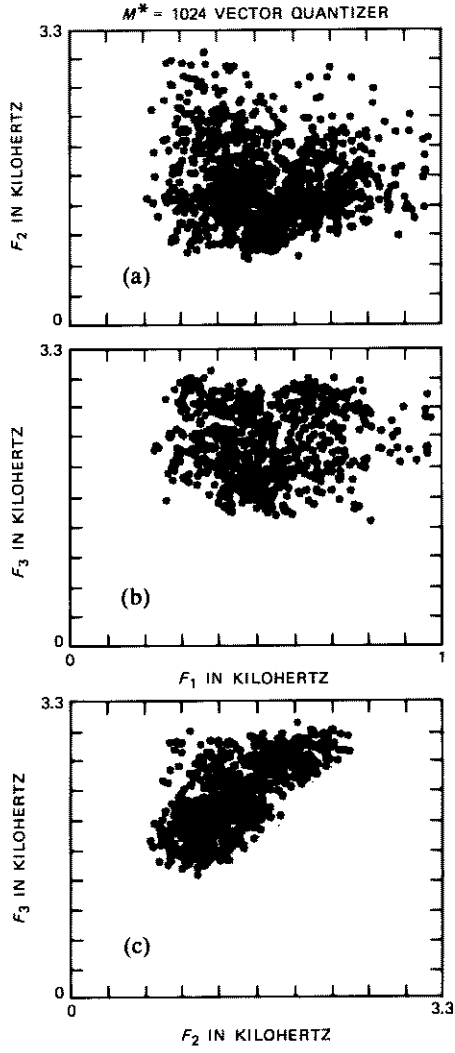



Fig. 8—Plots of code-word coverage in the $F_1$-$F_2$, $F_1$-$F_3$, and $F_2$-$F_3$ planes for an $M^*$ = 1024 vector quantizer.

as a scatter plot in $F_1$-$F_2$-$F_3$ space. Projections of this scatter diagram on the $F_1$-$F_2$, $F_1$-$F_3$, and $F_2$-$F_3$ planes are shown for a typical code book in Figs. 7 and 8 for the code books obtained from the binary-split training algorithm for $M^* = 64$ (Fig. 7) and $M^* = 1024$ (Fig. 8). It is seen that the code words cover the expected regions in the formant frequency planes fairly uniformly. The major difference between the coverage of the $M^* = 1024$ and the $M^* = 64$ code books is the density of coverage of the areas in the respective formant frequency planes. The coverage of the single-split algorithms for $M^* = 64$ was essentially identical to that of the binary-split algorithm.

## IV. DISCUSSION

Our overall conclusion from the tests that compared the fine and gross differences in clustering LPC vectors via a VQ training algorithm is that all the variations in the training procedure that we studied (i.e., different splitting procedures, different convergence criteria, etc.) lead to essentially indistinguishable differences in the set of VQ code book entries. Since the binary-split algorithm, as discussed by Linde et al.[1] requires the least amount of computation, it is the best of the algorithms considered.

In this paper we present the results of a series of experiments on a training set of 39,708 vectors. More recently we have experimented with the binary-split VQ training procedure on a number of different training sets whose size varied from 10,000 to 600,000 vectors. We found that the training procedure always rapidly and reliably converged to a set of code book vectors whose properties were similar to those described in this paper. We are currently using the VQ code book sets in work related to speech recognition and speech coding.

## V. ACKNOWLEDGMENTS

The authors gratefully acknowledge several fruitful discussions with Fred Juang of Bell Laboratories concerning the characteristic properties of the VQ clusters. Juang's insight into the training procedure and the resulting properties of the code book vectors coincided with the results we found in this study.

## REFERENCES

1. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," IEEE Trans. Commun., *COM-28*, No. 1 (January 1980), pp. 84–95.
2. B. Juang, D. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," IEEE Trans. on Acoust., Speech, and Signal Proc., *ASSP-30*, No. 2 (April 1982), pp. 294–303.
3. A. Buzo, et al., "Speech Coding Based Upon Vector Quantization," IEEE Trans. on Acoust., Speech, and Signal Proc., *ASSP-28*, No. 5 (October 1980), pp. 562–74.
4. D. Wong, B. Juang, and A. H. Gray, Jr., "An 800 Bit/S Vector Quantization LPC Vocoder," IEEE Trans. on Acoust., Speech, and Signal Proc., *ASSP-30*, No. 5 (October 1982), pp. 770–80.

5. A. Buzo, H. Martinez, and C. Rivera, "Discrete Utterance Recognition Based Upon Source Coding Techniques," Proc. ICASSP-82 (May 1982), pp. 539–42.
6. J. E. Shore and D. Burton, "Discrete Utterance Speech Recognition Without Time Normalization," Proc. ICASSP-82 (May 1982), pp. 907–10.
7. R. Billi, "Vector Quantization and Markov Source Models Applied to Speech Recognition," Proc. ICASSP-82 (May 1982), pp. 574–7.
8. L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker Independent, Isolated Word Recognition," B.S.T.J., *62*, No. 4 (April 1983), pp. 1075–105.
9. L. F. Lamel, et al. "An Improved Endpoint Detector for Isolated Word Recognition," IEEE Trans. on Acoust., Speech, and Signal Proc., *ASSP-29*, No. 4 (August 1981), pp. 777–85.

## AUTHORS

**Stephen E. Levinson,** B. A. (Engineering Sciences), 1966, Harvard; M.S. and Ph.D. (Electrical Engineering), University of Rhode Island, Kingston, 1972 and 1974, respectively; General Dynamics, 1966–1969; Yale University, 1974–1976; Bell Laboratories, 1976—. From 1966 to 1969, Mr. Levinson was a design engineer at Electric Boat Division of General Dynamics in Groton, Connecticut. From 1974 to 1976, he held a J. Willard Gibbs Instructorship in Computer Science at Yale University. In 1976, he joined the technical staff at Bell Laboratories, where he is pursuing research in the areas of speech recognition and cybernetics. Member, Association for Computing Machinery; Fellow, Acoustical Society of America; Senior Member, IEEE, editorial board of Speech Technology; Associate Editor, IEEE Transactions on Acoustics, Speech and Signal Processing.

**Lawrence R. Rabiner,** S.B. and S. M., 1964, Ph.D. (Electrical Engineering), The Massachusetts Institute of Technology: Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication of the Acoustical Society, ASSP Technical Committee on Speech Communication; Member, IEEE Proceedings Editoral Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

**Man Mohan Sondi,** B.Sc. (Physics), Honours degree, 1950, Delhi University, Delhi, India; D.I.I.Sc. (Communications Engineering), 1953, Indian Institute of Science, Bangalore, India; M.S., 1955; Ph.D. (Electrical Engineering), 1957, University of Wisconsin, Madison, Wisconsin; Bell Laboratories, 1962—. Before joining Bell Laboratories, Mr. Sondhi worked for a year at the Central Electronics Engineering Research Institute, Pilani, India and taught for a year at the University of Toronto. At Bell Laboratories his research has included work on speech signal processing, echo cancellation, adaptive filtering, modelling of auditory and visual processes, and acoustical inverse problems. From 1971 to 1972 Mr. Sondhi was a guest scientist at the Royal Institute of Technology, Stockholm, Sweden.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

# Upper Bounds on the Minimum Distance of Trellis Codes

By A. R. CALDERBANK,* J. E. MAZO,* and H. M. SHAPIRO†

(Manuscript received March 16, 1983)

A trellis code is a "sliding window" method of encoding a binary data stream into a sequence of real numbers that are input to a noisy transmission channel. When a trellis code is used to encode data at the rate of $k$ bits/channel symbol, each channel input will depend not only on the most recent block of $k$ data bits to enter the encoder but will also depend on, say, the $\nu$ bits preceding this block. The $\nu$ bits determine the state of the encoder and the most recent block of $k$ bits generates the channel symbol conditional on the encoder state. The performance of trellis codes, like that of block codes, depends on a suitably defined minimum-distance property of the code. In this paper we obtain upper bounds on this minimum distance that are simple functions of $k$ and $\nu$. These results also provide a lower bound on the number of states required to achieve a specific coding gain.

## I. INTRODUCTION

In this paper we are concerned with transmission of digital data using trellis codes to gain some noise immunity over standard uncoded methods. We assume pulse amplitude modulation whereby the values of the transmitted data are estimated from a sequence of samples $r^j$ generated by a receiver. These output samples are often modeled as

$$r^j = x^j + n^j, \tag{1}$$

where $x^j$ is a real number sequence determined by the source sequence of binary data and $n^j$ is an independent zero-mean white Gaussian

* Bell Laboratories. † Swarthmore College, Swarthmore, Pennsylvania.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

noise sequence of variance $\sigma^2$. For uncoded transmission at rate $k$ bits/symbol, $x^j$ takes on one of $2^k$ fixed values. Error performance may be improved using coding, but if we insist on transmitting at rate $k$ bits/symbol then we must increase the number of possible values taken by the $x^j$. We can choose either a block or tree (trellis) structure for the code. In this paper we consider only trellis codes. The performance of trellis codes, like that of block codes, depends on a suitably defined minimum-distance property of the code. We obtain upper bounds on this minimum distance, $d_{\min}$. The analogous problem for block codes is well studied, but little work has been done on distance properties of trellis codes.[1,2]

We assume the following model for encoding the binary data (i.e., choosing the $x^j$) prior to transmission over the Gaussian channel. Regard the incoming binary digits as partitioned into blocks of $k$ consecutive bits. The real number $x^j$ is to be a time-independent function of the most recent $k$-bit block and also of the $\nu$ bits preceding this block. Thus if $\{a_i\}$ is the binary data sequence, we assume

$$x^j = x(a_{jk}, a_{jk-1}, \cdots, a_{jk-(k-1)}; a_{(j-1)k}, \cdots, a_{(j-1)k-(\nu-1)}). \tag{2}$$

This is an example of a $k$-bit/symbol trellis code. We regard the $\nu$ "old" bits as determining the state of the encoder (there are $2^\nu$ possible states) and the $k$ "new" bits as generating the channel symbol (there are $2^k$ possible symbols) conditional on the encoder state. The trellis structure is made evident by drawing an example. Fig. 1 shows the case $k = 1$, $\nu = 2$.

If, in this example, the encoder is in state (00) at time $j$, and the next bit (block of $k = 1$ bits) to be transmitted is a 1, then we transmit the symbol $x(100)$ and move to state (10).

Other trellis codes exist. For example, we could define a code with just three trellis states or the symbols $x^j$ could also depend on the time index $j$. However, we shall only consider trellis codes determined by (2). The trellis structure of (2) is identical to that of linear algebraic

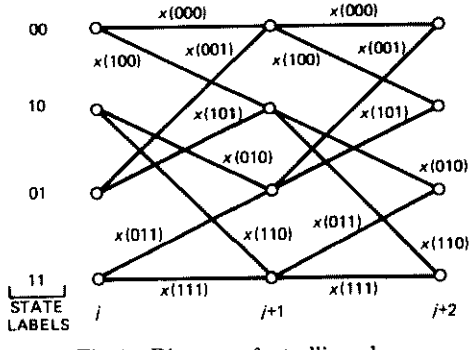



Fig. 1—Diagram of a trellis code.

convolutional codes. We use the term sliding window trellis codes for trellis codes determined by (2).

To simplify the discussion in the text we shall assume that $k$ divides $\nu$. The general case is treated in Appendix B.

The problem we consider involves certain distance properties of trellis codes. To motivate it, consider the decoding problem. Optimum decoding involves finding the most likely path through the trellis, given the observed sequence (1).[3] Typically, the path chosen will not coincide with the correct path for all time but will occasionally diverge from it and remerge at a later time. This is called an error event, and we generically denote it by the letter $E$. For example, with the trellis in Fig. 1, $x(000)$ may have been sent several times in succession, resulting in the straight path shown in Fig. 2, but noise may have caused the decoder to choose an alternate path. In Fig. 2 the decoder chose the symbols $x(100)$, $x(010)$, $x(001)$ instead of $x(000)$, $x(000)$, $x(000)$.

An error event $E$ of length $L$ lasts from time $i$ to time $i + L$, the decoder having decided upon the symbol sequence $\hat{x}^{i+1}, \cdots, \hat{x}^{i+L}$ instead of the correct sequence $x^{i+1}, \cdots, x^{i+L}$. The (squared) Euclidean distance $d^2$ $(= d^2(E))$ between the two paths of $E$ is given by

$$d^2 = \sum_{j=i+1}^{i+L} (x^j - \hat{x}^j)^2 \tag{3}$$

and is crucial to determining the probability $P(E)$ of an error event $E$. With the white noise assumption made in (1), $P(E)$ is easy to calculate and, when $d^2 \gg \sigma^2$, it is approximately given by

$$P(E) \approx \exp\left(-\frac{d^2}{8\sigma^2}\right). \tag{4}$$

Equation (4) leads us to expect that, for small noise, symbol error probabilities will be determined by error events having the smallest minimum distance between their two paths and it becomes of interest to design codes that have good minimum-distance properties in this sense. Such designs have recently been considered by Ungerboeck, who obtained on the order of 3-dB performance improvements (factor

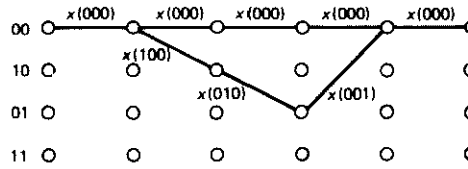



Fig. 2—Example of an error event.

of 2 in minimum distance) over the uncoded case for $k = 1, 2$, and four or eight states in the trellis.[4]

Ungerboeck based his designs on a computer search of binary convolutional codes with $2^\nu$ states, rate $k/(k+1)$, and a particular mapping of the output binary $(k+1)$ tuples to $2^{k+1}$ equally spaced channel symbols ($\pm 1$, $\pm 3$, etc.). His use of convolutional codes thus conforms to the general scheme of (2), which implies the same trellis structure as described herein. However, his a priori choice of only $2^{k+1}$ equally spaced channel symbols is certainly restrictive in principle. In this paper we consider the natural question of how large $d^2_{\min}/P$ can be made if these restrictions are removed. Here, $d_{\min}$ is the minimum distance between all pairs of paths associated with error events in the trellis, and $P$ is the average transmitted power.

Section II gives a detailed description of the trellis structure and of error events. If $S$ is a finite set of error events, then

$$\min_{E \in S} \{d^2(E)\} \leq \frac{1}{|S|} \sum_{E \in S} d^2(E), \tag{5}$$

since the minimum of a set of real numbers is bounded above by their average. This observation is the basis of our first two bounds. The first and simplest bound is

$$\frac{d^2_{\min}}{P} \leq 4\left(1 + \frac{\nu}{k}\right), \tag{6}$$

which is obtained in Section III. A more detailed analysis in Section IV gives

$$\frac{d^2_{\min}}{P} \leq \frac{2^{k+1}}{2^k - 1}\left(1 + \frac{\nu}{k}\right), \tag{7}$$

which is stronger than (6) provided $k > 1$. Let $T$ be another finite set of error events and let $r_1, r_2 \geq 0$ be real numbers satisfying $r_1 + r_2 = 1$. Then,

$$\min_{E \in S \cup T} \{d^2(E)\} \leq r_1\left(\frac{1}{|S|} \sum_{E \in S} d^2(E)\right) + r_2\left(\frac{1}{|T|} \sum_{E \in T} d^2(E)\right), \tag{8}$$

since the minimum of a set of real numbers is bounded above by any weighted average of those numbers. In Section V, by choosing $S$, $T$, $r_1$, and $r_2$, appropriately, we prove

$$\frac{d^2_{\min}}{P} \leq \left(\frac{2^{2k+1}}{2^{2k} - 1}\right)\left(2 + \frac{\nu}{k}\right). \tag{9}$$

This bound is stronger than (7) provided $\nu > k(2^k - 1)$. Combining (7)

and (9) we have

$$\frac{d_{\min}^2}{P} \leq \min\left[\frac{2^{k+1}}{2^k-1}\left(1+\frac{\nu}{k}\right), \frac{2^{2k+1}}{2^{2k}-1}\left(2+\frac{\nu}{k}\right)\right]. \tag{10}$$

Extensions of bounds (6), (7), and (9) to the case when $k$ does not divide $\nu$ are given in Appendix B.

## II. A GROUP ACTION ON THE TRELLIS

In later sections we obtain upper bounds on $d_{\min}^2/P$ by considering sets of error events that are fixed by a group of symmetries of the trellis. In this section we describe the group.

We consider trellis codes with $2^r$ states transmitting $k$ bits/channel symbol and for simplicity we assume that $k$ divides $\nu$. States are labelled with binary $\nu$ tuples, and edges of the trellis are labelled with binary $\nu + k$ tuples. We identify the binary $r$ tuple $(b_0, \cdots, b_{r-1})$ with the integer

$$b_0 2^0 + b_1 2^1 + \cdots + b_{r-1} 2^{r-1}.$$

The states are labelled with binary $\nu$ tuples $00\cdots0$, $10\cdots0$, $010\cdots0$, $110\cdots0$, $\cdots$, $11\cdots1$, in increasing order, from top to bottom as in Fig. 1. The edges are labelled with binary $\nu + k$ tuples $x_0 = x(0\cdots0)$, $x_1 = x(10\cdots0)$, $x_2 = x(010\cdots0)$, $x_3 = x(110\cdots0)$, $\cdots$, $x_{2^{\nu+k}-1} = x(11\cdots1)$, also in increasing order, from top to bottom as in Fig. 1. Set $N = (k+\nu)/k$. If we write an edge label as $x(s_0, \cdots, s_{N-1})$, then it will be understood that each $s_j$ is a binary $k$ tuple. A "+" appearing in the argument of a label means bit-by-bit modulo 2 addition. A similar notation will be used for states.

We define a group of symmetries of the trellis. These symmetries will map error events of length $L$ to error events of length $L$. For each binary $\nu + k$ tuple $t$, we define a permutation $g_t$ of the edge labels $x(s)$ by the rule

$$g_t(x(s)) = x(s + t). \tag{11}$$

For example, when $k = 1$, $\nu = 2$, and $t = (010)$,

$$\underline{x} = \begin{bmatrix} x(000) \\ x(100) \\ x(010) \\ x(110) \\ x(001) \\ x(101) \\ x(011) \\ x(111) \end{bmatrix} \mapsto g_{010}(\underline{x}) = \begin{bmatrix} x(010) \\ x(110) \\ x(000) \\ x(100) \\ x(011) \\ x(111) \\ x(001) \\ x(101) \end{bmatrix}. \tag{12}$$

This may also be written

$$g_{010}(\underline{x}) = T\underline{x}, \tag{13}$$

where $T$ is the permutation matrix

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & & & & \\ 1 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & & & & \\ & & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ & & & & 1 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{14}$$

If $\mathbf{G}_{k,\nu} = \{g_t \mid t \text{ is a binary } \nu + k \text{ tuple}\}$, then $\mathbf{G}_{k,\nu}$ is an abelian group of order $2^{k+\nu}$, and every element $g_t$ of $\mathbf{G}_{k,\nu}$ satisfies $g_t^2 = e$, where $e$ is the group identity.

*Lemma 1: Any pair of edge labels is interchanged by a unique group element.*

*Proof:* Edge labels $x(s)$ and $x(u)$ are interchanged only by $g_{s+u}$. □

We call the time sections (0, 1), (1, 2), ⋯ the components of the trellis. We shall now show how to choose binary $\nu + k$ tuples $t = t^0, t^1, \cdots$ so that if $g_{t^i}$ is applied to the edges in component $i$, then an error event of length $L$ is always mapped to another error event of length $L$. It is, in general, necessary to choose a different $g_t$ for each component since if we simply apply the same permutation $g_t$ to the edges in every component, then an error event $E$ need not be transformed to another error event. Thus, if $g_{010}$ is applied to each component of the error event shown in Fig. 2, then we obtain the edges shown in Fig. 3. The permutation $g_{010}$ transforms the edge labelled $x(uvw)$, joining state $vw$ and state $uv$, into the edge labelled $x(u(1 + v)w)$, joining state $(1 + v)w$ and state $u(1 + v)$. If $t = t^0 = 010$, then $g_{010}$ permutes the encoder states at time 0 by the rule

$$vw \mapsto (1 + v)w, \tag{15}$$

and permutes the encoder states at time 1 by the rule

$$uv \mapsto u(1 + v). \tag{16}$$

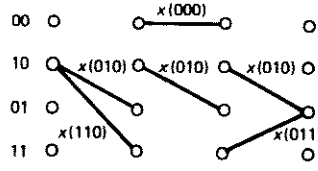



Fig. 3—Permutation $g_{010}$ applied to all edges of an error event.

Similarly, the permutation $g_{t^1}$ permutes encoder states at time 1 and encoder states at time 2. If we want to map error events to error events, then the action of $g_{t^1}$ on encoder states at time 1 must be given by (16). Choose $t^1 = 001$, $t^2 = 100$, $t^3 = 010$, $t^4 = 001$, $\cdots$. The action of $g_{t^0}$, $g_{t^1}$, and $g_{t^2}$ on components 0, 1, and 2 is shown in Fig. 4. Thus the sequence $(g_{t^0}, g_{t^1}, g_{t^2}, \cdots)$ transforms the error event shown in Fig. 2 to the error event shown in Fig. 5.

For general $k$ and $\nu$, let $t = t^0 = (t_0, \cdots, t_{N-1})$ where $N = (k + \nu)/k$ and $t_0, \cdots, t_{N-1}$ are binary $k$ tuples. Let $t^1 = (t_{N-1}, t_0, \cdots, t_{N-2})$ be the vector obtained from $t^0$ by cycling the blocks of $k$ bits to the right and moving the last block, $t_{N-1}$, to the front. Repeat this operation $i$ times to obtain $t^i = (t_{N-i}, \cdots, t_{N-1}, t_0, \cdots, t_{N-i-1})$. For $i \geq N$ we view $i$ as an integer modulo $N$. Thus $t^N = t^0 = t$, $t^{N+1} = t^1, \cdots$. The action of $g_{t^i}$ on encoder states at time $i$ coincides with that of $g_{t^{i-1}}$ being given by the rule

$$s \mapsto (t_{N-i+1}, \cdots, t_{N-1}, t_0, \cdots t_{N-i-1}) + s. \tag{17}$$

If $\mathbf{G}^*_{k,\nu} = \{(g_{t^0}, g_{t^1}, \cdots) \mid t^0 \text{ is a binary } k \text{ tuple}\}$, then $\mathbf{G}^*_{k,\nu}$ is a group of $2^{\nu+k}$ symmetries of the trellis. The group $\mathbf{G}^*_{k,\nu}$ is abelian, and every element has order 2. We denote $(g_{t^0}, g_{t^1}, \cdots)$ by $g^*_{t^0}$, since it is determined by $t^0$.

*Lemma 2: If $i \geq 0$ and if $x(s)$, $x(t)$ are any pair of edge labels in component $i$, then there is a unique element of $\mathbf{G}^*_{k,\nu}$ that interchanges $x(s)$ and $x(t)$.*

*Proof:* This follows from Lemma 1, since the restriction of $\mathbf{G}^*_{k,\nu}$ to the edges in component $i$ is just $\mathbf{G}_{k,\nu}$. □

A set $S$ of error events is said to be *fixed* by $\mathbf{G}^*_{k,\nu}$ if for all $g \in \mathbf{G}^*_{k,\nu}$ and all $E \in S$ we have $g(E) \in S$.

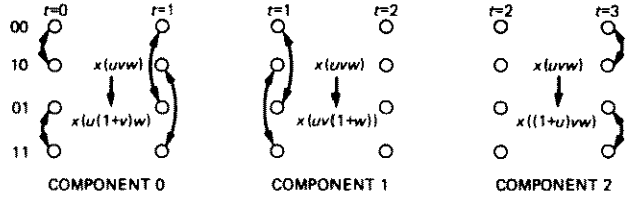



Fig. 4—Action of $g_{t^0}$, $g_{t^1}$, and $g_{t^2}$.

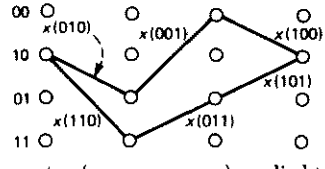



Fig. 5—The symmetry $(g_{t^0}, g_{t^1}, g_{t^2}, \cdots)$ applied to an error event.

*Lemma 3: Let $i \geq 0$ and let $S$ be a set of error events of the same length that is fixed by $\mathbf{G}^*_{k,\nu}$. If $m_i(x(a))$ is the total number of times the edge label $x(a)$ occurs in component $i$ of the error events of $S$, then*

$$m_i(x(a)) = \frac{2|S|}{2^{k+\nu}} \qquad \text{for all } \nu + k \text{ tuples } a.$$

*Proof:* Let $s, t$ be binary $\nu + k$ tuples. By Lemma 2 there is an element of $\mathbf{G}^*_{k,\nu}$ interchanging error events involving $x(s)$ in component $i$ with error events involving $x(t)$ in component $i$. Hence $m_i(x(s)) = m_i(x(t))$. Since the total number of edges in component $i$ is $2|S|$, we have $m_i(x(a)) = 2|S|/2^{k+\nu}$ for all $\nu + k$ tuples $a$.

An *orbit* $S$ of the group $\mathbf{G}^*_{k,\nu}$ is a set of error events satisfying

1. if $E \in S$ and $g \in \mathbf{G}^*_{k,\nu}$ then $g(E) \in S$, and
2. if $E_1, E_2 \in S$ then there exists $g \in \mathbf{G}^*_{k,\nu}$ such that $g(E_1) = E_2$.

Fig. 6 shows an orbit of $\mathbf{G}^*_{1,2}$. Observe that $m_i(x(a)) = 1$ for all $i$ and for all $a$.

## III. THE FIRST BOUND

In this section we derive the upper bound

$$\frac{d^2_{\min}}{P} \leq 4\left(1 + \frac{\nu}{k}\right).$$

This bound will be strengthened in later sections but it seems worth presenting the simpler argument here.

Observe that the average transmitted signal power is simply the average of the transmitted channel symbols, namely

$$P = \frac{1}{2^{\nu+k}} \sum_{i=0}^{2^{\nu+k}-1} x_i^2. \tag{18}$$

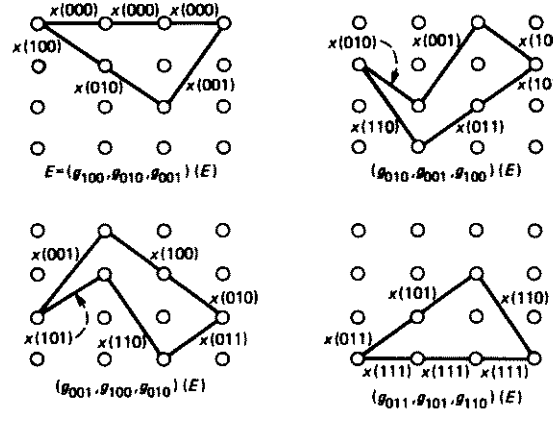



Fig. 6—An orbit of $\mathbf{G}^*_{1,2}$.

(Recall that the channel symbol $x(a_0 \cdots a_{\nu+k-1})$ is also denoted $x_i$ where $i = a_0 + a_1 2^1 + \cdots + a_{\nu+k-1}2^{\nu+k-1}$.) The Euclidean distance between the paths of the error event $E$ shown in Fig. 2 is

$$d^2(E) = (x_0 - x_1)^2 + (x_0 - x_3)^2 + (x_0 - x_5)^2,$$

which is a quadratic form in the variables $x_i$. In general we define

$$\underline{x}^T = (x_0, x_1, \cdots, x_{2^{\nu+k}-1}), \tag{19}$$

where the superscript $T$ denotes matrix transpose. Then the Euclidean distance $d^2(E)$ between the paths of an error event $E$ is given by

$$d^2(E) = \underline{x}^T A(E) \underline{x}, \tag{20}$$

where $A(E)$ is a symmetric, positive semi-definite matrix which we call the *distance matrix* of $E$. The distance matrix $A(E)$ has two properties that we wish to note:

Property I. The $i$th diagonal element of $A(E)$ counts the number of times the symbol $x_i$ occurs in the error event.

Property II. The rows of $A(E)$ sum to zero.

By (18) and (20),

$$\frac{d_{\min}^2}{P} = \min_E \frac{\underline{x}^T A(E) \underline{x}}{P} = 2^{\nu+k} \min_E \frac{\underline{x}^T A(E) \underline{x}}{\underline{x}^T \underline{x}}, \tag{21}$$

where we minimize over all error events $E$.

Although we will make no use of the fact in this work, we note that in (21) only a finite number of error events need be considered, for no error event need be considered that has a repeated pair of states. Thus, if the pair of states $u$ and $w$ occur at time $i$ and also at a later time $j$, all components between $i$ and $j$ may be eliminated and the remainder of the error event after time $j$ may be placed after time $i$. Since components cannot make a negative contribution to $d^2(E)$ the new error event has distance no greater than the original one. By (21) the best normalized minimum distance that can be achieved for any choice of channel symbols is

$$2^{\nu+k} \max_{\underline{x}} \min_E \frac{\underline{x}^T A(E) \underline{x}}{\underline{x}^T \underline{x}}. \tag{22}$$

Consider an error event $E$ with initial state (time $t = 0$) $a = (a_1, \cdots, a_{N-1})$ and final state $z = (z_1, \cdots, z_{N-1})$. If $k$ tuples $b_1, b_1^*$ are input at time 0, then at time 1 the two paths occupy states $(b_1, a_1, \cdots, a_{N-2})$ and $(b_1^*, a_1, \cdots, a_{N-2})$. There must be at least $N - 1$ further inputs before the paths can remerge. To remerge at $z$, the $k$ tuples $z_{N-1}, z_{N-2}, \cdots, z_1$ must be input in that order to both paths. We denote this error event by $E(a, z; b_1, b_1^*)$. Thus, the minimal length

of an error event is $N = (k + \nu)/k$. Fig. 2 shows the error event $E(00, 00; 0, 1)$ which has minimal length 3.

Given an arbitrary set $S$ of error events, define

$$Q(S) = \frac{1}{|S|} \sum_{E \in S} A(E). \tag{23}$$

Let $S^N$ be the set of all error events of length $N$. Note that $S^N$ is fixed by the group $\mathbf{G}^*_{k,\nu}$.

*Theorem 1: If $k$ divides $\nu$ then the normalized minimum distance of any sliding window trellis code with $2^\nu$ states and rate $k$ bits/channel symbol satisfies*

$$\frac{d^2_{\min}}{P} \leqslant 4\left(1 + \frac{\nu}{k}\right).$$

*Proof:* By (22),

$$\begin{aligned}
\frac{d^2_{\min}}{P} &\leqslant 2^{\nu+k} \max_{\underline{x}} \min_{E} \frac{\underline{x}^T A(E) \underline{x}}{\underline{x}^T \underline{x}} \\
&\leqslant 2^{\nu+k} \max_{\underline{x}} \min_{E \in S^N} \frac{\underline{x}^T A(E) \underline{x}}{\underline{x}^T \underline{x}} \\
&\leqslant \frac{2^{\nu+k}}{|S^N|} \max_{\underline{x}} \frac{\underline{x}^T \left( \sum_{E \in S^N} A(E) \right) \underline{x}}{\underline{x}^T \underline{x}}.
\end{aligned}$$

The last inequality simply states that the minimum is not more than the average. Setting $A_N = \sum_{E \in S^N} A(E)$, we have

$$\frac{2^{\nu+k}}{|S^N|} \max_{\underline{x}} \frac{\underline{x}^T A_N \underline{x}}{\underline{x}^T \underline{x}} = \frac{2^{\nu+k}}{|S^N|} \lambda_1(A_N),$$

where $\lambda_1(A_N)$ denotes the largest eigenvalue of $A_N$. By Property I, the $i$th diagonal entry of $A_N$ counts the total number of times the edge $x_i$ appears in some component of the error events of length $N$. By Lemma 3 all diagonal entries are equal to $2N|S^N|/2^{\nu+k}$. Property II implies that all row sums of $A_N$ are zero. By the Gersgorin Circle Theorem[5]

$$\lambda_1(A_N) \leqslant 2(\text{diagonal entry}) = 2\left(\frac{2N|S^N|}{2^{\nu+k}}\right),$$

and so

$$\frac{d^2_{\min}}{P} \leqslant 4N = 4\left(1 + \frac{\nu}{k}\right). \quad \square$$

*Remarks:* In Section IV we derive a formula for $Q(S^N)$, and, by computing $\lambda_1(A_N)$, we prove

$$\frac{d_{\min}^2}{P} \leqslant \frac{2^{k+1}}{2^k - 1}\left(1 + \frac{\nu}{k}\right).$$

In Appendix B we prove that if $\nu = (N-1)k + l$, where $0 \leqslant l < k$, then

$$\frac{d_{\min}^2}{P} \leqslant 4\left(1 + \left[\frac{\nu}{k}\right]\right),$$

where $\lfloor y \rfloor$ denotes the integer part of $y$.

## IV. A FORMULA FOR $Q(S^N)$ AND A SHARPER BOUND

In this section we derive a formula for $Q(S^N)$, the matrix obtained by averaging the distance matrices of all error events of minimal length $N = (k + \nu)/k$. We require a matrix representation of the group $\mathbf{G}_{k,\nu}$.

If $A$ is an $m \times n$ matrix and $B$ is an $m_1 \times n_1$ matrix, then the tensor product $A \otimes B$ (also called the Kronecker product) is the $mm_1 \times nn_1$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots\cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots\cdots & a_{2n}B \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m1}B & a_{m2}B & \cdots\cdots & a_{mn}B \end{bmatrix}.$$

Tensor products are discussed in Ref. 5, where they are called direct products. For appropriately sized matrices, $A$, $B$, $C$, and $D$, we have $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$. If $\lambda$ is an eigenvalue of $A$ with associated eigenvector $v$, and $\mu$ is an eigenvalue of $B$ with associated eigenvector $w$, then $\lambda\mu$ is an eigenvalue of $A \otimes B$ with eigenvector $v \otimes w$.

We denote the $n \times n$ identity matrix by $I_n$ and we abbreviate $I_2$ to $I$. Set

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{24}$$

Consider the $2^{\nu+k} \times 2^{\nu+k}$ matrix

$$P_i = \underbrace{I \otimes \cdots \otimes I}_{j \text{ terms}} \otimes A \otimes \underbrace{I \otimes \cdots \otimes I}_{i \text{ terms}}$$

$$= I_{2^j} \otimes A \otimes I_{2^i}, \tag{25}$$

where $i + j + 1 = \nu + k$. This is the matrix

$$P_i = \begin{bmatrix} \begin{matrix} 0 & I_{2^i} \\ I_{2^i} & 0 \end{matrix} & & & \\ & \begin{matrix} 0 & I_{2^i} \\ I_{2^i} & 0 \end{matrix} & & 0 \\ & & \ddots & \\ & & & \begin{matrix} 0 & I_{2^i} \\ I_{2^i} & 0 \end{matrix} \\ 0 & & & \end{bmatrix}$$

with the indicated block repeated $2^j$ times along the main diagonal. Define $u_i$, $i = 0, 1, \cdots, 2^{\nu+k} - 1$, to be the binary $\nu + k$ tuple with a 1 in position $i$ and 0's elsewhere. Let

$$\underline{x} = (x_0, \cdots, x_{2^{\nu+k}-1})^T = (x(0 \cdots 0), \cdots, x(1 \cdots 1))^T.$$

The permutation $g_{u_i}$ maps $x(s)$ to $x(u_i + s)$ and so it interchanges edges with subscripts differing by $2^i$. But this is precisely the effect of the transformation $\underline{x} \to P_i \underline{x}$. If $t$ is an arbitrary $\nu + k$ tuple then the matrix describing the permutation $g_t$ is obtained by multiplying the appropriate matrices $P_i$. For $t = (t_0, t_1, \cdots, t_{\nu+k-1})$ we define

$$M(t) = M_{\nu+k-1} \otimes \cdots \otimes M_1 \otimes M_0, \tag{26}$$

where

$$M_j = \begin{cases} I & \text{if } t_j = 0 \\ A & \text{if } t_j = 1. \end{cases} \tag{27}$$

Note that the subscript order in (26) is the reverse of the subscript order in the vector $t$. We have now proved the following lemma.

*Lemma 4: If $t$ is a $\nu + k$ tuple, then the permutation $g_t$: $x(s) \to x(s + t)$ is represented by $\underline{x} \to M(t)\underline{x}$.*

As an example, the permutation $g_{010}$ given in (12) is represented by the matrix $P = I \otimes A \otimes I$ given in (14). By Lemma 4 we may regard $\mathbf{G}_{k,\nu}$ as the following group of matrices:

$$\mathbf{G}_{k,\nu} = \{M_{\nu+k-1} \otimes \cdots \otimes M_1 \otimes M_0 \mid M_j = I \text{ or } A,$$

$$j = 0, \cdots, \nu + k - 1\}. \tag{28}$$

We shall prove that $Q(S^N)$ is a particular linear combination of

matrices $M(t)$ in $\mathbf{G}_{k,\nu}$. To calculate $\lambda_1(Q(S^N))$ we need to work with eigenvectors and eigenvalues of the matrices $M(t)$.

The matrices $M(t)$ are symmetric and they all commute; hence, they can be simultaneously diagonalized. Let $H$ be the tensor product of $\nu + k$ copies of

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}.$$

Observe that $H^{-1} = H^T$. Since $(1, 1)^T$ and $(1, -1)^T$ are eigenvectors of $A$, the columns of $H$ are eigenvectors of $M(t)$ for all $\nu + k$ tuples $t$. Thus, $H^TM(t)H$ is diagonal for every matrix $M(t)$ in $\mathbf{G}_{k,\nu}$. If $p = (p_0, \cdots, p_{\nu+k-1})$ is a binary $\nu + k$ tuple, define

$$w(p) = w_{\nu+k-1} \otimes \cdots \otimes w_1 \otimes w_0,$$

where

$$w_j = \begin{cases}(1, 1)^T, & \text{if } p_j = 0\\ (1, -1)^T, & \text{if } p_j = 1.\end{cases} \tag{29}$$

The vectors $w(p)$ are the columns of $H$. Note that $w(p)$ is formed by *reversing* the vector $p$. We have $A(1, 1)^T = (1, 1)^T$ and $A(1, -1)^T = -(1, -1)^T$. If $t = (t_0, \cdots, t_{\nu+k-1})$ then by (27) and (29)

$$M(t)w(p) = \bigotimes_{j=0}^{\nu+k-1} M_j w_j = \left(\prod_{j=0}^{\nu+k-1} (-1)^{t_j p_j}\right) w(p)$$

$$= (-1)^{p\cdot t} w(p), \tag{30}$$

where $p \cdot t$ is the dot product of the vectors $p$ and $t$.

*Lemma 5: Suppose $R$ is a diagonable matrix that commutes with every matrix $M(t)$ in $\mathbf{G}_{k,\nu}$. Then $R$ is a linear combination of the matrices $M(t)$ in $\mathbf{G}_{k,\nu}$.*

*Proof:* If $s$, $t$ are different $\nu + k$ tuples, then by Lemma 1, $g_s(x_0) \neq g_t(x_0)$. The permutation matrices $M(t)$ are therefore linearly independent because the 1's in row 0 are in different positions. Thus we have $2^{\nu+k}$ linearly independent diagonal matrices $H^{-1}M(t)H$. Since $R$ commutes with every matrix $M(t)$, $H^{-1}RH$ commutes with every matrix $H^{-1}M(t)H$, and therefore $H^{-1}RH$ is diagonal. The matrices $H^{-1}M(t)H$ span the set of diagonal matrices so $H^{-1}RH$ is a linear combination of matrices $H^{-1}M(t)H$ and the lemma follows. □

*Lemma 6: If $S$ is a set of error events fixed by $\mathbf{G}^*_{k,\nu}$ then $(\sum_{E\in S} A(E))$ is a linear combination of the matrices $M(t)$ in $\mathbf{G}_{k,\nu}$.*

*Proof:* The distance matrix $A(E)$ of an error event is the sum of contributions from each component:

$$A(E) = \sum_c A_c(E), \tag{31}$$

where we sum over the components $c$ of $E$. The restriction of $\mathbf{G}_{k,\nu}^*$ to the edges in any component $c$ is just the group $\mathbf{G}_{k,\nu}$. If edges $x(s)$, $x(\hat{s})$ appear in component $c$ of error event $E$, then edges $g_t(x(s))$, $g_t(x(\hat{s}))$ appear in component $c$ of error event $E' = g_t(E)$. We have

$$A_c(E') = M(t)^T A_c(E) M(t). \tag{32}$$

Since $M(t)$ is a permutation matrix and $M(t)^2 = I$, we have $M(t)^T = M(t)^{-1}$. Now $g_t$ merely permutes the error events in $S$, so that by (32),

$$\sum_{E \in S} A_c(E) = \sum_{E \in S} A_c(g_t(E)) = \sum_{E \in S} M(t)^{-1} A_c(E) M(t)$$

$$= M(t)^{-1} \left( \sum_{E \in S} A_c(E) \right) M(t) \tag{33}$$

for all matrices $M(t)$ and for all components $c$. Summing (33) over all components $c$ finishes the proof. □

*Example:* If $S$ is the orbit of error events shown in Fig. 6 then

$$\sum_{E \in S} A(E) = \left[\begin{array}{cc|cc|cc|cc} 3 & -1 & -1 & & -1 & & & \\ -1 & 3 & & -1 & & -1 & & \\ \hline -1 & & 3 & -1 & & & -1 & \\ & -1 & -1 & 3 & & & & -1 \\ \hline -1 & & & & 3 & -1 & -1 & \\ & -1 & & & -1 & 3 & & -1 \\ \hline & & -1 & & -1 & & 3 & -1 \\ & & & -1 & & -1 & -1 & 3 \end{array}\right]$$

$$= 3I \otimes I \otimes I - (I \otimes I \otimes A + I \otimes A \otimes I + A \otimes I \otimes I). \tag{34}$$

Consider $S^N$, the set of all error events $E(a, z; b_1, b_1^*)$ of minimal length $N = (k + \nu)/k$. Recall that $a = (a_1, \cdots, a_{N-1})$ is the initial state, $z = (z_1, \cdots, z_{N-1})$ is the final state, and $b_1, b_1^*$ are the first pair of inputs. We have $|S^N| = \binom{2^k}{2} 2^\nu \cdot 2^\nu$.

*Lemma 7:*

(1) *Let* $t = (t_0, \cdots, t_{N-1})$ *and let* $t' = (t_1, \cdots, t_{N-1})$ *where* $t_i$, $i = 0, 1, \cdots, N-1$, *is a binary* $k$ *tuple. If* $g_t^* = (g_{t^0}, g_{t^1}, \cdots, g_{t^{N-1}}) \in \mathbf{G}_{k,\nu}^*$, *then*

$$g_t^*(E(a, z; b_1, b_1^*)) = E(a + t', z + t', b_1 + t_0, b_1^* + t_0). \tag{35}$$

(2) *The group* $\mathbf{G}_{k,\nu}^*$ *partitions the set* $S^N$ *of error events of length* $N$ *into* $2^\nu(2^k - 1)$ *orbits each of size* $2^{\nu+k-1}$.

*Proof:* Part (1) follows from the definition of $g_{t^i}$ given in (17). To verify part (2) we note that $E(a, z; b_1, b_1^*)$ is fixed only by the symmetry $g_b^*$, where $b = (b_1 + b_1^*, 0, 0, \cdots, 0)$. Hence, each orbit consists of

$2^{\nu+k-1}$ distinct error events. Since the total number of error events in $S^N$ is $(2^\nu \cdot 2^\nu(2^k)(2^k-1))/2$, we see that there are $2^\nu(2^k-1)$ orbits. □

The orbit containing the error event $E(a, z; b_1, b_1^*)$ is determined by $a + z$ and $b_1 + b_1^*$. Setting $f = b_1 + b_1^*$, we denote this orbit by $S(a+z; f)$. This orbit contains $E(a, z; 0, f)$; note that $f \neq 0$ because $b_1 \neq b_1^*$. Recall that if $f$ is a $k$ tuple, then the $\nu + k$ tuple $(f0 \cdots 0)^i$ equals $(y_0, y_1, \cdots, y_{N-1})$ where $y_i = f$ and $y_j = 0$ for $j \neq i$.

*Lemma 8: Let $S^N$ be the set of all error events of length $N$ and let $S(a+z; f)$ be the orbit of $\mathbf{G}_{k,\nu}^*$ containing the error event $E(a, z; 0, f)$. Then*

$$(1)\quad 2^{\nu+k}Q(S(a+z; f)) = 2NI_{2^{\nu+k}} - 2\sum_{i=0}^{N-1} M((f0 \cdots 0)^i) \tag{36}$$

$$(2)\quad 2^{\nu+k}(2^k-1)Q(S^N) = 2(2^k-1)NI_{2^{\nu+k}} - 2\sum_{f\neq 0}\sum_{i=0}^{N-1} M((f0 \cdots 0)^i). \tag{37}$$

*Proof:* We calculate the contribution to $Q(S(a+z; f))$ made by pairs of edges in component 0. Since the restriction of $\mathbf{G}_{k,\nu}^*$ to the edges in any component is just the group $\mathbf{G}_{k,\nu}$, this distance contribution is

$$\begin{aligned}
&\frac{1}{2^{\nu+k}}\sum_t [g_t(x(0a_1 \cdots a_{N-1})) - g_t(x(fa_1 \cdots a_{N-1}))]^2 \\
&= \frac{1}{2^{\nu+k}}\sum_t [x(t + (0a_1 \cdots a_{N-1})) - x(t + (fa_1 \cdots a_{N-1}))]^2 \\
&= \frac{1}{2^{\nu+k}}\left[2\sum_t x(t)^2 - 2\sum_t x(t)x(t + (f0 \cdots 0))\right] \\
&= \frac{1}{2^{\nu+k}}\,\underline{x}^T[2I_{2^{\nu+k}} - 2M(f0 \cdots 0)]\underline{x}.
\end{aligned}$$

In general, the distance contribution made by edges in component $i$ is

$$\begin{aligned}
&\frac{1}{2^{\nu+k}}\sum_t [g_t(x(z_{N-i} \cdots z_{N-1}0a_1 \cdots a_{N-i-1})) \\
&\qquad - g_t(x(z_{N-i} \cdots z_{N-1}fa_1 \cdots a_{N-i-1}))]^2 \\
&= \frac{1}{2^{\nu+k}}\sum_t [x(t + (z_{N-i} \cdots z_{N-1}0a_1 \cdots a_{N-i-1})) \\
&\qquad - x(t + (z_{N-i} \cdots z_{N-1}fa_1 \cdots a_{N-i-1}))]^2 \\
&= \frac{1}{2^{\nu+k}}\,\underline{x}^T[2I_{2^{\nu+k}} - 2M((f0 \cdots 0)^i)]\underline{x}.
\end{aligned} \tag{38}$$

Summing (38) over all components $i$, we obtain (36). Since (36) is independent of $a + z$, we obtain the formula for $Q(S^N)$ by summing (36) over all nonzero $k$ tuples $f$. □

*Remark:* When $k = 1$, there is only one choice for $f$, namely $f = 1$, and so every form $Q(S(a + z; f))$ is equal to $Q(S^N)$. For $k = 1$, $\nu = 2$, we have

$$Q(S(00; 1)) = Q(S(10; 1)) = Q(S(01; 1)) = Q(S(11; 1)) = Q(S^3)$$
$$= 1/4[3I_8 - (I \otimes I \otimes A + I \otimes A \otimes I + A \otimes I \otimes I)]$$

[see the matrix given as (34)]. However, for $k > 1$, the form $Q(S(a + z; f))$ will change with $f$. Thus, for $k = 2$, $\nu = 4$, we have

$$Q(S(a + z; 11)) = 1/32[3I_{64} - (I_4 \otimes I_4 \otimes (A \otimes A)$$
$$+ I_4 \otimes (A \otimes A) \otimes I_4 + (A \otimes A) \otimes I_4 \otimes I_4)],$$

while

$$Q(S(a + z; 10)) = 1/32[3I_{64} - (I_4 \otimes I_4 \otimes (I \otimes A)$$
$$+ I_4 \otimes (I \otimes A) \otimes I_4 + (I \otimes A) \otimes I_4 \otimes I_4)].$$

*Theorem 2: If $k$ divides $\nu$, then the normalized minimum distance of any sliding window trellis code with $2^\nu$ states and rate $k$ bits/channel symbol satisfies*

$$\frac{d_{\min}^2}{P} \leq \frac{2^{k+1}}{2^k - 1}\left(1 + \frac{\nu}{k}\right).$$

*Proof:* From the proof of Theorem 1, we have

$$\frac{d_{\min}^2}{P} \leq 2^{\nu+k}\lambda_1[Q(S^N)]$$
$$= \frac{1}{2^k - 1}\lambda_1(Q_N), \tag{39}$$

where $Q_N = (2^k - 1)2^{\nu+k}Q(S^N)$. Let $c = (c_0, \cdots, c_{N-1})$ be a binary $\nu + k$ tuple and let $\gamma$ be the number of nonzero $k$ tuples $c_i$. Then by (30), the eigenvalue of $Q_N$ associated with $w(c)$ is

$$2(2^k - 1)N - 2\sum_{f \neq 0}\sum_{i=0}^{N-1}(-1)^{c_i \cdot f}$$
$$= 2(2^k - 1)N - 2\sum_{i=0}^{N-1}\sum_{f \neq 0}(-1)^{c_i \cdot f}, \tag{40}$$

where we sum over all nonzero $k$ tuples $f$. Since

$$\sum_{f \neq 0} (-1)^{c_i \cdot f} = \begin{cases} 2^k - 1, & \text{if } c_i = 0 \\ -1, & \text{if } c_i \neq 0, \end{cases}$$

eq. (40) becomes

$$2(2^k - 1)N - 2[(2^k - 1)(N - \gamma) - \gamma] = 2^{k+1}\gamma. \tag{41}$$

The largest eigenvalue of $Q_N$ is obtained when $\gamma = N = (1 + (\nu/k))$. The theorem now follows from (39). □

*Remarks:* Observe that the largest eigenvalue of $Q(S^N)$ is associated with $w(c)$, where $c_0, c_1, \cdots, c_{N-1}$ are all nonzero. For example, with $k = 1$, the largest eigenvalue, $4(1 + (\nu/k))$ has multiplicity one and is associated with the eigenvector $(1, -1)^T \otimes (1, -1)^T \otimes \cdots \otimes (1, -1)^T$. When $k > 1$, there will be several linearly independent eigenvectors associated with $\lambda_1(Q(S^N))$ because there are several choices for $c$ with all $c_i \neq 0$. Also, note that Theorem 2 gives the same bound as Theorem 1 when $k = 1$. For $k \geq 2$, the bound of Theorem 2 is an improvement.

In Appendix B we prove that if $\nu = (N - 1)k + l$ where $0 \leq l < k$, then

$$\frac{d_{\min}^2}{P} \leq \frac{2^{k-l+1}}{2^{k-l} - 1}\left(1 + \left\lfloor \frac{\nu}{k} \right\rfloor\right),$$

where $\lfloor y \rfloor$ denotes the integer part of $y$.

## V. A FINAL BOUND OBTAINED FROM A WEIGHTED AVERAGE

Let $S^{N+1}$ be the set of all error events of length $N + 1 = 2 + (\nu/k)$. Let $Q(S^{N+1})$ be the matrix obtained by averaging the distance matrices of all error events of length $N + 1$. In this section we derive a formula for $Q(S^{N+1})$ and we prove

$$\frac{d_{\min}^2}{P} \leq \frac{2^{2k+1}}{2^{2k} - 1}\left(2 + \frac{\nu}{k}\right)$$

using a weighted average of $Q(S^N)$ and $Q(S^{N+1})$.

An error event $E$ of length $N + 1$ is determined by the initial state $a = (a_1, \cdots, a_{N-1})$, the final state $z = (z_1, \cdots, z_{N-1})$, the inputs $b_1$, $b_1^*$ at time 0, and the inputs $b_2$, $b_2^*$ at time 1. Since the two paths diverge at time 0, we must have $b_1 \neq b_1^*$. To remerge at $z$ the last $N - 1$ inputs must be the $k$ tuples $z_{N-1}, z_{N-2}, \cdots, z_1$ in that order. After $N$ inputs the two paths occupy states $z_2 \cdots z_{N-1}b_2$ and $z_2 \cdots z_{N-1}b_2^*$. At this stage the two paths must be disjoint so $b_2 \neq b_2^*$. We denote this error event $E$ by $E(a, z; b_1, b_1^*; b_2, b_2^*)$ [equivalently $E(a, z; b_1^*, b_1; b_2^*, b_2)$].

The group $\mathbf{G}_{k,\nu}^*$ maps error events of length $N + 1$ to error events of

length $N+1$. To be specific, let $t=(t_0, \cdots, t_{N-1})$ be a $\nu+k$ tuple and set $t'=(t_1, \cdots, t_{N-1})$, $t''=(t_0, \cdots, t_{N-2})$. If $g_t^*=(g_t, g_{t^1}, \cdots, g_{t^{N-1}}, g_t)$ then it follows from the definition of $g_{t^i}$ given in (7) that

$$g_t^*(E(a, z; b_1, b_1^*; b_2, b_2^*))$$
$$= E(a+t', z+t''; b_1+t_0, b_1^*+t_0; b_2+t_{N-1}, b_2^*+t_{N-1}). \quad (42)$$

This group action does not preserve $a+z$ but it does preserve $b_1+b_1^*$ and $b_2+b_2^*$. Set $g=b_1+b_1^*$ and $f=b_2+b_2^*$. We denote the orbit of $\mathbf{G}_{k,\nu}^*$ containing the error event $E(a', z'; 0, g; 0, f)$ by $S(a', z'; g, f)$ (in the discussion above, $a'=(a_1, \cdots, a_{N-2}, a_{N-1}+b_2)$ and $z'=(z_1+b_1, z_2, \cdots, z_{N-1})$). Note that $f, g \neq 0$.

If $f, g$ are $k$ tuples, then the $\nu+k$ tuple $(fg0 \cdots 0)^0=(fg0 \cdots 0)$ and $(fg0 \cdots 0)^i$ is obtained from $(fg0 \cdots 0)^{i-1}$ by cycling the blocks of $k$ bits to the right and moving the last block to the front. Thus $(fg0 \cdots 0)^{N-2}=(0 \cdots 0fg)$. Define matrices $M_i(fg0 \cdots 0)$, $i=0, \cdots, N$ in $\mathbf{G}_{k,\nu}$ as follows:

$$M_0(fg0 \cdots 0)=M(g0 \cdots 0)$$
$$M_i(fg0 \cdots 0)=M((fg0 \cdots 0)^{i-1}) \qquad i=1, \cdots, N-1,$$
$$M_N(fg0 \cdots 0)=M(0 \cdots 0f). \quad (43)$$

*Example:* For $k=1$, $\nu=2$, the orbit $S(00, 00; 1, 1)$ is shown in Fig. 7. The quadratic form $Q(S(00, 00; 1, 1))$ is given by

$$Q(S(00, 00; 1, 1))=\frac{1}{8}\left[\begin{array}{cc|cc|cc|cc}
8 & -2 & & -2 & -2 & & -2 & \\
-2 & 8 & -2 & & & -2 & & -2 \\
\hline
 & -2 & 8 & -2 & -2 & & -2 & \\
-2 & & -2 & 8 & & -2 & & -2 \\
\hline
-2 & & -2 & & 8 & -2 & & -2 \\
 & -2 & & -2 & -2 & 8 & -2 & \\
\hline
-2 & & -2 & & & -2 & 8 & -2 \\
 & -2 & & -2 & -2 & & -2 & 8
\end{array}\right]$$

$$= \frac{1}{8}(8I_8 - 2(I \otimes I \otimes A + I \otimes A \otimes A + A \otimes A \otimes I + A \otimes I \otimes I))$$
$$= \frac{1}{8}\left(8I_8 - 2\sum_{i=0}^{3} M_i(110)\right). \quad (44)$$

*Lemma 9: Let $S^{N+1}$ be the set of all error events of length $N+1$ and let $S(a, z; g, f)$ be the orbit of $\mathbf{G}_{k,\nu}^*$ containing the error event $E(a, z; 0, g; 0, f)$. Then,*

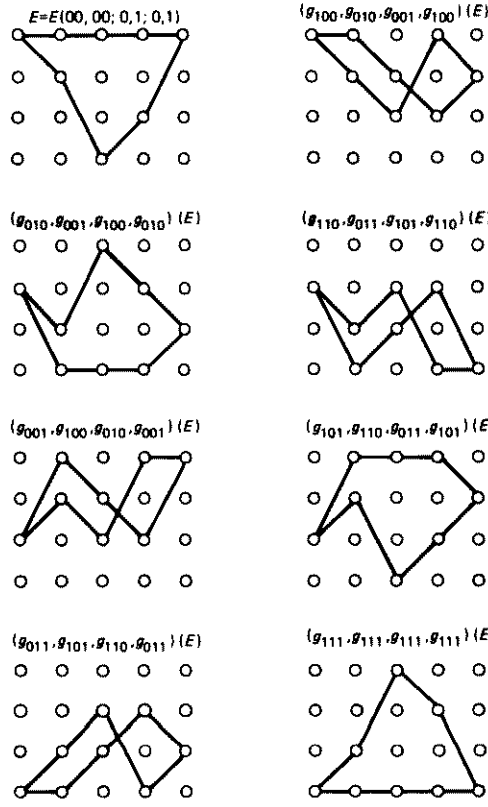



Fig. 7—The orbit $S(00, 00; 1, 1)$.

$$(1)\quad 2^{\nu+k}Q(S(a, z; g, f)) = 2(N+1)I_{2^{\nu+k}} - 2\sum_{i=0}^{N} M_i(fg0\ \cdots\ 0). \tag{45}$$

$$(2)\quad 2^{\nu+k}(2^k-1)^2Q(S^{N+1}) = 2(2^k-1)^2(N+1)I_{2^{\nu+k}} - 2\sum_{f,g\neq 0}\sum_{i=0}^{N} M_i(fg0\ \cdots\ 0). \tag{46}$$

*Proof:* We calculate the contribution to $Q(S(a, z; g, f))$ made by pairs of edges in component 0. This distance contribution is

$$\frac{1}{2^{\nu+k}}\sum_{t}[x(t+(0a_1\ \cdots\ a_{N-1})) - x(t+(ga_1\ \cdots\ a_{N-1}))]^2$$

$$= \frac{1}{2^{\nu+k}}\,\underline{x}^T[2I_{2^{\nu+k}} - 2M(g0\ \cdots\ 0)]\underline{x}$$

as found in the proof of Lemma 8. Similarly, the contribution made

by pairs of edges in component $N$ (the last component of the error events) is

$$\frac{1}{2^{\nu+k}} \sum_t [x(t + (z_1 \cdots z_{N-1}0)) - x(t + (z_1 \cdots z_{N-1}f))]^2$$

$$= \frac{1}{2^{\nu+k}} \underline{x}^T[2I_{2^{\nu+k}} - 2M(0 \cdots 0f)]\underline{x}.$$

For $i = 1, \cdots, N-1$, the contribution made by pairs of edges in component $i$ is

$$\frac{1}{2^{\nu+k}} \sum_t [x(t + (z_{N-i+1} \cdots z_{N-1}00a_1 \cdots a_{N-i-1}))$$
$$- x(t + (z_{N-i+1} \cdots z_{N-1}fga_1 \cdots a_{N-i-1}))]^2$$
$$= \frac{1}{2^{\nu+k}} \left[ 2 \sum_t x(t)^2 - 2 \sum_t x(t)x(t + (fg0 \cdots 0)^{i-1}) \right]$$
$$= \frac{1}{2^{\nu+k}} \underline{x}^T[2I_{2^{\nu+k}} - 2M((fg0 \cdots 0)^{i-1})]\underline{x}.$$

The sum of the contributions from all $N+1$ components is

$$Q(S(a, z; g, f)) = \frac{1}{2^{\nu+k}} \underline{x}^T \left[ 2(N+1)I_{2^{\nu+k}} - 2 \sum_{i=0}^{N} M_i(fg0 \cdots 0) \right] \underline{x}.$$

This proves part (1). Observe that (45) is independent of $a$ and $z$. We obtain $Q(S^{N+1})$ by summing (45) over all pairs $g, f$ of nonzero $k$ tuples. Since there are $(2^k - 1)^2$ such pairs,

$$(2^k - 1)^2 2^{\nu+k} Q(S^{N+1})$$
$$= 2(2^k - 1)^2(N+1)I_{2^{\nu+k}} - 2 \sum_{f,g \neq 0} \sum_{i=0}^{N} M_i(fg0 \cdots 0)$$

as required. □

*Remarks:* When $k = 1$, we must have $f = g = 1$ and so every form $Q(S(a, z; g, f))$ is equal to $Q(S^{N+1})$. In this case, $N = 1 + \nu$ and

$$Q(S^{N+1}) = \frac{1}{2^{\nu+1}} \left[ 2(2+\nu)I_{2^{\nu+1}} - 2 \sum_{i=0}^{1+\nu} M_i(110 \cdots 0) \right]$$

[see the matrix given as (44)]. For $k > 1$, there are several choices for $f$ and $g$. Thus, for $k = 2$, $\nu = 4$, we have, with $g = (1, 1)$ and $f = (0, 1)$,

$$Q(S(a, z; 11, 01)) = 1/64[8I_{64} - 2(I_4 \otimes I_4 \otimes (A \otimes A)$$
$$+ I_4 \otimes (A \otimes A) \otimes (A \otimes I)$$
$$+ (A \otimes A) \otimes (A \otimes I) \otimes I_4$$
$$+ (A \otimes I) \otimes I_4 \otimes I_4)],$$

while with $g = (01)$ and $f = (10)$ we get

$$\begin{aligned} Q(S(a, z; 01; 10)) = 1/64[8I_{64} &- 2(I_4 \otimes I_4 \otimes (A \otimes I) \\ &+ I_4 \otimes (A \otimes I) \otimes (I \otimes A) \\ &+ (A \otimes I) \otimes (I \otimes A) \otimes I_4 \\ &+ (I \otimes A) \otimes I_4 \otimes I_4)]. \end{aligned}$$

*Theorem 3: If $k$ divides $\nu$ then the normalized minimum distance of any convolutionally derived trellis code with $2^\nu$ states and rate $k$ bits/channel symbol satisfies*

$$\frac{d_{\min}^2}{P} \leq \frac{2^{2k+1}}{2^{2k} - 1}\left(2 + \frac{\nu}{k}\right).$$

*Proof:* If $\bar{Q}$ is any weighted average of $Q(S^N)$ and $Q(S^{N+1})$, then by (8) we have

$$\frac{d_{\min}^2}{P} \leq 2^{\nu+k}\lambda_1(\bar{Q}). \tag{47}$$

Let $\delta = 1/(2^{2k} - 1)$. Then $2(2^k - 1)\delta + (2^k - 1)^2\delta = 1$. Define $\bar{Q}$ to be the following weighted average of $Q(S^N)$ and $Q(S^{N+1})$:

$$\bar{Q} = 2(2^k - 1)\delta Q(S^N) + (2^k - 1)^2\delta Q(S^{N+1}).$$

Set

$$Q_N = 2^{\nu+k}(2^k - 1)Q(S^N)$$

and

$$Q_{N+1} = 2^{\nu+k}(2^k - 1)^2 Q(S^{N+1}).$$

Then by (47)

$$\frac{d_{\min}^2}{P} \leq \delta\lambda_1(2Q_N + Q_{N+1}). \tag{48}$$

The eigenvectors, $w(c)$, of $Q_N$ and $Q_{N+1}$ are in 1-1 correspondence with binary vectors $c = (c_1, \cdots, c_N)$, where $c_i$, $i = 1, \cdots, N$ are $k$ tuples. By (41)

$$Q_N w(c) = 2^{k+1}\gamma(c)w(c), \tag{49}$$

where $\gamma(c)$ is the number of nonzero $k$ tuples $c_i$. Introduce $k$ tuples $c_0 = c_{N+1} = 0$ and define

$$\alpha(c) = |\{i \,|\, c_i = 0, c_{i+1} \neq 0 \text{ or } c_i \neq 0, c_{i+1} = 0\}|$$

and

$$\beta(c) = |\{i \mid c_i \neq 0 \text{ and } c_{i+1} \neq 0\}|. \tag{50}$$

There are $(N+1) - \alpha(c) - \beta(c)$ indices $i$, $0 \leq i \leq N$, for which $c_i = c_{i+1} = 0$. By (30) the eigenvalue of $Q_{N+1}$ associated with $w(c)$ is

$$2(2^k - 1)^2(N+1) - 2 \sum_{f,g \neq 0} \sum_{i=0}^{N} (-1)^{c_i \cdot f + c_{i+1} \cdot g}$$

$$= 2(2^k - 1)^2(N+1) - 2 \sum_{i=0}^{N} \left( \sum_{f \neq 0} (-1)^{c_i \cdot f} \right) \left( \sum_{g \neq 0} (-1)^{c_{i+1} \cdot g} \right). \tag{51}$$

Recall that the sum $\sum_{f \neq 0} (-1)^{c_i \cdot f}$ is $(2^k - 1)$ when $c_i = 0$, but equal to $-1$ whenever $c_i \neq 0$. Hence (50) is equal to

$$\begin{aligned} &2(2^k - 1)^2(N+1) - 2[((N+1) - \alpha(c) - \beta(c))(2^k - 1)^2 \\ &\qquad - \alpha(c)(2^k - 1) + \beta(c)] \\ &= 2(2^k - 1)^2(\alpha(c) + \beta(c)) + 2(2^k - 1)\alpha(c) - 2\beta(c) \\ &= 2^{k+1}(2^k(\alpha(c) + \beta(c)) - 2(\alpha(c) + \beta(c)) + \alpha(c)) \\ &= 2^{k+1}[2^k(\alpha(c) + \beta(c)) - (\alpha(c) + 2\beta(c))]. \end{aligned} \tag{52}$$

Now, $\gamma(c)$ is the number of nonzero $c_i$'s. Since each nonzero $c_i$ appears in two pairs, $(c_{i-1}, c_i)$ and $(c_i, c_{i+1})$, we have

$$\alpha(c) + 2\beta(c) = 2\gamma(c). \tag{53}$$

Substitution in (52) shows that the eigenvalue in (51), of $Q_{N+1}$ associated with $w(c)$ is

$$2^{k+1}[2^k(2\gamma(c) - \beta(c)) - 2\gamma(c)]. \tag{54}$$

By (49) and (54) we have

$$\begin{aligned} &(2Q_N + Q_{N+1})w(c) \\ &\quad = 2^{k+1}(2\gamma(c) + 2^k(2\gamma(c) - \beta(c)) - 2\gamma(c))w(c) \\ &\quad = 2^{2k+1}(2\gamma(c) - \beta(c))w(c). \end{aligned} \tag{55}$$

There are $N - \gamma(c)$ indices $i$, $1 \leq i \leq N$, for which $c_i = 0$. Since every $c_j$, $1 \leq j \leq N$, appears in the two pairs $(c_{j-1}, c_j)$ and $(c_j, c_{j+1})$, there are at most $2 + 2(N - \gamma(c))$ indices $i$, $0 \leq i \leq N$, for which $c_i = 0$ or $c_{i+1} = 0$. Hence

$$\beta(c) \geq (N+1) - 2 - 2(N - \gamma(c)) = 2\gamma(c) - N - 1$$

and

$$2(\gamma(c)) - \beta(c) \leq N + 1. \tag{56}$$

Now (48), (55), and (56) imply

$$\frac{d_{\min}^2}{P} \leq \frac{2^{2k+1}}{2^{2k}-1}(N+1) = \frac{2^{2k+1}}{2^{2k}-1}\left(2+\frac{\nu}{k}\right). \quad \square$$

*Remarks:* Equality can hold in (56). If $N$ is odd, set $c_0 = c_2 = \cdots = c_{N+1} = 0$ and $c_1, c_3, \cdots, c_N \neq 0$. Then $\gamma(c) = (N+1)/2$ and $\beta(c) = 0$. (Observe that for $k = 1$, $\nu = 2$, the largest eigenvalue of the form $2Q_N + Q_{N+1}$ is associated with eigenvector $(1, -1)^T \otimes (1, 1)^T \otimes (1, -1)^T$.) If $N$ is even, set $c_0 = c_3 = c_5 = c_7 = \cdots = c_{N+1} = 0$ and $c_1, c_2, c_4, c_6, \cdots, c_N \neq 0$ to get $\gamma(c) = (N+2)/2$ and $\beta(c) = 1$. Setting

$$\frac{2^{k+1}}{2^k-1}\left(1+\frac{\nu}{k}\right) = \frac{2^{2k+1}}{2^{2k}-1}\left(2+\frac{\nu}{k}\right)$$

yields $\nu = k(2^k - 1)$. If $\nu < k(2^k - 1)$, then Theorem 2 gives the stronger bound; if $\nu > k(2^k - 1)$ then Theorem 3 gives the stronger bound. In particular, for $k = 1$, Theorem 3 gives a stronger bound for any $\nu > 1$.

The bound given by Theorem 3 is obtained from the largest eigenvalue of a particular weighted average of $Q(S^N)$ and $Q(S^{N+1})$. In Appendix A we use the duality theorem of linear programming to prove that no other weighted average of $Q(S^N)$ and $Q(S^{N+1})$ gives a stronger bound.

In Appendix B we prove that if $\nu = (N-1)k + l$, where $0 \leq l < k$, then

$$\frac{d_{\min}^2}{P} \leq \frac{2^{2(k-l)+1}}{2^{2(k-l)}-1}\left(2+\left\lfloor\frac{\nu}{k}\right\rfloor\right),$$

where $\lfloor y \rfloor$ denotes the integer part of $y$.

## VI. CONCLUSIONS

Three upper bounds on the normalized minimum distance, $(d_{\min}^2/P)$, have been given for trellis codes. The bound

$$\frac{d_{\min}^2}{P} \leq 4\left(1+\frac{\nu}{k}\right)$$

given in Theorem 1 is typical. This certainly provides nontrivial information. For example, is it possible to gain 10 dB in minimum distance using $2^6 = 64$ states at rate 1 bit/symbol? The answer is no. Theorem 1 bounds the gain at 8.4 dB; Theorem 3 bounds the gain at 7.3 dB. Nevertheless, there still remain the questions of how tight these bounds are and if they exhibit the "right" dependence on the

Table I—Possible gains at rate 1 bit/symbol

| $\nu$ | Lower Bound (Ungerboeck) | Upper Bounds | |
|---|---|---|---|
| | | Theorems 1 and 2 | Theorem 3 |
| 2 | 2.5 db | 4.7 db | 4.3 db |
| 3 | 3 | 6.0 | 5.2 |
| 4 | 3.4 | 7.0 | 6.0 |
| 5 | 4.2 | 7.8 | 6.7 |
| 6 | 4.5 | 8.4 | 7.3 |
| 7 | 5.1 | 9.0 | 7.8 |
| 8 | 5.3 | 9.5 | 8.2 |
| 9 | 5.6 | 10.0 | 8.6 |
| 10 | 5.8 | 10.4 | 9.0 |
| 11 | 6 | 10.7 | 9.4 |

parameters $\nu$ and $k$. For example, consider the normalized minimum distance for block codes of length $n$, having $2^{nk}$ code words ($k$ bits/symbol). In that case, known upper bounds behave, for large $n$, like $d^2/P \lesssim 2n/4^k$. Thus the linear dependence on $\nu$, a quantity analogous to block length, appears correct. However, the true dependence on $k$ may be different from our bound. Table I gives upper and lower bounds on the gain (in dB) that is possible at rate 1 bit/channel symbol. The lower bounds arise from codes constructed by Ungerboeck.[4]

Also minimum distance is by no means the complete story with regard to error rate. The heuristics leading to the claim that terms involving $d_{\min}$ would dominate an upper bound on the error rate make the assumption that the infinite series determining the upper bound converges. Even if a code with a good $d_{\min}$ were found, an upper bound on error rate should still be computed for that particular code. As an example of a catastrophe that may occur, consider the assignment of edge labels $\underline{x}^T = (1, -1, -1, 1, -1, 1, 1, -1)$ to the trellis of Fig. 1. One observes that a pair of edges leaving a node always contributes $(1 - (-1))^2 = 4$ to the distance and similarly for a pair of edges merging into a node. One immediately concludes that no error event has distance less than 8 for this edge assignment. Since $P = 1$, this is a 3 dB gain over the uncoded $\pm 1$ situation. How could this happen with only $\pm 1$ symbols? One answer is that we forgot to include unmerged events, events which go on forever. We had implicitly assigned infinity to their distance, but now some have distance 4. However, this could be rectified by perturbing the $\pm 1$ edge labels by small amounts. A more serious trouble with this code is that an infinite number of error events have (essentially) the minimum distance and so a coefficient that we did not explicitly consider turns out to be infinite for this particular code.

## REFERENCES

1. A. D. Wyner, "On Coding and Information Theory," SIAM Review, *11*, No. 3 (July 1969), pp. 317–46.
2. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting-Codes*, Amsterdam: North-Holland and New York: Elsevier/North-Holland, 1977.
3. G. D. Forney, "The Viterbi Algorithm," Proc. IEEE, *61*, No. 3 (March 1973), pp. 268–78.
4. G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," IEEE Trans. Inform. Theory, *IT-28*, No. 1 (January, 1982), pp. 55–67.
5. Peter Lancaster, *Theory of Matrices*, New York: Academic Press, 1969.


## APPENDIX A

The upper bound of Theorem 3 is obtained from the largest eigenvalue of a particular weighted average of the quadratic forms $Q(S^N)$ and $Q(S^{N+1})$. In this appendix we prove that no other weighted average gives a stronger bound. We shall assume throughout that $\nu \geq k(2^k - 1)$ since the bound given in Theorem 3 improves upon that given in Theorem 2 only for $\nu$ in this range.

If $r_1, r_2 \geq 0$ and $r_1 + r_2 = 1$, then

$$\frac{d_{\min}^2}{P} \leq 2^{\nu+k}\lambda_1(r_1 Q(S^N) + r_2 Q(S^{N+1})).$$

Recall from (29) that the eigenvectors $w(c)$ of $Q(S^N)$ and $Q(S^{N+1})$ are 1:1 correspondence with binary $\nu + k$ tuples $c$. Let $c = (c_1, \cdots, c_N)$, where $c_i$, $i = 1, \cdots, N$ is a binary $k$ tuple and let $c_0 = c_{N+1} = 0$. Recall that

$$\alpha(c) = |\{i | c_i = 0,\ c_{i+1} \neq 0 \text{ or } c_i \neq 0,\ c_{i+1} = 0\}|,$$

$$\beta(c) = |\{i | c_i \neq 0 \text{ and } c_{i+1} \neq 0\},$$

and

$$\gamma(c) = |\{i | c_i \neq 0\}|.$$

Define $\phi_N(c)$ and $\phi_{N+1}(c)$ by $2^{\nu+k}(2^k - 1)Q(S^N)w(c) = \phi_N(c)w(c)$ and $2^{\nu+k}(2^k - 1)^2 Q(S^{N+1})w(c) = \phi_{N+1}(c)w(c)$. Then by (49) and (54)

$$\phi_N(c) = 2^{k+1}\gamma(c) \tag{57}$$

and

$$\phi_{N+1}(c) = 2^{k+1}[2^k(2\gamma(c) - \beta(c)) - 2\gamma(c)]. \tag{58}$$

To find the optimal weighted average we have to solve the following linear programming problem.

Choose real variables $r_1, r_2, r \geq 0$ so as to minimize $r$ subject to the inequalities

$$-(r_1 + r_2) \leq -1 \tag{59}$$

and

$$r_1 \frac{\phi_N(c)}{2^k - 1} + r_2 \frac{\phi_{N+1}(c)}{(2^k - 1)^2} - r \leqslant 0, \qquad \text{for all} \quad \nu + k \text{ tuples } c.$$

In Theorem 3 we proved that a feasible solution to (43) is

$$r_1 = \frac{2}{2^k + 1}, \qquad r_2 = \frac{2^k - 1}{2^k + 1}, \qquad r = \frac{2^{2k+1}}{2^{2k} - 1}\left(2 + \frac{\nu}{k}\right). \tag{60}$$

The linear program (59) is the dual of the primal linear program given below.

Choose real variables $a_c$, $a \geqslant 0$, where the index $c$ runs through all binary $\nu + k$ tuples, so as to maximize $a$ subject to the inequalities

$$\frac{1}{2^k - 1}\left(\sum_c \phi_N(c) a_c\right) - a \geqslant 0$$

$$\frac{1}{(2^k - 1)^2}\left(\sum_c \phi_{N+1}(c) a_c\right) - a \geqslant 0$$

$$-\left(\sum_c a_c\right) \geqslant -1. \tag{61}$$

If we can find a feasible solution to (61) with

$$a = \frac{2^{2k+1}}{2^{2k} - 1}\left(2 + \frac{\nu}{k}\right),$$

then by the duality theorem of linear programming,[2] (60) is an optimal solution to (59). We consider two cases.

***Case 1. N odd***

Pick $f = (f_1, \cdots, f_N)$, where $f_i$, $i = 1, \cdots, N$ is a binary $k$ tuple and every $f_i$ is nonzero. Pick $g = (g_1, \cdots, g_N)$, where $g_i$, $i = 1, \cdots, N$ is a binary $k$ tuple and $g_i \neq 0$ if and only if $i$ is odd. Then $\gamma(f) = N$, $\beta(f) = N - 1$ and $\gamma(g) = (N + 1)/2$, $\beta(g) = 0$. By (57) and (58), $\phi_N(f)$, $\phi_N(g)$, $\phi_{N+1}(f)$, and $\phi_{N+1}(g)$ are as follows:

| | $f$ | $g$ |
|---|---|---|
| $\phi_N$ | $2^{k+1}N$ | $2^k(N + 1)$ |
| $\phi_{N+1}$ | $2^{k+1}[2^k(N + 1) - 2N]$ | $2^{k+1}[(2^k - 1)(N + 1)]$ |

Set

$$a_c = \begin{cases} \left(\frac{2^k - 1}{2^k + 1}\right)\left(\frac{N + 1}{N - 1}\right), & \text{if } c = f \\ 1 - a_f = \frac{2(N - 2^k)}{(2^k + 1)(N - 1)}, & \text{if } c = g \\ 0, & \text{otherwise,} \end{cases}$$

$$a = \frac{2^{2k+1}}{2^{2k} - 1}\left(2 + \frac{\nu}{k}\right). \tag{62}$$

Direct calculation shows that (62) is a feasible solution to (61). (Since $\nu \geq k(2^k - 1)$, the variables $a_c$ are all nonnegative.)

***Case 2. N even***

Pick $h = (h_1, \cdots, h_N)$, where $h_i$, $i = 1, \cdots, N$, is a binary $k$ tuple, $h_2 = h_5 = h_7 = h_9 = \cdots = h_{N-1} = 0$, and $h_1, h_3, h_4, h_6, h_8, \cdots, h_N$ are nonzero. Then $\gamma(h) = (N + 2)/2$, $\beta(h) = 1$ and, by (57) and (58),

$$\phi_N(h) = 2^k(N + 2) \quad \text{and} \quad \phi_{N+1}(h) = 2^{k+1}[(2^k - 1)(N + 2) - 2^k].$$

Set

$$a_c = \begin{cases} \frac{(2^k - 1)N - 2}{(2^k + 1)(N - 2)}, & \text{if } c = f \\ 1 - a_f = \frac{2(N - 2^k)}{(2^k + 1)(N - 2)}, & \text{if } c = g \\ 0, & \text{otherwise,} \end{cases}$$

$$a = \frac{2^{2k+1}}{2^{2k} - 1}\left(2 + \frac{\nu}{k}\right). \tag{63}$$

Direct calculation shows that (63) is a feasible solution to (61). (Again since $\nu \geq k(2^k - 1)$, the variables $a_c$ are all nonnegative.)

We have now shown that (60) is an optimal solution to (59).

## APPENDIX B

In this appendix we extend Theorems 1, 2, and 3 to the case when $k$ does not divide $\nu$. Setting $\nu = (N - 1)k + l$, where $0 \leq l < k$, we have $N = \lfloor(\nu + k)/k\rfloor$ where $\lfloor y \rfloor$ denotes the integer part of $y$.

Encoder states are labelled with binary $\nu$ tuples in the way described in Section II. Edges of the trellis are labelled with real numbers $x(s)$, where $s$ is a binary $\nu + k$ tuple. The group $\mathbf{G}_{k,\nu}$ is defined in the way

described in Section II; for each binary $\nu + k$ tuple $t$, we define a permutation of the edge labels $x(s)$ by the rule

$$g_t(x(s)) = x(s + t).$$

The symmetry $g_t^*$ is the sequence

$$g_t^* = (g_{t^0}, g_{t^1}, g_{t^2}, \cdots),$$

where $t^0 = t$ and $t^i$ is obtained from $t^{i-1}$ by cycling the entries $k$ bits to the right and moving the last $k$ bits to the front. When $k = 2$ and $\nu = 3$,

$$g_{11001}^* = (g_{11001}, g_{01110}, g_{10011}, g_{11100}, g_{00111}, g_{11001}, \cdots).$$

In general, $t^i = t^{d+i}$ where $d = (k + \nu)/gcd(k, k + \nu)$. Given any component of the trellis and any pair of edges $x(s)$, $x(t)$ in that component there is a unique element of $\mathbf{G}_{k,\nu}^*$ interchanging $x(s)$ and $x(t)$. The proof of Theorem 1 goes through without change and we have

$$\frac{d_{\min}^2}{P} \leqslant 4N_0, \tag{64}$$

where $N_0$ is the minimal length of an error event.

To see that $N_0 = N$, consider an error event $E$ with initial state (time $t = 0$) $a = (a_1 \cdots a_{N-1}a_N)$, where $a_1, \cdots, a_{N-1}$, are $k$ tuples and $a_N$ is an $l$ tuple. If $k$ tuples $b_1, b_1^*$ are input at time 0 then at time 1 the two paths occupy states $(b_1, a_1, \cdots, a_{N-2}, \tilde{a}_{N-1})$ and $(b_1^*, a_1, \cdots, a_{N-1}, \tilde{a}_{N-1})$, where $\tilde{s}$ denotes the $l$ tuple $(s_1 \cdots s_l)$ obtained from the $k$ tuple $(s_1 \cdots s_k)$ by deleting the last $k - l$ bits. At time 1 the $k$-tuple $z_Nc$ is input to both paths, where $c$ is a fixed but arbitrary $k - l$ tuple. At time 2 the two paths occupy states $(z_Nc, b_1, a_1, \cdots, a_{N-3}, \tilde{a}_{N-2})$ and $(z_Nc, b_1^*, a_1, \cdots, a_{N-3}, \tilde{a}_{N-2})$. At time $N$, after inputs $z_{N-1}, \cdots, z_2$, the two paths occupy states $(z_2, z_3, \cdots, z_{N-1}, z_Nc, \tilde{b}_1)$ and $(z_2, z_3, \cdots, z_{N-1}, z_Nc, \tilde{b}_1^*)$. If $\tilde{b}_1 = \tilde{b}_1^*$ then the two paths remerge at time $N$ in state $z = (z_2, z_3, \cdots, z_{N-1}, z_Nc, \tilde{b}_1)$. We denote this error event by $E(a, z; b_1, b_1^*)$. Thus by (64)

$$\frac{d_{\min}^2}{P} \leqslant 4\left(1 + \left[\frac{\nu}{k}\right]\right) \tag{65}$$

for general $k$ and $\nu$.

Let $S(a, z; b_1, b_1^*)$ be the orbit of $\mathbf{G}_{k,\nu}^*$ containing the error event $E(a, z; b_1, b_1^*)$. We calculate the contribution to $Q(S(a, z; b_1, b_1^*))$ made by pairs of edges in component 0 in the same way as Lemma 8. Setting

$f = b_1 + b_1^*$ this distance contribution is

$$\frac{1}{2^{\nu+k}} \sum_t [g_t(x(b_1 a_1 \cdots a_{N-1} a_N)) - g_t(x(b_1^* a_1 \cdots a_{N-1} a_N))]^2$$

$$= \frac{1}{2^{\nu+k}} \underline{x}^T [2I_{2^{\nu+k}} - 2M(f0 \cdots 0)] \underline{x}.$$

Similarly, the distance contribution made by pairs of edges in component $N-1$ is

$$\frac{1}{2^{\nu+k}} \underline{x}^T [2I_{2^{\nu+k}} - 2M((f0 \cdots 0)^{N-1})] \underline{x}.$$

Note that the first $l$ bits of $f$ are zero and that the last $l$ bits of $(f0 \cdots 0)^i$ are zero for $0 \leq i \leq N-1$. Arguing as in Lemma 8 we obtain

$$2^{\nu+k} Q(S(a, z; b_1, b_1^*)) = 2NI_{2^{\nu+k}} - 2 \sum_{i=0}^{N-1} M((f0 \cdots 0)^i).$$

There are $(2^{k-l} - l)$ $k$ tuples $f$ for which $f \neq 0$ and $\tilde{f} = 0$. Hence

$$2^{\nu+k}(2^{k-l} - 1) Q(S^N) = 2(2^{k-l} - 1) NI_{2^{\nu+k}} - 2 \sum_{\substack{f \neq 0 \\ \tilde{f}=0}} \sum_{i=0}^{N-1} M((f0 \cdots 0)^i).$$

Setting $Q = 2^{\nu+k}(2^{k-l} - 1) Q(S^N)$ we obtain

$$\frac{d_{\min}^2}{P} \leq \frac{1}{2^{k-l} - 1} \lambda_1(Q), \tag{66}$$

which reduces to (39) when $l = 0$. The proof of Theorem 2 goes through (change "$c_i = 0 (\neq 0)$" to "the last $k - l$ digits of $c_i$ are zero (nonzero)")

$$\lambda_1(Q) = 2^{k-l+1} N. \tag{67}$$

By (66) and (67)

$$\frac{d_{\min}^2}{P} \leq \frac{2^{k-l+1}}{2^{k-l} - 1} \left(1 + \left\lfloor \frac{\nu}{k} \right\rfloor\right) \tag{68}$$

for general $k$ and $\nu$.

Finally we consider the set $S'$ of all error events of length $N+1$ for which the $k$ tuples $b_1$, $b_1^*$ input at time 0 satisfy $\tilde{b}_1 = \tilde{b}_1^* = 0$ and for which the $k$ tuples $b_2$, $b_2^*$ input at time 1 satisfy $\tilde{b}_2 = \tilde{b}_2^* = 0$. Let $E \in S'$ with initial state $a = (a_1, \cdots, a_{N-1}, a_N)$ and final state $z = (z_1, \cdots, z_{N-1}, z_N)$, where $a_i$, $z_i$, $i = 1 \cdots, N-1$ are $k$ tuples and $a_N$, $z_N$ are $l$ tuples. At time 2 the two paths occupy states $((z_N 0 \cdots 0) + b_2$,

$b_1, a_1, \cdots, a_{N-3}, \tilde{a}_{N-2}$) and $((z_N 0 \cdots 0) + b_2^*, b_1^*, a_1, \cdots, a_{N-3}, \tilde{a}_{N-2})$. At time $N$ the two paths occupy states $(z_2, z_3, \cdots, z_{N-1}, (z_N 0 \cdots 0) + b_2, \tilde{b}_1)$ and $(z_2, z_3, \cdots, z_{N-1}(z_N 0 \cdots 0) + b_2^*, b_1^*)$. Set $f = b_2 + b_2^*$ and $g = b_1 + b_1^*$ and define $M_i(fg)$ $i = 0, \cdots N$ as in (43). Arguing as in Lemma 9 we obtain

$$2^{\nu+k}(2^{k-l})(2^{k-l}-1)Q(S') = 2(2^{k-l}-1)(2^{k-l}-1)(N+1)I_{2^{\nu+k}} - 2\sum_{\substack{f\neq 0\\ \tilde{f}=0}}\sum_{\substack{g\neq 0\\ \tilde{g}=0}}\sum_{i=0}^{N} M_i(fg0\cdots 0).$$

Let $\delta = 1/(2^{2(k-l)}-1)$ and let $\bar{Q} = 2(2^{k-l}-1)\delta Q(S^N) + (2^{k-l}-1)^2\delta Q(S')$. Then $2(2^{k-l}-1)\delta + (2^{k-l}-1)^2\delta = 1$ and so

$$\frac{d_{\min}^2}{P} \leqslant 2^{\nu+k}\lambda_1(\bar{Q}).$$

The proof of Theorem 3 goes through [change "$c_i = 0$ ($\neq 0$)" to "the last $k-l$ digits of $c_i$ are zero (nonzero)"] and we obtain

$$\frac{d_{\min}^2}{P} \leqslant \frac{2^{2(k-l)+1}}{2^{2(k-l)}-1}\left(2 + \left[\frac{\nu}{k}\right]\right), \tag{69}$$

for general $k$ and $\nu$.

## AUTHORS

**Helene Shapiro,** B.A. (Mathematics), 1975, Kenyon College; Ph.D. (Mathematics), 1979, California Institute of Technology; Assistant Professor of Mathematics, University of Wisconsin-Madison, 1979–80; Assistant Professor of Mathematics, Swarthmore College, 1980—. Ms. Shapiro's research has been in matrix theory and deals with generalized commutativity problems.

**Robert Calderbank,** B.Sc. (Mathematics), 1975, University of Warwick; M.Sc. (Mathematics), 1976, University of Oxford; Ph.D. (Mathematics), 1980, California Institute of Technology; Bell Laboratories, 1980—. Mr. Calderbank is presently employed in the Mathematics and Statistics Research Center.

**James E. Mazo,** B.S. (Physics), 1958, Massachusetts Institute of Technology; M.S. (Physics), 1960, and Ph.D. (Physics), 1963, Syracuse University; Research Associate, Department of Physics, University of Indiana, 1963–1964; Bell Laboratories, 1964—. At the University of Indiana, Mr. Mazo worked on studies of scattering theory. At Bell Laboratories, he has been concerned with problems in data transmission and is now working in the Mathematics and Statistics Research Center. Member, IEEE.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Benes V., Karatzas I., **On the Relation of Zakai Equations and Mortensen Equations.** Siam J Con 21(3):472–489, 1983.

Bos A., Conway J. H., Sloane N. J. A., **Further Lattice Packings in High Dimensions.** Mathematika 29(58):171–180, 1982.

Bremner A. et al., **Two-Weight Ternary Codes and the Equation $Y^2 = 4 \times 3^4 + 13$.** J Number Th 16(2):212–234, 1983.

Burman D. Y., Smith D. R., **A Light-Traffic Theorem for Multi-Server Queues.** Math Oper R 8(1):15–25, 1983.

Chung F. R. K. **On a Ramsey-Type Problem.** J Graph Th 7(1):79–83, 1983.

Chung F. R. K. et al., **A Survey of Bounds for Classical Ramsey Numbers.** J Graph Th 7(1):25–37, 1983.

Chung F. R. K., Graham R. L., **On Universal Graphs for Spanning-Trees.** J Lond Math 27(Apr):203–211, 1983.

Cleveland W. S., McGill R., **A Color-Caused Optical Illusion on a Statistical Graph.** Am Statistn 37(2):101–105, 1983.

Coffman E. G., Garey M. R., Johnson D. S., **Dynamic Bin Packing.** Siam J Comp 12(2):227–258, 1983.

Cunningham W. H., Klincewicz J. G., **On Cycling in the Network Simplex-Method.** Math Progr 26(2):182–189, 1983.

Dalal S. R., Phadia E. G., **Nonparametric Bayes Inference for Concordance in Bivariate Distributions.** Comm St-The 12(8):947–963, 1983.

Graham R. L., **Euclidean Ramsey Theorems on the *N*-Sphere.** J Graph Th 7(1):105–114, 1983.

Halfin S., **The Sphere Method and the Robustness of the Ellipsoid Algorithm.** Math Progr 26(1):109–116, 1983.

Derosa F., **Commodore Pet Computer Used as a Programmable Gated Integrator.** Rev Sci Ins 54(5):630–632, 1983.

Haddon R., Lamola A., **The Organic Computer—Can Microchips be Built by Bacteria?.** Science 23(3):40–44, 1983.

Hennessy J. L. et al., **Complex Versus Reduced Instruction Set Computers.** ISSCC Diges 26:218–219, 1983.

Johnson D. S., Niemi K. A., **On Knapsacks, Partitions, and a New Dynamic-Programming Technique for Trees.** Math Oper R 8(1):1–14, 1983.

Kruskal J. B., **An Overview of Sequence Comparison—Time Warps, String Edits, and Macromolecules.** Siam Rev 25(2):201–237, 1983.

Kruskal J. B., Landwehr J. M., **Icicle Plots—Better Displays for Hierarchical-Clustering.** Am Statistn 37(2):162–168, 1983.

Lagarias J. C., Odlyzko A. M., Shearer J. B., **On the Density of Sequences of Integers the Sum of No Two of Which is a Square. 2. General Sequences.** J Comb Th A 34(2):123–139, 1983.

Lagarias J. C., Aminzade F., **Multistage Planning and the Extended Linear-Quadratic-Gaussian Control Problem.** Math Oper R 8(1):42–63, 1983.

Leon R. V., Lynch J., **Preservation of Life Distribution Classes Under Reliability Operations.** Math Oper R 8(2):159–169, 1983.

Reilly T. M., **The Extended Classroom.** Datamation 29(4):199–200, 1983.

Riffelmacher D., **Multiplicative Complexity and Algebraic Structure.** J Comput Sy 26(1):92–106, 1983.

Sandberg I. W., **Volterra-Like Expansions for Solutions of Non-Linear Integral-Equations and Non-Linear Differential-Equations.** IEEE Circ S 30(2):68–77, 1983.

Stuck B. W., **Calculating the Maximum Mean Data Rate in Local Area Networks.** Computer 16(5):72–76, 1983.

Witsenhausen H. S., **Team Guessing With Lacunary Information.** Math Oper R 8(1):110–121, 1983.

Yu Y. K., Banerji R. B., **Periodicity of Sprague-Grundy Function in Graphs With Decomposable Nodes.** Cybern Syst 13(4):299–310, 1982.

## ELECTRICAL ENGINEERING

Alavi K., Pearsall T. P., Forrest S. R., Cho A. Y., **$Ga_{0.47}In_{0.53}As/Al_{0.48}In_{0.52}$ Multiquantum-Well LEDs Emitting at 1.6 $\mu$m.** Electr Lett 19(6):227–229, 1983.

Allara D. L., Nuzzo R. G., **The Application of Reflection Infrared and Surface Enhanced Raman-Spectroscopy to the Characterization of Chemisorbed Organic Disulfides on Au.** J Elec Spec 30(Feb):11, 1983.

Antler M., Ratliff E. T., **Sliding Wear of Inlay Clad Metals and Electrodeposited Cobalt-Gold.** IEEE Compon 6(1):3–7, 1983.

Auston D. H., **Impulse-Response of Photoconductors in Transmission-Lines.** IEEE J Q El 19(4):639–648, 1983.

Arnold H. W., Bodtmann W. F., **A Hybrid Multichannel Hardware Simulator for Frequency-Selective Mobile Radio Paths.** IEEE Commun 31(3):370–377, 1983.

Banu M., Tsividis Y., **Fully-Integrated Active RC-Filters in MOS Technology.** ISSCC Diges 26:244+, 1983.

Bergmann E. E., **Polarization Independent Beam Splitter.** Rev Sci Ins 54(5):579–581, 1983.

Blum E. B., Siemann R. H., Auston D. H., Freeman R. R., **Bunch Length Measurements in CeSr Using an X-Ray Sensitive Photoconducting Detector.** Nucl Instru 207(3):321–324, 1983.

Bobeck A. H., **Citation Classic—Properties and Device Applications of Magnetic Domains in Ortho-Ferrites.** CC/Eng Tech 1983(20):24, 1983.

Boll H. J. et al., **Practical Versus Theoretical Limits of VLSI (Discussion).** ISSCC Diges 26:212–216, 1983.

Chu T. S., **A Semi-Empirical Formula for Microwave Depolarization Versus Rain Attenuation on Earth-Space Paths (Letter).** IEEE Commun 30(12):2550–2554, 1982.

Coffey B. J., Lax M., Elliott C. J., **Three-Dimensional Propagation in Free-Electron Laser-Amplifiers.** IEEE J Q El 19(2):297–305, 1983.

Dorros I., **Telephone Nets Go Digital.** IEEE Spectr 20(4):48–53, 1983.

Ducorps A. M., Yashinovitz C. J., **CAMAC Controller for a Conventional and Pseudo-Random Time-of-Flight System.** Rev Sci Ins 54(4):444–453, 1983.

Dutta N. K., Deimel P. P., **Optical-Properties of a GaAlAs Superluminescent Diode.** IEEE J Q El 19(4):496–498, 1983.

Ericson T., Korner J., **Successive Encoding of Correlated Sources.** IEEE Info T 29(3):390–395, 1983.

Feldman M., White A. D., White D. L., **Application of Zone Plates to Alignment in X-Ray-Lithography.** Opt Eng 22(2):203–207, 1983.

Fork R. L., Shank C. V., Yen R., Hirlimann C. A., **Femtosecond Optical Pulses.** IEEE J Q El 19(4):500–506, 1983.

Foschini G. J., Gopinath B., **Sharing Memory Optimally.** IEEE Commun 31(3):352–360, 1983.

Fraser D. L., Williams G. F., Jindal R. P., Kushner R. A., Owen B., **A Single-Chip NMOS Pre- Amplifier for Optical Fiber Receivers.** ISSCC Diges 26:80–81, 1983.

Friesem A. A. et al., **Parallel Transmission of Images Through Single Optical Fibers.** P IEEE 71(2):208–221, 1983.

Fuls E. N., **X-Ray-Lithography Applied to the Fabrication of One Micrometer *N*-Channel Metal-Oxide Semiconductor Circuits.** Opt Eng 22(2):199–202, 1983.

Guancial E., Kohl R. W., Dunbar J. J., Milton J. L., Dasgupta S., **Qualification of Connectors Manufactured With Diffused Gold RI56 Inlay Contacts.** IEEE Compon 6(1):100–107, 1983.

Haque C. A., Drozdowicz M. H., Frank R. A., Hanlon J. T., **Extraneous Metal Deposits From Production Processes on Contact Materials.** IEEE Compon 6(1):55–60, 1983.

Hartman D. H., **Multiple Reflection Noise on Transmission-Systems With Periodic or Quasi-Periodic Impedance Disturbances—A Frequency-Domain Approach.** IEEE Circ S 30(4):242–246, 1983.

Hatamian M., Anderson D. J., **Design Considerations for a Real-Time Ocular Counterroll Instrument.** IEEE Biomed 30(5):278–288, 1983.

Johnson P. D., Perlman M. L., **Slit Width Calibration With a Radioactive Photon Source.** Nucl Instru 206(3):593–594, 1983.

Kadota T. T., **Piecewise Linear Random-Paths on a Plane and a Central Limit-Theorem.** IEEE Info T 29(2):241–245, 1983.

Kadota T. T., Seery J. B., **Detection of Weak Secondary Signals With the Aid of Already Detected Strong Primary Signals.** IEEE Acoust 31(2):426–437, 1983.

Kaminow I. P., Eisenstein G., Stulz L. W., **Measurement of the Modal Reflectivity of an Anti-Reflection Coating on a Superluminescent Diode.** IEEE J Q El 19(4):493–495, 1983.

Macrander A., Barr D., Wagner A., **Resist Possibilities in Ion-Beam Lithography.** Opt Eng 22(2):215–219, 1983.

O'Connor K. J., Kushner R. A., **A Five ns 4K × 1 NMOS Static RAM.** ISSCC Diges 26:104+, 1983.

Prabhu K. A., Netravali A. N., **Motion Compensated Component Color Coding.** IEEE Commun 30(12):2519–2527, 1982.

Prabhu K. A., Netravali A. N., **Motion Compensated Composite Color Coding.** IEEE Commun 31(2):216–223, 1983.

Prabhu V. K., **Error Rate Bounds for Differential PSK (Letter).** IEEE Commun 30(12):2547–2550, 1982.

Saari V. R., **Low-Power High-Drive CMOS Operational-Amplifiers.** IEEE J Soli 18(1):121–127, 1983.

Sandberg I. W., **On Volterra Expansions for Time-Varying Non-Linear Systems.** IEEE Circ S 30(2):61–67, 1983.

Sharma S. P., Hines L. L., **Oxidation of Ruthenium.** IEEE Compon 6(1):89–92, 1983.

Sloane N. J. A., Thompson J. G., **Cyclic Self-Dual Codes.** IEEE Info T 29(3):364–366, 1983.

Stone J., Chraplyvy A. R., **Spontaneous Brillouin Noise in Long-Distance High-Bandwidth Optical-Fibre Transmission.** Electr Lett 19(8):275–277, 1983.

Swanson E. J., Starke R. J., Cross G. F., Olson K. H., Waldron C. J., Copeland R. A., Surek S. A., Ribble R. J., Vera A. J., **A Fully Adaptive Transversal Canceler and Equalizer Chip.** ISSCC Diges 26:20+, 1983.

Tucker R. S., Pope D. J., **Microwave Circuit Models of Semiconductor Injection-Lasers.** IEEE Micr T 31(3):289–294, 1983.

Uman M. A., Master M. J., Krider E. P., **A Comparison of Lightning Electromagnetic-Fields With the Nuclear Electromagnetic Pulse in the Frequency-Range 104–107 Hz.** IEEE Elmagn 24(4):410–416, 1982.

Winters J. H., **Switched Diversity With Feedback for DPSK Mobile Radio Systems.** IEEE Veh T 32(1):134–150, 1983.

Wittmer N. C., Michejda J. A., Gannett J. W., Bechtold P. F., Taylor G. W., Lifshitz N., Dennis D. C., Bayruns R. J., **A NMOS LSI 16 × 16 Multiplier.** ISSCC Diges 26:32–33, 1983.

Wong W. C., Steele R., Glance B., Horn D., **Time Diversity With Adaptive Error-Detection to Combat Rayleigh Fading in Digital Mobile Radio.** IEEE Commun 31(3):378–387, 1983.

Wu F. F., Tsai Y. K., **Probabilistic Dynamic Security Assessment of Power-Systems. 1. Basic Model.** IEEE Circ S 30(3):148–159, 1983.

Yue O. C., **Spread Spectrum Mobile Radio, 1977–1982.** IEEE Veh T 32(1):98–105, 1983.

## MANAGEMENT/ECONOMICS

Porter R. H., **On The Rationality of Buying From a Price Predator.** Econ Lett 11(4):385–389, 1983.

Whitt W., **Untold Horrors of the Waiting Room—What the Equilibrium Distribution Will Never Tell About the Queue-Length Process.** Manag Sci 29(4):395–408, 1983.

## PHYSICAL SCIENCES

Abrahams S. C., et al., **Dielectric and Thermal-Properties, Expansion, and High-Temperature Plastic-Deformation of $Cd_4Na(VO_4)_3$.** J Appl Crys 16(FEB):133–135, 1983.

Adams A. C., **Plasma Deposition of Inorganic Films.** Sol St Tech 26(4):135–139, 1983.

Anderson P. W., **Localization Redux.** Physica B&C 117(MAR):30–36, 1983.

Antonio M. R., Teo B. K., Cleland W. E., Averill B. A., **Iron and Molybdenum Extended X-Ray Absorption Fine-Structure Studies of Double-Cubane Clusters Containing $MoFe_3S_4$.** J Am Chem S 105(11):3477–3484, 1983.

Aspnes D. E., **Surface Transition Regions and Visible-Near UV Optical-Properties of Some Semiconductors.** Physica B&C 117(MAR):359–361, 1983.

Azbel M. Y., **Impurities in Semimetals.** Sol St Comm 46(2):113–115, 1983.

Bates F. S., Baker G. L., **Soluble Polyacetylene Graft Co-Polymers (Letter).** Macromolec 16(4):704–707, 1983.

Belanger R. M. et al., **Determination of the Group-Velocity of Magnetic Polaritons in $FeF_2$.** J Magn Magn31-4(FEB):681–682, 1983.

Benbow R. L., Smith N. V., **Photoemission Spectra and Band Structures of D-Band Metals. 10. Relativistic Momentum Matrix-Elements.** Phys Rev B 27(6):3144–3151, 1983.

Bertz S. H., Jelinski L. W., Dabbagh G., **Conjugate Addition of Lithium Isopropyl Dithioisobutyrate to 4-Acetoxycyclopent-2-enone. Assignment of Stereochemistry in Flexible 5-Membered Rings.** J Chem S CH1983(7):388–390, 1983.

Bethea C. G., Chen C. Y., Cho A. Y., Garbinski P. A., Levine B. F., **Picosecond $Al_xGa_{1-x}As$ Modulation-Doped Optical Field-Effect Transistor Sampling Gate.** Appl Phys L 42(8):682–684, 1983.

Blumenfeld K. S., Kozlowski S., Breslauer K. J., **Salt-Dependent Conformational Transitions in the Self-Complementary Deoxydodecanucleotide d(CGCAATTCGCG)-Evidence for Hairpin Formation.** Biopolymers 22(4):1247–1257, 1983.

Bokor J., Bucksbaum P. H., Freeman R. R., **Generation of 35.5-nm Coherent Radiation.** Optics Lett 8(4):217–219, 1983.

Bondybey V. E., Milton S. V., English J. H., Rentzepis P. M., **Spectroscopic and Picosecond Time-Resolved Studies of Vibrational-Relaxation in Naphthazarine in Solid Neon.** Chem P Lett 97(2):130–134, 1983.

Bondybey V. E., English J. H., Miller T. A., **Spectroscopic and Dynamical Studies of Vibrationally Unrelaxed Fluorescence From Halobenzene Cations in a Neon Matrix.** J Phys Chem 87(8):1300–1305, 1983.

Bondybey V. E., Heaven M., Miller T. A., **Laser Vaporization of Tin—Spectra and Ground-State Molecular-Parameters of $Sn_2$.** J Chem Phys 78(6):3593–3598, 1983.

Bosch M. A., Good E., Lemons R. A., **Electrically Amplified Optical-Recording in Polymer-Films.** Appl Phys L 42(6):484–486, 1983.

Bovey F. A., Cais R. E., Jelinski L. W., Schilling F. C., Starnes W. H., Tonelli A. E., **Structural and Dynamic Characterization of Polymers by C-13 and F-19 NMR (Review).** Adv Chem SE1983(203):421–439, 1983.

Bowden M. J., Keith H. D., **A Comment on Crystallinity in Poly(olefin Sulfones).** Eur Polym J 19(3):259–261, 1983.

Bowden M. J., Novembre T., **Initiation Considerations and Kinetics of Formation of Poly(2-Methyl-1-Pentene Sulfone) (Review).** ACS Symp S 1983(212):125–136, 1983.

Bowmer T. N., Hellman M. Y., Vocom W. I., **Radiation Crosslinking of Polyvinyl-Chloride With Trimethylolpropanetrimethacrylate. 2. Dependence on Radiation-Dose and Blend Composition.** J Appl Poly 28(6):2083–2092, 1983.

Brand H., Swift J., **Macroscopic Dynamics in Nematics Near the Uniaxial-Biaxial Phase-Transition.** J Phys Lett 44(9):L333–L337, 1983.

Bruinsma R., Aeppli G., **One-Dimensional Ising-Model in a Random Field.** Phys Rev L 50(19):1494–1497, 1983.

Capizzi M., Frova A., Martelli F., Modesti S., Quagliano L. G., Staehli J. L., Guzzi M., Logan R. A., Chiaretti G., **The E-H Plasma in Direct-Gap GaAlAs Under Three-D Confinement.** Physica B&C 117(Mar):333–335, 1983.

Chabal Y. J., Chaban E. E., Christman S. B., **High-Resolution Infrared Study of Hydrogen Chemisorbed on Si(100).** J Elec Spec 29(Jan):35–40, 1983.

Chandler D., Weeks J. D., Andersen H. C., **Vanderwaals Picture of Liquids, Solids, and Phase-Transformations.** Science 220(4599):787–794, 1983.

Chandross E. A., Reichmanis E., Wilkins C. W., Hartless R. L., **New Photoresists for Deep Ultraviolet (Less-Than 300 nm) Exposure.** Can J Chem 61(5):817–823, 1983.

Chang C. C., Sheng T. T., Shankoff T. A., **Phosphorus Depth Profiles in Thermally Oxidized P-Doped Polysilicon.** J Elchem So 130(5):1168–1171, 1983.

Chemla D. S., Damen T. C., Miller D. A. B., Gossard A. C., Wiegmann W., **Electro-Absorption by Stark-Effect on Room-Temperature Excitons in GaAs-GaAlAs Multiple Quantum Well Structures.** Appl Phys L 42(10):864–866, 1983.

Chemla D. S., Heritage J. P., Liao P. F., Isaacs E. D., **Enhanced Four-Wave Mixing From Silver Particles.** Phys Rev B 27(8):4553–4558, 1983.

Chen C. H., Fleming R. M., Petroff P. M., **Time-Dependent Strandlike Domains in the Charge-Density-Wave States of $NbSe_3$—an Effect Not Induced by Electric-Field.** Phys Rev B 27(7):4459–4462, 1983.

Chen H. S., Aust K. T., Waseda Y., **Effect of Low-Temperature Annealing on the Structure of $Cu_{57}Zr_{43}$ Glass.** J Mat Sci L 2(4):153–157, 1983.

Chin A. K., **A Simple, High Collection Efficiency Cathodoluminescence Imaging Technique—Application to GaSb and CdTe.** Scan Elec M1982(P3):1069–1075, 1982.

Chin, A. K., Camlibel I., Dutt B. V., Swaminathan. V., Bonner W. A., Ballman A. A., **Extremely Rapid Outdiffusion of N-Type Impurities in InP.** Appl Phys L 42(10):901–903, 1983.

Cho A. Y., **Growth of III-V Semiconductors by Molecular-Beam Epitaxy and Their Properties.** Thin Sol Fi 100(4):291–317, 1983.

Citrin P. H., Hamann D. R., Mattheiss L. F., Rowe J. E., **Comment on the Geometry and Electronic-Structure of Cl on Cu(100) and Ag(001)—Reply (Letter).** Phys Rev L 50(22):1824, 1983.

Citrin P. H., Rowe J. E., Eisenberger P., Comin F., **SEXAFS Studies of Adsorbate Structure on Si and Ge Surfaces.** Physica B&C 117(Mar):786–788, 1983.

Citrin P. H., Wertheim G. K., **Photoemission From Surface-Atom Core Levels, Surface Densities of States, and Metal-Atom Clusters—A Unified Picture (Review).** Phys Rev B 27(6):3176–3200, 1983.

Citrin P. H., Wertheim G. K., Baer Y., **Surface-Atom X-Ray Photoemission From Clean Metals—Cu, Ag, and Au.** Phys Rev B 27(6):3160–3175, 1983.

Copeland J. A., **Fluctuations in Semiconductor-Laser Emissions.** J Appl Phys 54(5):2813–1819, 1983.

Cottam M. G., So V., Lockwood D. J., Katiyar R. S., Guggenheim H. J., **Raman-Scattering From Magnon Pair Excitations in $FeF_2$.** J Phys C 16(9):1741–1755, 1983.

Cottrell P. T., Frankenthal R. P., Kammlott, G. W., Siconolfi D. J., Draper C. W., **Effect of Laser Surface Melting on Some Corrosion Characteristics of an Iron-Chromium Alloy.** J Elchem So 130(5):998–1001, 1983.

Danner D. A., Flamm D. L., Mucha J. A., **Downstream Atomic Monitoring for Absolute Etch Rate Determinations.** J Elchem So 130(4):905–907, 1983.

Defreez R. K. et al., **High-Output Room-Temperature Pulsed Operation for Broad Contact $InP/In_{0.53}Ga_{0.47}As/InP$ Lasers Grown By Molecular-Beam Epitaxy.** J Appl Phys 54(5):2177–2182, 1983.

Denbroeder F. J. A., Nakahara S., **Diffusion Induced Grain-Boundary Migration and Recrystallization in the Cu-Ni System.** Scrip Metal 17(3):399–404, 1983.

Dillon J. F., **Magneto-Optics and Its Uses.** J Magn Magn31–4(Feb):1–9, 1983.

Dimauro L. F., Wood D. E., Lloyd R. V., **Polarized Emission From Doublet State Organic Free-Radicals.** J Chem Phys 78(10):5920–5925, 1983.

Donovan E. P., Spaepen F., Turnbull D., Poate J. M., Jacobson D. C., **Heat of Crystallization and Melting-Point of Amorphous-Silicon.** Appl Phys L 42(8):698–700, 1983.

Dubois L. H., Schwartz G. P., **High-Resolution EELS Studies of Clean and Oxide Covered Semiconductor Surfaces.** J Elec Spec 29(Jan):175–180, 1983.

Dudderar T. D. et al., **Achieving Stability in Remote Holography Using Flexible Multimode Image Bundles.** Appl Optics 22(7):1000–1005, 1983.

Edelson D., **Sensitivity Analysis of Proposed Mechanisms for the Briggs-Rauscher Oscillating Reaction.** J Phys Chem 67(7):1204–1208, 1983.

Eisenberger P, Okamura M. Y., Feher G., **The Electronic-Structure of Fe-2+ in Reaction Centers From Rhodopseudomonas-Sphaeroides. 2. Extended X-Ray Fine-Structure Studies.** Biophys J 37(2):523–538, 1982.

Englert T., Maan J. C., Tsui D. C., Gossard A. C., **A Study of Intersubband Scattering in GaAs-$Al_xGa_{1-x}As$ Heterostructures by Means of a Parallel Magnetic-Field.** Sol St Comm 45(11):989–991, 1983.

Englert T., Maan J. C., Uihlein C., Tsui D. C., Gossard A. C., **Oscillations of the Cyclotron-Resonance Linewidth With Landau-Level Filling Factor in GaAs/AlGaAs Heterostructures.** Physica B&C 117(Mar):631–633, 1983.

Englert T., Maan J. C., Uihlein C., Tsui D. C., Gossard A. C., **Observation of Oscillatory Linewidth in the Cyclotron-Resonance of GaAs-$Al_xGa_{1-x}As$ Heterostructures.** Sol St Comm 46(7):545–548, 1983.

Fisher D. S., **Threshold Behavior of Charge-Density Waves Pinned by Impurities.** Phys Rev L 50(19):1486–1489, 1983.

Fisher D. S., Weeks J. D., **Shape of Crystals at Low-Temperatures—Absence of Quantum Roughening.** Phys Rev L 50(14):1077–1080, 1983.

Flamm D. L., Ibbotson D. E., Mucha J. A., Donnelly V. M., **$XeF_2$ and F-Atom Reactions with Si—Their Significance for Plasma-Etching.** Sol St Tech 26(4):117–121, 1983.

Flamm D. L., **Plasma Processing (Editorial).** Sol St Tech 26(4):115, 1983.

Frankenthal R. P., Siconolfi D. J., **The Equilibrium Surface-Composition of Some Indium Lead and Indium Tin Alloys.** Surf Sci 125(2):595–603, 1983.

Fulton T. A., Dolan G. J., **New Approach to Electron-Beam Lithography.** Appl Phys L 42(8):752–754, 1983.

Gibson J. M., Chen C. H., McDonald M. L., **Ultrahigh-Resolution Electron-Microscopy of Charge-Density Waves in 2H-$TaSe_2$ Below 100-K.** Phys Rev L 50(18):1403–1406, 1983.

Glarum S. H., Marshall J. H., **An Admittance Study of the Tin Electrode.** J Elchem So 130(5):1088–1096, 1983.

Graebner J. E., Allen L. C., Golding B., Kane A. B., **Acoustic Saturation in a Glass at Low-Temperatures.** Phys Rev B 27(6):3697–3708, 1983.

Greywall D. S., **Specific Heat of Normal Liquid $^3He$.** Phys Rev B 27(5):2747–2766, 1983.

Gyorgy E. M., McWhan D. B., Dillon J. F., Walker L. R., Waszczak J. V., Musser D. P., Willens R. H., **Structural and Magnetic-Properties of Compositionally Modulated Cu-Ni Films.** J Magn Magn31-4(Feb):915–919, 1983.

Haavasoja T., Narayanamurti V., Chin M. A., **Propagation of Near-Zone-Boundary, Acoustic Phonons in Solid $^4He$ With the Use of Superconducting Tunnel-Junctions.** Phys Rev B 27(5):2767–2769, 1983.

Hall P. M., **Solder Attachment of Leadless Ceramic Chip Carriers.** Sol St Tech 26(3):103–107, 1983.

Hamann D. R., **Computers in Physics—An Overview (Review).** Phys Today 36(5):24–33, 1983.

Haque C. A., **Durability of Solid Fluoropolymer Thin-Films on Metal-Surfaces for Silicone Oil Creep.** Appl Surf S 15(1–4):293–306, 1983.

Harley R. T., Lyons K. B., Fleury P. A., Smith S. R. P., **Fluctuation Dynamics at the Cooperative Jahn-Teller Phase Transition in $TbVO_4$.** J Phys C 16(8):1407–1422, 1983.

Hasegawa A., **Recent Topics in Physics of Hot Plasmas in Space Environment.** Phys Scr T2(NSI):223–227, 1982.

Hauser J. J., **Effect of Crystal Structure on the Magnetic-Properties of Semiconducting $Cd_{0.5}Mn_{0.5}Te$.** Phys Rev B 27(6):3460–3464, 1983.

Hebard A. F., Fiory A. T., **Critical-Exponent Measurements of a Two-Dimensional Superconductor.** Phys Rev L 50(20):1603–1606, 1983.

Helfand E., **Brownian Dynamics Simulation of Systems With Transitions. Particularly Conformational Transitions in Chain Molecules.** Physica A., 118(1–3):123–135, 1983.

Hemmerle W. J., Carey M. B., **Some Properties of Generalized Ridge Estimators.** Comm St-Sim 12(3):239–256, 1983.

Hilinski E. F., Milton S. V., Ramana C., Rentzepis, P. M., **On the Chemistry of Metastable Species.** Can J Chem 61(5):999–1006, 1983.

Hilinski E. F., Rentzepis P. M., **Biological Applications of Picosecond Spectroscopy (Review).** Nature 302(5908):481–487, 1983.

Hong M. et al., **Superconducting Properties of a Liquid-Infiltration Nb-$Nb_3Sn$ Composite Formed During a Low-Temperature Reaction.** Appl Phys L 42(7):621–623, 1983.

Hopfel R. A., Gornik E., Gossard A. C., Wiegmann W., **Plasmon Excitation by Coulomb Scattering of Electrons in Two-D-Systems.** Physica B&C 117(MAR):646–648, 1983.

Hu E. L., Howard R. E., Grabbe P., Tennant D. M., **Ion-Beam Processing Using Metal on Polymer Masks.** J Elchem So 130(5):1171–1173, 1983.

Jackel J. L., Rice C. E., Veselka J. J., **Composition Control in Proton-Exchanged $LiNbO_3$.** Electr Lett 19(10):387–388, 1983.

Jelinski L. W., Dumais J. J., Engel A. K., **Motions About 3 Bonds in Polymers—Solid-State Deuterium NMR-Studies (Letter).** Macromolec 16(3):492–496, 1983.

Jelinski L. W., Dumais J. J., Engel A. K., **Multitechnique Solid-State NMR Approach to Assessing Molecular-Motion—Poly(Butylene Terephthalate) and Poly(Butylene Terephthalate)-Containing Segmented Co-Polymers. 4.** Macromolec 16(3):403–409, 1983.

Jelinski L. W., Dumais J. J., Watnick P. I., Engel A. K., Sefcik M. D., **NMR Relaxation in Poly(Butylene Terephthalate) and Poly(Butylene Terephthalate)-Containing Segmented Co-Polmers. 5.** Macromolec 16(3):409–415, 1983.

Johnson D. W. et al., **Use of Mn-Based and Mg-Based Ferrites for Negative-Temperature-Coefficient Thermistors.** Am Ceram S 62(5):597–600, 1983.

Johnson L. F., Ingersoll K. A., **Ion Polishing With the Aid of a Planarizing Film.** Appl Optics 22(8):1165–1167, 1983.

Jopson R. M., Darack S., Freeman R. R., Bokor J., **Laser-Plasma-Induced Extreme-Ultraviolet Radiation From Liquid Mercury.** Optics Lett 8(5):265–267, 1983.

Kamgar A., Johnston R. L., **Delay Times in Si MOSFETs in the 4.2-400 K Temperature-Range.** Sol St Elec 26(4):291–294, 1983.

Kang K. S., Shay J. L., **Blue Sputtered Iridium Oxide-Films (Blue SIROFS).** J Elchem So 130(4):766–769, 1983.

Kelley D. F., Milton S. V., Huppert D., Rentzepis P. M., **Formation Kinetics and Spectra of Aromatic Radicals in Solution (Letter).** J Phys Chem 87(11):1842–1843, 1983.

Kelley D. F., Rentzepis P. M., **Picosecond Emission-Spectroscopy of Two Predissociative Isomeric Methylenepyrazolines and Product Trimethylenemethane.** J Am Chem S 105(7):1820–1824, 1983.

Kelso S. M., Aspnes D. E., Olson C. G., Lynch D. W., **Optical Determination of Exciton Sizes in Semiconductors.** Physica B&C 117(MAR):362–364, 1983.

Kinsbron E. et al., **Crystallization of Amorphous-Silicon Films During Low-Pressure Chemical Vapor-Deposition.** Appl Phys L 42(9):835–837, 1983.

Klauder J. R., Ezawa H., **Remarks on a Stochastic Quantization of Scalar Fields.** Prog T Phys 69(2):664–673, 1983.

Kodama Y., Hasegawa A., **Amplification and Reshaping of Optical Solitons in Glass-Fiber. 3. Amplifiers With Random Gain.** Optics Lett 8(6):342–344, 1983.

Kolodner P., Katzir A., Hartsough N., **Non-Contact Surface-Temperature Measurement During Reactive-Ion Etching Using Fluorescent Polymer-Films.** Appl Phys L 42(8):749–751, 1983.

Kurkjian C. R., Krause J. T., Paek U. C., **Tensile-Strength Characteristics of Perfect Silica Fibers.** J Physique 43(NC-9):585–586, 1982.

Levine B. F., Bethea C. G., Tsang W. T., Capasso F., Thornber K. K., Fulton R. C., Kleinman D. A., **Measurement of High Electron-Drift Velocity in a Sub-Micron, Heavily Doped Graded GaP $Al_xGa_{1-x}As$ Layer.** Appl Phys L 42(9):769–771, 1983.

Levinson M., Benton J. L., Kimerling L. C., **Electronically Controlled Metastable Defect Reaction in InP.** Phys Rev B 27(10):6216–6221, 1983.

Levinson M., Temkin H., **Defect States in Electron-Irradiated InGaAsP.** Appl Phys L 42(7):605–607, 1983.

Lewis J. A., **Thermal-Convection and Equilibrium Liquid-Phase Epitaxial-Growth.** Ann NY Acad 404(MAY):301, 1983.

Lin C., **Non-Linear Optics in Fibers and Near-Infrared Frequency-Conversion.** P Soc Photo 355:17–26, 1982.

Lin P. S. D., Leamy H. J., **Tunneling Current Microscopy.** Appl Phys L 42(8):717–719, 1983.

Lines M. E., **Topical Review—Computer Modeling of Amorphous Magnetic-Materials (Review).** J Magn Magn 36(1–2):1–25, 1983.

Lines M. E., Eibschutz M., **Influence of Isomer-Shift Distribution on Mossbauer Quadrupole Spectra in Amorphous Magnetic Insulators.** Phys Rev B 27(9):5308–5312, 1983.

Lovinger A. J., **Ferroelectric Polymers.** Science 220(4602):1115–1121, 1983.

Lum R., Wilkins C. W., Robbins M., Lyons A. M., Jones R. P., **Thermal-Analysis of Graphite and Carbon Phenolic Composites by Pyrolysis-Mass Spectrometry.** Carbon 21(2):111–116, 1983.

Lynn J. W., Erwin R. W., Chen H. S., Rhyne J. J., **Evolution From Ferromagnetism to Spin-Glass Behavior.** Sol St Comm 46(4):317–320, 1983.

Lynn J. W., Erwin R. W., Rhyne J. J., Chen H. S., **Ferromagnetic and Spin-Glass Behavior Near the Critical Concentration in Amorphous $(Fe_xNi_{1-x})_{75}Gl_{25}$.** J Magn Magn 31-4(FEB):1397–1398, 1983.

MacDonald J. R., Davies J. A., Jackman T. E., Feldman L. C., **How Well Does $^4He$ Backscattering From Low-$Z$ Nuclei Obey the Rutherford Formula?** J Appl Phys 54(4):1800–1803, 1983.

Marcus M., Dynes R. C., Griffiths J. E., McWhan D., **Effect of Ion-Beam Damage on X-Ray-Diffraction and Raman-Spectra of Amorphous $GeSe_2$.** J Appl Phys 54(4):1792–1794, 1983.

Marcus M. A., Tsai C. L., Giessen B. C., **Interstitial Site of Cu in Ca.** Sol St Comm 46(6):455–456, 1983.

Mccroryjoy C., Joy D. C., **Chemical and Instrumental Analysis of Ferrites (Review).** Talanta 30(5):299–315, 1983.

McShera C., Colleran P. J., Glynn T. J., Imbusch G. F., Remeika J. P., **Luminescence Study of $LiGa_5$-$xFe_xO_8$.** J Luminesc 28(1):41–52, 1983.

Miller B. et al., **Applications of Hydrodynamic Modulation at Rotating-Disk Electrodes to Semiconductor Electrochemistry.** Ann NY Acad 404(MAY):473–474, 1983.

Miller R. C., Gossard A. C., Tsang W. T., **Extrinsic Photo-Luminescence From GaAs Quantum Wells.** Physica B&C 117(MAR):714–716, 1983.

Mollenauer L. F., Stolen R. H., Gordon J. P., Tomlinson W. J., **Extreme Picosecond Pulse Narrowing by Means of Soliton Effect in Single-Mode Optical Fibers.** Optics Lett 8(5):289–291, 1983.

Mollenauer L. F., Vieira N. D., Szeto L., **Optical-Properties of the $T1^0(1)$ Center in KC1.** Phys Rev B 27(9):5332–5346, 1983.

Mucha J. A., Hess D. W., **Plasma-Etching (Review).** ACS Symp S 1983(219):215–285, 1983.

Munch R. et al. **High-Pressure and Extended X-Ray-Absorption Fine-Structure Study of Cr-Rich Cr-Ge Alloys.** Phys Rev L 50(20):1619–1622, 1983.

Murarka S. P., Retajczyk T. F., **Effect of Phosphorus Doping on Stress in Silicon and Polycrystalline Silicon.** J Appl Phys 54(4):2069–2072, 1983.

Murnick D. E., **Intense Polarized $^3He$ Source for Use in Fusion Reactors.** Appl Phys L 42(6):544–545, 1983.

Murray C. A., **Using Surface Enhanced Raman-Scattering to Study Vibrations**

**of Adsorbates on Thin Metallic Overlayers on Silver.** J Elec Spec 29(JAN):371–382, 1983.

Nakahara S., **Diffusion-Induced Grain-Boundary Migration in Heated Cu-Ni Alloy Targets During Sputtering.** Scrip Metal 17(5):607–610, 1983.

Nakahara S., Okinaka Y., **Microstructure and Ductility of Electroless Copper-Deposits.** Act Metall 31(5):713–724, 1983.

Nassau K., Chadwick D. L., **Multicomponent Glasses of $GeO_2$ and $Sb_2O_3$ With $Bi_2O_3$, $Tl_2O$, and/or PbO.** J Am Ceram 66(5):332–337, 1983.

Nazarathy M., Hardy A., Shamir J., **Generalized-Mode Theory of Conventional and Phase-Conjugate Resonators.** J Opt Soc 73(5):576–586, 1983.

Nazarathy M, Shamir J., Hardy A., **Nonideal Phase-Conjugate Resonators—A Canonical Operator Analysis.** J Opt Soc 73(5):587–593, 1983.

Nelson D. F., Abraham D. L., Cooper J. A., **Measurement of Cube-Root Broadening of Charge Packets in Metal-Oxide-Semiconductor Structures.** Appl Phys L 42(9):815–817, 1983.

Nicholas R. J., Brummell M. A., Portal J. C., Cheng K. Y., Cho A. Y., Pearsall T. P., **An Experimental-Determination of Enhanced Electron g-Factors in GaInAs-AlInAs Heterojunctions.** Sol St Comm 45(10):911–914, 1983.

Nicholas R. J., Brummell M. A., Portal J. C., Cho A. Y., Cheng K. Y., Pearsall T. P., **Quantum Transport in GaInAs-AlInAs Heterojunctions.** Lect N Phys 177:110–113, 1983.

Nicholas R. J., Brummell M. A., Portal J. C., Razeghi M., Poisson M. A., **Magneto-transport in GaInAs-InP Heterojunctions and Super-Lattices.** Lect N Phys 177:98–109, 1983.

Niklasson G. A., Craighead H. G., **Optical-Properties of Codeposited Aluminum-Silicon Composite Films.** Appl Optics 22(8):1237–1240, 1983.

Nitzan A., Tully J. C., **Stochastic Classical Trajectory Approach to Relaxation Phenomena. III. Comparison of Trajectory Results to Quantum-Mechanical Perturbation-Theory.** J Chem Phys 78(6):3959–3963, 1983.

Nordblad P. et al., **Critical Behavior of Induced Linear Birefringence in Isotropic Anti-Ferromagnets.** J Magn Magn 31-4(FEB):1093–1094, 1983.

Olego D., Pinczuk A., Ballman A. A., **Resonant Inelastic Light-Scattering by Free-Electrons in InP.** Sol St Comm 45(11):941–944, 1983.

Patterson G. D., Carroll P. J., Pearson D. S., **Rayleigh-Brillouin Spectroscopy of Normal-Tetracosane—Are There Transitions in the Melt?** J Chem S F2 79(P5):677–685, 1983.

Patterson G. D., Carroll P. J., Stevens J. R., **Photon-Correlation Spectroscopy of Poly(Methyl Methacrylate) Near the Glass-Transition.** J Pol Sc Pp 21(4):613–623, 1983.

Patterson G. D., Carroll P. J., Stevens J. R., **Photon-Correlation Spectroscopy of Polystyrene as a Function of Temperature and Pressure.** J Pol Sc Pp 21(4):605–611, 1983.

Pfeiffer L., Kovacs T., Celler G. K., Gibson J. M., Lines M. E., **Mössbauer-Effect of $^{73}$Ge in Laser-Processed Si and Ge Crystals.** Phys Rev B 27(7):4018–4026, 1983.

Phillips J. C., **Separable Spectral Model of the Metal-Insulator Transition in Si:P.** Phil Mag B 47(4):407–418, 1983.

Pinczuk A., Worlock J. M., **Light-Scattering by Two-Dimensional Electron-Systems.** Physica B&C 117(MAR):637–642, 1983.

Rabinovich E. M., **On the Structural Role of Fluorine in Silicate-Glasses (Letter).** Phys C Glas 24(2):54–56, 1983.

Raghavachari K., **Theoretical-Study of Conformational Processes in Sulfur Diimides.** J Phys Chem 87(8):1308–1312, 1983.

Raghavachari K., Haddon R. C., Roth H. D., **Theoretical Studies in the Norbornadiene Quadricyclane System.** J Am Chem S 105(10):3110–3114, 1983.

Raghavan P., Raghavan R. S., **Nuclear g Factors of the $17^-/2$ and $21^+/2$ Isomeric States of $^{91}$Mo.** Phys Rev C 27(4):1532–1535, 1983.

Reichmanis E., Gooden R., Wilkins C. W., Schonhorn H., **A Study of the Photochemical Response of Ortho-Nitrobenzyl Cholate Derivatives in P(MMA-MAA) Matrices.** J Pol Sc Pc 21(4):1075–1083, 1983.

Reimer J. A., Duncan T. M., **Local Bonding Configuration of Phosphorus in Doped and Compensated Amorphous Hydrogenated Silicon.** Phys Rev B 27(8):4895–4901, 1983.

Robinson I. K., **Direct Determination of the Au(110) Reconstructed Surface by X-Ray-Diffraction.** Phys Rev L 50(15):1145–1148, 1983.

Rosamilia J. M., Miller B., **Amplitude Enhancement and Automatic Scanning in Electroanalysis With Hydrodynamically Modulated Rotating-Disk Electrodes (Review).** Analyt Chem 55(7):1142–1145, 1983.

Roth H.D., Schilling M. L., **Radical Cations of Strained Ring Hydrocarbons-Cyclopropane Derivatives.** Can J Chem 61(5):1027–1035, 1983.

Rowe J. E., Pearsall T. P., Logan R. A., **A Study of the Valence Band-Structure of $Ga_{0.47}In_{0.53}As$ by Angle Resolved Photoemission.** Physica B&C 117(MAR):347–349, 1983.

Scalco E., Moore W. F., **A New Radiolytically Crosslinkable Polyvinyl-Chloride Insulation For Telecommunications Wire.** Radiat Ph C 21(4):389–396, 1983.

Schilling F. C., Bovey F. A., Tseng S., Woodward A. E., **C13NMR of Partially Epoxidized 1,4-*Trans*-Polybutadiene Crystals.** Macromolec 16(5):808–816, 1983.

Schwenk H., Andres K., Wudl F., Aharon-Shalom E., **Direct Observation of a Large London Penetration Depth in an Organic Superconductor.** Sol St Comm 45(8):767–769, 1983.

Shockley W., **Do Dislocations Hold Technological Promise?** Sol St Tech 26(1):75–78, 1983.

Simpson J. R., Mac Chesney J. B., **Optical Fibers with an $Al_2O_3$-Doped Silicate Core Composition.** Electr Lett 19(7):261–262, 1983.

Skocpol W. J., Jackel L. D., Howard R. E., Hu E. L., Fetter L. A., **Non-Metallic Localization and Interaction in One-Dimensional (0.1-μm) Si MOSFETs.** Physica B&C 117(MAR):667–669, 1983.

Solomon P. M., Stark A. A., Sanders D. B., **CO Emission in the Outer Galaxy Between Longitudes 50 Degrees and 72 Degrees.** Astrophys J 267(1):L29-L35, 1983.

Soto L., Franey J. P., Graedel T. E., Kammlott G. W., **On the Corrosion-Resistance of Certain Ancient Chinese Bronze Artifacts.** Corros Sci 23(3):241–250, 1983.

Spalink J. D. et al., **Bathorhodopsin Intermediates From 11-CIS-Rhodopsin and 9-CIS-Rhodopsin.** P Nas Biol 80(7):1887–1891, 1983.

Starnes W. H., Schilling F. C., Plitz I. M., Cais R. E., Freed D. J., Hartless R. L., Bovey F. A., **Branch Structures in Polyvinyl-Chloride and the Mechanism of Chain Transfer to Monomer During Vinyl-Chloride Polymerization.** Macromolec 16(5):790–807, 1983.

Stillinger F. H., **Variational Model for Micelle Structure.** J Chem Phys 78(7):4654–4661, 1983.

Stillinger F. H., Weber T. A., **Two-Dimensional Gaussian Core Model at High-Temperature.** Phys Rev A 27(5):2642–2648, 1983.

Stormer H. L., Chang A., Tsui D. C., Hwang J. C. M., Gossard A. C., Wiegmann W., **Fractional Quantization of the Hall-Effect.** Phys Rev L 50(24):1953–1956, 1983.

Stormer H. L., Tsui D. C., Gossard A. C., Hwang J. C. M., **Observation of Quantized Hall-Effect and Vanishing Resistance at Fractional Landau-Level Occupation.** Physica B&C 117(Mar):688–690, 1983.

Stormer H. L., Tsui, D. C., **The Quantized Hall-Effect.** Science 220(4603): 1241–1246, 1983.

Svensson C. et al., **Structural Temperature-Dependence in Alpha-Lithium Iodate—Neutron and X-Ray Study Between 20-K and 500-K.** J Chem Phys 78(12):7343–7352, 1983.

Swaminathan V., Anthony P. J., Pawlik J. R., Tsang W. T., **Temperature and Excitation Dependences of Active Layer Photo-Luminescence in (Al, Ga)As Laser Heterostructures.** J Appl Phys 54(5):2623–2629, 1983.

Tan S. L., Anderson P. W., **Long-Range Vanderwaals Forces Between Restricted-Dimensional Metals.** Chem P Lett 97(1):23–25, 1983.

Taniuti T., Hasegawa A., **Non-Linear Drift Waves.** Phys Scr T2(NSI):529–533, 1982.

Temkin H., Alavi K., Wagner W. R., Pearsall T. P., Cho A. Y., **1.5–1.6-μm**

**$Ga_{0.47}In_{0.53}As/Al_{0.48}In_{0.52}As$ Multiquantum Well Lasers Grown by Molecular-Beam Epitaxy.** Appl Phys L 42(10):845–847, 1983.

Thomas G. A., **Experimental Tests of Localization in Semiconductors.** Physica B&C 117(Mar):81–83, 1983.

Thompson L. F., **Advances in Microlithography (Editorial).** Sol St Tech 26(5):85, 1983.

Thompson L. F., **An Introduction to Lithography (Review).** ACS Symp S 1983(219):1–13, 1983.

Thompson L. F., Bowden M. J., **Resist Processing (Review).** ACS Symp S 1983(219):161–214, 1983.

Thompson L. F., Bowden M. J., **The Lithographic Process—The Physics (Review).** ACS Symp S 1983(219):15–85, 1983.

Tonelli A. E., **Stereosequence-Dependent C-13 NMR Chemical-Shifts in Polystyrene.** Macromolec 16(4):604–607, 1983.

Tonelli A. E., Schilling F. C., Cais R. E., **C-13-NMR and F-19-NMR Chemical-Shifts and the Microstructures of Fluoropolymers (Review).** Adv Chem SE 1983(203):441–454, 1983.

Tsang W. T., Hartman R. L., **cw Electro-Optical Characteristics of Graded-Index Wave-Guide Separate-Confinement Heterostructure Lasers With Proton-Delineated Stripe.** Appl Phys L 42(7):551–553, 1983.

Tsang W. T., Olsson N. A., **New Large Optical Cavity Laser With Distributed Active Layers.** Appl Phys L 42(10):850–852, 1983.

Tsang W. T., Olsson N. A., Logan R. A., **High-Speed Direct Single-Frequency Modulation With Large Tuning Rate and Frequency Excursion in Cleaved-Coupled-Cavity Semiconductor-Lasers.** Appl Phys L 42(8):650–652, 1983.

Tsang W. T., Olsson N. A. Logan R. A., Linke R. A., **Demonstration of Multilevel Multichannel Optical Frequency-Shift Keying (FSK) With Cleaved-Coupled-Cavity ($C^3$) Semiconductor-Lasers.** Electr Lett 19(9):341–343, 1983.

Tsuda T., Yasuoka H., Kitaoka Y., DiSalvo F. J., **$^{51}V$ NMR-Study of the Phase-Transition in 1*T*-$VS_2$.** J Magn Magn 31–4(Feb):1101–1102, 1983.

Tung R. T., Gibson J. M., Poate J. M., **Growth of Single-Crystal Epitaxial Silicides on Silicon by the Use of Template Layers.** Appl Phys L 42(10):888–890, 1983.

Vansaarloos W., **Use of the Star-Triangle Transformation for the Application of Differential Real-Space Renormalization-Group Theory.** Phys Rev B 27(9):5678–5685, 1983.

Van Uitert L. G., **Chelate Compound Stability Constant Calculations on Metal Diketonate Complexes.** Polyhedron 2(4):285–289, 1983.

Van Uitert L. G., **Relations Between Normal Boiling-Point and Melting-Point for Simple Halides and Oxides.** J Am Ceram 66(5):380–382, 1983.

Wagner A., **Applications of Focused Ion-Beams to Microlithography.** Sol St Tech 26(5):97–103, 1983.

Walker L. R., Walstedt R. E., **Computer Studies of RKKY Spin-Glasses.** J Magn Magn 31–4(Feb):1289–1292, 1983.

Walsh W. M., **Magnetic-Resonance Studies of Itinerant Antiferromagnetism in Organic Superconductors.** J Magn Magn 31–4(Feb):1273–1276, 1983.

Warren W. W., Schönherr G., Hensel F., **Intervalence Excitations of a Mixed-Valence Salt: NMR and Optical Absorption in Molten $InCl_2$.** Chem P Lett 96(4):505–508, 1983.

Warren W. W., Sotier S., Brennert G. F., **Resolution of the Conductivity Dilemma in Liquid Solutions of Alkali-Metals in Alkali-Halides.** Phys Rev L 50(19):1505–1508, 1983.

Weeks J. D. et al., **Vanderwaals Theory of Melting in Two-Dimensions and Three-Dimensions.** J Chem Phys 78(6):4197–4205, 1983.

Wiesenfeld J. M., Stone J., **Chirp in Picosecond Film Lasers and Pulse-Compression by Linear Dispersion in Optical Fibers.** Optics Lett 8(5):262–264, 1983.

Wilson B. A., Hu P., Harbison J. P., Jedju T. M., **Subnanosecond Total-Light Decay and Spectra in *a*-Si:H.** Phys Rev L 50(19):1490–1493, 1983.

Wilson R. J., Hahn E. L., **Local-Field-Induced Multiple-Pulse Free-Induction Decay.** Phys Rev B 27(7):4129–4134, 1983.

Wilson R. J., Mills A. P., **Positron and Positronium Emission From Tungsten (111).** Phys Rev B 27(7):3949–3954, 1983.

Worlock J. M., Maciel A. C., Perry C. H., Tien Z. J., Aggarwal R. L., Gossard A. C., Wiegmann W., **Inelastic Light-Scattering by Two-Dimensional Electron-Systems in High Magnetic-Fields.** Lect N Phys 177:186–193, 1983.

Wright C. E., Meyer D. E., **Conditions for a Linear Speed-Accuracy Trade-Off in Aimed Movements.** Q J Exp P-A 35(May):279–296, 1983.

Wu C. C., Leventhal M., Sarazin C. L., Gull T. R., **High-Velocity Iron-Absorption Lines in Super-Nova Remnant-1006 (Letter).** Astrophys J 269(1):L 5-L 9, 1983.

Yoshioka D., **Hall Conductivity of Two-Dimensional Electrons in a Periodic Potential.** Phys Rev B 27(6):3637–3645, 1983.

Yoshioka D., Halperin B. I., Lee P. A., **Ground-State of Two-Dimensional Electrons in Strong Magnetic-Fields and 1/3 Quantized Hall-Effect.** Phys Rev L 50(16):1219–1222, 1983.

Yoshioka D., Lee P. A., **Ground-State Energy of a Two-Dimensional Charge-Density-Wave State in a Strong Magnetic-Field.** Phys Rev B 27(8):4986–4996, 1983.

Yurke B., Denker J. S., Johnson B. R., Bigelow N., Levy L. P., Lee D. M., Freed J. H., **NMR-Induced Recombination of Spin-Polarized Hydrogen.** Phys Rev L 50(15):1137–1140, 1983.

## SOCIAL/LIFE SCIENCES

Culligan N., Gelperin A., **One-Trial Associative Learning by an Isolated Molluscan CNS—Use of Different Chemoreceptors for Training and Testing.** Brain Res 266(2):319–327, 1983.

Hoch S. J., Tschirgi J. E., **Cue Redundancy and Extra Logical Inferences in a Deductive Reasoning Task.** Mem Cognit 11(2):200–209, 1983.

Julesz B., **Textons, Rapid Focal Attention Shifts, and Iconic Memory.** Behav Brain 6(1):25–27, 1983.

Lamola A. A., Flores J., Blumberg W. E., **Binding of Photobilirubin to Human-Serum Albumin—Estimate of the Affinity Constant.** Eur J Bioch 132(1):165–169, 1983.

Lund A. M. et al., **Frequency Judgment Accuracy as a Function of Age and School-Achievement (Learning-Disabled Versus Non-Learning-Disabled) Patterns.** J Exp C Psy 35(2):236–247, 1983.

Miller J., **Can Response Preparation Begin Before Stimulus-Recognition Finishes.** J Exp Psy P 9(2):161–182, 1983.

Ondrias M. R., Friedman J. M., Rousseau D. L., **Metastable Species of Hemoglobin—Room-Temperature Transients and Cryogenically Trapped Intermediates.** Science 220(4597):615–617, 1983.

Ondrias M. R., Rousseau D. L., Simon S. R., **Resonance Raman-Spectra of Photodissociated Carbonmonoxy Hemoglobin and Deoxy Hemoglobin at 10-K.** J Biol Chem 258(9):5638–5642, 1983.

Patel D. J., Ikuta S., Kozlowski S., Itakura K., **Sequence Dependence of Hydrogen-Exchange Kinetics in DNA Duplexes at the Individual Base Pair Level in Solution.** P Nas Biol 80(8):2184–2188, 1983.

Patel D. J., Kozlowski S. A., Ikuta S., Itakura K., Bhatt R., Hare D. R., **NMR-Studies of DNA Conformation and Dynamics in Solution.** Cold S Harb 47(P1):197–206, 1982.

Pattipati K. R., Kleinman D. L., Ephrath A. R., **A Dynamic Decision-Model of Human Task Selection Performance.** IEEE Syst M 13(2):145–166, 1983.

Radin D. I., **Experimental Attempts to Influence Pseudo-Random Number Sequences.** J Am S Psyc 76(4):359–374, 1982.

Shaw M. L., **Two-State Versus Continuous State Stimulus Representations—A Test Based on Attentional Constraints.** Perc Psych 33(4):338–354, 1983.

## SPEECH/ACOUSTICS

Allen J. et al., **Speech Recognition (Discussion).** ISSCC Diges 26:214–215, 1983.

Cordell R. R., **Phase Intermodulation Distortion Instrumentation and Measurements.** J Aud Eng S 31(3):114–124, 1983.

Cox R. V., Crochiere R. E., Johnston J. D., **Real-Time Implementation of Time Domain Harmonic Scaling of Speech for Rate Modification and Coding.** IEEE J Sol I 18(1):10–24, 1983.

Cox R. V., Crochiere R. E., Johnston J. D., **Real-Time Implementation of Time Domain Harmonic Scaling of Speech for Rate Modification and Coding.** IEEE Acoust 31(1):258–272, 1983.

Streeter L. A., MacDonald N. H., Apple W., Krauss R. M., Galotti K. M., **Acoustic and Perceptual Indicators of Emotional Stress.** J Acoust So 73(4):1354–1360, 1983.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 November 1983 Number 9, Part 1

# Calculation of Modes in an Optical Fiber Using the Finite Element Method and EISPACK

By T. A. LENAHAN*

(Manuscript received April 20, 1983)

This paper presents a method for computing the propagation modes of a circular optical fiber. Finite element analysis reduces Maxwell's equations to standard eigenvalue equations involving symmetric tridiagonal matrices. Routines from the Eigensystem Package (EISPACK) compute their eigenvalues and eigenvectors, and from these the waveforms, propagation constants, and delays (per unit length) of the modes are obtained. An extension allows loss of leaky modes to be calculated. Examples indicate that the method is reliable, economical, and comprehensive, applying to both single and multimode fibers.

## I. INTRODUCTION

This paper presents a method for calculating the propagation modes of a circular optical fiber. The modal quantities, essential for telecommunications, include the waveforms (which describe the radial distribution of propagating power), the propagation constants (which determine cutoff conditions), and the delays per unit length (which

*Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

determine pulse dispersion along the fiber). The technique combines the Finite Element Method (FEM) and routines from the Eigensystem Package (EISPACK)[1] to achieve an efficient calculation of these modal quantities for both single and multimode fibers.

The most popular approach to modal calculations for multimode fibers has been the Wentzel, Kramers, Brillouin (WKB) method, where Maxwell's equations are approximated by an easily integrated first-order differential equation.[2] WKB analysis provides a simple model for understanding optical transmission through a fiber[3] and has guided fiber design.[4]

For fibers with index of refraction profiles described by a power law, the WKB method is equivalent to geometric optics and to the model of a fiber with unlimited radial extension.[5] For such fibers the effect of the outer cladding is missed by the WKB and equivalent methods, and the bandwidth capability is often overestimated.[6]

Accuracy of the WKB method declines substantially with the number of propagating modes. For single-mode fibers the WKB method is not suitable, and instead, various analytic and numerical techniques have been used.

Analytic calculations have centered about the step-index profile and the infinitely extending parabolic profiles. Modal quantities have been expressed in terms of Bessel functions for the single step[7] and, also, the double step.[8] Parabolic profiles, analyzed by analogy with the harmonic oscillator,[9] have well-known expressions for their modal quantities. Coupled with perturbation analysis,[10] these results cover a broad range of profiles. But numerical approaches permit an even more comprehensive treatment.

Several numerical procedures have extended the analysis of the step-index profile. The solution in the core comes from a numerical integration; the solution in the cladding (where the index is assumed constant) is well known. Boundary conditions at the core-cladding interface link the two solutions and lead to a set of linear equations involving the propagation constant as a parameter. A search of the corresponding determinant for zeroes gives the propagation constants and, subsequently, the waveforms and delays.

In Ref. 11 this procedure is applied to Maxwell's first-order vector equations, and important results have been obtained for multimode[12,13] and single-mode[14,15] fibers. A similar scheme[16] deals with two coupled second-order differential equations equivalent to Maxwell's equations. The second-order scalar wave equation approximates Maxwell's equations by neglecting the relatively small gradient index terms. In Ref. 17 the basic procedure is applied to the scalar wave equation, and a variety of results have been obtained for single-mode fibers.[18,19]

In Ref. 20 the FEM is developed in terms of a variational principle

for the vector equations. Approximate solutions in the core and cladding are matched at the interface by performing a determinant search, as described before. The FEM has also been applied[21] to diverse single-mode waveguides by approximating the index profile by piecewise constant functions and then enforcing boundary conditions across the numerous interfaces. The result is a matrix eigenvalue equation in generalized form. Both of these FEM approaches seem limited to a small number of modes for economical operation.

The approach in this paper is characterized by a sequence of three steps. First, Maxwell's equations are transformed to ordinary differential equations in eigenvalue form. In one case, gradient index terms are neglected, and in the second, they are considered to first order so that their effect can be monitored. Next, finite element analysis, using the Galerkin technique,[22] reduces these differential equations to matrix equations in standard eigenvalue form. The matrices are symmetric and tridiagonal, and their positive eigenvalues correspond to the propagation modes. The routine BISECT in the EISPACK[1] library delivers the eigenvalues and TINVIT, also in EISPACK, delivers the corresponding eigenvectors. The eigenvalues give the propagation constants of the modes, the eigenvectors give the waveforms, and a combination of the two give the delays.

Like many other numerical techniques, this method can treat any uniform, circular fiber and meet usual standards of accuracy. Also, the effect of gradient index terms can be monitored. But by casting the equations in standard eigenvalue form, modern techniques of computational linear algebra (as used in EISPACK) can achieve substantial cost advantage over other numerical approaches. Typically, this method will process for a multimode fiber 25 modes per second on the *Cray-1** computer.

The calculation procedure is derived in the next section. Effects of material dispersion are incorporated into the analysis, and calculation of loss for leaky modes is also considered. Results are given in Section III for a variety of single and multimode examples. These results, as discussed in Section IV, illustrate the reliability, economy, and scope of the method.

## II. ANALYSIS

This section derives from Maxwell's equations an algorithm that computes the propagation modes of an optical fiber. The algorithm is straightforward and can be carried out on any computer that has access to the EISPACK[1] or similar routines.

* Registered service mark of Cray Research, Inc.

### 2.1 Reduction of Maxwell's equations

The fiber is assumed perfectly straight and circular, and uniform along its length. The cylindrical coordinate system $(r, \theta, z)$ is defined so that the $z$-axis coincides with the fiber axis. The index of refraction can then be expressed by the function $n(r)$, the index profile. The profile can be any bounded function in the core ($r \leq R_1$), but it is constant ($n_{cl}$) in the cladding. The geometry is shown in Fig. 1.

The permittivity is

$$\epsilon = \epsilon_o n^2(r), \tag{1}$$

where $\epsilon_o$ denotes the free-space permittivity. The permeability $\mu$ is assumed throughout to be $\mu_o$, the free-space value.

Maxwell's equations relate the electric field $E$ and the magnetic field $H$ by

$$\begin{aligned} \text{curl } E &= -i\omega\mu H \\ \text{curl } H &= i\omega\epsilon E, \end{aligned} \tag{2}$$

where $\omega$ denotes the frequency of excitation (in rad/s). Taking the curl of the second equation eliminates $E$ to give

$$\nabla^2 H + \nabla g \times (\nabla \times H) + k^2 H = 0, \tag{3}$$

where

$$g = \ln(k^2), \tag{4}$$

and the wave number $k$ has the forms

$$k = \omega(\mu\epsilon)^{1/2} = \omega n(\mu_o\epsilon_o)^{1/2} = \omega n/c = 2\pi n/\lambda, \tag{5}$$

with $n$ the index of refraction, $c$ the velocity of light in vacuum, and $\lambda$ the free-space wavelength of the excitation.

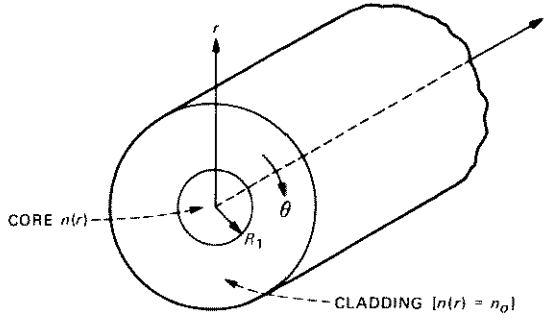



Fig. 1—Geometry of a uniform circular fiber with index profile $n(r)$.

Propagation modes are solutions of the form

$$H = H_o \exp(i\beta z), \tag{6}$$

where $H_o$ is a vector field independent of $z$, and $\beta$ is the propagation constant of the mode. Substituting this into the field equation gives, in cylindrical coordinates,

$$\begin{bmatrix} \Delta^2 - \frac{1}{r^2} + k^2 - \frac{g_r}{r}\frac{\partial}{\partial r} r & \left(\frac{2}{r^2} + \frac{g_r}{r}\right)\frac{\partial}{\partial \theta} & 0 \\ -\frac{2}{r^2}\frac{\partial}{\partial \theta} & \nabla^2 - \frac{1}{r^2} + k^2 & 0 \\ 0 & i\beta g_r & \nabla^2 - g_r\frac{\partial}{\partial r} + k^2 \end{bmatrix} \begin{pmatrix} H_{o\theta} \\ H_{or} \\ H_{oz} \end{pmatrix} = \beta^2 \begin{pmatrix} H_{o\theta} \\ H_{or} \\ H_{oz} \end{pmatrix}, \tag{7}$$

where $g_r$ means $\partial g/\partial r$. The transverse (i.e., $r$ and $\theta$) components of $H$ are uncoupled from the longitudinal (i.e., $z$) component and satisfy an eigenvalue equation with eigenvalue $\beta^2$. The corresponding operators are indicated as a $2 \times 2$ submatrix in the full $3 \times 3$ matrix.

The angular dependence of the transverse field is given by

$$\begin{pmatrix} H_{o\theta} \\ H_{or} \end{pmatrix} = \begin{pmatrix} h_\theta \cos\, m'\theta \\ h_r \sin\, m'\theta \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} h_\theta \sin\, m'\theta \\ h_r \cos\, m'\theta \end{pmatrix}$$

$$\text{for} \quad m' = 0, 1, 2, \cdots, \tag{8}$$

where the functions $h_\theta$ and $h_r$ depend only on $r$. The two forms correspond to different polarizations. Substituting these into eq. (7) gives

$$\begin{bmatrix} \left(\frac{1}{r}\frac{d}{dr} r \frac{d}{dr} - \frac{g_r}{r}\frac{d}{dr} r - \frac{m'^2 + 1}{r^2} + k^2\right) \pm m' \left(\frac{2}{r^2} + \frac{g_r}{r}\right) \\ \pm \frac{2m'}{r^2} \qquad \left(\frac{1}{r}\frac{d}{dr} r \frac{d}{dr} - \frac{m'^2 + 1}{r^2} + k^2\right) \end{bmatrix} \cdot \begin{pmatrix} h_\theta \\ h_r \end{pmatrix} = \beta^2 \begin{pmatrix} h_\theta \\ h_r \end{pmatrix}, \tag{9}$$

where the $\pm$ sign depends on the polarization.

The case where $m' = 0$ reduces to two uncoupled equations:

$$\left(\frac{k^2}{r}\frac{d}{dr}\frac{r}{k^2}\frac{d}{dr} - \frac{1}{r^2} - \frac{g_r}{r} + k^2\right) h_\theta = \beta^2 h_\theta \tag{10}$$

for the Transverse Magnetic (TM) modes for which $H_z$ is identically zero, and

$$\left(\frac{1}{r}\frac{d}{dr}\,r\,\frac{d}{dr} - \frac{1}{r^2} + k^2\right) h_r = \beta^2 h_r \tag{11}$$

for the Transverse Electric (TE) modes for which $E_z$ is identically zero. Often, gradient index terms (those involving $g_r$) are neglected on the basis that profile variations are relatively small.[4] The difference or splitting in $\beta^2$ for the TM and TE modes measures the accuracy of this practice.

When $m' \neq 0$, eq. (9) is expressed as

$$\begin{pmatrix} A & B \\ B & A \end{pmatrix} \begin{pmatrix} h_\theta \\ h_r \end{pmatrix} + \frac{1}{2}\begin{pmatrix} a & b \\ -b & -a \end{pmatrix} \begin{pmatrix} h_\theta \\ h_r \end{pmatrix} = \beta^2 \begin{pmatrix} h_\theta \\ h_r \end{pmatrix}, \tag{12}$$

where

$$A = \frac{1}{4}\frac{d}{dr}\,r\,\frac{d}{dr} - \frac{m'^2 + 1}{r^2} + k^2 + \frac{a}{2}, \qquad a = -\frac{g_r}{r}\frac{d}{dr}\,r$$

$$B = \pm\frac{2m'}{r^2} + \frac{b}{2}, \qquad b = \pm\frac{m' g_r}{r}. \tag{13}$$

The initial term has eigenvectors of the form,

$$\begin{pmatrix} f_1 \\ f_1 \end{pmatrix}, \tag{14}$$

where $(A + B)f_1 = \beta^2 f_1$, and

$$\begin{pmatrix} f_2 \\ -f_2 \end{pmatrix}, \tag{15}$$

where $(A - B)f_2 = \beta^2 f_2$. The second term of eq. (12) is neglected because its first order perturbation contribution is zero, as in general

$$\begin{pmatrix} a & b \\ -b & -a \end{pmatrix} \begin{pmatrix} f \\ \pm f \end{pmatrix} \text{ is orthogonal to } \begin{pmatrix} f \\ \pm f \end{pmatrix}, \tag{16}$$

respectively. If two eigenvalues are equal or nearly equal, a degenerate-perturbation calculation may be required.

To first order, the modes when $m' \neq 0$ are either the $EH_{m',n}$ with transverse $H$ fields of the form

$$H_{ot} = \begin{pmatrix} f \sin m'\theta \\ f \cos m'\theta \end{pmatrix} \text{ and } \begin{pmatrix} f \cos m'\theta \\ -f \sin m'\theta \end{pmatrix}, \tag{17}$$

where by eqs. (14) and (15) $f$ satisfies

$$\left(\frac{k}{r}\frac{d}{dr}\frac{r}{k}\frac{d}{dr} - \frac{(m'+1)^2}{r^2} - \frac{(m'+1)}{2r}g_r + k^2\right)f = \beta^2 f, \tag{18}$$

or the $HE_{m',n}$ with

$$H_{ot} = \begin{pmatrix} f\cos m'\theta \\ f\sin m'\theta \end{pmatrix} \text{ and } \begin{pmatrix} f\sin m'\theta \\ -f\cos m'\theta \end{pmatrix}, \tag{19}$$

where $f$ satisfies

$$\left(\frac{k}{r}\frac{d}{dr}\frac{r}{k}\frac{d}{dr} - \frac{(m'-1)^2}{r^2} + \frac{(m'-1)}{2r}g_r + k^2\right)f = \beta^2 f. \tag{20}$$

With $m = m' \pm 1$, eqs. (18) and (20) can be combined as

$$\left(\frac{k}{r}\frac{d}{dr}\frac{r}{k}\frac{d}{dr} - \frac{m^2}{r^2} \mp \frac{m}{2r}g_r + k^2\right)f = \beta^2 f. \tag{21}$$

The modes are indexed by the angular parameter $m$. For $m = 0$ there are two polarizations for each $HE_{1n}$ mode. This group includes the fundamental mode $HE_{11}$, which propagates in single-mode fibers. For $m = 1$ there are two polarizations for the $HE_{2n}$ modes, also the TM modes and the TE modes. For $m > 1$ there are two polarizations for the $HE_{m+1,n}$ modes and two for the $EH_{m-1,n}$ modes.

When gradient index terms are neglected, eq. (9) reduces to the scalar wave equation

$$\left(\frac{1}{r}\frac{d}{dr}r\frac{d}{dr} - \frac{m^2}{r^2} + k^2\right)f = \beta^2 f. \tag{22}$$

For $m = 0$ each solution represents two polarizations, i.e., two modes; for $m \neq 0$ each solution represents two polarizations and two angular harmonics, $m' = m \pm 1$, i.e., four modes.

The scalar wave equation and its counterparts [eqs. (10) and (21)] assume the form of a time-independent Schroedinger equation in two dimensions. In particular, the scalar wave equation can be expressed as

$$\left(\frac{1}{r}\frac{d}{dr}r\frac{d}{dr} - \frac{m^2}{r^2} + (k^2 - k_{cl}^2)\right)f = (\beta^2 - k_{cl}^2)f, \tag{23}$$

where

$$k_{cl} = 2\pi n_{cl}/\lambda. \tag{24}$$

The quantity $(k^2 - k_{cl}^2)$ plays the role of a potential that is 0 in the outer cladding and beyond; $(\beta^2 - k_{cl}^2)$ is an eigenvalue.

Results on Schroedinger's operators[23] imply that the propagation

modes correspond on a one-to-one basis to the positive eigenvalues and the number of such modes is finite (see Ref. 23, p. 366). This fact implies that increasingly accurate estimates of the propagation modes can be obtained by ever finer discretization of the equations. By contrast, finer discretizations introduce ever more negative eigenvalues corresponding to the continuum of radiation or unbound modes.

### *2.2 Finite element reduction*

The Finite Element Method (FEM) can solve to desired accuracy the differential equations just derived. The present discussion specializes to the scalar wave equation [eq. (23)]; modifications needed for the equations containing gradient index terms are indicated in the appendix.

The solution function $f(r)$ is approximated by a piecewise linear function. This can be expressed in terms of interpolation functions (shown in Fig. 2) as

$$f(r) \cong \sum_{l=0}^{L+1} f(r_l) N_l(r), \tag{25}$$

where $r_l = l\delta$ for $l = 0, 1, \cdots, L + 1$ are evenly spaced sample points.

The first and last terms of the series are affected by end conditions on $f(r)$. At the center ($r = 0$) of the fiber $f(r)$ and its radial derivative must be bounded and well defined; so $f(0) = 0$ when $m > 0$ and $df/dr$ $(0) = 0$ or equivalently $f(0) = f(\delta)$ to first order in $\delta$ when $m = 0$.

At the other end, $R = L\delta$ represents a truncation radius in the cladding, and $R + \delta$ represents the truncated part in a way that will now be explained. Assuming that the cladding extends indefinitely, the solution there is

$$f(r) = aK_m(\eta r), \tag{26}$$

where $K_m$ denotes the $m$th order modified Bessel function of the second kind,[24] $a$ is an unknown coefficient,

$$\eta = (\beta^2 - k_{c1}^2)^{1/2}, \tag{27}$$

and $m$ denotes the azimuthal mode number used in the scalar wave

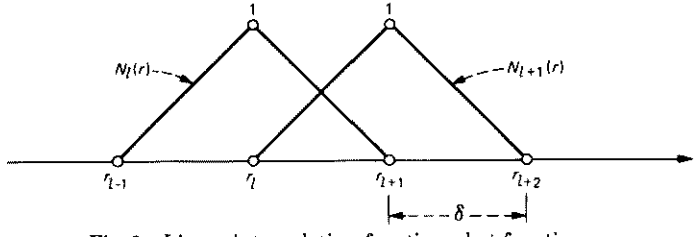



Fig. 2—Linear interpolation functions: hat functions.

equation [see eq. (23)]. It follows that the end condition for $f$ at truncation is

$$f'/f = \eta K_m'(\eta R)/K_m(\eta R). \tag{28}$$

A Taylor's expansion of $f$ to first order about $r = R$ yields

$$\begin{aligned} f(R + \delta) &= f(R) + \delta f'(R) \\ &= f(R)(1 + \delta\eta K_m'(\eta R)/K_m(\eta R)). \end{aligned} \tag{29}$$

Most modes of a multimode fiber are unaffected by the end condition in eq. (29) because their waveforms are negligibly small beyond the core. But the waveform of a single-mode fiber can extend beyond the core and then the end condition needs to be enforced.

Sample values of $f(r)$, the coefficients in eq. (25), are estimated by the Galerkin weighted residual method.[22] Although a piecewise linear approximation cannot satisfy the scalar wave equation, it can satisfy weighted averages of the equation. In the Galerkin technique the weightings are chosen as the basis functions $N_j(r)$ $j = 1, \cdots, L$. In terms of inner products, defined for any functions $p(r)$ and $q(r)$ as

$$(p, q) \equiv \int_o^{R+\delta} p(r)q(r)r\,dr, \tag{30}$$

Galerkin's technique yields

$$\begin{aligned} -\left(\frac{df}{dr}, \frac{dN_j}{dr}\right) - m^2\left(\frac{f}{r^2}, N_j\right) + \left((k^2 - k_{cl}^2)f, N_j\right) \\ = (\beta^2 - k_{cl}^2)(f, N_j) \end{aligned} \tag{31}$$

for $j = 1, \cdots, L$. The first term comes from an integration by parts.

Substituting the piecewise linear approximation of $f$ [from eq. (25)] into the Galerkin equations gives $L$ equations for $L + 2$ sample values, but the end conditions eliminate two of these values. The result is $L$ equations in $L$ unknowns, expressed as the matrix equation,

$$(A - m^2B + C)\mathbf{f} = \delta^2(\beta^2 - k_{cl}^2)D\mathbf{f}. \tag{32}$$

The column vector $\mathbf{f}$ denotes the sample values,

$$\mathbf{f} = [f(r_1), \cdots, f(r_L)]^t. \tag{33}$$

The $L \times L$ matrix $A$ has $jl$ element

$$a_{jl} = -\left(\frac{dN_l}{dr}, \frac{dN_j}{dr}\right)\delta^2 \tag{34}$$

for all $j$ and $l$ except, to accommodate the end conditions, $a_{10}$ must be added to $a_{11}$ when $m = 0$ and $a_{L,L+1}f_{L+1}/f_L$ must be added to $a_{LL}$ for all

$m$. The $L \times L$ matrices $B$, $C$, and $D$ have $jl$ elements

$$b_{jl} = \left(\frac{1}{r^2} N_l, N_j\right) \delta^2,$$

$$c_{jl} = ((k - k_{cl}^2)N_l, N_j)\delta^2, \tag{35}$$

and

$$d_{jl} = (N_l, N_j).$$

The latter three matrices are evaluated by "lumping the masses," meaning that the integrals are evaluated numerically and the integration points are chosen as the sample points.[25] The trapezoidal rule then yields 0 for the off-diagonal elements and for the main diagonal gives

$$b_{ll} = \delta^2/l \; c_{ll} = l[k^2(r_l) - k_{cl}^2]\delta^4 \; d_{ll} = l\delta^2 \tag{36}$$

for $l = 1, \cdots, L$. The matrix $A$ can be evaluated exactly. It is symmetric and tridiagonal (i.e., $a_{jl} = 0$ if $|l - j| > 1$) and has

$$a_{ll} = -2l\delta^2 \; a_{l,l+1} = (l + \tfrac{1}{2})\delta^2 \tag{37}$$

for $l = 1, \cdots, L$. When $m = 0$, $a_{11}$ is $-1.5\delta^2$.

Eq. (32) converts to a standard symmetric eigenvalue problem by multiplying both sides by the diagonal matrix $D^{-1/2}$ and putting

$$\mathbf{g} = D^{1/2}\mathbf{f}. \tag{38}$$

The result is

$$T\mathbf{g} \equiv D^{-1/2}(A - m^2B + C)D^{-1/2}\mathbf{g} = \delta^2(\beta^2 - k_{cl}^2)\mathbf{g}, \tag{39}$$

standard form relative to the vector $\mathbf{g}$.

The key matrix, $T$, is symmetric and tridiagonal. For $m \neq 0$

$$t_{ll} = -2 - \frac{m^2}{l^2} + \delta^2[k^2(r_l) - k_{cl}^2]$$

$$t_{l+1,l} = t_{l,l+1} = \frac{1}{2}\left[\left(\frac{l+1}{l}\right)^{1/2} + \left(\frac{l}{l+1}\right)^{1/2}\right] \tag{40}$$

for $l = 1, \cdots, L - 1$; for $m = 0$

$$t_{11} = -\tfrac{3}{2} + \delta^2[k^2(r_l) - k_{cl}^2]; \tag{41}$$

for all $m$

$$t_{LL} = -2 - \frac{m^2}{L^2} + \delta^2[k^2(r_L) - k_{cl}^2] + t_{L,L+1}g_{L+1}/g_L. \tag{42}$$

The end condition in eq. (29) combined with eq. (38) gives

$$g_{L+1}/g_L = \frac{L+1}{L}\,[1 + \delta\eta K_m'(\eta R)/K_m(\eta R)]. \tag{43}$$

Equation (39) represents the FEM reduction of the scalar wave equation. The other differential equations, as indicated in the appendix, reduce to the same form, with $T$ symmetric and tridiagonal. The sample distance $\delta$ is discussed in Section III.

### *2.3 Calculation of the propagation modes*

The propagation modes are calculated by solving for the positive eigenvalues ($\mu$) of $T$. The associated propagation constants are

$$\beta = (\mu/\delta^2 + k_{\text{cl}}^2)^{1/2} \tag{44}$$

with effective index of refraction

$$n_e = \beta/(2\pi/\lambda) = \beta(c/\omega). \tag{45}$$

When the end condition is enforced, the $T$ matrix depends on $\beta$ in its $(L, L)$ entry, but a simple iteration procedure yields the proper $\beta$. The associated waveforms are given by $\mathbf{f}$ or $\mathbf{g}$.

The modal delays per unit length ($\tau_g$) are the reciprocal of the group velocities,

$$\tau_g = \frac{d\beta}{d\omega}. \tag{46}$$

An efficient calculation of these uses the formula,

$$\frac{d\beta}{d\omega} = \left(\frac{\omega}{\beta}\right)\left(\frac{d\beta^2}{d\omega^2}\right) = \left(\frac{c}{n_e}\right)\left[\frac{dk_{\text{cl}}}{d\omega^2} + \frac{1}{\delta^2}\left(\frac{dT}{d\omega^2}\,\mathbf{g}\right)\cdot\mathbf{g}\right], \tag{47}$$

where $\mathbf{g}$ is assumed to be normalized (i.e., $\mathbf{g}\cdot\mathbf{g} = 1$). This follows standard procedure for taking the derivative of an eigenvalue with respect to a parameter.[26] Equations (40), (41), and (42) for $T$ imply that $dT/d\omega^2$ is a diagonal matrix. Using the equivalence between $\omega^2 d/d\omega^2$ and $-\lambda 2d/d\lambda^2$, eq. (47) becomes

$$\tau_g = \sum_{l=1}^{L}\left[n^2(r_l) - \lambda^2\,\frac{dn^2}{d\lambda^2}\,(r_l)\right] g_l^2/cn_e. \tag{48}$$

To form the $T$ matrix, the index profile $n(r)$ must be available for any excitation wavelength ($\lambda$). Sellmeier expansions[27] of the form

$$n^2 = 1 + \sum_{i=1}^{3} A_i/[1 - (l_i/\lambda)^2] \tag{49}$$

have been fitted to measured values of refractive index over the range of wavelengths 0.8 $\mu$m to 1.5 $\mu$m for bulk samples of pure $SiO_2$, 13.5-percent Ge doped $SiO_2$, and 1 percent F doped $SiO_2$ (denoted here as

$a$, $b$, and $c$, respectively). Also, results in Ref. 28 indicate that the index is approximately linear with concentration. Therefore, the index profile of an optical fiber having a graded Ge dopant is taken as

$$n(r, \lambda) = a(\lambda) + C_n(r)[b(\lambda) - a(\lambda)], \tag{50}$$

where $C_n(r)$ denotes the concentration profile for Ge, relative to 13.5 percent. For dual dopants, Ge and F, the index profile is

$$n(r, \lambda) = a(\lambda) + C_n(r)[b(\lambda) - a(\lambda)] + \tilde{C}_n(r)[c(\lambda) - a(\lambda)], \tag{51}$$

where $\tilde{C}_n(r)$ denotes the concentration profile for F, relative to 1 percent.

The concentration profiles in these equations may be specified or may be deduced from the index profile at a reference wavelength. Once the concentration profiles are available, the index profile and its $\lambda^2$ derivative can be determined from eq. (50) or eq. (51) for any wavelength.

The $T$ matrix can be expressed in dimensionless form by replacing the sample spacing by

$$\delta = R_1/L_1, \tag{52}$$

where $R_1$ denotes the core radius and $L_1$ the number of samples in the core. This gives

$$\delta^2[k^2(r) - k_{\mathrm{cl}}^2] = (2\pi R_1/\lambda)^2(n^2(r) - n_{\mathrm{cl}}^2)/L_1^2, \tag{53}$$

and from eq. (50), (with $a = n_{\mathrm{cl}}$),

$$n^2(r) - n_{\mathrm{cl}}^2 = C_n(n_1^2 - n_{\mathrm{cl}}^2) + (C_n^2 - C_n)(n_1 - n_{\mathrm{cl}})^2, \tag{54}$$

where now $C_n(r)$ represents the Ge concentration normalized with respect to $n_1$, the index at the center. The latter term in eq. (54) can usually be neglected because $n_1 \sim n_{\mathrm{cl}}$ and often, $C_n(r) \sim 1$ for most $r$. Then the $T$ matrix can be expressed in terms of the usual $V$ number,

$$V = (2\pi R_1/\lambda)(n_1^2 - n_{\mathrm{cl}}^2)^{1/2} = R_1(k_1^2 - k_{\mathrm{cl}}^2)^{1/2} \tag{55}$$

and the effective $V$ number,

$$V_e = (2\pi R_1/\lambda)(n_e^2 - n_{\mathrm{cl}}^2)^{1/2} = R_1(\beta^2 - k_{\mathrm{cl}}^2)^{1/2} \tag{56}$$

for the $(L, L)$ element. The FEM reduction becomes

$$T(V, V_e)\mathbf{g} = (V_e/L_1)^2\mathbf{g}, \tag{57}$$

where now an iteration on $V_e$ is required. A similar expression, involving three $V$ numbers, holds for dual dopants.

The effective index ($n_e$) is obtained from $V_e$ in eq. (56). The delay

is

$$\tau_g = \frac{\omega}{\beta}\frac{d\beta^2}{d\omega^2} = \frac{1}{R_1^2}\frac{dV_e 2}{dV^2}\frac{dV^2}{d\omega^2} + \frac{dk_{cl}^2}{d\omega^2}$$

$$= \frac{1}{cn_e}\left\{n_{cl}^2 - \lambda^2\frac{dn_{cl}^2}{d\lambda^2} + \frac{V_e}{V}\frac{dV_e}{V}\right.$$

$$\left.\cdot\left[(n_1^2 - n_{cl}^2) - \lambda^2\frac{d}{d\lambda^2}(n_1^2 - n_{cl}^2)\right]\right\}. \tag{58}$$

Matrix $T$ is always symmetric and tridiagonal. The routine BISECT in the EISPACK[1] library computes the positive eigenvalues for such matrices and TINVIT, also in EISPACK, computes the associated eigenvectors **g**. These routines operate efficiently and reliably.

### *2.4 Leaky modes*

When the index profile drops below $n_{cl}$ over part of its range, then leaky modes may exist. These modes have $n_e < n_{cl}$ and complex propagation constants. The corresponding attenuation accounts for radiation loss as the waveform spreads out radially. As is well known, leaky modes also can arise when $m \neq 0$ from the negative term, $-(m^2/r^2)$, in the propagation equation.

Leakage loss is calculated by truncating the waveform at the beginning of the outer cladding and enforcing the proper end condition. In terms of the complex propagation constant $\gamma$, eq. (27) for $\eta$ changes to

$$\eta = (-\gamma^2 - k_{cl}^2)^{1/2}, \tag{59}$$

and the end condition [in eq. (43)] becomes complex. Now, the $T$ matrix has a real ($T_r$) and an imaginary ($T_I$) part, but matrix $T_I$ consists of all zeroes except the ($L$, $L$) diagonal element (call it $x$). First-order perturbation theory estimates an eigenvalue of $T$ as

$$\mu = \mu_r + ixg_L^2 = \delta^2\eta^2, \tag{60}$$

where $\mu_r$ is an eigenvalue of $T_r$ and $g_L$ is the $L$th component of the associated normalized eigenvector **g**. Again, an iteration is required because $T$ depends on $\gamma$.

## III. EXAMPLES OF MODAL CALCULATIONS

In this section modal calculations based on the results of Section II are illustrated in 10 examples. The next section summarizes and evaluates the results.

### 3.1 Infinite parabolic profile

The parabolic profile that extends without radial limit has index

$$n(r) = n_1[1 - 2\Delta(r/R)^2]^{1/2} \tag{61}$$

for all $r$, where $R$ denotes the nominal radius of the fiber core. Under the scalar wave approximation, its modal waveforms are Laguerre Gaussian functions, its propagation constants are

$$\beta_{\mu m} = [k_{cl}^2 n_1^2 - 2Qk_{cl}n_1(2\Delta)^{1/2}/R]^{1/2}, \tag{62}$$

and its modal delays (neglecting material dispersion) are

$$\tau_{\mu m} = (\beta_{\mu m} + k_{cl}^2 n_1^2/\beta_{\mu m})/2\omega, \tag{63}$$

where the mode number is

$$Q = 2\mu + m + 1 \qquad m, \mu = 0, 1, \cdots, \tag{64}$$

with $m$ the angular harmonic and $\mu$ the radial harmonic. Each $Q$ represents a group of modes that have the same propagation constant and delay (see Ref. 9).

Modal delays were calculated assuming parameter values $\Delta = 0.013$, $R = 25$ $\mu$m, and $\lambda = 1.3$ $\mu$m. Convergence with respect to truncation radius (TR) was complete (within 0.1 ps/km) for each mode for $TR = 1.5R$. Convergence with respect to the number ($L_1$) of sample points in the core was within 1 ps/km in the rms delay for $L_1 = 400$, as indicated in Table I. An accuracy of 1 ps/km determines the bandwidth within 5 percent for a 10-GHz × km fiber and 0.5 percent for a 1-GHz × km fiber. The percent errors of the computed delays (with $L_1 = 500$) are shown in Fig. 3.

The most accurately computed mode within a mode group had radial number $\mu = 0$; their waveforms have no zero-crossings and are most easily approximated by piecewise linear functions. The least accurately computed mode (which was off by 5.5 ps/km) had the largest $\mu$. As another measure of accuracy, the spread in the computed delays for each $Q$ is also given in Table I.

To test the single-mode case, the parameters were changed to $\Delta =$

Table I—Convergence results for infinite parabolic profile in units of ps/km*

| $L_1$ | $\tau_{RMS}$ | Spread (1) | Spread (2) |
|---|---|---|---|
| 100 | 144.0 | 136.1 | 96.5 |
| 200 | 129.4 | 33.9 | 23.7 |
| 300 | 127.1 | 15.1 | 10.5 |
| 400 | 126.3 | 8.5 | 5.9 |
| 500 | 125.9 | 5.5 | 3.8 |

* ($\Delta = 0.013$, $R = 25$ $\mu$m, $\lambda = 1.3$ $\mu$m)

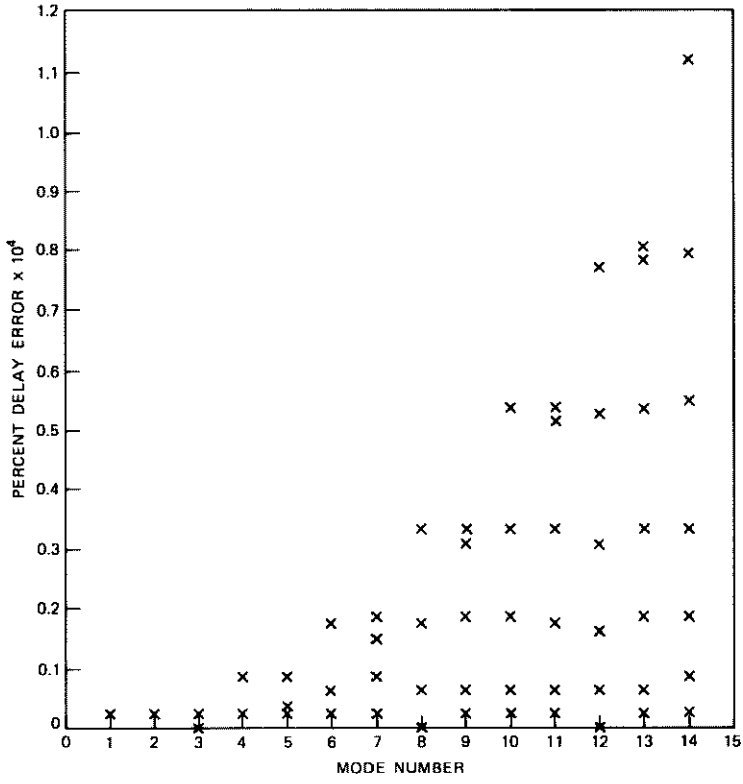



Fig. 3—Computation errors in modal delays for infinite parabolic profile.

0.005 and $R = 5$ $\mu$m, but again $\lambda = 1.3$ $\mu$m. The calculated effective index of the $HE_{11}$ mode was accurate to seven decimal places and the delay to 0.1 ps/km when $TR$ was $2.5R$ and $L_1$ was 300. This precision translates to nm accuracy in the zero-dispersion wavelength. The beam radius (BR) defined by the condition

$$f^2(BR) = f^2(0)/e \tag{65}$$

was accurate to four decimal places, as indicated in Table II. The number of samples in the core can be less in the single-mode case because the $HE_{11}$ waveform does not oscillate; the truncation radius needs to be greater because the waveform extends farther into the cladding.

### *3.2 Parabolic fiber*

Power-law fibers have the index profile

$$n(r) = \begin{cases} n_1(1 - 2\Delta(r/R)^\alpha)^{1/2} & r \le R \\ n_1(1 - 2\Delta)^{1/2} & r \ge R. \end{cases} \tag{66}$$

Assuming parameter values of $\Delta = 0.013$, $R = 25$ $\mu$m, $\lambda = 1.3$ $\mu$m, and $\alpha = 2$ (the parabolic case), effective index ($n_e$) and delay ($\tau_g$) were computed for each propagation mode under the scalar wave approximation. The rms delay ($\tau_{RMS}$) converged to 1.793 ns/km within 0.3 percent and the rms delay with the two highest-order-mode groups deleted ($\tau'_{RMS}$) converged to 113 ps/km within 0.9 percent when $L_1 = 300$ and $TR = 0.7R$.

Figure 4 shows $n_e$ and $\tau_g$ for each mode arranged by mode group.

Table II—Modal quantities for $HE_{11}$ mode of parabolic profile*

| $L_1$ | $TR$ | $n_e$ | $\tau_g$ ($\mu$s/km) | $BR$ ($\mu$m) |
|---|---|---|---|---|
| 300 | $3R/2$ | 1.4531412 | 4.8581500 | 2.6561930 |
| 300 | $2R$ | 1.4531608 | 4.8577058 | 2.6644067 |
| 400 | $2R$ | 1.4531608 | 4.8577059 | 2.6644135 |
| 300 | $5R/2$ | 1.4531609 | 4.8577024 | 2.6644397 |
| Actual Values | | 1.4531609 | 4.8577025 | 2.6643516 |

* ($\Delta = 0.013$, $R = 25$ $\mu$m, $\lambda = 1.3$ $\mu$m)

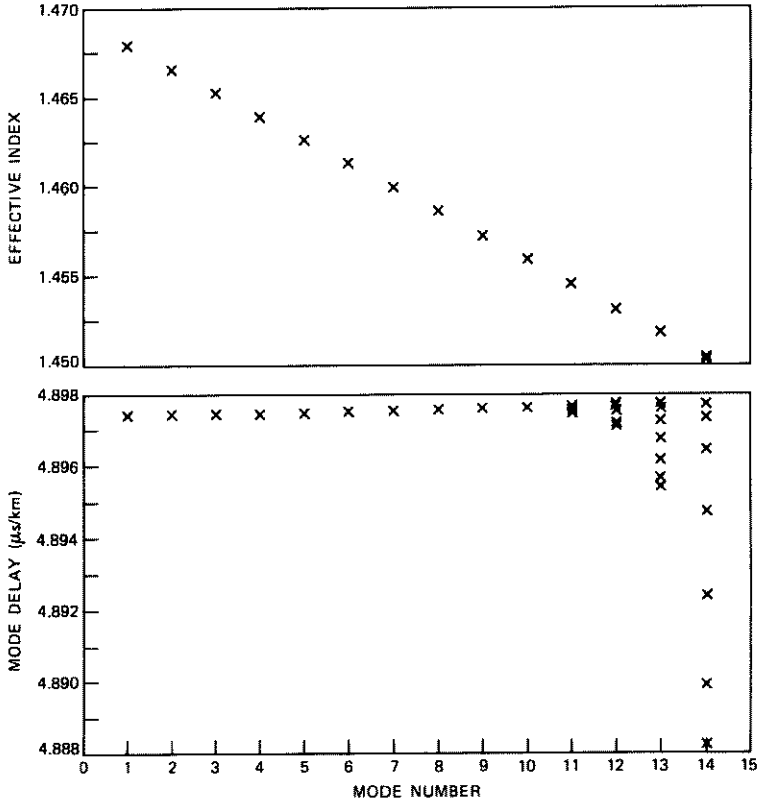



Fig. 4—Plots of $n_e$ and $\tau_g$ versus mode number for parabolic fiber.

The delays spread almost 10 ns/km for $Q = 14$ and 3 ns/km for $Q = 13$, but less than 1 ns/km for the others. The spread in $n_e$ is negligible, reaching a maximum of 0.01 percent.

Figure 5 shows $n_e$ and $\tau_g$ plotted jointly on an expanded delay scale with outliers excised. To compare with WKB results, $n_e$ and $\tau_g$ for the corresponding infinite parabola are indicated. The reference delays exceed all in their mode group: by as little as 0.2 ps/km for each of the first nine mode groups, but by more than 34 ps/km and 86 ps/km for $Q = 13$ and 14, respectively.

### *3.3 Contribution of gradient index terms*

Gradient index terms cause modal delays to spread still further. Figure 6 shows the delay differences caused by these terms for each mode of the parabolic fiber. Of the 112 modal delays only 14 differed by more than 20 ps/km from the corresponding scalar wave values. The TM mode nearest cutoff had the largest difference (33 ps/km).

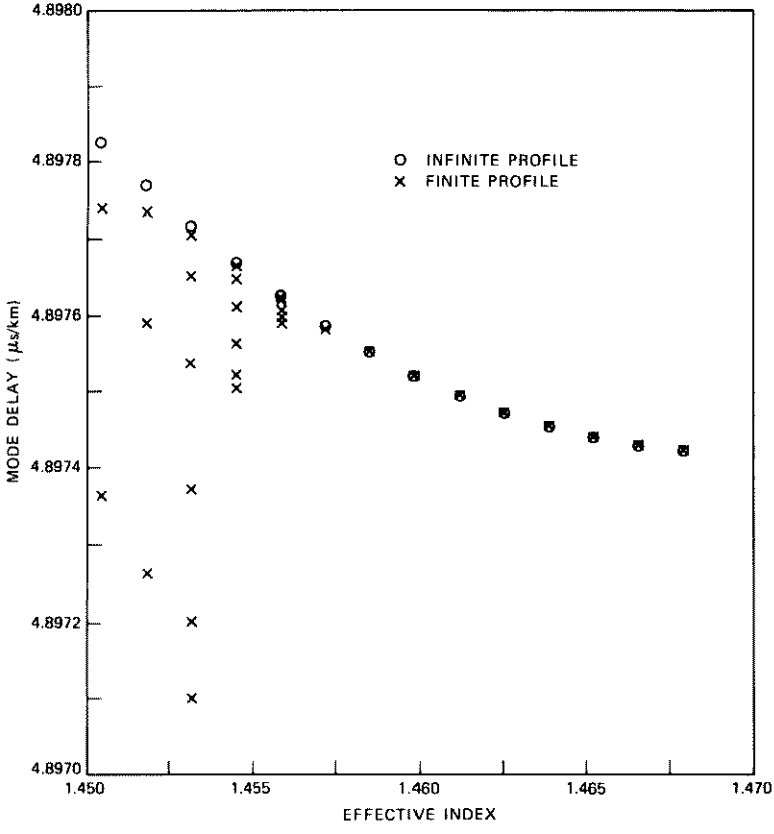



Fig. 5—Plots of $\tau_g$ versus $n_e$ for cladded and infinite parabolic profiles.

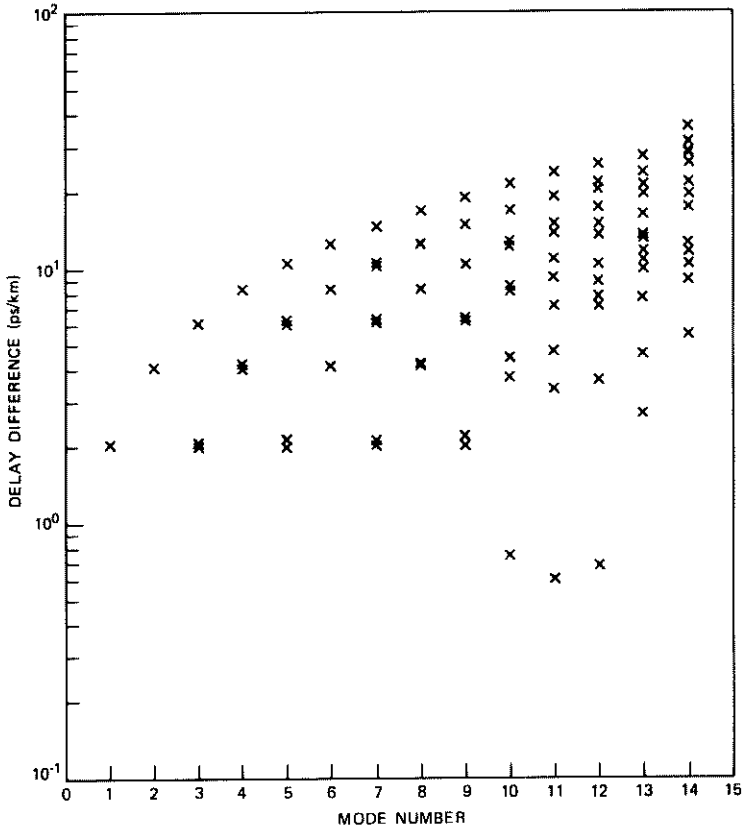



Fig. 6—Absolute differences in modal delays due to gradient index terms for parabolic fiber.

The change in $\tau_{RMS}$ was 5.9 ps/km and in $\tau'_{RMS}$ it was less than 1.5 ps/km. The error is less than 1.5 percent in either case. In the examples that follow gradient index terms are neglected.

### *3.4 Bandwidth of the parabolic fiber vs. λ*

Material dispersion causes the bandwidth of power-law fibers to depend sharply on the excitation wavelength ($\lambda$). Here and in the following examples, the profile is assumed known at the reference wavelength of $\lambda = 0.8$ $\mu$m; the concentration profile and the index profile at other wavelengths are determined, as discussed in the previous section.

Bandwidth is defined as the half-power frequency of the transfer function over some distance. When intramodal dispersion is neglected, the transfer function is

$$G(\omega) = \sum_n a_n \exp(i\omega\tau_n), \tag{67}$$

where $\tau_n$ denotes the delay and $a_n^2$ the power of the $n$th mode. Defining the rms delay ($\tau_{\mathrm{RMS}}$) by

$$\tau_{\mathrm{RMS}} = \left[\sum_n a_n(\tau_n - \tau_{\mathrm{av}})^2 \Big/ \sum_n a_n\right]^{1/2} \tag{68}$$

and the average delay by

$$\tau_{\mathrm{av}} = \sum_n a_n\tau_n \Big/ \sum_n a_n, \tag{69}$$

then to second order in the $\omega(\tau_n - \tau_{\mathrm{av}})$, the bandwidth is

$$BW = 1/(2\pi\tau_{\mathrm{RMS}}), \tag{70}$$

in units of GHz × km for delays in ns/km.

Bandwidth was calculated on this basis over a range of wavelengths for the parabolic fiber of example 2, assuming the tapered modal-power distribution shown in Fig. 7. Lower-order modes are weighted more heavily than the higher and the highest are omitted as the higher modes are increasingly vulnerable to microbending and cladding absorption.

Figure 8 shows the spectral plot of bandwidth. The peak value occurs at $\lambda \simeq 0.963$ $\mu$m, compared to 0.983 $\mu$m reported in Ref. 13.

The spectral plot is also shown for the case where material dispersion is neglected. The figure shows that the bandwidth is lower and essen-

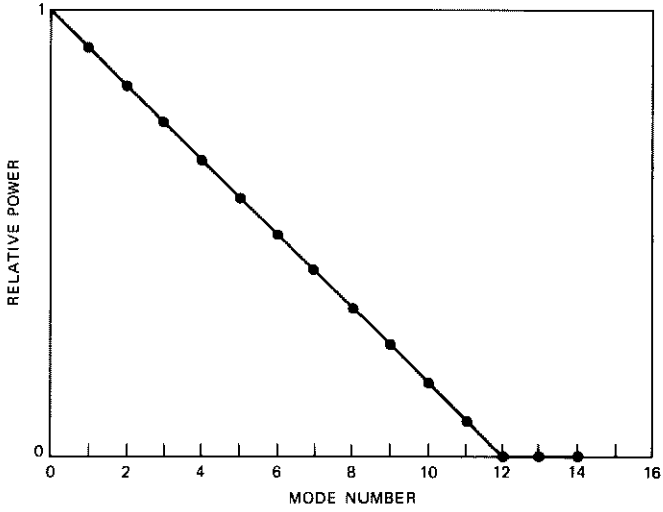



Fig. 7—A plot of the tapered modal-power distribution.

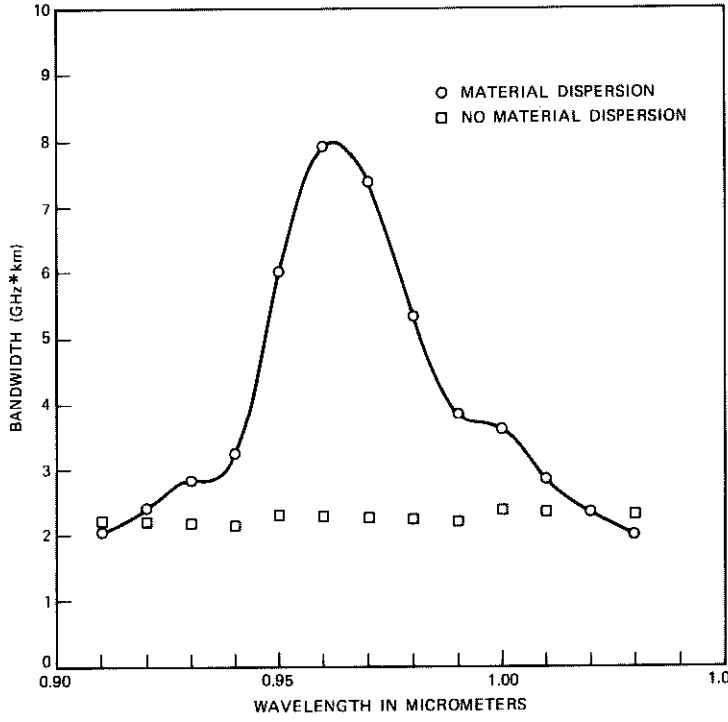



Fig. 8—Plots of bandwidth versus wavelength for parabolic fiber with and without material dispersion, using tapered modal-power distribution.

tially constant except for small discontinuities associated with new modes cutting in. Other calculation indicates that bandwidth increases to a peak at 9.5 GHz × km for $\alpha \simeq 1.98$ compared to an optimum $\alpha \simeq 1.97$ estimated by WKB analysis.[4]

### 3.5 Optimum $\alpha$ for power-law fibers

Material dispersion causes the optimum $\alpha$ of power-law fibers to depend on excitation wavelength. Figure 9 shows bandwidth versus $\alpha$ when $\Delta = 0.013$ and $R = 25$ $\mu$m for $\lambda = 0.82$ $\mu$m and $\lambda = 1.32$ $\mu$m. The optimum $\alpha$'s are 2.07 for the former and 1.88 for the latter, compared to values of 2.081 and 1.884, respectively, reported in Ref. 13. The peak bandwidths are 5.48 GHz × km and 6.02 GHz × km, respectively.

### 3.6 An alternate modal-power distribution

The fractional power that propagates in the cladding can be used to derive an alternate modal-power distribution. Modes with more than 0.1 percent of their power in the cladding will lose most of their power over 1 km for typical cladding losses of 1 dB/m. A realistic modal

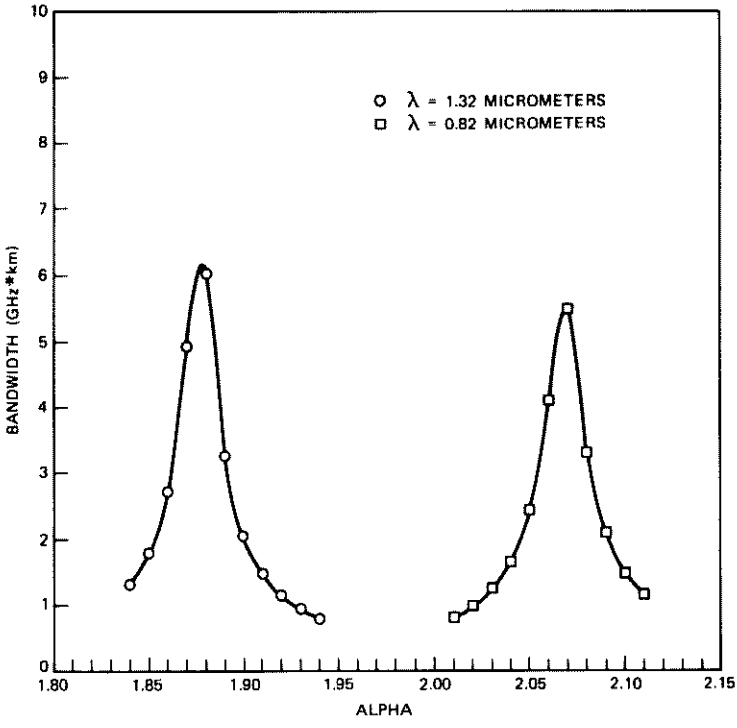



Fig. 9—Plots of bandwidth (in power-law fibers) versus $\alpha$ for $\lambda = .82$ $\mu$m and $\lambda = 1.32$ $\mu$m, using tapered modal-power distribution.

distribution would neglect these modes and for simplicity could weight the others equally.

Figure 10 shows bandwidth versus $\alpha$ for the two cases of example 5, assuming the new distribution. The optimum $\alpha$ is lower by 0.01 and the peak bandwidth is increased by 40 to 50 percent in either case. Table III indicates the modes neglected in the bandwidth calculation when $\lambda = 1.32$ $\mu$m and $\alpha = 1.88$.

### *3.7 Layer structure*

Layer structure in actual fibers perturbs the ideal profile. In this example power-law profiles are approximated by steps that span equal areas and match the ideal profile at the mid-area points of the layers (see Fig. 11). Bandwidth at $\lambda = 1.32$ $\mu$m (using the modal distribution in example 6) is shown versus $\alpha$ and the number of steps (NS) in Fig. 12. The graphs indicate that the optimum $\alpha$ is essentially independent of NS, but the peak bandwidth and its sharpness decrease for smaller NS.

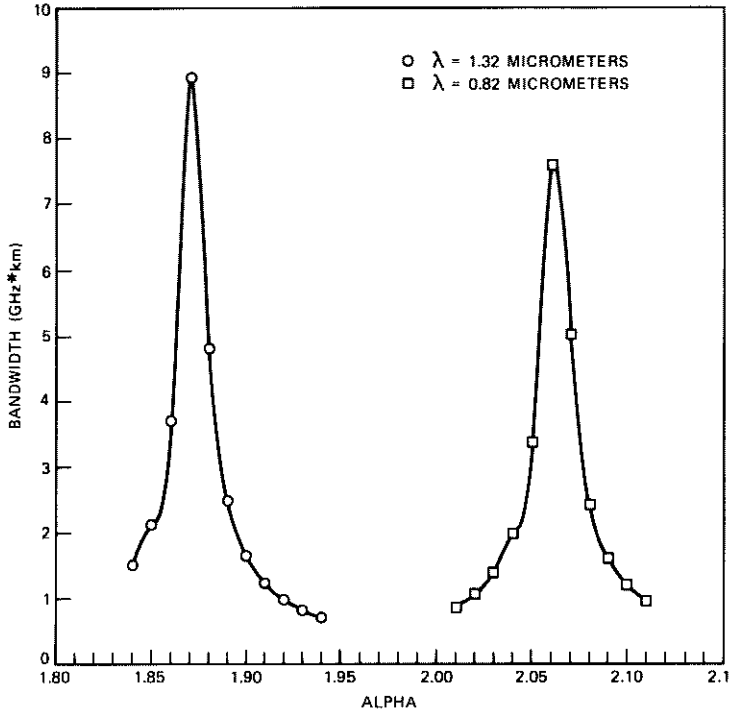



Fig. 10—Plots of bandwidth versus $\alpha$ for $\lambda = .82$ $\mu$m and $\lambda = 1.32$ $\mu$m assuming equal excitation of modes having less than 0.1 percent power in cladding.

### *3.8 Delay vs. λ for a single-mode fiber*

Delay (per km) of the $HE_{11}$ mode was computed versus $\lambda$ for an actual single-mode fiber based on its measured index profile. The profile measurement was performed on the associated preform and is shown in Fig. 13. The depression of the inner cladding is due to F doping.

As the calculation presumes radially symmetric profiles, the right and left sides were considered separately. About 750 sample points were used in either case.

The radial scale of the fiber profile was assumed to depend linearly on that of the preform. The fiber core radius ($R$) was estimated from the preform core radius ($RP$), fiber diameter ($DF$), and preform diameter ($DP$) as

$$R = 1.01(RP)(DF/DP), \tag{71}$$

where the 1-percent addition estimates the $SiO_2$ loss in the outer cladding during the draw process.

Table III—Modes with less than 0.1 percent of their power in cladding*

| | <0.1 Percent | | >0.1 Percent | |
|---|---|---|---|---|
| $Q$ | $m$ | $\mu$ | $m$ | $\mu$ |
| 14 | 1 | 6 | | |
| | 3 | 5 | | |
| 13 | 0 | 6 | 12 | 0 |
| | 2 | 5 | | |
| | 4 | 4 | | |
| | 6 | 3 | | |
| | 8 | 2 | | |
| | 10 | 1 | | |
| 12 | 1 | 5 | 11 | 0 |
| | 3 | 4 | | |
| | 5 | 3 | | |
| | 7 | 2 | | |
| | 9 | 1 | | |
| 11 | 0 | 5 | 8 | 1 |
| | 2 | 4 | 10 | 0 |
| | 4 | 3 | | |
| | 6 | 2 | | |
| 10 | 1 | 4 | 3 | 3 |
| | | | 5 | 2 |
| | | | 7 | 1 |
| | | | 9 | 0 |

* ($\Delta = 0.013$, $R = 25$ $\mu$m, $\lambda = 1.32$ $\mu$m, $\alpha = 1.88$)

Figure 14 shows computed differential delay versus $\lambda$ for the right and left profiles together with measurements. It was assumed that measurement matched calculation exactly at $\lambda = 1.32$ $\mu$m, because only relative delays were measured. The zero-dispersion wavelength (at the delay minimum) was computed as $\lambda_o = 1.3127$ $\mu$m for the left profile and $\lambda_o = 1.3113$ $\mu$m for the right, compared to measurement of $\lambda_o = 1.3114$ $\mu$m.

### *3.9 Leakage loss vs. $\lambda$ for a single-mode fiber*

Leakage loss (in dB/m) of the TE mode of the single-mode fiber in the preceding example was calculated for both sides of the profile as a function of $\lambda$ and is shown in Fig. 15. The difference between the left and right profiles becomes evident in this figure. A loss of 4 dB/m has been identified[29] with effective cutoff, and corresponds to cutoff wavelengths, $\lambda_c = 1.17$ $\mu$m for the left profile and $\lambda_c = 1.23$ $\mu$m for the right, compared to a measurement of 1.28 $\mu$m. This discrepancy is discussed in the next section.

The equivalent step approximation (where the index of the core and the inner cladding are area-weighted averages of the profile) gives $\lambda_o = 1.308$ $\mu$m and $\lambda_c = 1.21$ $\mu$m. The $\lambda_c$ value straddles the previous calculated values, but the $\lambda_o$ value is somewhat less.

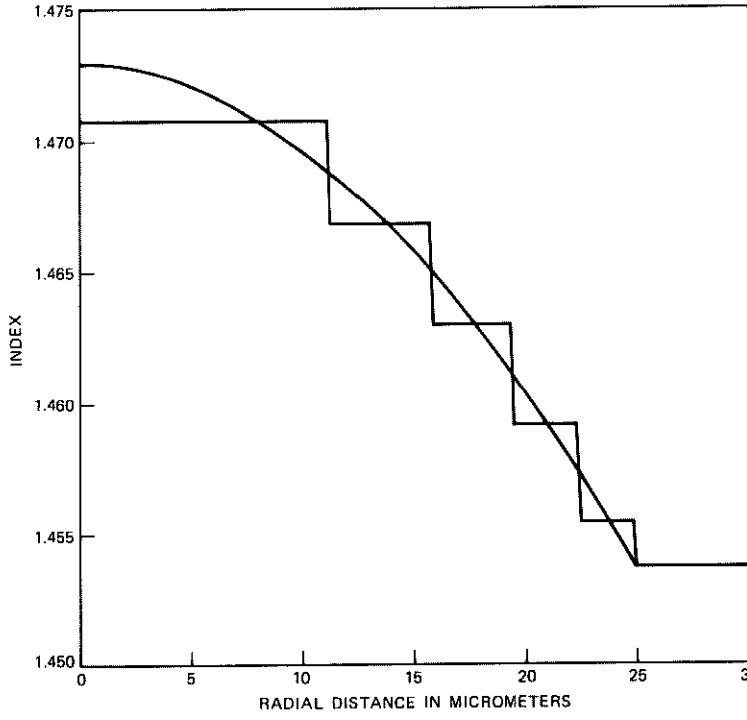



Fig. 11—A plot of a multilayered fit to an $\alpha$ profile (NS = 5, $\alpha = 1.90$).

### *3.10 Design curves for single-mode fibers*

The dimensionless formulation allows design curves to be generated efficiently. Figure 16 shows $V_e$ versus $V$ for the $HE_{11}$ mode of three power-law profiles. The specific parameters ($R$, $\Delta$, $\lambda$) determine $V$ and, hence, $V_e$ and $dV_e/dV$. Delay versus $\lambda$ is computed according to eq. (58) and $\lambda_o$ is estimated at the minimum of the delay curve. Figure 17 shows $\lambda_o$ versus $R$ for two values of $\Delta$ for triangular profiles ($\alpha = 1$).

## IV. SUMMARY AND CONCLUSIONS

A method for computing modes of a circular optical fiber has been presented. The finite element method reduces Maxwell's equations to the standard eigenvalue problem, involving tridiagonal matrices. Routines from EISPACK exploit the tridiagonal form to compute the eigenvalues and eigenvectors efficiently. From these the modal quantities are obtained.

Using a piecewise linear approximation of the waveform is necessary to get tridiagonal form. Although piecewise quadratic and cubic approximations of the waveform can lead to smaller matrices, tridiagonal

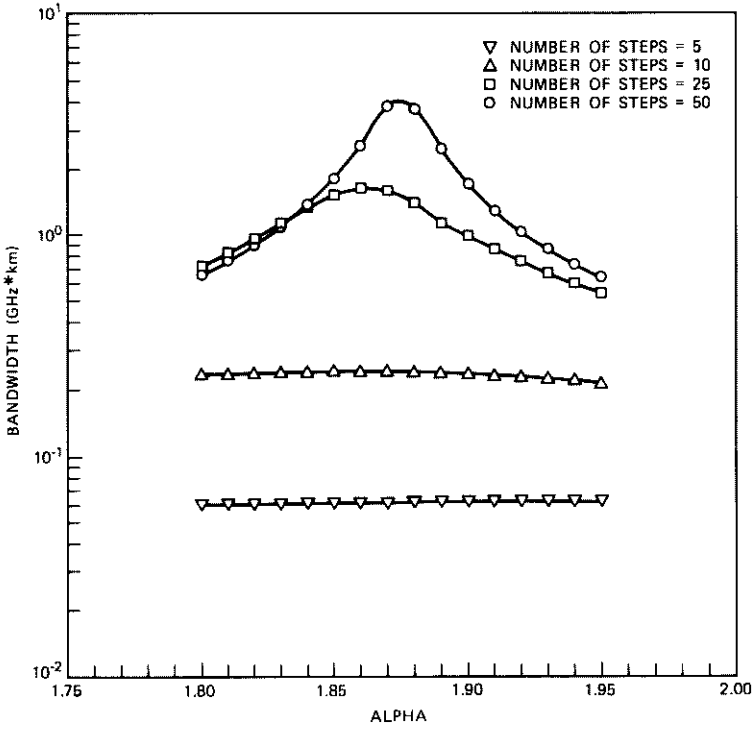



Fig. 12—Plots of bandwidth versus $\alpha$ at $\lambda = 1.32$ $\mu$m for four values of NS.

form is lost and the eigencalculations would be less efficient. Extension to elliptical and other nonradially symmetric fibers leads to similar difficulty in the eigen calculations.

The method applies to any circular fiber and can account for gradient index terms to first order. As illustrated in the 10 examples of Section III, the calculations can provide information on the radial distribution of propagating power, on pulse dispersion (arising from material, intermodal, and intramodal dispersion), and on leakage loss.

The accuracy of the method was tested for the infinite parabolic profile of the first example. After convergence was established, the modal quantities were compared with the actual values known through analysis. Agreement was excellent for both a multimode and a single-mode profile.

The WKB and scalar wave approximations were tested for the parabolic fiber in the next two examples. The cladding of the fiber altered the delays of the higher-order modes substantially, a facet missed by WKB analysis. The gradient index terms contributed less than 1.5 percent to the rms delay; and so, for most purposes the scalar wave approximation suffices for this fiber. The validity of these

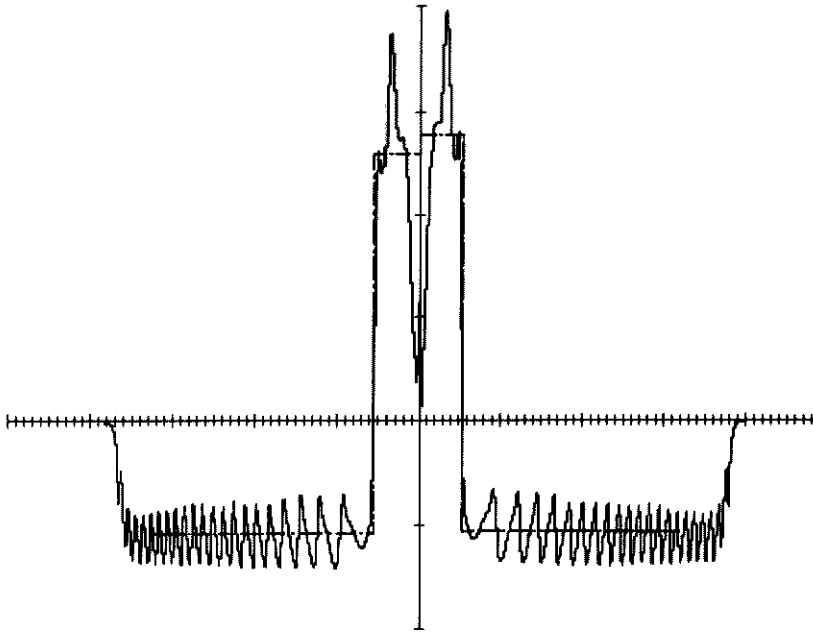


Fig. 13—The preform profile for a single-mode optical fiber.

approximations may change in other fibers; the value of the method is that it permits the test.

Material dispersion was incorporated into the calculation in the remaining examples. The variation of refractive index with wavelength and dopant concentration was modeled by Sellmeier expansions based on measurements of bulk samples with certain dopant concentration.[27] A linear dependence of index on concentration was assumed, but is not essential to the method. Irreversible thermal and stress effects[30] might also be incorporated.

Bandwidth was estimated in examples 4 and 5 for power-law fibers assuming a tapered modal-power distribution. With the higher-order mode groups deemphasized, results essentially agreed with WKB analysis. The usual sharp peak in bandwidth occurred in the spectral plot for the parabolic ($\alpha = 2$) fiber in example 4 and in the $\alpha$-dependence in example 5. Calculation of optimum $\alpha$ agreed reasonably well in these cases with other numerical work and WKB calculations.

The bandwidth calculation was repeated in example 6 using a different modal-power distribution. Modes with more than 0.1 percent of their power in the cladding were neglected, as being too lossy to maintain power. The optimum $\alpha$'s were essentially the same, but peak

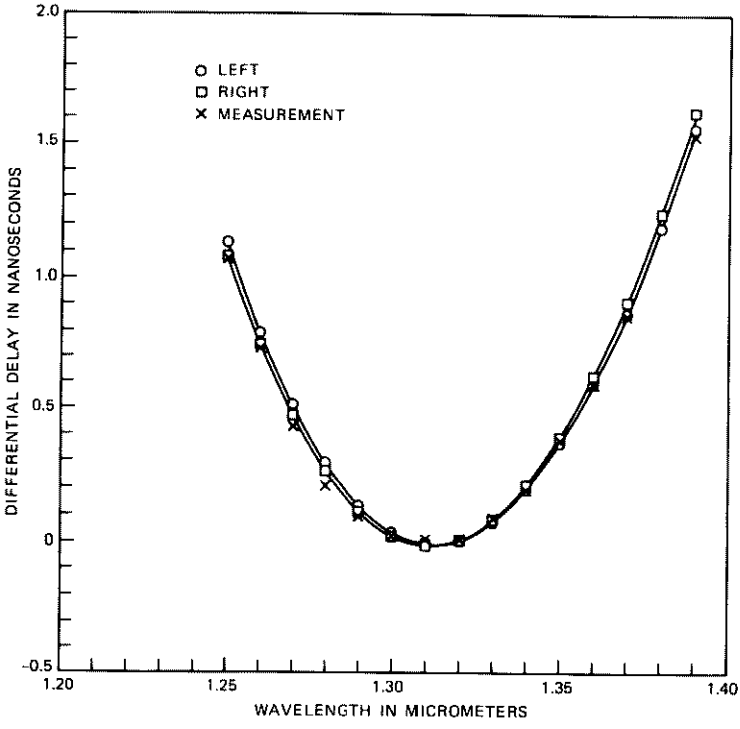



Fig. 14—Measurement and calculations for left and right profiles of differential delay versus wavelength.

bandwidths were somewhat higher. This example could be expanded to simulate differential mode attenuation, mode mixing, concatenation of dissimilar fibers, and other longitudinal variations.

Perturbations of $\alpha$ profiles usually lower bandwidth. Example 7 concerned the effect of layer structure on bandwidth and showed the expected decline with a smaller number of layers. Other perturbations such as ripple can also be treated. As a measure of its efficiency, the method used about two seconds per profile on the *Cray-1* for this example.

The last three examples concerned single-mode fibers. The first two of these involved an actual fiber whose profile was measured in the preform stage. Calculation of delay (per km) versus $\lambda$ matched measurement extremely well, but calculated leakage loss exceeded measurement, giving a 7-percent average underestimate of cutoff wavelength ($\lambda_c$). Other calculations of leakage involving only slightly depressed inner claddings, where $\lambda_c$ matches measurement to within 1 percent, suggest that the discrepancy may involve material changes in the F-doped glass caused by the draw process (consistent with obser-

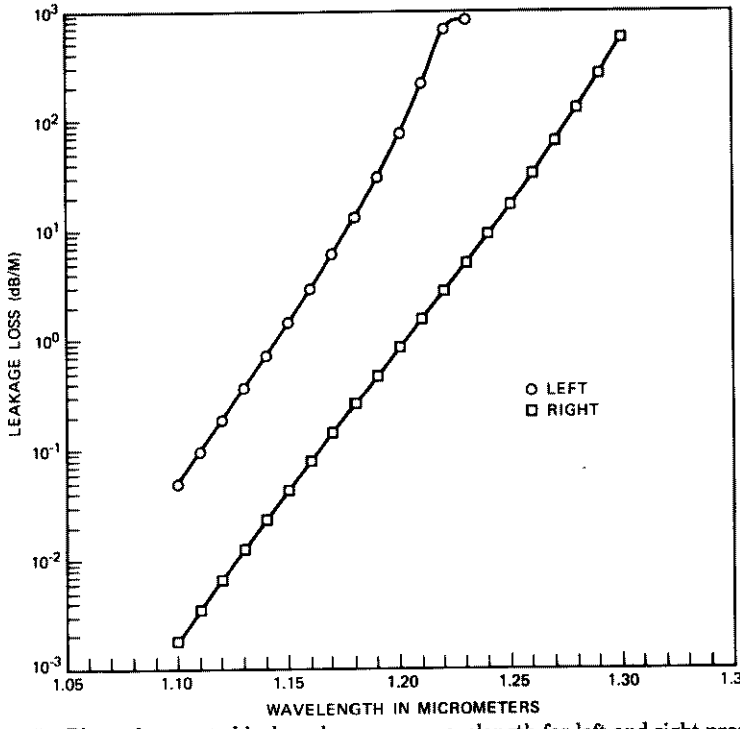



Fig. 15—Plots of computed leakage loss versus wavelength for left and right profiles.

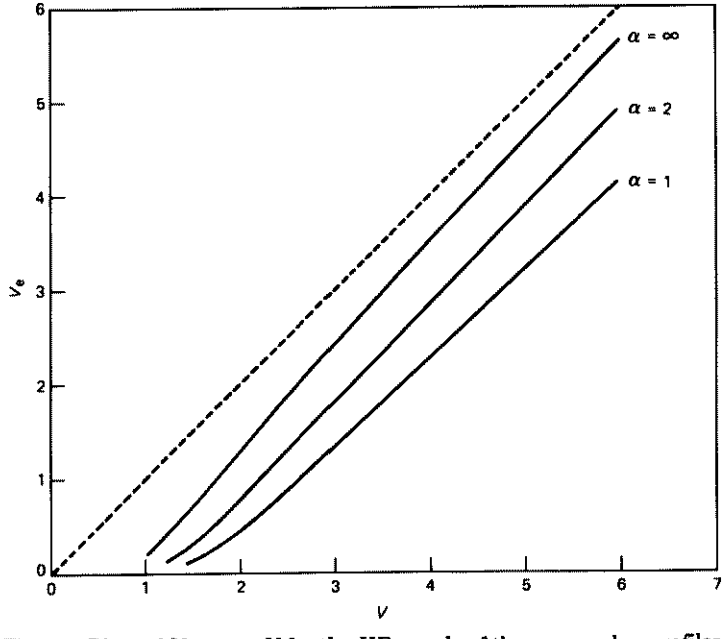



Fig. 16—Plots of $V_e$ versus $V$ for the $HE_{11}$ mode of three power-law profiles.

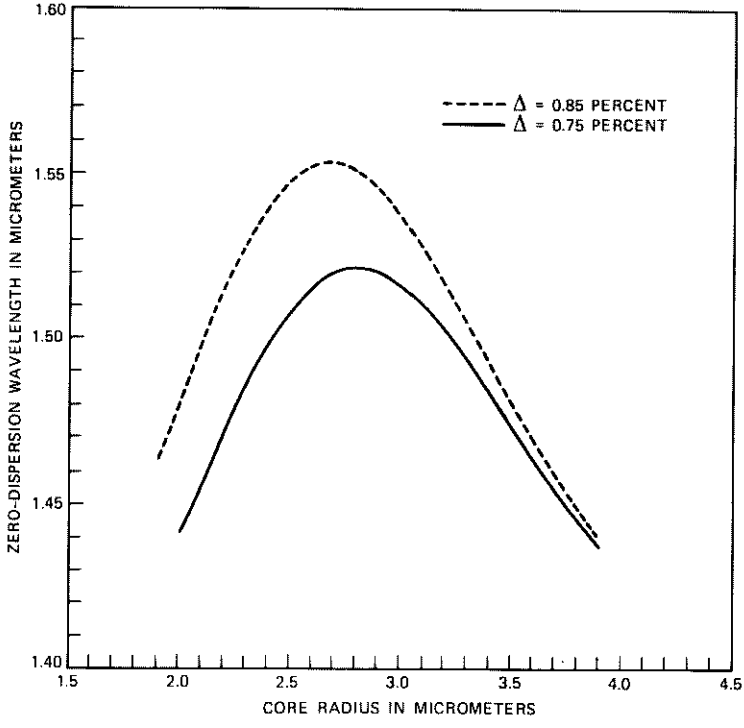



Fig. 17—Plots of computed zero-dispersion wavelength ($\lambda_o$) versus core radius ($R$) for two values of $\Delta$, for triangular profiles.

vations in Ref. 31). Such changes need to be understood when predicting transmission performance from a preform profile. Similar values for $\lambda_o$ and $\lambda_c$ were obtained for the equivalent step profile (with depressed inner cladding), but the lower $\lambda_o$ indicates the effect of profile structure.

The dimensionless formulation permits the greatest efficiency in getting design curves. Zero-dispersion wavelength ($\lambda_o$) for triangular profiles is calculated in example 10 showing the expected shift[32] to higher wavelengths. Spot size or cutoff wavelength can be obtained with similar efficiency.

In summary, the calculation method is reliable, and relatively inexpensive. In the context of circular fibers it is comprehensive, capable of simulating diverse effects.

## V. ACKNOWLEDGMENTS

The author is grateful for useful discussions with I. A. White and A. J. Ritger on multimode fibers, W. T. Anderson and P. F. Glodis on

single-mode fibers, D. S. Burnett on the FEM, and L. Kaufman on EISPACK.

The author is also grateful to J. S. Nobles and D. L. Philen for fiber measurements and to H. W. Friedrichsen for programming support.

## REFERENCES

1. B. T. Smith et al., *Matrix Eigensystem Routines—EISPACK Guide*, New York: Springer Verlag, 1976.
2. P. M. Morse and H. Feshback, *Methods of Theoretical Physics*, New York: McGraw-Hill, 1953, p. 1092.
3. J. E. Midwinter, *Optical Fibers for Transmission*, New York: Wiley, 1979, Chapter 6.
4. D. Gloge and E. A. J. Marcatili, "Multimode Theory of Graded-Core Fibers," B.S.T.J., *52*, No. 9 (November 1973), pp. 1563–78.
5. C. Pask, "Exact Expressions for Scalar Modal Eigenvalues and Group Delays in Power-Law Optical Fibers," J. Opt. Soc. Am., *69*, No. 11 (November 1979), pp. 1599–1603.
6. G. A. E. Crone and J. M. Arnold, "Anomalous Group Delay in Optical Fibers," Opt. Quant. Elect., *12*, No. 6 (November 1980), pp. 511–7.
7. E. Snitzer, "Cylindrical Dielectric Waveguide Modes," J. Opt. Soc. Am., *51*, No. 5 (May 1961), pp. 491–8.
8. S. Kawakami and S. Nishida, "Characteristics of a Doubly Clad Optical Fiber with a Low-Index Inner Cladding," IEEE Trans., *QE-11* (December 1974), pp. 879–87.
9. D. Marcuse, *Light Transmission Optics*, New York: Van Nostrand, 1972, pp. 267–72.
10. R. A. Sammut and A. K. Ghatak, "Perturbation Theory of Optical Fibers with Power-Law Core Profile," Opt. Quant. Elect., *10*, No. 6, (November 1978), pp. 475–82.
11. G. E. Peterson, A. Carnevale, U. C. Paek, and D. W. Berreman, "An Exact Numerical Solution to Maxwell's Equations for Lightguides," B.S.T.J., *59*, No. 7 (September 1980), pp. 1175–96.
12. G. E. Peterson, A. Carnevale, and U. C. Paek, "Comparison of Vector and Scalar Modes in a Lightguide with a Hyperbolic Secant Index Distribution," B.S.T.J., *59*, No. 9 (November 1980), pp. 1681–91.
13. G. E. Peterson, A. Carnevale, U. C. Paek, and J. W. Fleming, "Numerical Calculation of Optimum $\alpha$ for a Germania-Doped Silica Lightguide," B.S.T.J., *60*, No. 4 (April 1981), pp. 455–70.
14. U. C. Paek, G. E. Peterson, and A. Carnevale, "Dispersionless Single-Mode Lightguides with $\alpha$ Index Profiles," B.S.T.J., *60*, No. 5 (June 1981), pp. 583–98.
15. U. C. Paek, G. E. Peterson, and A. Carnevale, "Electromagnetic Fields, Field Confinement, and Energy Flow in Dispersionless Single-Mode Lightguides with Graded-Index Profiles," B.S.T.J., *60*, No. 8 (October 1981), pp. 1727–43.
16. Y. Kokubun and K. Iga, "Mode Analysis of Graded-Index Optical Fibers Using a Scalar Wave Equation Including Gradient-Index Terms and a Direct Numerical Integration," J. Opt. Soc. Am., *70*, No. 4 (April 1980), pp. 388–94.
17. W. L. Mammel and L. G. Cohen, "Numerical Prediction of Fiber Transmission Characteristics from Arbitrary Refractive-Index Profiles," Appl. Opt., *21*, No. 4 (February 1982), pp. 699–703.
18. S. J. Jang, L. G. Cohen, W. L. Mammel, and M. A. Saifi, "Experimental Verification of Ultra-Wide Bandwidth Spectra in Doubly-Clad Single-Mode Fiber," B.S.T.J., *61*, No. 3 (March 1982), pp. 385–90.
19. L. G. Cohen, W. L. Mammel, and S. Lumish, "Tailoring the Shapes of Dispersion Spectra to Control Bandwidths in Single-Mode Fibers," Optics Letters, *7*, No. 4 (April 1982), pp. 183–5.
20. K. Ikamoto and T. Okoshi, "Vectorial Wave Analysis of Inhomogeneous Optical Fibers Using Finite Element Method," IEEE Trans., *MTT-26* (February 1978), pp. 109–14.
21. C. Yeh, K. Ha, S. B. Dong, and W. P. Brown, "Single-Mode Optical Waveguides," Appl. Opt., *18*, No. 10 (May 1979), pp. 1490–504.
22. D. S. Burnett, *Finite Element Analysis, From Concept to Applications*, New York: Addison-Wesley, to be published.
23. M. Reed and B. Simon, *Methods of Modern Mathematical Physics IV: Analysis of Operators*, New York: Academic Press, 1978.

24. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Washington, D. C.: National Bureau of Standards, 1970, Chapter 9.
25. O. C. Zienkiewiez, *The Finite Element Method, 3rd Edition*, England: McGraw-Hill, 1977, p. 537.
26. T. Kato, *Perturbation Theory for Linear Operators*, New York: Springer-Verlag, 1966, p. 125 and p. 391.
27. J. W. Fleming, "Material Dispersion in Lightguide Glasses," Elect. Lett., *14*, No. 11 (May 1978), pp. 326–8.
28. S. E. Miller and A. G. Chynoweth, ed., *Optical Fiber Telecommunications*, New York: Academic Press, 1979, Chapter 7, p. 188.
29. W. T. Anderson, private communication.
30. P. L. Chu and T. Whitbread, "Measurement of Stresses in Optical Fiber and Preform," Appl. Opt., *21*, No. 23 (December 1, 1982), pp. 4241–5.
31. M. J. Saunders, "A Comparison of Single-Mode Refractive Index Profiles in Preforms and Fibers," 8th ECOC, paper C-17 (September 1982).
32. K. I. White, "Design Parameters for Dispersion-Shifted Triangular-Profile Single-Mode Fibers," Elect. Lett., *18*, No. 17 (August 19, 1982), pp. 725–7.

## APPENDIX

### *Gradient Index Terms*

Gradient index terms, involving the quantity $q_r = d/dr \ln k^2$, occur in eqs. (10) and (21). Although the index or its derivative may be discontinuous, they can be approximated by smooth functions, so $k^2$ and $g_r$ can be taken as well defined, continuous functions of $r$. This appendix concerns the changes needed in the T matrix to accommodate these terms.

Equation (10) converts to symmetric form when $h_\theta = kf$ and both sides are divided by $k$. The result is

$$\left(\frac{k}{r}\frac{d}{dr}\frac{r}{k^2}\frac{d}{dr}k - \frac{1}{r^2} - \frac{g_r}{r} + k^2\right)f = \beta^2 f. \tag{72}$$

Applying the Galerkin technique gives for the first term,

$$\begin{aligned} A' \sim a'_{jl} &= -\left[\frac{1}{k^2}\frac{d}{dr}(kN_l), \frac{d}{dr}(kN_j)\right]\delta^2 \\ &= \left[\left(\frac{dN_l}{dr}, \frac{dN_j}{dr}\right) + \frac{1}{4}(g_r^2 N_l, N_j)\right. \\ &\quad \left. + \frac{1}{2}\left(g_r \frac{dN_l}{dr}, N_j\right) + \frac{1}{2}\left(g_r N_l, \frac{dN_j}{dr}\right)\right]\delta^2 \end{aligned} \tag{73}$$

in place of matrix $A$ specified in eq. (34). The first quantity is $a_{jl}$ as before, the second adds to the $C$ matrix of eq. (35), and the last two when evaluated by the trapezoidal rule contribute only to the first side diagonals. The term $g_r/r$ is handled in the same way as $k^2$. Combining these contributions gives a new $T$ matrix that is symmetric and tridiagonal.

Likewise, eq. (21) converts to symmetric form when $k^{1/2}f$ replaces $f$.

The rest proceeds in the same way to give a symmetric, tridiagonal $T$ matrix.

Material dispersion is incorporated by expressing the index in terms of concentration functions as before. Radial derivatives needed for the gradient index terms themselves involve derivatives of the concentration function. The $\omega^2$ derivatives of these terms, needed for the modal delays, are found by differentiating the coefficients of the concentration functions.

## AUTHOR

**Terrence A. Lenahan,** S.B. and S.M. (Electrical Engineering) 1964, Massachusetts Institute of Technology; Ph.D. (Applied Mathematics) 1970, University of Pennsylvania; Bell Laboratories, 1970—. Mr. Lenahan has done studies of various areas of mathematical physics including electromagnetic propagation, elasticity, and fluid mechanics. He has recently been interested in the propagation of light in optical fibers. Member, American Mathematical Society, SIAM, Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

# Measurements of 800-MHz Radio Transmission Into Buildings With Metallic Walls

By D. C. COX,* R. R. MURRAY,* and A. W. NORRIS*

(Manuscript received January 5, 1983)

In this paper we describe an experiment conducted to measure 800-MHz attenuation into buildings. This information is needed for refining the configuration and design of portable radiotelephone systems that will accommodate low-power portable sets. Signal levels have been measured in and around three small buildings and a house, using an instrumentation van with an erectable 27-foot-high antenna. These buildings and the house all have metallic materials in their walls and thus are expected to exhibit high attenuation. The van was parked at one location with respect to the three buildings, and at nine different locations, at distances ranging from 400 feet to 1600 feet, from the house. We found that small-scale signal envelope variations are approximately Rayleigh distributed. For the house, large-scale distributions of the small-scale signal medians are approximately log-normal. Median signal levels outside the house decrease as $d^{-4.5}$, where $d$ is the distance from the van. Inside the house, levels decrease as $d^{-3.9}$ for first-floor locations and as $d^{-3.0}$ for second-floor locations. Average signal levels at 1000 feet, relative to free space, are −12.5 dB outside, −18.5 dB for the first floor, −16.5 dB for the second floor, and −28.9 dB for the basement. Other statistics of the signal levels and of attenuation into the house are also given in the paper. Cross-polarization couplings of −10 dB to 0 dB were measured in and around the three small buildings. The small-scale signal medians inside the three buildings range from 2 dB above to 24 dB below the averages of the signal medians outside buildings.

## I. INTRODUCTION

The attenuation of radio signals propagating into buildings has a significant effect on the performance of portable radiotelephone sys-

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

tems.[1] Frequencies in the 800-MHz to 900-MHz range are good candidates for such systems. Most earlier propagation measurements at these frequencies in the shadowing and multipath environment around buildings were made for mobile radio systems.[2] Ranges for the mobile radio data are generally greater than one mile. Battery power limitations on portable systems will likely restrict such systems to ranges on the order of 1000 feet. Earlier measurements made in buildings are limited in scope,[3–6] are oriented towards lower frequencies,[7–11] or are directed toward high-power portable sets with ranges greater than one mile.

One important portable radiotelephone environment comprises suburban residential areas characterized by discrete houses and other small buildings with densities ranging from less than one house per acre to a few houses per acre. Houses and small buildings with metallic materials in their walls are expected to exhibit high extreme values of attenuation.

In the residential environment, fixed radio terminals that communicate with portable sets could be placed at convenient locations outside buildings. These fixed terminals will be referred to as Portable Radiotelephone Terminals or PORTs.

We have implemented an experiment to provide 800-MHz attenuation information needed for refining the configuration and design of portable radiotelephone systems that will accommodate low-power portable sets. Measurements were made in and around three small buildings and a house. The buildings and the house all have metallic materials in their walls. An instrumentation van was parked at different locations to simulate different PORTs with distances ranging from 200 to 1600 feet. A 27-foot-high erectable antenna on the van simulated an unobtrusive PORT antenna. A portable signal source was moved in and around a building and signal levels were received and recorded in the van.

Section II of this paper describes the instruments used and how measurements were taken. Section III summarizes building attenuations and cross-polarization couplings measured in the three small buildings. Section IV contains statistical descriptions of the building attenuation for the house.

## II. THE MEASUREMENTS

### *2.1 Instrumentation*

#### *2.1.1 Signal source*

The portable signal source is a modified 815-MHz* handie-talkie.

* The actual frequency of the measurements is 815 MHz; however, the statistical results are not sensitive to small changes in frequency. Therefore, when frequency is referred to relative to the measurements, it will be rounded to 800 MHz.

The transmitting antenna is a half-wavelength coaxial sleeve dipole attached to the top of the hand-held unit. The dc power is provided by a self-contained nickel cadmium battery through a voltage regulator. The regulator minimizes output power drift due to normal battery discharge. The transmitter output is 0.8 watt. The output varies less than 0.3 dB and 700 Hz over continuous one-hour periods that include ambient temperature changes of 0°C to 25°C.

#### *2.1.2 Instrumentation van*

The instrumentation van (movable PORT) is a modified motor home, containing a 5-kw ac generator. An uninterruptable power supply isolates the instrumentation from generator voltage fluctuations that otherwise could affect measurement accuracy. The van is shown in Fig. 1.

Two 27-foot antenna masts are installed in pivoting mounts so they can be stored horizontally for transport and can be erected at the test site. They are 14.2 feet apart and are adjusted to vertical using built-in bubble levels. An 815-MHz collinear receiving antenna is mounted on one mast. A bracket on the other mast holds the 815-MHz signal source to provide a reference signal for calibrating the receiver. The centers of the two antennas are at the same height when erected.

#### *2.1.3 Measuring receiver*

The measuring receiver is an 815-MHz frequency-modulated (FM) communications receiver, modified to detect the received signal envelope. The receiving antenna is a collinear array (four dipole elements, 5.8-dB gain over dipole, 18-degree vertical beam width) mounted vertically at the top of the tilt-over mast on the instrumentation van. In the receiver modification, an 11.7-MHz intermediate frequency (IF) output is extracted before the limiter, down-converted to 13 kHz, bandpass-filtered ($BW_{3dB}$ = 8 KHz), and linearly detected. The modified receiver has a −123 dBm sensitivity for 0 dB output s/n from the linear envelope detector and a 45-dB measuring range between levels 6 dB above the noise level and 3 dB below saturation. The measuring range is linear within ±1 dB over a 35-dB range at high signal levels. The entire receiver is enclosed in a sealed brass box to provide radio frequency (RF) isolation. A variable RF attenuator reduces the input signal in 1-dB steps to prevent overloading of the receiver.

#### *2.1.4 Data acquisition*

The analog receiver output drives a 12-bit-resolution digital storage oscilloscope and an integral 5-1/4-inch flexible disc drive for data storage. The oscilloscope is set to record 2048 samples in a 20-second measurement period. Up to 16 tracks of 2048 samples each can be

Fig. 1—The instrumentation van with the receiving antenna mast and the reference signal mast erected. The center of the four-element collinear receiving antenna is 27 feet above ground at the top of the mast mounted on the right side of the van near the front. The signal source is at the top of its mast mounted on the left side near the rear. The center of the signal-source dipole is also 27 feet above ground.

recorded on each disc. Data for each parked position of the van, i.e., one PORT location, are stored on a separate disc.

The recorded data are transferred to a desktop computer. At the time of transfer the following steps are performed:

1. Hand-recorded log information is appended to the signal strength data. The log information includes such items as the address of the house, the position of the van, the path azimuth, the path length, and the receiver input attenuator setting.
2. The data are scaled to convert the recorded signal voltages to dB relative to 0 dB at the reference location 14.2 feet from the receiving antenna. The scaling takes into account recorded dc offsets, recorded reference levels, and the receiver input attenuator settings.
3. Medians and cumulative distributions of signal level are calculated from the scaled data.
4. The scaled data are stored on flexible discs within the computer for further analysis.

### *2.2 Procedure*

Van locations were selected using tax maps covering the immediate vicinity of the house. Points were chosen equally spaced in azimuth around the house for each of three radii at about 400, 800, and 1600 feet. Road layout and terrain irregularities influenced the final choices of vehicle placement. A single van location was used for the measurements in the other three small buildings.

Measurements were coordinated between the van and the measurement location over a 450-MHz voice link. The link comprises a 25-watt FM transceiver in the van and a 2-watt handie-talkie carried by the person making the measurements.

A typical procedure for measuring a building starts with the van parked at an appropriate position. The van location must be fairly level. If necessary, wooden ramps are put under wheels to aid in leveling. The portable transmitter is installed on its mast as a local reference source. The mast is erected and is plumbed to vertical. Similarly, the receiving antenna on the opposite side of the vehicle is erected and plumbed. Keeping both masts plumbed on the level van assures a fixed distance between the reference and receiving antennas for calibration purposes. The linear detector IF input is grounded, and the dc level is adjusted and recorded on a disc track. Next, the detector input is ungrounded and the receiver RF attenuator is adjusted so the RF reference level is within the receiver operating range. This level, which serves as a calibration reference at the fixed 14.2 foot distance, is then recorded.

The signal source is removed from the mast and taken to the building for signal level measurements. The unit is hand-held at arm's length,

for a scan height of 4.5 feet.[6] At a selected location within the building, a 20-second raster scan is made by moving the transmitter in a horizontal plane at 2.5 ft/s.[6] The 4-foot square scanned area consists of 12 parallel linear scans separated by 4-inch increments. During the scan period 2048 data points are taken at 100 samples/s.

During the scanning period, the oscilloscope is monitored to see that signal amplitudes are within the receiver operating range. If not, the RF input attenuation is adjusted and the scan is repeated. The remaining locations within the building and immediately outside are similarly scanned and recorded. After the measurements are made, the transmitter is reinstalled on the reference mast, erected and plumbed, and the dc level and RF signal reference level are again recorded. The closure error between beginning and ending reference-level recordings is usually <0.5 dB. If the closure is >1 dB, the measurements are repeated. Such high closure error occasionally occurs when the transmitter battery has discharged below the regulation limit of the voltage regulator.

### *2.3 Received signal characteristics and attenuation definitions*

Motion of the signal source through the 4-foot-square areas inside and outside of buildings results in small-scale signal variations. The variations are caused by multipath propagation.[2,6] Inside houses and in areas shadowed from the van, where propagation is dominated by reflection and scattering, the variations in the received signal envelope are approximately Rayleigh distributed (see Ref. 6 and Section IV of this paper). Received signal minima are separated on the order of one half wavelength.[6] The medians (or means) of these small-scale variations are approximately stationary over the small areas but the medians for areas in different rooms in a building can be significantly different. Thus, the signal statistics can be modeled as a combination of a small-scale, quasi-stationary process (multipath) superimposed on a large-scale process (shadowing). This model is like the models used for mobile radio propagation.[2,12]

Only a single parameter is needed to describe the small-scale Rayleigh-distributed signal variation. The median can be determined if somewhat over half of the samples are above the measurement threshold. The mean, however, will be biased by the receiver noise level unless significantly more than half the samples are above the threshold. Therefore, medians of received signal variations for the 4-ft-square areas are used to characterize the small-scale signal variations at different measurement locations. The lowest median signal measured was 44 dB above the receiver noise level for the data presented in Sections III and IV. The lowest median cross-polarized signal measured was 38 dB above the noise level.

Building attenuation for a location inside a building can be defined as the difference between the signal level at the location and a "representative" signal level outside the building. Both levels are taken to be small-scale medians, measured in decibels, relative to a common reference level.

There are at least two possibilities that could be used for the representative level outside. The first choice is to use the average level of the signal in which the building is immersed. This level can be approximated by taking the average of several median levels, in decibels, measured at different locations surrounding the building. This decibel averaging is appropriate for large-scale variations of the median that are log-normally distributed. Large-scale variations are shown in Section IV to be approximately log-normally distributed. This definition of representative level seems appropriate for system considerations because service would have to be provided all around the exterior of a building.

The second possible choice for a representative level is to use the level of the signal incident on the building from the PORT (or received at the PORT from the incident region, since reciprocity holds). The incident level can be determined readily when the PORT is in line-of-sight of the building and there is little multipath from surrounding buildings. This was the case for the measurements in Ref. 6. However, when there are many intervening buildings between the PORT and the subject building and there is considerable multipath from surrounding buildings, the signal level incident on the building is not easily defined. This second choice of representative level appears least appropriate at the longest distances where the shadowing and multipath are most significant. These are also the distances of most concern in system considerations.

In Section III, comparisons are made of attenuations determined using both choices for representative levels. However, the more appropriate average outside level is used for the statistics in Section IV.

## III. ATTENUATIONS AND CROSS-POLARIZATION COUPLINGS FOR THREE BUILDINGS

### *3.1 Building descriptions*

The first building, shown in Fig. 2, is 20 feet by 38 feet, of corrugated steel construction and mounted on a concrete platform. There are metal screened doors inside the main metal doors, and all of the nine windows are metal screened, with the exception of three in the rear. The building inside is a single large area and is about one-third filled with equipment.

The second building is shown in Fig. 3. It is 21.5 feet by 28 feet, is covered with aluminum siding and has one solid door and nine win-

Fig. 2—Corrugated steel building on Crawford Hill as seen from the instrumentation van position.

Fig. 3—Antenna Control building on Crawford Hill. The photograph was taken looking toward the building along the path from the van.

dows. All except the three largest windows are metal screened. The building interior is divided roughly in half on the long dimension and is about half filled with equipment.

The third building is of wooden construction, is 16 feet by 54 feet

Fig. 4—Wooden building on Crawford Hill. The propagation path from the van was into the corner to the left in the photograph.

and contains vapor barrier foil insulation in the walls. This building, shown in Fig. 4, was empty. In front it has two doors with unscreened windows and three other unscreened windows. The room with double doors to the right in Fig. 4 is attached to an exterior wall. Measurements were not made in the attached room.

### *3.2 Attenuations for the three buildings*

Signal levels were measured at four locations outside and two or three locations inside of the three buildings from a single van (PORT) location. These levels are summarized in Table I. The distances indicated in the table (i.e., 225 feet, 330 feet, and 845 feet) are the path length from the van antenna to the fronts of the buildings. The centers of the four outside locations for each building were about 5 to 10 feet away from the midpoints of the four sides of the building. The outside locations are numbered clockwise looking down on the building from above, starting with the side closest to the van. The small-scale median signal levels in decibels and the decibel average for the four outside locations are tabulated in the column labeled "Level/dB". These levels are relative to 0 dB at the signal level reference 14.2 feet from the van receiving antenna. The column labeled $1/r^2$/dB contains the signal levels that would occur in free space ($1/r^2$) at the distance of the measurement location. These levels are also relative to 0 dB at the 14.2 foot reference location. The $1/r^2$ level in the row labeled Avg. is the free space value at the building midpoint.

For the wooden building positioned at 330 feet from the van and for the building with metal siding positioned at 845 feet, the signal medians measured in front of the buildings are 1 to 2 dB greater than

Table I—Building attenuations for three buildings

| Building Material | Location Outside | Level (dB) | $1/r^2$ (dB) | Location Inside | Relative to Average Attenuation (dB) | Relative to Incident Attenuation (dB) |
|---|---|---|---|---|---|---|
| All metal (225 ft) | Avg. | −29.0 | −24.3 | | | |
| | 1 | −24.9 | −23.7 | 1 | 22.3 | 26.0 |
| | 2 | −27.8 | −24.1 | 2 | 18.7 | 22.3 |
| | 3 | −36.2 | −25.0 | 3 | 24.0 | 27.3 |
| | 4 | −27.1 | −24.1 | | | |
| Metal siding (845 ft) | Avg. | −45.8 | −35.7 | | | |
| | 1 | −34.3 | −35.6 | | | |
| | 2 | −43.2 | −35.5 | 1 | 9.0 | 20.4 |
| | 3 | −49.5 | −35.8 | 2 | 0.3 | 11.8 |
| | 4 | −56.2 | −35.8 | | | |
| Wood (330 ft) | Avg. | −31.3 | −27.5 | | | |
| | 1 | −24.8 | −26.8 | | | |
| | 2 | −40.6 | −27.7 | 1 | −1.8 | 3.9 |
| | 3 | −34.5 | −27.8 | 2 | 2.6 | 9.0 |
| | 4 | −25.4 | −27.2 | | | |

the free-space values. The signal median in front of the all-metal building positioned at 225 feet is about 1 dB below the free-space value. These values are all consistent with the effects of a single reflection from the relatively flat dry ground between the van and the buildings.[13] The 27-foot van antenna height and a reflection coefficient phase angle of nearly 180 degrees, appropriate for small angles between the incident wave and smooth dry ground, yield signal maxima at distances above the ground of about 3 feet for 330 feet from the van and of about 8 feet for 845 feet from the van. For the same conditions, a minimum occurs about 4 feet above the ground at 225 feet from the van. Recall, the signal source is scanned about 4.5 feet above the ground. Therefore, at 330 feet, the scan is near a broad maximum that could be as much as 6 dB above free space if the ground were perfectly reflecting. The measured level at 330 feet is 2 dB above free space. At 845 feet, the scan should be down on the side of an 8-foot-high maximum. The measured level at 845 feet is 1.3 dB above free space. At 225 feet, the scan is near the first minimum above the ground. The measured level at 225 feet is 1.2 dB below free space.

The interior locations of the buildings are arbitrarily numbered in the column labeled "Location Inside". The column labeled "Relative to Average Attenuation in decibels" contains the differences between the median signal levels measured at the locations and the average of the four outside median levels for the building. Positive attenuation indicates the signal level inside is smaller than the average level outside.

Within a period of a few minutes, three separate measurements

were made at Location 1 inside the metal-sided building. The spread of the resulting three medians was only ±0.5 dB around the average value listed in the table. Two separate measurements were made at Location 2 outside the wooden building. The resulting two medians were within ±0.3 dB of the average value listed. Thus, the measurement repeatability is good and is consistent with the ±0.5 dB standard deviation of the medians expected for the statistical fluctuation resulting from the limited number (≈150) of independent samples in a 4-foot-square area. Measurements were made at the same locations in and around the metal building on three different days that had different ground moisture conditions, etc. The three medians for each location were averaged and the difference was taken between the location average and the individual medians. The standard deviation of the differences was 1 dB.

The column labeled "Relative to Incident Attenuation in decibels" contains the differences between the median levels at the inside locations and the median level for the outside location that is closest to the van. These outside closest locations have the largest signal levels measured for their corresponding buildings. The outside level is corrected for free space ($1/r^2$) for the distance between the outside location and the inside location being considered. This is the second definition of building attenuation described in Section 2.3. This second definition of attenuation is essentially the same definition of building attenuation that was used in Ref. 6 for the same all-metal building listed first in Table I (the values in Ref. 6 were not corrected for $1/r^2$). The second definition attenuation values are within 1 or 2 dB of the values in Ref. 6 for Locations 1 and 3 inside the metal building. At Location 2, however, the attenuation in Table I is 6 dB less than the earlier value. Items inside the building have been rearranged since the earlier measurements, but no reason for such a large change is evident. The second definition attenuation into the all-metal building is greater than attenuation from the front (van side) to the back of that building (10.0 dB front-to-back attenuation including $1/r^2$ correction). The second definition attenuation into the other buildings is less than the front-to-back attenuation (21.7 dB for the metal-sided bulding and 15.0 dB for the wooden one). This suggests that the dominant mechanism for signal propagation behind the metal building is scatter and/or reflection from objects behind and to the side of that building rather than passage through the building itself. The dominant mechanism for propagation behind the other two buildings is not evident from these simple measurements since either passage through multiple walls or scatter and/or reflection is consistent with the result.

For Location 1 in the wooden building, the first definition attenuation is negative. This indicates that the median level inside at that

location is greater than the average of the four outside medians. This is a reasonable situation because the signal levels at one side and at the back of that building are much lower than the signal level inside Location 1. This inside location has unscreened windows between it and the outside in the direction of the van. As mentioned in Section II, this first definition of attenuation seems more appropriate for use in system analysis since a system would have to serve the outside locations at all sides of a building. Of course, the distribution of outside levels relative to the outside average is also needed for a complete assessment of system performance.

The building attenuations by either definition are greatest for the all-metal building with metal screened windows. Since both of the other buildings have metal in their walls, attenuation into them probably depends strongly on coupling through the nonscreened windows.

### *3.3 Cross-polarization couplings for the three buildings*

Cross-polarization coupling in multipath propagation is significant in reducing the effects of the random orientation of portable radiotelephones.[14] Cross-polarization coupling is defined as 20 $\log(E_x/E_t)$, where $E_t$ is the average or median field magnitude of the polarization aligned with the transmitted polarization and $E_x$ is the average (or median) field magnitude of the polarization orthogonal (crossed) to $E_t$. An indication of the cross-polarization coupling can be obtained by orienting the signal-source dipole antenna horizontally and scanning a measurement location with the dipole pointed towards the van (end-on orientation). The scan can be repeated with the dipole perpendicular to the direction of the van (broadside orientation).

If the multipath propagation were uniformly distributed in azimuth around the measurement location and were confined to a horizontal plane, horizontally polarized multipath would produce a median received signal level in a scanned horizontal dipole that was 3 dB less than the median that would be produced by the same multipath in a scanned loop oriented with its plane horizontal. The 3-dB decrease results from the nonuniform directivity pattern of the dipole in any plane containing the dipole. The median level received by the loop in the horizontally polarized multipath would be the same as the median level received by a scanned vertical dipole in vertically polarized multipath of the same average intensity. The multipath signal variations in all cases would be Rayleigh distributed.

For the measurement situation, equal median signal levels for end-on and broadside horizontal scans would be consistent with multipath having a uniform azimuthal distribution. Then, under the assumption that the propagation directions are confined to a horizontal plane, the

cross-polarization coupling would be 3 dB greater than the signal difference $\Delta = L_v - L_h$, where $L_v$ is the median level of the signal from a scan with the source oriented vertically, and $L_h$ is the median level of the signal from a scan with the source oriented horizontally. The medians have a statistical fluctuation with a standard deviation on the order of ±0.5 dB because of the limited number of independent samples. Therefore, for differences between the medians of the two horizontal scans of one or two decibels, they can be taken as equal and their average value can be used to determine Δ.

Table II summarizes the cross-polarization measurements made in and around the three buildings. The medians of the end-on and broadside scans are tabulated in columns labeled End-on and Broadside. The values are in decibels relative to the median of the signal scan at the same location with the source antenna oriented vertically. That is, columns End-on and Broadside indicate Δ for end-on and broadside scans. The column labeled cross-polarization is 3 dB greater than the average Δ for end-on and broadside scans.

The locations inside the metal building would yield a small positive value for cross-polarization coupling. This probably indicates a breakdown of the assumption that the multipath propagation is confined to a horizontal plane, so the coupling is taken as 0 dB. If the multipath propagation were uniformly distributed in all directions in three dimensions, the orientation of the antenna would be irrelevant. Since the data show only a small bias towards stronger median signal for the vertical antenna in the metal building, an alternative assumption for that building would seem to be uniformly distributed multipath propagation in all directions in three dimensions.

The cross-polarization values in Table II are all greater than −10 dB and most are greater than −6 dB. Another point worth noting is that the locations with the lowest signal levels (greatest attenuation) also have the largest cross-polarization coupling with values greater than −6 dB and usually greater than −3 dB. Since cross-polarization

Table II—Cross-polarization coupling for three buildings

| Building Material | Location Outside | Location Inside | Broadside Scan (dB) | End-on Scan (dB) | Cross-polarization (dB) |
|---|---|---|---|---|---|
| All metal | 3 | — | −9.3 | −6.8 | −5.0 |
| | — | 1 | −1.0 | +0.4 | 0 |
| | — | 2 | −4.4 | −0.9 | 0 |
| | — | 3 | −1.0 | — | 0 |
| Metal siding | 2 | — | −13.6 | −10.6 | −9.1 |
| | 4 | — | −5.0 | −6.2 | −2.6 |
| | — | 1 | −4.0 | −4.5 | −1.3 |
| Wood | 2 | — | −5.7 | — | −2.7 |
| | — | 1 | −8.4 | −10.2 | −6.6 |

coupling increases the effectiveness of diversity in mitigating the effects of random portable set orientation and multipath propagation,[14] this trend could be significant in determining system performance.

## IV. ATTENUATION STATISTICS FOR A HOUSE

### *4.1 House description*

The house is a two-story colonial located on a level, one acre lot. It is in a newly developed area with a density of one house per acre and with relatively few trees. The house has an area of 2400 square feet, consisting of living room, dining room, kitchen, and den on the first floor, and four bedrooms on the second floor. It also has a basement and a two-car attached garage. It is constructed with wood, with aluminum siding on three sides and nonmetallic siding on the front. All exterior walls contain insulation faced with a metal foil vapor barrier.

### *4.2 Small-scale statistics*

The cumulative distributions of the envelope variations of multipath propagation for a scan of a 4-foot-square location are expected to be Rayleigh.[2,6] Figure 5 shows the distributions measured at four locations in and around the house. A Rayleigh distribution is a straight line with the particular slope indicated on the figure. The distributions are good approximations to the Rayleigh distribution for many locations inside the house and behind the house from the van. A few locations on the same side of the house as the van and only 400 feet from it experience essentially line-of-sight propagation. At these few locations the signal variation is small and the envelope distribution significantly departs from Rayleigh, as indicated by the example plotted as squares on the figure. The medians of the four distributions on the figure have been normalized to 0 dB. The two distributions that show the greatest departure from Rayleigh were selected as the extremes that have the greatest and least spread in attenuation of any of the distributions in the data set.

### *4.3. Large-scale statistics of small-scale medians for outside locations*

The open data points in Fig. 6 are the medians of the measured signal envelopes for the small-scale locations outside the house. The medians are plotted versus the distance between the van antenna and the location. The median levels are in decibels relative to the signal level at the van reference. The outside locations are in front of the midpoints of the four outside walls of the house. The decibel averages of the four locations for each van position are indicated by the solid data points.

A strong dependence of signal level on distance is evident in Fig. 6.

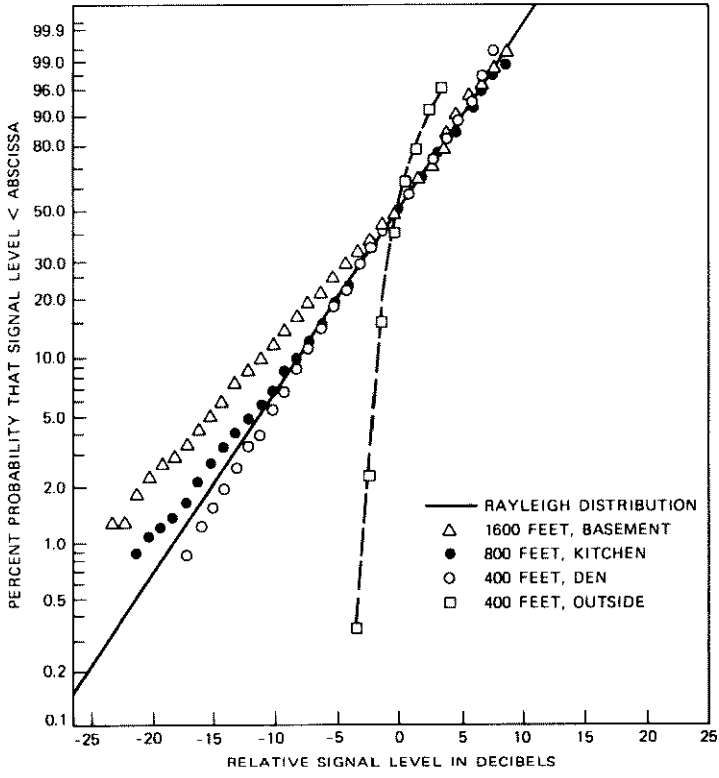



Fig. 5—Measured signal envelope distributions from four small (4-ft square) areas. On these coordinates, Rayleigh distributions are straight lines with the slope indicated. Distances between the van and the measurement location are indicated in the key.

The solid line is the linear-least-squares regression fit to the 36 medians from the four locations for each of the nine van positions. The least-squares regression fit to the averages of the four locations for each van position yields the same solid line. Signal level varies with distance as $d^{-4.5}$ for the solid line. The dotted line represents free-space propagation ($d^{-2}$) relative to 0 dB at the van reference. At 1000 feet from the van, the signal level represented by the solid line is 12.5 dB lower than the level would be in free space, i.e., the average excess attenuation over free space is 12.5 dB at 1000 feet. The distance dependence exponent is probably reliable only to several tenths because of the large spread in the data and the relatively small number of data points. The signal level outside was correlated to distance with a coefficient of 0.8.

The lower dashed line is the linear regression line for the data near 400 feet and 800 feet only. The upper dashed line is for the data near 800 feet and 1600 feet. The distance dependence for the 400-foot to

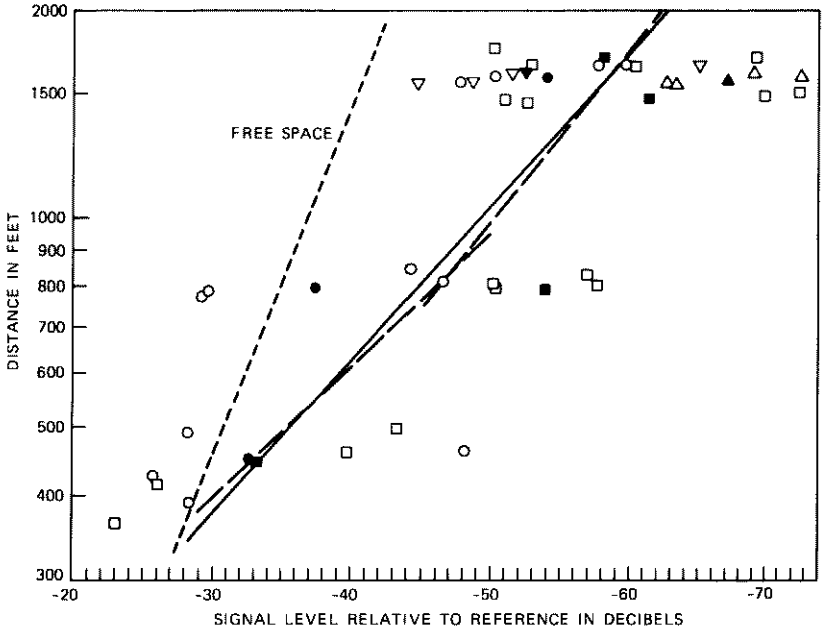



Fig. 6—Medians of the small-scale envelope variations for different outside measurement locations plotted versus distances between the locations and the van antenna. Open data points are medians for individual outside locations. Solid points are averages of the medians for the four measurement locations for each of the van locations. Points represented by the same symbols at nearly the same distance designate the four measurements associated with a particular van position. Signal levels are with respect to 0 dB at the signal reference located 14.2 feet from the van receiving antenna. The dotted line represents free-space propagation ($1/r^2$) relative to 0 dB at the signal reference. The solid line and the dashed lines are linear-least-squares regression lines, as discussed in the text.

800-foot distances is $d^{-5.1}$; the dependence for the 800-foot to 1600-foot distances is $d^{-4.1}$. Thus, the signal decreases faster with distance in the first several hundred feet than it does farther from the house. This is reasonable because, for the first few hundred feet, there are few obstructions along the path and propagation approaches line-of-sight. After several hundred feet, the number of intervening houses increases. Reflection from the ground also causes the signal to decrease more rapidly with distance than it does in free space.

The data points at 400 feet that are greater than free space values deserve comment. These points are from locations that are not shadowed from the van, i.e., they are within line-of-sight. A single ground reflection from a 27-foot-high antenna 400 feet away would produce a signal maximum about 4 feet above the ground. The signal maximum would be greater than the single-path free-space value. (It would be 6 dB greater for a reflection coefficient of unity.) The signal levels a few dB above free space are not unexpected 4.5 feet above the ground in

locations that have large reflecting surfaces (walls of houses) nearby in addition to the ground.

Figure 7 is the cumulative distribution of the data from Fig. 6 after the distance dependence, $d^{-4.5}$, is removed. The calculated standard deviation for the data points is 9.0 dB. A straight line on the coordinates in Fig. 7 represents a log-normal distribution of signal level. The measured distribution is a reasonable approximation to a log-normal distribution.

### *4.4 Large-scale statistics of small-scale medians for inside locations*

The distance dependences for the linear-least-square fits to the medians for measurement locations inside the house were somewhat different for the different floors of the house. The variation with distance was $d^{-3.9}$ for the first floor, $d^{-3.0}$ for the second floor, and $d^{-3.2}$ for the basement. Since only one location was measured in the basement, this value for the basement may contain considerable statistical

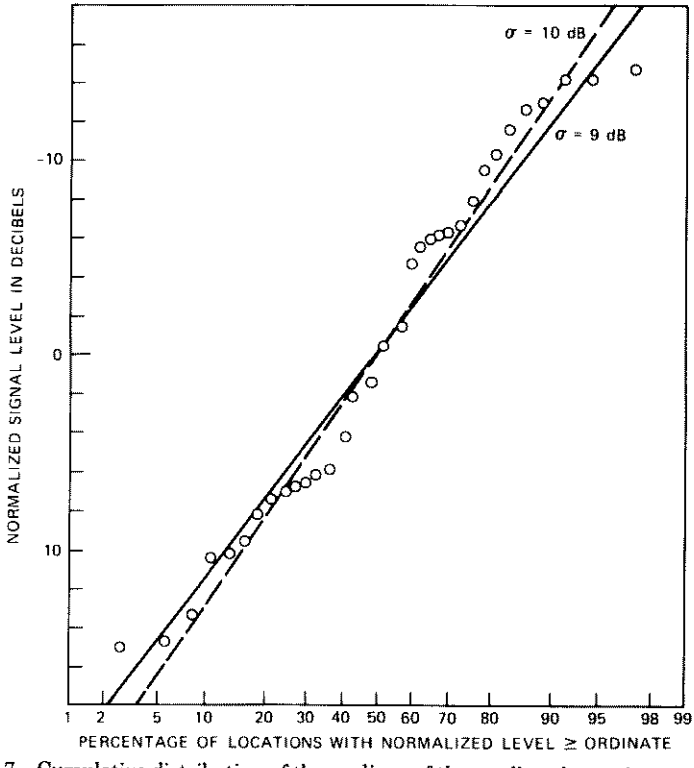



Fig. 7—Cumulative distribution of the medians of the small-scale envelope variations for outside locations after the values for the solid regression line in Fig. 6 are subtracted out. The solid line and the dashed line in this figure represent log-normal distributions with standard deviations of 9 dB and 10 dB, respectively.

error. The variation with distance for the composite set of data from the entire house (eight measurement locations inside the house and nine van positions) was $d^{-3.5}$. The less rapid decrease in signal level with distance for the second floor compared with the first floor is reasonable because the height of the second floor results in less attenuation from intervening houses. Since the decrease in signal level with distance is less rapid for the locations inside the house than outside, the apparent building attenuation will decrease somewhat with distance.

At 1000 feet, the excess attenuation over the free space values are 18.5 dB for the first floor, 16.5 dB for the second floor, 29.9 dB for the basement, and 19 dB for the composite data set for the entire house. The average building attenuation at 1000 feet, which is the difference between the linear regression values outside and inside, is then 6.0 dB for the first floor, 4.0 dB for the second floor, 16.4 dB for the basement, and 6.5 dB for the entire house.

#### *4.5 Large-scale statistics of the building attenuation*

The building attenuation considered in this section uses the first definition in Section 2.3, i.e., attenuation is with respect to the average of the four outside median levels. Building attenuations for the house are plotted versus distance in Fig. 8. The open data points represent data from the first floor; the solid data points and the points marked B represent data from the second floor and the basement, respectively. As noted in the previous section, the attenuation is weakly dependent on distance. Also, the attenuation into the basement is significantly greater than the attenuation into the other two floors. The linear regression lines are labeled on the figure for several combinations of the data. The variation of attenuation with distance is $d^{-0.6}$ for the first floor, $d^{-1.5}$ for the second floor, $d^{-1.4}$ for the basement, and $d^{-1.0}$ for the first and second floor together and for the entire house. The attenuation at 1000 feet taken from the regression lines is 6.0 dB for the first floor, 4.0 dB for the second floor, and 16.4 dB for the basement. These 1000-ft attenuation values are the same as those obtained in the previous section from the separate linear regression lines to the signal levels inside and outside.

Means ($m$) and standard deviations ($\sigma$) for several different groupings of the attenuation data are tabulated in Table III. It appears that most of the difference in the distance dependence for the first and second floors results from the different attenuation averages at 1600 feet.

The different distance dependences and different average attenuation for the different floors present a dilemma if they are real effects not confined to this data set. For radio system analysis, a single

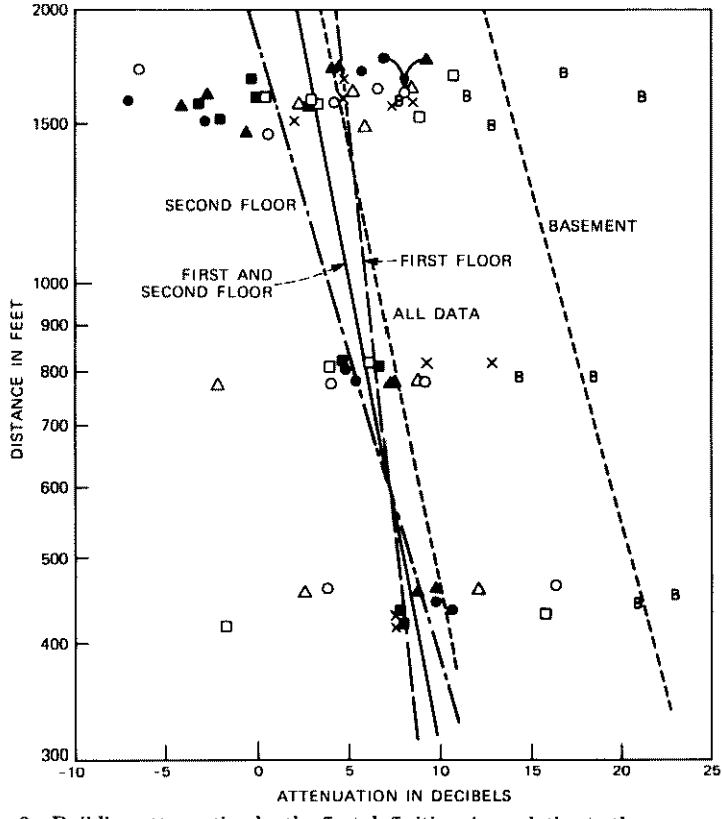



Fig. 8—Building attenuation by the first definition, i.e., relative to the average of the median levels for the four outside locations, plotted versus distance. Open data points and X points are from the first floor, solid data points are from the second floor, and points marked B are from the basement. All points marked with a particular symbol are from the same measurement location in the house but are for different van positions. The lines are linear-least-squares regression lines through different groupings of points.

Table III—Attenuation and standard deviations for the house (table values are in dB)

| Floor | $m$ / $\sigma$ | 400 ft | 800 ft | 1600 ft |
|---|---|---|---|---|
| First | $m$ | 8.1 | 6.5 | 4.8 |
| | $\sigma$ | 6.4 | 4.5 | 3.9 |
| Second | $m$ | 9.1 | 6.1 | 0.7 |
| | $\sigma$ | 1.1 | 1.3 | 4.6 |
| Basement | $m$ | 22.1 | 16.4 | 13.8 |
| | $\sigma$ | 1.7 | 2.8 | 5.1 |
| First and second | $m$ | 8.5 | 6.3 | 3.1 |
| | $\sigma$ | 4.8 | 3.4 | 4.7 |
| First, second and basement | $m$ | 10.2 | 7.6 | 4.4 |
| | $\sigma$ | 6.4 | 4.7 | 5.9 |

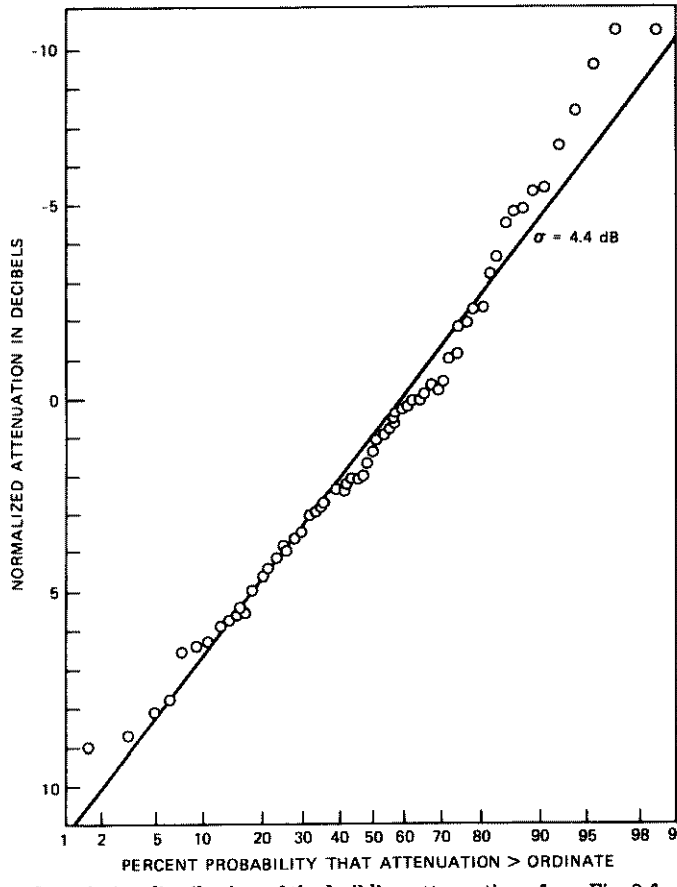



Fig. 9—Cumulative distribution of the building attenuations from Fig. 8 for the first and second floors after the values for the first- and second-floor regression line (solid line in Fig. 8) are subtracted out. The straight line represents a log-normal distribution with a standard deviation of 4.4 dB.

distance dependence and average attenuation are desirable; however, if only the composite regression line for the whole house were removed from the data, the variation remaining in the residual attenuation data would have a significantly larger standard deviation ($\sigma = 6.2$ dB) than if the individual regression lines for each floor were removed separately ($\sigma = 4.1$ dB). Also, the composite line for the entire house depends on the number of measurements made on each floor. For systems analysis, these numbers of measurements should be related to the probability of calls being made on those floors (or at those locations); but these probabilities are unknown.

The separation in decibels between the regression lines on Fig. 8 for the first and second floors is not extreme over the 4 to 1 distance

covered. Also, the attenuation averages for these floors for the three distances are not grossly different (all within ±2 dB). Therefore, it seems reasonable to simplify the description of the attenuation by combining the first and second floor data. The composite regression line for this combination is also shown on the figure. When this composite regression line is removed from the first and second floor data, the resulting cumulative distribution of attenuation is as shown in Fig. 9. The standard deviation of the attenuation data plotted in Fig. 9 is 4.4 dB, not significantly different from the 4.1-dB standard deviation for all the data with the regression line for each floor removed separately. The distribution in Fig. 9 is a reasonable fit to a log-normal distribution, especially considering the limited size of the data set. Combining the basement data with the data from the other floors is not reasonable because of the large difference in average attenuations. Since this difference can be expected to exist for all houses, the attenuation into basements will probably have to be described separately from the attenuation for the rooms on other floors.

## V. CONCLUSIONS

Signal levels were measured in and around three small buildings from a single position of an instrumentation van. Measurements were also made in and around a house from nine different van positions ranging from 400 feet to 1600 feet away. The received signal envelope is approximately Rayleigh distributed over most small-scale areas about 4 feet square. For the three buildings, building attenuation of the small-scale signal medians ranged from −2 to 24 dB relative to the average signal level outside each building.

For the house, some of the parameters of the medians of the small-scale envelope variations and of the building attenuation defined in this paper are summarized in Table IV. After the distance dependence is removed from the median signal levels outside of the house, the distribution of the residual large-scale signal-level variations is ap-

Table IV—Summary of signal-level and attenuation parameters for the house

| Parameter | Outside | First Floor | Second Floor | Basement | First & Second Floors | Units |
|---|---|---|---|---|---|---|
| Signal-level distance dependence exponent | −4.5 | −3.9 | −3.0 | −3.2 | — | — |
| Signal-level relative to free space at 1000 ft. | −12.5 | −18.5 | −16.5 | −28.9 | — | dB |
| Building attenuation exponent | — | −0.6 | −1.5 | −1.4 | −1.0 | — |
| Average building attenuation at 1000 ft. | — | +6.0 | +4.0 | +16.4 | +5.1 | dB |

proximately log-normal with a standard deviation of 9 dB. The standard deviation of the building attenuation for the first and second floors is 4.4 dB after distance dependence is removed. Because of the large average attenuation into the basement of the house, it does not appear reasonable to combine the basement attenuation statistics with the statistics for the rooms above ground level.

Cross-polarization coupling is strong inside and outside of the three buildings. For all locations measured, the coupling is greater than −10 dB and for most locations it is −6 dB or greater.

## VI. ACKNOWLEDGMENTS

Initial design of the van masts and mounts was done by W. I. Tohlman and H. H. Hoffman. We wish to thank H. W. Arnold for permitting the measurements reported in this paper to be made in his house. The continued support of L. J. Greenstein and D. O. Reudink is greatly appreciated.

## REFERENCES

1. D. C. Cox, "Co-Channel Interference Considerations in Frequency-Reuse Small Coverage-Area Radio Systems," IEEE Trans. Commun., *COM-30* (January 1982), pp. 135–42.
2. W. C. Jakes, *Microwave Mobile Communications*, New York: John Wiley and Sons, 1974.
3. J. M. Durante, "Building Penetration Loss at 900 MHz," Conference Proceedings, IEEE VTG Conference, 1973, p. 1–7.
4. P. I. Wells and P. V. Tryon, "The Attenuation of UHF Radio Signals by Houses," U. S. Dept. of Commerce Report, OT Report 76-98, August 1976; and IEEE Trans. Veh. Technol., *VT-26* (November 1977), pp. 358–62.
5. J. Shefer, "Propagation Statistics of 900 MHz and 450 MHz Signals Inside Buildings," Microwave Mobile Radio Symposium, March 7–9, 1973, Boulder, Colorado.
6. H. H. Hoffman and D. C. Cox, "Attenuation of 900 MHz Radio Waves Propagating Into a Metal Building," IEEE Trans. Ant. Propag., *AP-30* (July 1982), pp. 808–11.
7. L. P. Rice, "Radio Transmission into Buildings on 35 and 150 MHz," B.S.T.J., *38*, No. 1 (January 1959), pp. 197–210.
8. D. Mitchell and K. G. Van Wynen, "A 150 MC Personal Radio Signaling System," B.S.T.J., *40*, No. 5 (September 1961), pp. 1239–57.
9. G. V. Waldo, "Report on the Analysis of Measurements and Observations of New York City UHF-TV Project," IEEE Trans. Broadc., *9* (1963), pp. 7–36.
10. K. Tsujimura and M. Kuwabara, "Cordless Telephone System and its Propagation Characteristics," IEEE Trans. Veh. Technol., *VT-26* (November 1977), pp. 367–71.
11. M. Komura, T. Hogihira, and M. Ogasawara, "New Radio Paging System and Its Propagation Characteristics," IEEE Trans. Veh. Technol., *VT-26* (November 1977), pp. 362–6.
12. D. C. Cox, "Multipath Delay Spread and Path Loss Correlation for 910 MHz Urban Mobile Radio Propagation," IEEE Trans. Veh. Technol., *VT-26* (November 1977), pp. 340–4.
13. P. A. Matthews, *Radio Wave Propagation, VHF and Above*, London: Chapman and Hall, Ltd., 1965, Chapter 2.
14. D. C. Cox, Antenna Diversity Performance in Mitigating the Effects of Portable Radiotelephone Orientation and Multipath Propagation, IEEE Trans. Commun., *COM-31* (May 1983), pp. 620–8.

## AUTHORS

**Donald C. Cox**, B.S., M.S., University of Nebraska, Lincoln, 1959 and 1960, respectively; Ph.D., Stanford University, 1968, all in Electrical Engineering;

Honorary Dr. of Science, University of Nebraska, Lincoln, 1983; Stanford University, 1963–1968; Bell Laboratories, 1968—. From 1960 to 1963, Mr. Cox worked on microwave communications system design at Wright-Patterson AFB, Ohio. From 1963 to 1968 he was at Stanford University doing tunnel diode amplifier design and research on microwave propagation in the troposphere. From 1968 to 1973 he was a Member of Technical Staff at Bell Laboratories, Holmdel, New Jersey, doing research in mobile radio propagation and on high-capacity mobile radio systems. He is now Supervisor of a group doing propagation and systems research for portable-radio telephony and for millimeter-wave satellite communications. Fellow, IEEE; member, Commissions B, C, and F of USNC/URSI, Sigma Xi, Sigma Tau, Eta Kappa Nu and Pi Mu Epsilon; Registered Professional Engineer, Ohio and Nebraska.

**Roy R. Murray,** B.S. (Electronic Engineering), 1975, Monmouth College; Bell Laboratories, 1965—. Mr. Murray has worked on high-speed multi-level digital modulators and, more recently, on UHF radio propagation into buildings. Currently, he is a member of the Telecommunication Systems Research Department. Member, Eta Kappa Nu.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

# Penetration of Radio Signals Into Buildings in the Cellular Radio Environment

By E. H. WALKER*

(Manuscript received April 4, 1983)

Penetration of radio signals from Advanced Mobile Phone Service cell site transmitters has been measured in fourteen office and industrial buildings in the Chicago area. Signal levels on the first floor of buildings averaged 14 dB less than reference levels in the adjacent streets. This penetration loss was found to decrease with increasing height. Standard deviations of penetration loss ranging from 5 to 11 dB attest to the diversity of architecture and floor arrangements. Other relationships useful in planning for portable phone terminal applications are derived from the data and are presented in this paper. The data include over 4000 measurements, each being the average of 1024 samples of the local field taken over an eight-second period as the instrumentation traveled about 20 feet over the measurement path. The measurement path in each building was traversed for each of the several cell sites that transmitted signals of adequate strength into the buildings.

## I. INTRODUCTION

The use of portable and hand-held terminals in 850-MHz cellular radio systems involves a radio field environment inside buildings that differs from the more familiar highway and street propagation environment of the mobile terminal. The laws that define propagation over open and urban terrain are complicated by a penetration loss in the transmission of the signal into the building's interior. The characterization of this loss, as encountered in large office and commercial

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

buildings, was the purpose of an extensive series of measurements made in fourteen buildings served by the Advanced Mobile Phone Service (AMPS) Developmental Cellular System in Chicago. These measurements were then compared with the penetration loss in a single-family residence.

The measurements were made in January 1980 to serve as a basis for evolving system requirements related to the use of portable phone terminals in AMPS systems.

Indoor portable service in cellular systems offers a possible improvement over conventional systems. The multiplicity of serving cell sites should in many cases illuminate all sides of those buildings that enjoy a relatively exposed or unshadowed environment. It was expected that this effect would be somewhat less pronounced in buildings that are more sheltered in the urban core; the data support this conjecture. The data also support expectations that areas with windows would have lower penetration losses than areas without windows; the difference is about 6 dB. First-floor penetration losses averaged 14.2 dB. The loss decreased with height at 1.9 dB per floor, very close to the 2.0- and 2.5-dB rates reported in Refs. 1 and 2. The penetration loss measured in the aluminum-sided ranch house, as a matter of interest, is 7.3 dB, a value close to the findings of Cox et al. in Ref. 3.

Most of the measurements made in the Cellular Test Bed (Ref. 4) and reported in Ref. 5 were of the signal as received by a mobile transceiver. The measurements reported here include the entire distribution of setup channel signal levels throughout the measurement areas for each cell site transmitter that qualified as a server. As such, the data are perfectly applicable to conventional mobile portable and paging applications. The effect is that of taking measurements with the base station in several different locations.

## II. ENVIRONMENT AND TEST METHOD

The Chicago AMPS Developmental Cellular System, as described in Ref. 4, uses ten separate setup channels for paging and mobile call origination in the Chicago service area. The channel signals are radiated continuously from omnidirectional antennas. The instrumentation used for the measurements reported in this paper can be tuned manually to any of these frequencies to measure one channel at a time. Figure 1 shows nominal coverage contours for the setup channels of the seven cell sites that served the selected buildings. The related voice-channel coverage is shown in Fig. 2 of the paper by D. L. Huff.[4] The center of each contour represents a cell site location. In general, the cell sites are spaced about 15 miles apart.

A group of buildings in the Chicago area was selected to represent a range of physical characteristics including location, architecture,

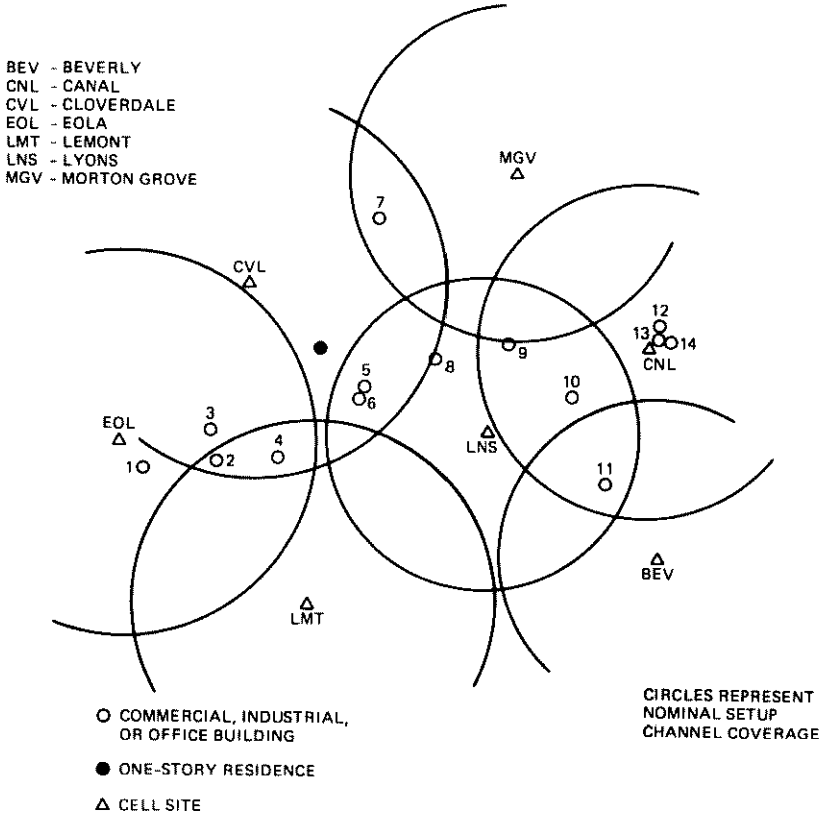



Fig. 1—AMPS Developmental Cellular System locations of 15 buildings and seven serving cell sites.

and local environment. Signal strength measurements were made along planned routes on selected floors of these buildings. A Propagation Measuring Set (PMS) was used to measure the strength of signals transmitted from cell sites.

Building penetration loss for a given floor area is defined to be the difference between the average of these measurements and an average of measurements made on the outside at street level.

Outside signal strength was measured at street level around the perimeter of the building, along the closest available path to the building's outside walls. These paths included driveways, streets, or parking lots, as required to achieve proximity to the building under test.

The PMS data output was transcribed for subsequent analysis by a computer program. The analysis was performed in the AMPS cellular test bed data processing facility described by DiPiazza, Plitkins and Zysman in Ref. 4.

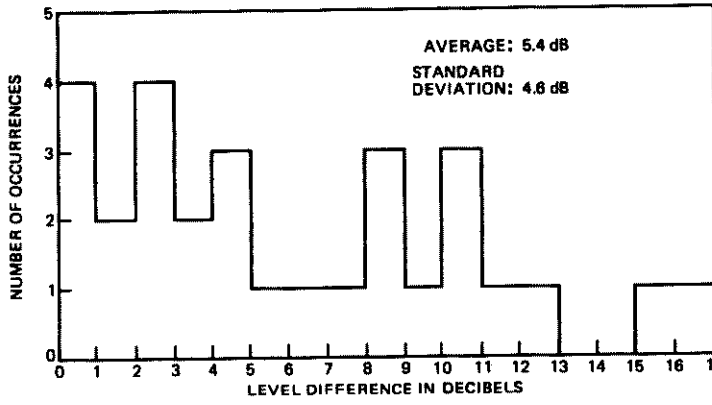



Fig. 2—Differences in level between the strongest and weakest serving channels on the same floor.

The purpose of the measurement program, as stated previously, was to determine the extent to which signals available from AMPS cell sites penetrate buildings. A first-order determination was desired; effects of 5 dB or more on the loss measurements were considered to be significant. A detailed investigation of the structure of interior fields in small buildings is reported by Cox in Ref. 3.

The measurements were subject to the effects of the following different conditions:

1. Different types of outside wall construction, such as steel-framed glass, brick, block masonry, etc.
2. Urban vs. suburban areas to identify difference between buildings in exposed locations and buildings sheltered in the dense urban core.
3. Changes in floor elevation.
4. Different building orientations with respect to serving cell sites.
5. Different percentages of window areas in the outside walls.
6. Different types of window treatment currently used to reflect sunlight and heat.

Time was an important factor in selecting the coverage objective for each building. Test runs were chosen to require about three hours of operations in any given building. The total time of recorded data is a small percentage of that.

Six significant categories of interior areas were identified. These were defined as open, enclosed, and hallways—each with and without windows. Typical of the areas that were considered to be "open" were cafeterias, lobbies, and large office spaces or conference rooms. Typical of "enclosed" areas were small, walled areas with space for only a few occupants, such as small office areas or workrooms.

Measurements in the first floors and second floors, together, were

thought to represent a worst case. Higher floors, thought to represent better penetration, were included with less regularity. Test routes were designed to distribute measurements over the floor areas in a uniform manner.

## III. BUILDING SELECTION

The buildings chosen in the Chicago AMPS service area were those where multiple cell sites could provide outside building signal levels of at least −90 dBm.

In the downtown section of Chicago, tests were made in three buildings. These represent the urban group. The downtown buildings in which tests were made can be characterized as "single tenant buildings".

In the suburban area around Chicago, tests were made in eleven commercial buildings and one residential dwelling. Only the data from commercial buildings were included in subsequent analyses.

From the standpoint of architectural variety the program provided measurements in a warehouse, a manufacturing plant, open-space offices, cafeterias, lobbies, small offices, and hallways. Outside wall construction varied from unwindowed concrete slab to steel-framed windows occupying about 90 percent of the outside surface.

The locations of the buildings where tests were conducted are shown on the map of Fig. 1. The circles show the idealized boundaries of regions of cell site control channel coverage. Table I shows the distribution of floor levels in buildings where tests were conducted and the cell sites whose transmissions served the various floor levels. A more detailed description of each of the buildings is given in the appendix.

Several factors were considered when selecting floors and floor areas for measurement. From a test sampling standpoint the first and second floors were chosen when available since building penetration was expected to be poorest at the lower levels. Higher floor levels were included in the tests to provide data on the sensitivity of building penetration to height. On any particular floor, areas were chosen that represent the different types of interiors; test paths were chosen to be uniformly distributed over the floor. In two instances measurements were attempted in basement areas but signal strength levels were below the threshold of the PMS.

## IV. MEASUREMENT TECHNIQUE

A portable Propagation Measurement Set (PMS) designed for field strength measurements was adapted for this test program by mounting it in a small "tea" cart equipped with a battery, battery charger, and a mobile antenna mounted on a 2-foot by 2-foot ground plane. The

Table I—Buildings, floors, and serving cell sites contributing to measurements

| | Buildings | Floors | Cell Sites* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | CVL | EOL | LMT | LNS | CNL | MGV | BEV |
| 1 | Nabisco Naperville | 1 | | X | | | | | |
| | | 2 | | X | X | | | | |
| | | 4 | X | X | X | | | | |
| 2 | Metropolitan Naperville | 1 | X | X | | | | | |
| | | 2 | X | X | X | | | | |
| | | 3 | X | X | X | | | | |
| 3 | Bell Labs Naperville | 1 | X | X | | | | | |
| | | 3 | X | X | | | | | |
| | | 5 | X | X | X | | | | |
| 4 | BSCTE Lisle | 1 | X | X | X | | | | |
| | | 3 | X | X | X | | | | |
| | | 5 | X | X | X | X | | | |
| | | 9 | X | X | X | X | | | |
| 5 | Illinois Bell Oakbrook | 1 | X | X | X | X | | | |
| | | 2 | X | X | X | X | | | |
| 6 | McDonalds Oakbrook | 1 | X | | X | X | | | |
| | | 2 | | | | X | | | |
| | | 8 | X | X | X | X | | | |
| 7 | Chrysler | 1 | X | | | | | X | |
| 8 | Allied Van Lines Hillside | 1 | | | | X | | | |
| | | 2 | X | | X | X | | | |
| | | 4 | X | | X | X | X | | |
| 9 | Illinois Bell Oak Park | 3 | | | | X | X | X | |
| | | 4 | | | | X | X | X | |
| 10 | Western Electric Cicero | 1 | | | | X | | | |
| | | 2 | | | | X | | | |
| | | 3 | | | | X | X | X | |
| 11 | Illinois Bell Kedzie | 1 | | | | X | X | | X |
| | | 4 | | | | X | X | | X |
| 12 | American Medical Assn | 1 | | | | X | X | X | |
| | | 2 | | | | X | X | X | |
| 13 | Illinois Bell Headquarters Randolph | 1 | | | | | X | | |
| | | 3 | | | | | X | | |
| | | 5 | | | | | X | X | X |
| 14 | Illinois Bell Adams | 1 | | | | | X | | |
| | | 2 | | | | | X | | |
| | | 12 | | | | | X | | |
| | | 15 | | | | | X | | |
| One-story residence (data not included in analyses) | | 1 | X | X | X | X | | | |

* CVL–Cloverdale; EOL–Eola; LMT–Lemont; LNS–Lyons; CNL–Canal; MGV–Morton Grove; BEV–Beverly.

ground plane was positioned 4-1/2 feet above floor level, at approximately shoulder height.

The design of the PMS is based upon the instrumentation receiver of the mobile communications laboratory described by DiPiazza, Plitkins, and Zysman in Ref. 4.

The PMS has an internal calibration system that calibrates the receiver over an 80-dB range in 1-dB increments. In the measurement mode, the basic unit of measurement is a one-second power average derived from a log-amplifier output sampled at a 128-Hz rate. The data output of the set can be adjusted to provide one average value, also derived as a power average, for a group of 1 to 128 of these one-second averages, selectable in binary increments. The measurement range of the set is −40 dBm to approximately −122 dBm.

The PMS uses a paper tape printer for hard copy output. The output data include frequency, a sequential "major marker," the power average, and the one-second sample variance for the group of one-second samples contributing to each longer-term mean. In this application, a mean value was generated each eight seconds. The "major marker" is used to key measurements to the time of day and to logged operator observations.

Prior to a series of measurements, the PMS is calibrated and the calibration results are printed on paper tape. A received signal strength meter in the PMS provides a visual indication of the receiver output without operating the paper tape recorder. The measurement team uses it to verify test set operation and to choose the group of control channels for measurement. The set also contains start, stop, and pause controls to limit data output to areas of interest.

To increase the confidence level of outside measurements, two methods were used where possible to determine outside signal strength. One method used the Mobile Telephone Laboratory (MTL) vehicle, which has the capability of power averaging instantaneous signal strength samples. This facility is described by Huff in Ref. 4. The MTL receivers were sampled at a rate of 32 samples per second. The short-term (1/2-second) MTL power averages were grouped in segments representing building faces and then further power averaged to provide a single mean value for each face segment. The segment means were then decibel averaged over all faces to develop the outside reference.

The second method used the PMS at the time of the in-building measurements to spot check MTL data as a guard against system variations.

MTL measurements were not available for the buildings in the Chicago downtown area, the AMA building, the IBT Headquarters building at Randolph, and the Bell Training Center at West Adams. The PMS was used to collect both inside and outside measurements.

## V. TEST PROCEDURES

The PMS was transported to the building locations in a step van. While the PMS was in the van, the van rooftop antenna was connected to it, and street-level data were recorded along the building perimeter.

The measurements collected along the entire perimeter were decibel averaged to provide the outside reference.

The largest possible variety of areas was included in the routing chosen for the PMS measurements in each particular building. In measuring areas within a building the data were marked so that the different interior types could be distinguished. The PMS has a provision for including a marker number in the data output. As the PMS measurements were made along each route, an accompanying commentary was made on a cassette recorder. This commentary provided information about the type of building interior area being measured and the associated major marker number. The major marker information and the interior type were used subsequently in coding the data for computer program input.

Also, this commentary was useful in flagging data that should be discarded and in coordinating the data with the floor area to which it applied.

As we noted earlier, the PMS has the capability to generate average values that can be selected to represent from 1 to 128 seconds of real time. The 8-second average was chosen for output data in these measurements. That value represents a compromise between the volume of data points and the number of feet of travel per data point. With the operational procedures used in these tests, the 8-second averages result in one data point for about each 20 feet of travel. A total data volume of approximately 4000 data points was accumulated for the entire series of in-building measurements.

During the initial planning phase, the expected setup channel signal strength in the vicinity of each building was predicted from propagation contours of the cell sites in the Chicago AMPS System (see Fig. 1). Before measuring each building floor, the operator of the PMS noted the received signal strength indicator (RSSI) value at several locations on the floor. If it was determined that a high percentage of the area was served at signal strengths above the PMS receiver threshold, the operator proceeded to collect data at each setup channel frequency over the entire prescribed route for the selected building floor.

In the measurements made in the first building in the measurements program (Bell Laboratories, Naperville), a high degree of repeatability was noted when the measurements from PMS output tapes were compared for repeated runs. Based on these visual observations it was decided that a single pass at each frequency of interest would provide sufficient accuracy for measurements in the remaining areas.

A comparison of repeated measurements over a common route is shown in Table II. The results indicate run averages are repeatable within approximately 1 dB.

Table II—Repeatability of measurements*

| Floor | Channel | First Run Avg. Level (dBm) | First Run Std. Dev. (dB) | Second Run Avg. Level (dBm) | Second Run Std. Dev. (dB) | Level Diff. (dB) |
|---|---|---|---|---|---|---|
| 3 | 780 | −93.7 | 7.2 | −94.6 | 7.4 | 0.9 |
| 3 | 798 | −109.3 | 6.5 | −108.6 | 8.0 | −0.7 |
| 5 | 780 | −77.4 | 10.4 | −78.0 | 11.2 | 0.6 |
| 5 | 788 | −94.3 | 9.0 | −94.7 | 9.6 | 0.4 |
| 5 | 798 | −85.5 | 9.5 | −85.0 | 9.0 | −0.5 |

* Measurements made at Bell Laboratories, Naperville, IL.

## VI. DATA PROCESSING

As the first step in data processing, all measurements are coded to associate them with major marker numbers. The coding identifies the:

1. Building
2. Floor number (or outside street-level data)
3. Serving frequency
4. Time of day
5. Number of 1-second samples included in each average
6. Type of interior area (six possible types)
7. Flags indicating:
   (a) repeated measurements
   (b) data to be rejected because of operational trouble
   (c) omission of $N$ data points (beginning or end) for reasons of ambiguity.

The time-of-day information permits verification of the major marker sequence and data group duration.

Data become ambiguous when the transition from one major marker to the next is accomplished without halting the measurements system. The ambiguous data point (8-second sample) contains portions of data from two major markers; it is deleted from the database.

The outside street-level data are used for developing a reference for building penetration loss calculations. After the data are encoded, the processing is performed in four steps (or passes) as described below:

1. Initial processing of the data (Pass I) develops statistics for all data for each separate category (building, floor, area type, and channel frequency). Invalid data points are eliminated. A summary listing is provided that includes the number of entries, the number of entries below PMS threshold, the average power level, and the standard deviation for each group of data from categories of areas. In addition to this summary, histograms for each group are developed.

2. The second pass introduces the outside street-level reference information and combines it with the Pass I data. This produces a display of the data for each frequency with its differential level relative to the outside reference (penetration loss).

3. The third pass combines the penetration loss for all the measurement frequencies on common floors, as generated in Pass II. This allows penetration loss to be used as a base for further combining to classify losses by area type, floor level, etc.

4. The fourth pass combines the penetration loss data from Pass III across different buildings. In the combining process similar data from different buildings are weighted to compensate for differences in the data volume collected in each specific building. The weighted average loss for a class of data from several buildings represents equal contribution from each building. Pass IV generates information about the effects of classes of building locations (suburban vs. urban) and penetration loss by floor and area type, for all buildings.

## VII. SIGNIFICANT CONCLUSIONS

The conclusions made as a result of our extensive measurements are as follows:

1. Urban vs. suburban—First-floor penetration loss measurements were processed for three different groupings of buildings. The urban (downtown Chicago) building penetration for three buildings shows an average value of 18.0 dB. The penetration loss distribution for suburban buildings has an average value of 13.1 dB. These statistics are listed in Table III. The comparison indicates a loss for the urban group that is approximately 5 dB greater.

2. Penetration loss sensitivity to transmitter location—Different penetration losses are measured on common floors of common buildings as a function of the specific serving-site identity. These differences arise when a building is served by multiple cell sites. The distribution of the "max-min" decibel values for such multisite penetration loss differences is plotted in Fig. 2. The average max-min penetration loss difference for all the data is 5.4 dB with a sigma of 4.6 dB. This

Table III—Penetration loss

| Buildings | Floor | Loss (dB) | St. Dev. (dB) | No. of Bldgs. |
|---|---|---|---|---|
| Urban | 1 | 18.0 | 7.7 | 3 |
| Suburban | 1 | 13.1 | 9.5 | 10 |
| All | 1 | 14.2 | 9.3 | 13 |
| | 2 | 10.6 | 8.5 | 8 |
| | 3 | 6.8 | 9.8 | 6 |
| | 4 | −0.8 | 10.7 | 4 |
| | 5 | 2.9 | 9.2 | 3 |
| | 8 | −4.3 | 9.0 | 1 |
| | 9 | −1.4 | 8.8 | 1 |
| | 12 | 15.3 | 4.9 | 1 |
| | 15 | 10.9 | 5.0 | 1 |

suggests that building penetration is a relatively strong function of the direction of illumination.

3. Penetration loss sensitivity to interior type—Table IV compares all interiors with and without windows for all buildings and all floors. The presence of windows is shown to reduce the average penetration loss by 6 dB, with a sigma of 5.2 dB. Not included in the data is the office area of the Chrysler building in Elk Grove where the copper-sputtered glass windows blocked out all measurable signal levels. For the windowed data category, Table IV lists penetration loss for three combinations of interior types. Open interior areas are shown to have 3 dB less penetration loss than hallways.

4. Floor height effect—A list of building penetration loss values ranked by floor height is shown in Table II. The high values of loss for the twelfth and fifteenth floors of a single building are the result of the shadowing effect of adjacent buildings. While these buildings effectively sandwiched the subject building, the adjacent street areas were relatively open. This may be a relatively common occurrence in the urban core and would appear to substantiate the observations of Durante.[1] Figure 3 is a scatter diagram with a straight line fit for penetration loss vs. floor level. The dots indicate data points that represent decibel averages of penetration losses. For each floor, there is a data point for each channel that served that floor. Only floor levels having three or more data points are included in the diagram and the building described above was omitted. The mean values for all data on each floor are indicated by "X's." The "least-squares" straight line fit to these means is also shown. The slope of the line is −1.9 dB per floor. The first floor intercept is 10.4 dB. While this loss rate agrees closely with the findings in Refs. 1 and 2, the first-floor penetration loss is about 10 dB lower. Subsequent measurements by Bell Laboratories in the Newark, N.J., Cellular Test Bed have confirmed the ranges of loss values and loss rates presented in this paper. It is assumed that differences in penetration loss with respect to values reported previously are probably due to differences in the methods of establishing the outside street-level reference.

Table IV—Penetration loss comparisons

| | |
|---|---|
| Windowed/nonwindowed areas = 6.0 dB | (Standard Deviation = 5.2 dB) |
| For windowed areas: | |
| where enclosed areas are in conjunction with hallways | |
| Enclosed/Hallways = 0 dB | (Standard Deviation = 6.0 dB) |
| where open areas are in conjunction with enclosed areas | |
| Open/Enclosed = 1.0 dB | (Standard Deviation = 4.8 dB) |
| where open areas are in conjunction with hallways | |
| Open/Hallways = 3.1 dB | (Standard Deviation = 5.2 dB) |

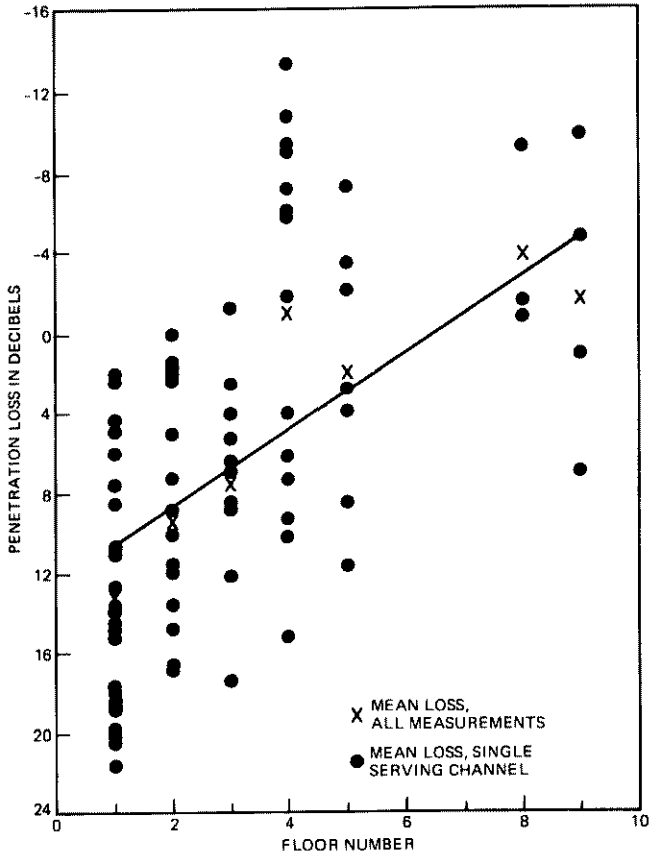



Fig. 3—Penetration loss measurements.

## VIII. ACKNOWLEDGMENTS

The test team members for operations and data measurement were M. Patarca, L. F. Smith, and J. G. Rebele. Data keypunching and coding from PMS output tapes and corresponding annotation was performed by M. Patarca. The software and the processing of data to produce the results of the measurements program was completed by A. Some. K. K. Kelly collaborated in the selection and location of candidate buildings. Messrs. J. T. Kennedy, G. A. Lothian and J. R. Nevarez arranged basing the operations in Chicago and obtained the permissions necessary to access the buildings for measurements.

## REFERENCES

1. J. M. Durante, "Building Penetration Loss at 900 MHz," Conference Proceedings, IEEE VTG Conf., 1973, pp. 1–7.

2. J. Dietz et. al., "Examination of the Feasibility of Conventional Land Mobile Operations at 950 MHz," FTC Office of the Chief Engineer, Research Division, Report R7102.
3. D. C. Cox, R. R. Murray, and A. W. Norris, "Measurements of 800-MHz Radio Transmissions into Buildings With Metallic Walls," B.S.T.J., this issue.
4. Special issue entitled "Advanced Mobile Phone Service," B.S.T.J., *58*, No. 1 (January 1979).
5. AT&T, Advanced Mobile Phone Service Developmental System Reports, Numbers 1–11, transmitted to the FCC by Illinois Bell, June 8, 1977–November 26, 1979.
6. L. P. Rice, "Radio Transmission into Buildings at 35 and 150 mc," B.S.T.J., *38*, No. 1 (January 1959), pp. 197–210.

## APPENDIX

### *Building Descriptions*

A.1 *Nabisco, Naperville*

| | |
|---|---|
| Surroundings: | Open area, without nearby multistory buildings. |
| Construction: | Masonry with some corrugated siding. The interior has some windowed office space but a large portion is devoted to cereal processing equipment. |
| Cell site signals: | 1st floor—Eola<br>2nd floor—Eola, Lemont<br>4th floor—Cloverdale, Eola, Lemont. |

A.2 *Metropolitan Life Insurance Co., Naperville*

| | |
|---|---|
| Surroundings: | Open area without nearby structures. |
| Construction: | Large glass sections alternated with large concrete slab sections. |
| Cell site signals: | 1st floor—Cloverdale, Eola<br>2nd floor—Cloverdale, Eola, Lemont<br>3rd floor—Cloverdale, Eola, Lemont. |

A.3 *Bell Laboratories, Naperville*

| | |
|---|---|
| Surroundings: | Open area without nearby structures. |
| Construction: | Low light transmission glass in metal framework. Mostly windowed halls with interior office space. |
| Cell site signals: | 1st floor—Cloverdale, Eola<br>3rd floor—Cloverdale, Eola<br>5th floor—Coverdale, Eola, Lemont. |

A.4 *Bell System Center for Technical Education, Lisle*

| | |
|---|---|
| Surroundings: | Open area without nearby structures. Located at the bottom of a hill to the East. |
| Construction: | Brick and glass with high-rise dormitory area. Windowed office space. |
| Cell site signals: | 1st floor—Cloverdale, Eola, Lemont<br>3rd floor—Cloverdale, Eola, Lemont<br>5th floor—Cloverdale, Eola, Lemont, Lyons<br>9th floor—Cloverdale, Eola, Lemont, Lyons |

A.5 *Illinois Bell, Oak Brook*

| | |
|---|---|
| Surroundings: | Stand-alone with other multistory buildings spaced several hundred yards distant. |
| Construction: | Brick and glass windowed office space and work space. |
| Cell site signals: | 1st floor—Cloverdale, Eola, Lemont, Lyons<br>2nd floor—Cloverdale, Eola, Lemont, Lyons. |

A.6 *McDonalds, Oak Brook*

| | |
|---|---|
| Surroundings: | Stand-alone with other multistory buildings spaced several hundred yards distant. |
| Construction: | Narrow windows with concrete in a ratio of approximately 50 percent. Interior office space uses half-height dividers. First floor is a lobby area. |
| Cell site signals: | 1st floor—Cloverdale, Lemont, Lyons<br>2nd floor—Lyons<br>8th floor—Cloverdale, Eola, Lemont, Lyons. |

A.7 *Chrysler, Elk Grove*

| | |
|---|---|
| Surroundings: | Open area without nearby multistory buildings. |
| Construction: | Concrete slabs, only one floor. The interior is a large warehouse with metal storage bins for automobile parts. An attached office building had no measurable signal because copper-sputtered windows constituted 100 percent of outside surface. |
| Cell site signals: | Cloverdale and Morton Grove. |

A.8 *Allied Van Lines, Hillside*

| | |
|---|---|
| Surroundings: | Residential area without nearby multistory structures. |
| Construction: | Concrete and glass with windowed office and work space. |
| Cell site signals: | 1st floor—Lyons, Canal<br>2nd floor—Cloverdale, Lemont, Lyons<br>4th floor—Cloverdale, Lemont, Lyons, Canal, Morton Grove. |

A.9 *Illinois Bell, Oak Park*

| | |
|---|---|
| Surroundings: | Corner buildings in area with other multistory buildings of about same height. |
| Construction: | Brick and glass. Windowed office space and work space. |
| Cell site signals: | 3rd floor—Lyons, Canal, Morton Grove<br>4th floor—Lyons, Canal, Morton Grove. |

A.10 *Western Electric, Hawthorne*

Surroundings: Corner building without nearby structures.

Construction: Brick and smaller glass windows typical of factory construction. First floor is open area with heavy machinery; second floor is open area with assembly lines; third floor is office space with windowed space.

Cell site signals: 1st floor—Lyons
2nd floor—Lyons
3rd floor—Lyons, Canal, Morton Grove.

A.11 *Illinois Bell, Kedzie and 61st Street*

Surroundings: Corner building in area with other multistory buildings of similar height.

Construction: Brick and glass. Windowed office space and work space.

Cell site signals: 1st floor—Lyons, Canal, Beverly
4th floor—Lyons, Canal, Beverly.

A.12 *American Medical Association, State Street (Downtown)*

Surroundings: Bounded on all sides by streets or parking area.

Construction: Reflective glass windows and concrete. The interior office area is open with three-quarter partitions.

Cell site signals: 1st floor—Lyons, Canal, Morton Grove
2nd floor—Lyons, Canal, Morton Grove.

A.13 *Illinois Bell, Randolph (Downtown)*

Surroundings: Bounded on four sides by city streets. Neighboring buildings are multistory.

Construction: Concrete and glass with lobby on first floor and windowed work and office spaces on upper floors. Office partitions are three-quarter-high dividers.

Cell site signals: 1st floor—Canal
3rd floor—Canal
5th floor—Canal, Morton Grove, Beverly.

A.14 *Bell Training Center, West Adams (Downtown)*

Surroundings: Neighboring multistory buildings in a business district. The building front is open to the street and both sides abut neighboring buildings. The rear of the building opens into a service courtyard.

Construction: Concrete and glass. Interior upper floors have windowed office space.

Cell site signals: 1st floor—Canal
2nd floor—Canal
12th floor—Canal
15th floor—Canal.

A.15 *Private Home, Lombard*

Surroundings: Residential neighborhood.
Construction: Aluminum siding.
Cell site signals: Cloverdale, Eola, Lemont, Lyons.

The outside reference measurements for this building were incomplete because a complete perimeter path was not available. The measurements for this building are not combined with the results from other buildings.

The penetration loss for this building based on the available street-level reference measurements is 7.3 dB with a sigma of 6.7 dB.

## AUTHOR

**Edward H. Walker,** B.S.E.E., 1956, Newark College of Engineering; Western Electric Company, 1940–1957; Bell Laboratories, 1958–1981. Mr. Walker has been involved in the design of radio circuits and the development of military radar systems. Since 1973 he has been responsible for the experimental evaluation of audio quality of the AMPS System and of mobile and portable terminals. His work on building penetration was completed prior to his retirement from Bell Laboratories in 1981. Member, IEEE.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

# Transmission Errors and Forward Error Correction in Embedded Differential Pulse Code Modulation

By D. J. GOODMAN* and C.-E SUNDBERG†

(Manuscript received December 11, 1982)

We have derived formulas for the combined effects of quantization and transmission errors on the performance of embedded Differential Pulse Code Modulation (DPCM), a source code that can be used for variable-bit-rate speech transmission. Our analysis is more general and more precise than previous work on transmission errors in digital communication of analog signals. Special cases include conventional DPCM and Pulse Code Modulation (PCM). Our main result is a signal-to-noise ratio formula in which the effects of source characteristics (input signal, codec design parameters) and the effects of transmission characteristics (modulation, channel, forward error correction) are clearly distinguishable. We also present, in computationally convenient forms, specialized formulas that apply to uncoded transmission through a random-error channel, transmission through a slowly fading channel, and transmission with part or all of the DPCM signal protected by an error-correcting code. Numerical results show how channel coding can have different effects on conventional and embedded DPCM. They also show how the binary-number representation of quantizer outputs influences performance.

## I. INTRODUCTION

### *1.1 Embedded Differential Pulse Code Modulation*

Embedded coding can play a valuable role in variable-bit-rate speech transmission. With an embedded code the analog-to-digital (a/d) and

* Bell Laboratories. † University of Lund, Sweden.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

digital-to-analog (d/a) converters operate at a constant, high bit rate, and the transmission system controls the instantaneous rate. Proposed applications for variable-bit-rate operation include a digital private branch exchange,[1] digital speech interpolation,[2] packet-switched voice transmission,[3] and mobile radio.[4]

Sophisticated versions of Differential Pulse Code Modulation (DPCM) are promising speech codes for these and other environments.[5-7] However, conventional DPCM is not suited to variable-bit-rate transmission because the decoder amplifies the effects of bit-rate adjustments. On the other hand, a slightly modified form of DPCM avoids this problem and produces an embedded code.[8]

Figure 1 shows the codec (coder, decoder) structure of embedded DPCM. Although up to $E$ bits/sample can be transmitted, the signals presented to the two integrators have a resolution of only $M$ bits/sample, the minimum bit rate of the channel. While Fig. 1 is a useful guide to practical implementations, Fig. 2, which is equivalent, is easier to analyze. It shows the quantizer at the encoder as a successive-approximation combination of two quantizers: a "minimal" quantizer with $M$ bits/sample and a "supplemental" quantizer with $E-M$ bits/sample, operating on the error signal of the minimal quantizer.*

In embedded DPCM, all of the bits from the minimal quantizer arrive at the decoder; the transmission system can delete some or all of the supplemental bits. With $S$ bits/sample of the supplemental quantizer transmitted to the decoder, the rate is $D = M + S$ bits/sample, and the quantizing distortion is very close to that of a conventional codec with $D$ bits/sample.

Errors in the two bit streams have different effects on the decoder output. Errors in the $M$, minimal bits, are enhanced by the decoder integrator, which has no effect on errors in the $S$, supplemental bits. This situation compares favorably with conventional DPCM, where all errors are integrated at the decoder. It also has implications for forward error correction in embedded DPCM. Figures 1 and 2 will be further explained in Section II.

### *1.2 The scope of this paper*

Our principal contribution in this paper is an analysis of the combined effects of granular quantizing distortion and transmission errors on the mean-square error of embedded DPCM. The analysis is quite general: special cases include Pulse Code Modulation (PCM) ($M = 0$) and conventional DPCM ($S = 0$). The formulas for the noisy-

* Based on the structure of Fig. 2, Jayant has recently described an enhanced supplemental quantizer (called an explicit noise coder) that uses memory and delay to improve speech quality.[9]

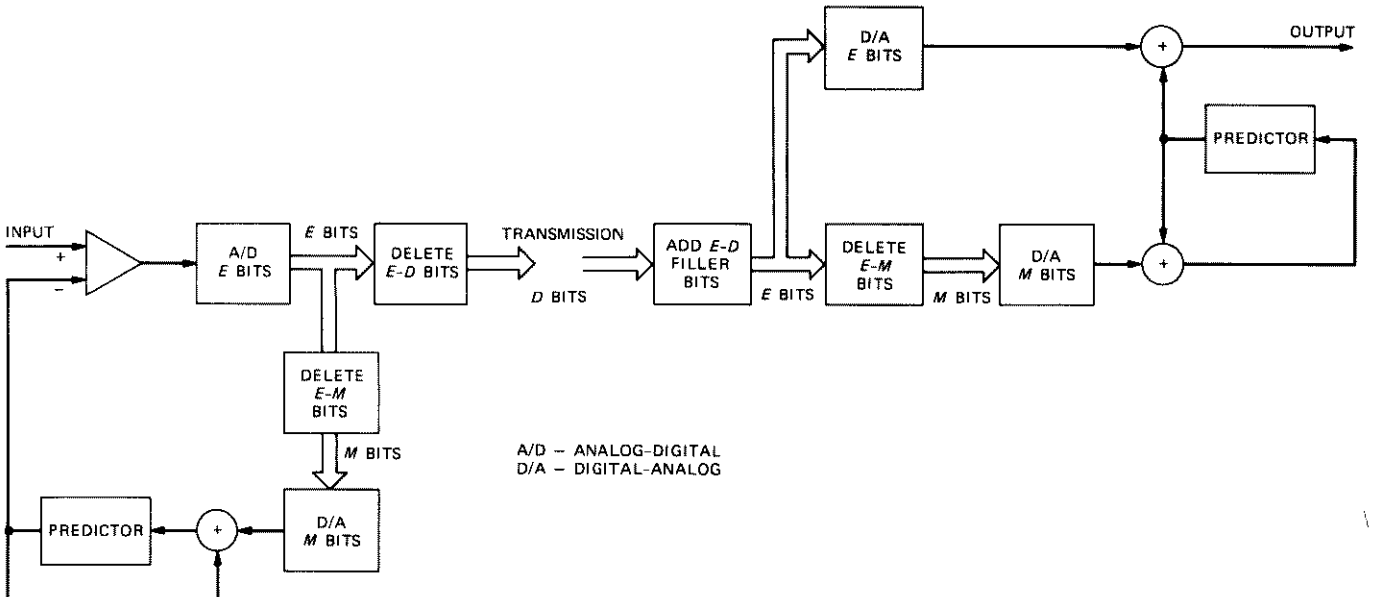



Fig. 1—Embedded DPCM encoder and decoder. Because both predictors operate on the same signal (with resolution $M$ bits/sample) performance is unaffected by errors due to bit-rate adjustment. The channel can transmit between $M$ bits/sample and $E$ bits/sample.

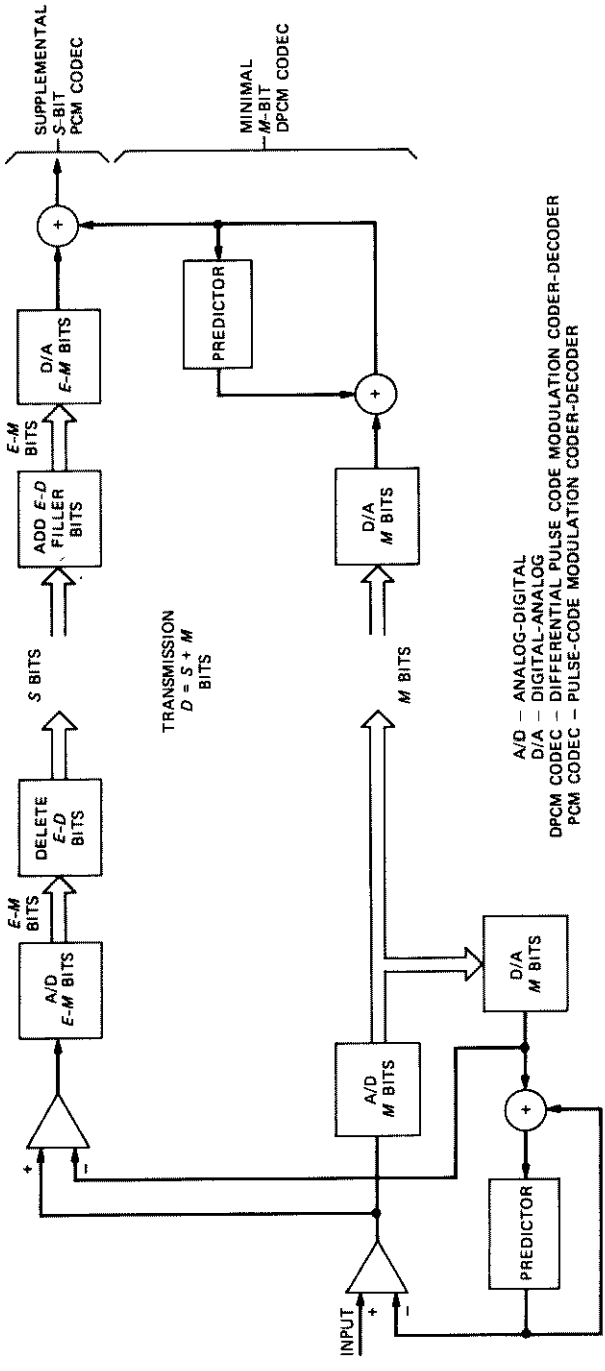



Fig. 2—Embedded DPCM encoder and decoder. $E$-bit analog-to-digital conversion is shown as a two-stage, successive-approximation process. All $M$ minimal bits are transmitted. $S$, the number of supplemental bits transmitted, can range from 0 to $E$–$M$.

channel performance of conventional DPCM are new, and the specialization to PCM is more precise than previous work on the subject,[10-12] which includes several approximations that are accurate for multibit ($\geq$ 6-bit) quantizers, but are rather imprecise at lower bit rates. The method of analysis has the advantage of separating source effects from transmission effects. The source effects include the characteristics of the analog input signal and the codec design parameters. The transmission effects include modulation and demodulation, the channel, forward error correction, and diversity reception.

The main result is eq. (61), in which the transmission effects are contained in the discrete probability function $P(l)$, where $l$ is an index of binary error patterns. The other symbols in (61) are source parameters and functions of source parameters. After deriving (61) we apply it to specific transmission environments and present, in Table VI, specialized formulas that are convenient for numerical computation.

In all, there are 78 formulas in Sections III through VI, most of them intermediate steps in derivations of a few key results. Anticipating that few readers will require all these details we provide here a summary of the analysis and we display a few numerical results. Sections I, II, and VII contain the main ideas of our work and sufficient information to allow readers to perform, on hand calculators, computations similar to the ones we present.

Section III introduces the notations for the signals and errors in the $M$-bit minimal DPCM codec and the $S$-bit supplementary PCM codec of Fig. 2. The analysis of Section III leads to (2), which expresses the sampled-data error sequence as a function of quantization errors, transmission errors, and integrator characteristics. Section IV begins the analysis of the mean-square value of (2) by deriving (35), the ratio of the mean-square codec input to the mean-square value of the encoder difference signal. Section V defines $A$ factors, which are conditional mean squares of the errors due to specific binary error patterns, and derives (61), the general signal-to-noise ratio (s/n) formula. Section VI adapts (61) to specific transmission models and provides guides to numerical computation.

### *1.3 Examples of numerical results*

Figure 3 shows the performance of embedded DPCM in four transmission environments, all of them employing Coherent Phase Shift Keying (CPSK) modulation at 32 kb/s in a white-Gaussian-noise channel. The encoder operates at 32 kb/s (8-kHz sampling, 4 bits/sample), and in format 1 all of this information is transmitted. Figure 3 indicates that when the channel s/n falls below 10 dB, the audio s/n deteriorates rapidly. In format 2, the least significant bit of each DPCM code word is deleted, and the remaining 3 bits/sample are

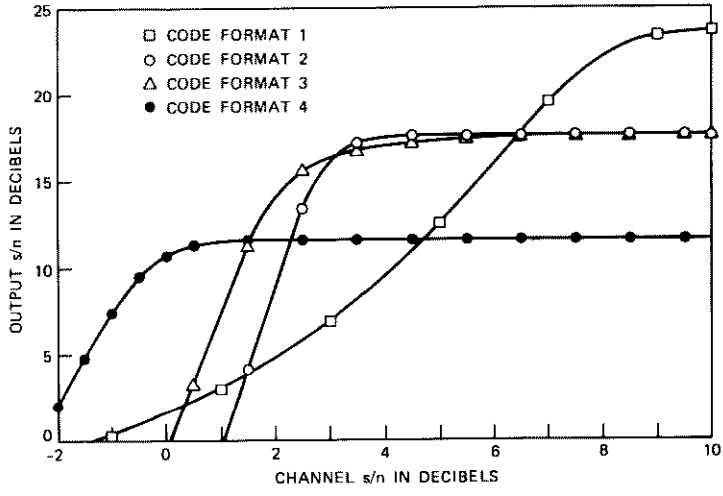



Fig. 3—Performance of embedded DPCM in four transmission environments.

protected by a rate 3/4 convolutional code. Although there is more quantizing noise than in format 1 (the s/n is 6 dB lower in the absence of transmission errors), the convolutional code permits accurate reception of the transmitted bit stream at channel s/n's down to 3 dB. Going one step further with this approach to channel coding, we have format 4, in which 16 kb/s of speech data are transmitted under the protection of a rate 1/2 code. The threshold of essentially error-free performance is now extended down to a channel s/n of about 0 dB.

In code format 3 the speech transmission rate is 24 kb/s, as in format 2, but now only 2 of the 3 bits/sample are protected by the convolutional code, which has rate 2/3. The threshold of curve 3 in Fig. 3 is about 1 dB lower than that of curve 2. On the other hand, format 3 is slightly worse than format 2 in intermediate channel conditions (s/n's between 3 and 5 dB). Over this range, format 2 is essentially error free, while format 3 is affected by errors in the unprotected third bit of each code word. The effect is small, however, because these errors are not amplified at the decoder.

With conventional, rather than embedded, DPCM, the corresponding picture, Fig. 4, is rather different, especially with respect to format 3. Here channel errors in the unprotected third bit are amplified by the integrator at the decoder. The result is a noticeably lower output s/n relative to format 2 (all three bits protected) when the channel s/n is between 3 and 6 dB. On the other hand, in clear channels the greater accuracy of prediction in the conventional encoder causes the output s/n of conventional DPCM at 24 kb/s (formats 2 and 3) and

32 kb/s (format 1) to be about 0.7 dB higher than that of embedded DPCM.

Figure 5, which applies to 24 kb/s speech transmission with a rate 2/3 code, summarizes the performance differences between conventional and embedded DPCM. Conventional DPCM has somewhat lower quantizing noise, which is reflected in the higher s/n in good channels. In intermediate conditions, when errors in the unprotected

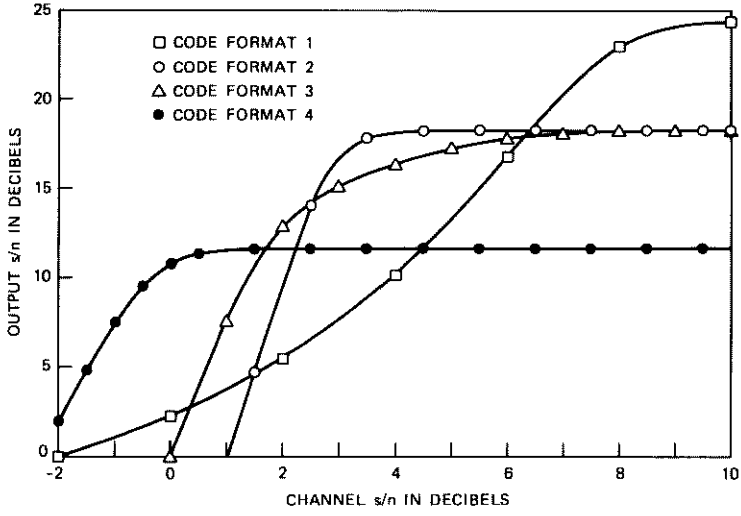



Fig. 4—Performance of conventional DPCM in four transmission environments.

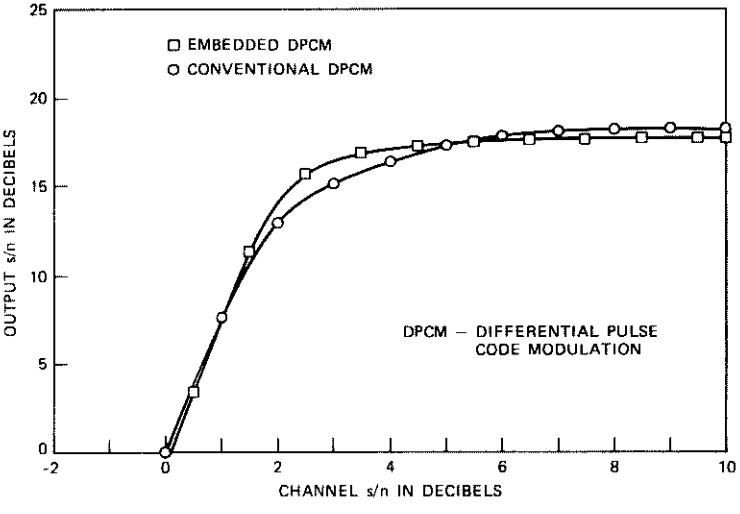



Fig. 5—A 24-kb/s speech transmission with a rate of 2/3 code (Format 3) used to summarize performance differences between embedded and conventional DPCM.

bit influence performance, embedded DPCM is better because the effects of these errors are not amplified at the decoder. In very difficult channels, errors in the two coded bits dominate performance and the s/n's of conventional and embedded DPCM are virtually equal.

## II. EMBEDDED DPCM SIGNAL PROCESSING

Figure 1 shows the signal processing operations that take place in embedded DPCM encoding, transmission, and decoding. While the analog-to-digital converter at the encoder generates $E$ bits/sample, the resolution of the signal presented to the integrator is limited to $M$ bits/sample, where $M$ is the minimum bit rate of the transmission system. The transmitted bit rate, $D$, can vary between $M$ and $E$. At the receiver $E$–$D$ filler bits are appended to the incoming signal. As in the encoder, $E$–$M$ bits are deleted at the integrator input so that in the absence of transmission errors the encoder integrator and the decoder integrator produce the same approximation signal. When this signal is added to the full-resolution ($D$ bits) quantizing error, the sum has nearly the quality of a conventional DPCM signal with $D$ bits/sample.

While Fig. 1 demonstrates practical implementations, Fig. 2, which is equivalent, is easier to analyze. It represents the analog-to-digital conversion as a two-step, successive-approximation process. First the input to the converter is represented by $M$ bits/sample. Then the error of this representation is processed by another analog-to-digital converter with $E$–$M$ bits/sample. Taken together, the two digital signals comprise an $E$ bits/sample representation of the DPCM difference signal. All of the $M$ bits of the minimal analog-to-digital converter are transmitted. The other $E$–$M$ bits are subject to deletion by the transmission system. At the receiver the minimal, $M$-bit signal is processed by a conventional DPCM decoder. The result is added to the supplemental, $S = D - M$ bit representation of the DPCM error signal to produce the system output.

## III. SIGNAL ANALYSIS

### *3.1 Error sequence*

To analyze Fig. 2, we introduce Fig. 6, which shows the signals that appear in the analysis and defines their notations. We are interested in the overall error signal

$$e(k) = x'(k) - x(k), \tag{1}$$

the difference between decoder output and encoder input. In particular we will derive the formula

$$e(k) = n_D(k) + e_D(k) + \sum_{i=1}^{\infty} b_i e_M(k - i), \tag{2}$$

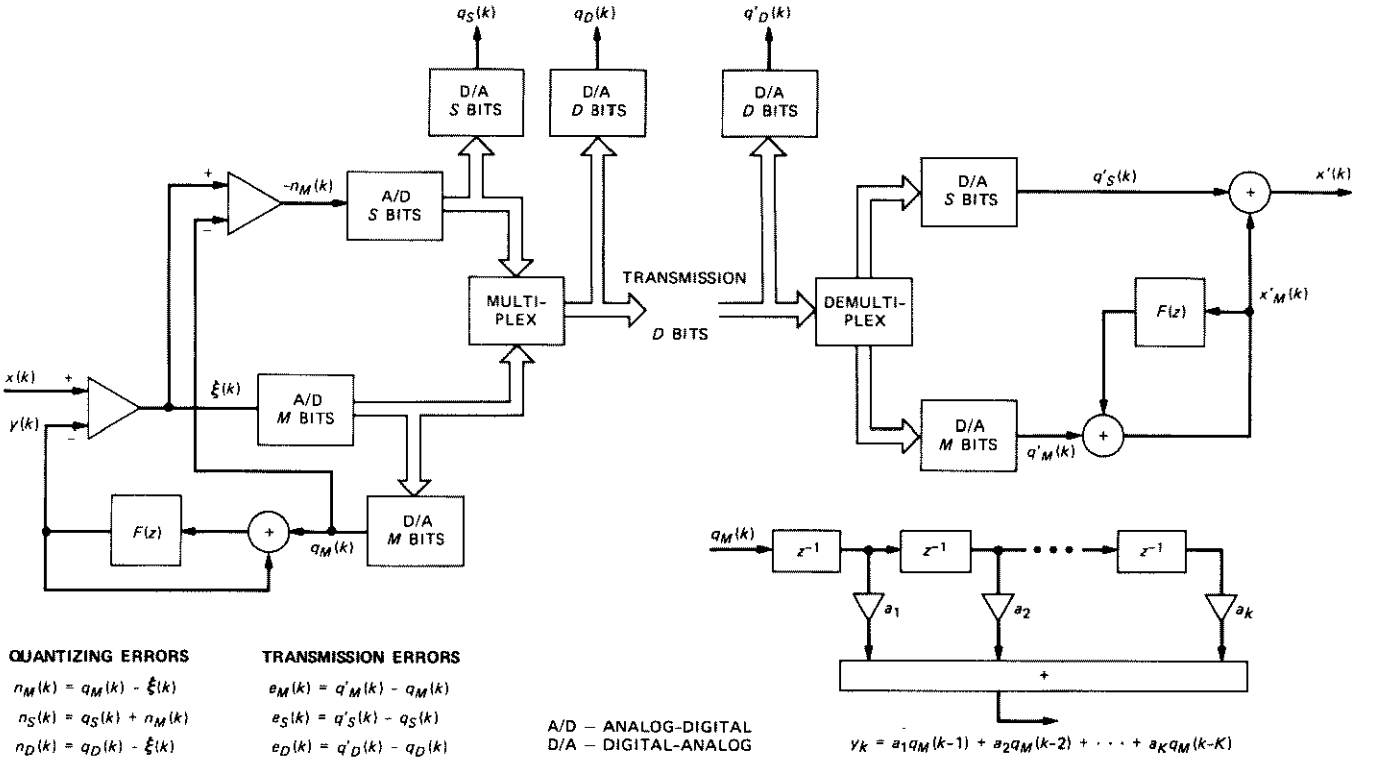



Fig. 6—The signal-processing operations of Fig. 2 and the notation used to analyze them. Three new digital-to-analog converters define signals that appear in the analysis.

where $n_D(k)$ is the quantization noise of the two-stage, $D$-bit analog-to-digital conversion, $e_D(k)$ is the effect of a transmission error on the entire $D$-bit transmitted code word, and $e_M(k)$ is the effect of a transmission error on the minimal, $M$-bit DPCM code word. The coefficients $b_i$ are related to the predictor coefficients $a_1, a_2, \cdots a_K$ according to (12).

Formally,

$$e_M(k) = q'_M(k) - q_M(k), \tag{3}$$

the difference between the quantized inputs to the decoder and encoder integrators. To define $n_D(k)$ and $e_D(k)$, we view the combined code word with $M + S = D$ bits as a digital representation of $\xi(k) = x(k) - y(k)$. A $D$-bit digital-to-analog converter would produce the quantized signal $q_D(k)$, and so we have the definition of quantization error:

$$n_D(k) = q_D(k) - \xi(k). \tag{4}$$

At the receiver, where the $D$ bits/sample are possibly corrupted by transmission errors, a digital-to-analog converter would produce $q'_D(k)$. The transmision error is

$$e_D(k) = q'_D(k) - q_D(k). \tag{5}$$

In the remainder of Section III we derive eq. (2); in Section IV and V we analyze its mean square.

### *3.2 Derivation of the error sequence*

Here the signal analysis is facilitated by the transform notation of Table I. The reader may verify that the output of the minimal encoder, $Q_M(z)$, is related to input and quantizing noise[8] by

$$Q_M(z) = [X(z) + N_M(z)][1 - F(z)]. \tag{6}$$

Table I—Transform notation for codec signals

| Encoder Signal | Description | Decoder Signal |
|---|---|---|
| $X(z)$ | Encoder input, decoder output | $X'(z)$ |
| | Minimal decoder output | $X'_M(z)$ |
| $Y(z)$ | Approximation signal | $Y'(z)$ |
| $Q_M(z)$ | $M$-bit representation of $X(z) - Y(z)$ | $Q'_M(z)$ |
| $N_M(z)$ | $Q_M(z) - [X(z) - Y(z)]$<br>Quantizing error in $Q_M(z)$<br>$Q'_M(z) - Q_M(z)$<br>Transmission error in $Q'_M(z)$ | $E_M(z)$ |
| $Q_S(z)$ | $S$-bit representation of $-N_M(z)$ | $Q'_S(z)$ |
| $N_D(z)$ | $Q_S(z) + N_M(z)$<br>Quantizing error in $Q_S(z)$<br>$Q'_S(z) - Q_S(z)$<br>Transmission error in $Q'_S(z)$ | $E_S(z)$ |
| $F(z)$ | Predictor $\sum_{i=1}^{K} a_i z^{-1}$ | $F(z)$ |

In the minimal decoder

$$X'_M(z) = \frac{Q'_M(z)}{1 - F(z)}, \tag{7}$$

which leads to

$$X'_M(z) = X(z) + N_M(z) + \frac{E_M(z)}{1 - F(z)}. \tag{8}$$

In the supplemental encoder,

$$Q_S(z) = -N_M(z) + N_D(z), \tag{9}$$

and at the decoder

$$Q'_S(z) = -N_M(z) + N_D(z) + E_S(z). \tag{10}$$

Combining (8) and (10) we have the output of the entire decoder,

$$X'(z) = X'_M(z) + Q'_S(z) = X(z) + N_D(z) + E_S(z) + \frac{E_M(z)}{1 - F(z)}. \tag{11}$$

To transform (11) to time-domain notation, we defined $b_i$, $i = 0, 1, 2, \cdots$, to be the inverse $z$ transform of $1/(1 - F)$, such that

$$\frac{1}{1 - F(z)} = \sum_{i=0}^{\infty} b_i z^{-i}. \tag{12}$$

Then we have

$$x'(k) = x(k) + n_D(k) + e_S(k) + \sum_{i=0}^{\infty} b_i e_M(k - i), \tag{13}$$

and the error is

$$\begin{aligned} e(k) &= x'(k) - x(k) \\ &= n_D(k) + e_S(k) + e_M(k) + \sum_{i=1}^{\infty} b_i e_M(k - i), \end{aligned} \tag{14}$$

where we have substituted $b_0 = 1$. Equation (14) is identical to (2) because

$$e_D(k) = e_M(k) + e_S(k). \tag{15}$$

## IV. MEAN-SQUARE ERROR

The square of (2) is

$$\begin{aligned} e^2(k) = [n_D(k) + e_D(k)]^2 &+ \sum_{i=1}^{\infty} b_i^2 e_M^2(k - i) \\ &+ 2 \sum_{i=1}^{\infty} b_i [n_D(k) + e_D(k)] e_M(k - i) \\ &+ 2 \sum_{i=1}^{\infty} \sum_{j=i+1}^{\infty} b_i b_j e_M(k - i) e_M(k - j). \end{aligned} \tag{16}$$

To analyze the mean value of (16), we assume that the sequence $\{x(k)\}$ is drawn from a stationary ergodic random process. In our derivations we ignore all correlations in (16) between nonsimultaneous samples. That is, we assume

$$E\{[n_D(k) + e_D(k)]e_M(k - i)\} \approx 0; \qquad i \geq 1 \tag{17}$$

and

$$E\{e_M(k - i)e_M(k - j)\} \approx 0; \qquad i \neq j. \tag{18}$$

Equation (17) indicates that the overall error (quantizing plus channel distortion) in the $k$th sample is uncorrelated with errors in other samples of the minimal $M$-bit quantized samples. Equation (18) states that errors in different minimal samples are uncorrelated. These approximations are accurate because the sequence of samples at the input to a DPCM quantizer is decorrelated by the differential coding process and because transmission errors affecting different code words are independent or only weakly correlated.

The approximations, (17) and (18), remove the last two sums from the expected value of (16), leaving

$$E\{e^2(k)\} = E\{[n_D(k) + e_D(k)]^2\} + b_P E\{e_M^2(k)\}, \tag{19}$$

in which we summarize the influence of the predictor in

$$b_P = \sum_{i=1}^{\infty} b_i^2. \tag{20}$$

The expectations in (19) are related to the quantization and transmission of $\xi(k)$, the DPCM difference signal. In Section V, we present a complete theory of the errors due to these operations. While this theory relates these errors to $\sigma_\xi^2 = E\{\xi^2(k)\}$, we are ultimately interested in the s/n of the codec input, $x(k)$:

$$s/n = E\{x^2(k)\}/E\{e^2(k)\} = \sigma_x^2/\sigma_e^2. \tag{21}$$

To find this quantity, we will now derive $\sigma_x^2/\sigma_\xi^2$ and then combine it with the results of Section V. To begin the derivation, we refer to Fig. 6 and verify

$$Y(z) = F(z)[X(z) + N_M(z)], \tag{22}$$

which leads to

$$\xi(k) = x(k) - y(k) = x(k) - \sum_{i=1}^{K} a_i x(k - i) - \sum_{i=1}^{K} a_i n_M(k - i). \tag{23}$$

We write the mean-square value of (23) as

$$\sigma_\xi^2 = E\left\{\left[x(k) - \sum_{i=1}^{K} a_i x(k - i)\right]^2\right\} + E\left[\sum_{i=1}^{K} a_i n_M(k - i)\right]^2 - 2E\left\{\left[x(k) - \sum_{i=1}^{K} a_i x(k - i)\right]\left[\sum_{i=1}^{K} a_i n_M(k - i)\right]\right\}. \tag{24}$$

The first term in (24) depends on the spectral properties of $x(k)$ and on the predictor. The ratio of $\sigma_x^2$ to this quantity is called the prediction gain,

$$G = \sigma_x^2 / E\left\{\left[x(k) - \sum_{i=1}^{K} a_i x(k-i)\right]^2\right\}. \tag{25}$$

It indicates that extent to which the predictor (in the absence of quantization) reduces the mean-square value of the signal to be quantized. Formally we have

$$G^{-1} = \sum_{j=0}^{K} \sum_{i=0}^{K} a_i' a_j' r_{i-j}, \tag{26}$$

in which $a_0' = 1$, $a_i' = -a_i$, $i = 1, 2, \cdots, K$, and $r_n$ is a normalized covariance coefficient of the stationary input,

$$r_n = E\{x(k)x(k+n)\}/\sigma_x^2. \tag{27}$$

In evaluating the second and third terms of (24), we ignore correlation between different quantizing-noise samples and correlations between quantizing-noise samples and samples of the codec input. Thus we use (18) and the approximation

$$E\{n_M(k)x(j)\} \approx 0. \tag{28}$$

This allows us to write

$$\sigma_\xi^2 = G^{-1}\sigma_x^2 + a_P E\{n_M^2(k)\}, \tag{29}$$

where we define

$$a_P = \sum_{i=1}^{K} a_i^2. \tag{30}$$

The noise component of (29), which depends on the quantizer overload point and on the statistical properties of $\xi(k)$, is analyzed in Section V, where we restrict our attention to granular quantizing noise and derive $\sigma_q^2(B)$, the noise power of a $B$-bit quantizer with unity overload point. If the actual overload point is $\xi_{max}$, the noise power of the $M$-bit minimal quantizer is

$$E\{n_M^2(k)\} = \xi_{max}^2 \sigma_q^2(M) \approx \Delta_M^2/12. \tag{31}$$

The quantizer step size is

$$\Delta_M = \xi_{max} 2^{-(M-1)}, \tag{32}$$

and the approximation would be exact if $n_M(k)$ were uniformly distributed over the range $-\Delta_M/2$ to $\Delta_M/2$. Table II presents $\sigma_q^2(B)$ numerically and indicates the fractional error due to the above approximation. A parameter in Table II is the dimensionless load factor

Table II—Quantizing noise

| Load Factor | | Gaussian | | Exponential | |
|---|---|---|---|---|---|
| | | 2 bits | 3 bits | 2 bits | 3 bits |
| 1.78 | Noise power $\sigma_q^2(B)$ | 0.02066 | 0.005198 | 0.02189 | 0.005276 |
| | Approximation error* | −0.01 | 0.00 | 0.05 | 0.01 |
| 3.16 | Noise power $\sigma_q^2(B)$ | 0.02082 | 0.005207 | 0.02394 | 0.005419 |
| | Approximation error* | 0.00 | 0.00 | 0.13 | 0.04 |
| 5.62 | Noise power $\sigma_q^2(B)$ | 0.02292 | 0.005209 | 0.02885 | 0.005836 |
| | Approximation error* | 0.09 | 0.00 | 0.28 | 0.11 |

* Relative error of the approximation $\sigma_q^2(B) \approx 2^{-2B}/3$.

$$L = \xi_{max}/\sigma_\xi. \tag{33}$$

Combining (29), (31), and (33), we arrive at

$$\sigma_\xi^2 = G^{-1}\sigma_x^2 + a_P L^2 \sigma_q^2(M)\sigma_\xi^2, \tag{34}$$

or the quantity we set out to derive:

$$\sigma_x^2/\sigma_\xi^2 = G[1 - a_P L^2 \sigma_q^2(M)]. \tag{35}$$

## V. QUANTIZATION NOISE AND TRANSMISSION NOISE IN PCM

### *5.1 Granular and overload conditions*

To analyze (19), we study, statistically, the quantization and transmission of the DPCM difference signal $\xi(k)$. In this type of study it is customary to separate the quantizing error into two components: overload distortion and granular noise. In speech communication this distinction is valuable for predicting subjective quality.[13,14] Moreover, in analyzing DPCM the distinction is essential because, except for a codec with an ideal integrator,[15] ($F(z) = z^{-1}$, which is pathologically vulnerable to transmission errors), there is no theory for computing the mean-square slope-overload distortion. Thus our analysis separates the transmission of clipped samples of $\xi(k)$ from samples subject to granular distortion. Our theory pertains only to the transmission of unclipped samples. For those samples we add two different distortions, quantizing noise and noise due to transmission errors. Unlike slope overload, both of these impairments are essentially uncorrelated with the signal. This gives us confidence that the mean-square sum is a reasonable quality measure.

Formally, we rewrite (19) as

$$\sigma_e^2 = p_{ov}E\{e^2(k) \mid |\xi(k)| > \xi_{max}\} + p_{gr}E\{e^2(k) \mid |\xi(k)| \leq \xi_{max}\}, \tag{36}$$

where $\xi(k)$ is the quantizer input and $\xi_{max}$ is the overload point of the uniform DPCM quantizer. The probability of overload is $p_{ov}$ and $p_{gr} = 1 - p_{ov}$ is the probability of granular quantization. By definition, the

quantizer is overloaded at time $k$ if the quantization error exceeds half of a quantization step, i.e., if

$$|q_D(k) - \xi(k)| > \Delta_D/2. \tag{37}$$

The step size of the $D$-bit uniform quantizer is

$$\Delta_D = \xi_{\max} 2^{-(D-1)} = \Delta_M/2^{D-M}. \tag{38}$$

Our remaining analysis will be confined to the second expectation on the right side of (36) and in particular to the ratio,

$$s/n = \sigma_x^2/E\{e^2(k) \mid |\xi(k)| \leqslant \xi_{\max}\}. \tag{39}$$

To be concise in the remainder of this paper, we will omit the granular condition, $|\xi(k)| \leqslant \xi_{\max}$, from our notation of expected values.

### *5.2 Transmission model, normalized quantizer*

To facilitate numerical evaluation of s/n's, we will present three tables of normalized error terms. The normalization relates these errors to a quantizer with a unity overload point and an input with probability density function $p_u(\cdot)$. If the quantizer of interest has an overload point of $\xi_{\max}$ and the input has the probability density $p_\xi(\cdot)$, the relevant errors are table entries scaled by $\xi_{\max}^2$. The two probability densities are related by

$$p_u(u) = \xi_{\max} p_\xi(\xi_{\max} u). \tag{40}$$

To confine our attention to the granular quantization condition, we perform our averages with respect to the conditional probability density

$$p_{gr}(u) = \frac{p_u(u)}{\int_{-1}^{1} p_u(u)du}, \qquad |u| \leqslant 1$$

$$= 0 \qquad |u| > 1. \tag{41}$$

The model is illustrated in Fig. 7. The signal $u = \xi/\xi_{\max}$ is processed by a $B$-bit analog-to-digital converter with overload point 1 and step size

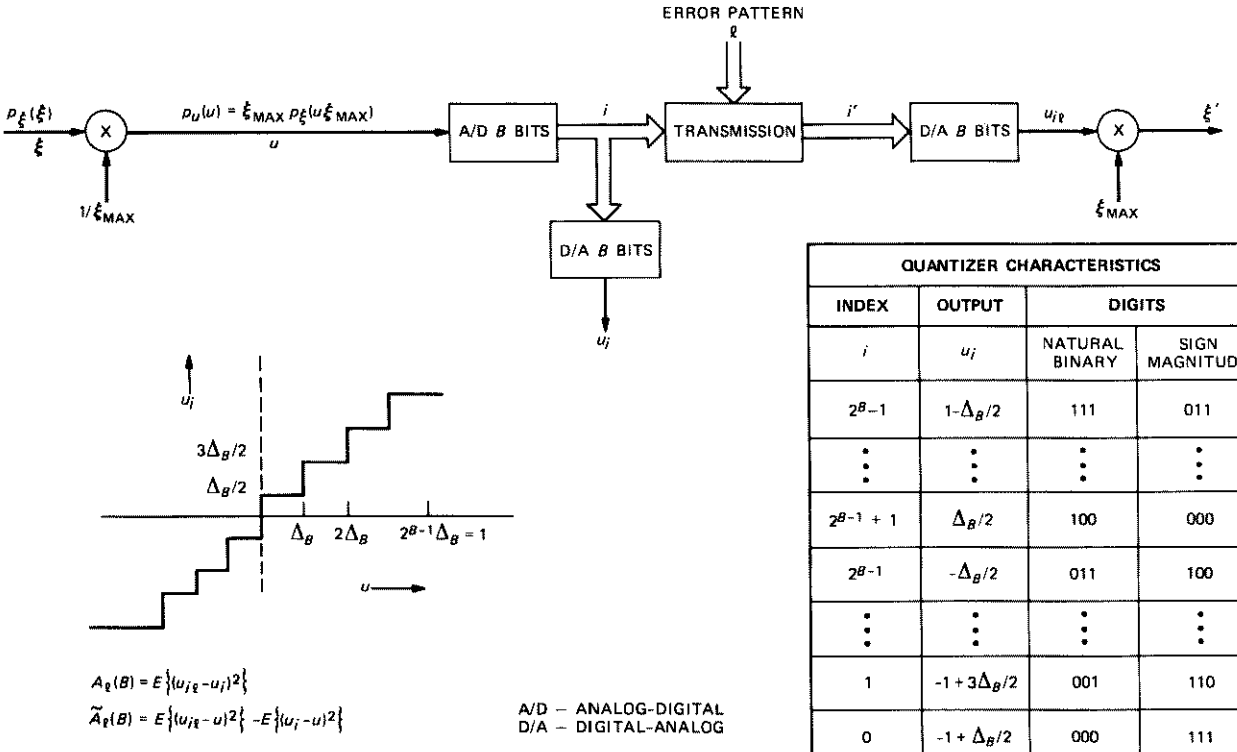
$$\Delta_B = 2^{-(B-1)}. \tag{42}$$

The digital output of the a/d is $i$, and the corresponding quantized signal is $u_i$, which is related to $u$ by the graph in Fig. 7 and by

$u_i = -1 + (i + 0.5)\Delta_B$ when

$$-1 + i\Delta_B \leqslant u < -1 + (i + 1)\Delta_B, \qquad i = 0, 1, \cdots, 2^B - 1. \tag{43}$$

In Fig. 7, the $B$-bit code word $i$ is transmitted, and $i'$ is received,




| QUANTIZER CHARACTERISTICS | | | |
|---|---|---|---|
| INDEX | OUTPUT | DIGITS | |
| $i$ | $u_i$ | NATURAL BINARY | SIGN MAGNITUDE |
| $2^B-1$ | $1-\Delta_B/2$ | 111 | 011 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| $2^{B-1}+1$ | $\Delta_B/2$ | 100 | 000 |
| $2^{B-1}$ | $-\Delta_B/2$ | 011 | 100 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | $-1+3\Delta_B/2$ | 001 | 110 |
| 0 | $-1+\Delta_B/2$ | 000 | 111 |

Fig. 7—Model for analyzing quantizing noise and the effects of transmission errors. The graph and table define the quantizer and two binary number representations.

with the transformation of $i$ to $i'$ characterized by a binary-error pattern with index $l$, $l$ being an integer in the range 0, $2^B - 1$. To relate the error effect to $l$, we refer to the natural-binary representation of $l$ and specify that a 1 in the $b$th *least*-significant position of $l$ causes an inversion of the $b$th *most*-significant* bit in the binary representation of $i$. Thus $l = 0$ refers to error-free transmission ($i = i'$); $l = 1$ refers to an error only in the most significant bit; $l = 5$ refers to errors in the first and third most significant bits; etc.

With $u$ the quantizer input and $l$ the binary error pattern, we denote the received sample in Fig. 7, $u_{il}$. It is helpful to separate the complete error $u_{il} - u$ into quantization-noise and transmission-noise components as follows,

$$u_{il} - u = (u_i - u) + (u_{il} - u_i). \tag{44}$$

### 5.3 Conditional expectations of transmission-error effects

#### 5.3.1 The general approach

Our goal is to evaluate the mean-square of (44) over the joint distribution of input statistics and binary-error patterns. The key to our analysis is the definition of $A$ factors, which are conditional mean-square errors, each related to a specific binary-error pattern, $l$. By analyzing these conditional errors, we separate the effects of source characteristics from the effects of transmission characteristics. The source effects are embodied in the $A$ factors; the transmission effects are embodied in probabilities of error patterns. These probabilities govern the weighted addition of the $A$ factors to produce the final result.

This approach to analyzing transmission impairments was introduced by Rydbeck and Sundberg,[10-12] who were mainly concerned with quantizers with 6 to 8 bits/sample. This high resolution admitted various approximations that are inaccurate in the 2- to 4-bit quantizers of greatest interest for embedded DPCM transmission. Thus we proceed to a precise calculation of two types of $A$ factors: conditional expectations related to the isolated effects of digital transmission errors and conditional expectations that include correlations between transmission errors and quantizing noise. In high-resolution quantizers this correlation is negligible, and the two types of $A$ factors are essentially equal.

#### 5.3.2 Analysis

To compute the mean-square value of (44), conditioned on error pattern $l$, we will identify three important quantities: $\sigma_q^2(B)$, the

* This reversal of the bit ordering of $l$ relative to the binary representation of $i$ will facilitate bookkeeping in subsequent computations.

granular-noise power of a $B$-bit quantizer; $A_l(B)$, the mean-square effect of error pattern $l$ on the quantized signal $u_i$; and $\tilde{A}_l(B)$, the overall effect of error pattern $l$ on the mean-square error of the analog output $u_{il}$.

To derive computationally convenient expressions for $\sigma_q^2(B)$, $A_l(B)$, and $\tilde{A}_l(B)$, we defined the integrals

$$p_i = \int_{v_i}^{v_{i+1}} p_{gr}(u)du \tag{45}$$

$$q_i = \int_{v_i}^{v_{i+1}} (u_i - u)p_{gr}(u)du \tag{46}$$

$$\sigma_{gr}^2 = \int_{-1}^{1} u^2 p_{gr}(u)du, \tag{47}$$

in which $v_i$ is the lower boundary and $v_{i+1}$ is the upper boundary of quantizing interval $i$:

$$v_i = -1 + i2^{-(B-1)}; \qquad i = 0, 1, \cdots, 2^B. \tag{48}$$

The first integral (45) is the probability of using interval $i$. The second integral (46) is the average quantization error in interval $i$. If $B$ is large, $\Delta_B$ is small, and $q_i \approx 0$ because $u_i$ is in the center of the quantization interval. The third integral (47) is the mean-square signal when the quantizer is in the granular condition.

Now we write the definitions followed by computational formulas for the quantizing noise and the effects of error pattern $l$:

$$\sigma_q^2(B) = E(u_i - u)^2 = \sigma_{gr}^2 + \sum_{i=0}^{2^B-1} (2q_i u_i - p_i u_i^2) \tag{49}$$

$$A_l(B) = E(u_{il} - u_i)^2 = \sum_{i=0}^{2^B-1} p_i(u_{il} - u_i)^2 \tag{50}$$

$$\tilde{A}_l(B) = E(u_{il} - u)^2 - E(u_i - u)^2 = A_l(B) + 2\sum_{i=0}^{2^B-1} q_i(u_{il} - u_i). \tag{51}$$

$\tilde{A}_l(B)$, the difference between the total noise and the quantizing noise, includes the correlation between quantization effects and transmission-error effects. In multibit quantizers ($B > 4$) this correlation is small, and $\tilde{A}_l(B) \approx A_l(B)$, an assumption inherent in previous work on PCM. Because low-resolution quantizers are of interest in DPCM, we take account of this correlation in our present work.

Finally, we combine (49), (50), and (51) to write the mean-square value of (44)

$$\epsilon_l^2 = E(u_{il} - u)^2 = \sigma_q^2(B) + \tilde{A}_l(B). \tag{52}$$

### 5.3.3 Computations

Table III displays formulas for $p_i$, $q_i$, and $\sigma_{gr}^2$ that apply to inputs with Gaussian and exponential probability density functions. Because the input, $u$, of the normalized quantizer is related to the input, $\xi$, of the quantizer with overload point $\xi_{max}$ by $u = \xi/\xi_{max}$, we have

$$\sigma_u^2 = \sigma_\xi^2/\xi_{max}^2 = 1/L^2, \tag{53}$$

where $L$ is the dimensionless load factor defined in (33). Because $L$ is a familiar quantizer design quantity, we have written the formulas in Table III as functions of $L$.

With the formulas in Table III, it is a simple matter to compute $\sigma_q^2(B)$ precisely. However, for $B \geq 4$ the approximation

$$\sigma_q^2(B) \approx \Delta_B^2/12 = 2^{-2B}/3 \tag{54}$$

is very accurate (within 3 percent of the exact value for $L \leq 5.6$). For $B = 2$ and 3, Table II shows the exact values of $\sigma_q^2(B)$ and the approximation errors

$$[\sigma_q^2(B) - \Delta_B^2/12]/\sigma_q^2(B) \tag{55}$$

for $L = 1.78, 3.16, 5.62$ ($L^2 = 10 \pm 5$ dB).

To compute $A_l(B)$, $\tilde{A}_l(B)$, it is necessary to know $u_{il} - u_i$, which depends on the binary number representation of $u_i$.

### 5.3.4 Binary number representations

We consider two representations: natural-binary and sign-magnitude, both defined in Fig. 7. Although in general the $A_l(B)$ and $\tilde{A}_l(B)$ depend on $p_i$ and $q_i$, there are some special cases that are important and illuminating. For example, in the natural-binary code, the single-error $A$ factors are independent of the signal statistics and of the quantizer. An error in the most significant bit causes $u_{il} - u_i = \pm 1$ provided it is the only error in the $B$-bit code word. Thus $A_1(B) = 1$. Likewise, any isolated error in the second most significant bit causes an output error of $\pm 1/2$, and in general a single binary error in position $b$ causes the mean-square error

$$A_l(B) = (1/4)^{b-1}; \qquad l = 2^{(b-1)}. \tag{56}$$

In the sign-magnitude code, isolated binary errors in positions $b = 2, 3, \cdots, B$ have the same effects as corresponding errors in the natural-binary code. However, an isolated error in the most significant position transforms $u_i$ to $u_{il} = -u_i$. The mean-square effect is

$$A_l(B) = 4E\{u_i^2\} \approx 4\sigma_u^2. \tag{57}$$

The approximation becomes more and more precise as $B$ increases.

Table III—Formulas for computing A factors and quantizing noise B-bit transmission

**General**

$v_i = -1 + i2^{-(B-1)}; u_i = v_i + 2^{-B}$

| | Gaussian | Exponential |
|---|---|---|
| Error integral, $Q(x)$ | $\int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du$ | |
| Probability density, $p(u)$ | $\frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(uL)^2}{2}\right]$ | $\frac{L}{\sqrt{2}} e^{-\lvert u \rvert L\sqrt{2}}$ |
| Conditional (granular) density, $p_{gr}(u)$ | $\frac{p(u)}{1-2Q(L)}; \lvert u \rvert \leq 1$<br>$0 \quad ; \lvert u \rvert > 1$ | $\frac{p(u)}{1-e^{-L\sqrt{2}}}; \lvert u \rvert \leq 1$<br>$0 \quad ; \lvert u \rvert > 1$ |
| Interval probability, $p_i$ | $\frac{Q(Lv_i) - Q(Lv_{i+1})}{1-2Q(L)}$ | $\frac{e^{-v_iL\sqrt{2}} - e^{-v_{i+1}L\sqrt{2}}}{2(1-e^{-L\sqrt{2}})}; 2^{B-1} \leq i \leq 2^B - 1$<br>$p_{2^B-1-i} \quad ; 0 \leq i \leq 2^{B-1} - 1$ |
| Interval mean noise, $q_i$ | $p_iu_i - \frac{e^{-(Lv_i)^2/2} - e^{-(Lv_{i+1})^2/2}}{L\sqrt{2\pi}[1-2Q(L)]}$ | $p_iu_i - \frac{e^{-v_iL\sqrt{2}}\left(v_i + \frac{1}{L\sqrt{2}}\right) - e^{-v_{i+1}L\sqrt{2}}\left(v_{i+1} + \frac{1}{L\sqrt{2}}\right)}{2(1-e^{-L\sqrt{2}})}; 2^{B-1} \leq i \leq 2^B - 1$<br>$-q_{2^B-1-i} \quad ; 0 \leq i \leq 2^{B-1} - 1$ |
| Conditional mean square, $\sigma_{gr}^2$ | $\frac{1-2Q(L) - L\sqrt{\frac{2}{\pi}} e^{-L^2/2}}{L^2[1-2Q(L)]}$ | $\frac{1-e^{-L\sqrt{2}} - e^{-L\sqrt{2}}(L^2 + L\sqrt{2})}{L^2(1-e^{-L\sqrt{2}})}$ |

### 5.4 The s/n of the embedded codec

Referring to (52) and Fig. 7, we have the mean-square difference between $\xi'$ and $\xi$, over all possible error patterns

$$E\{(\xi' - \xi)^2\} = \xi_{\max}^2 E\{\epsilon_l^2\} = \xi_{\max}^2 \left[\sigma_q^2(B) + \sum_{l=1}^{2^B-1} P(l)\hat{A}_l(B)\right], \quad (58)$$

where $P(l)$ is the probability of error pattern $l$. The effect of the transmission errors on the quantized version of $\xi$ is $\xi_{\max}^2 \sum_{l=1}^{2^B-1} P(l)A_l(B)$. Returning to (19), we have two expectations: the first is the combined (quantizing and transmission) noise of a $D$-bit signal; the second expectation is the noise due to transmission errors in the $M$-bit minimal signal. Thus, we can write (19) as

$$\sigma_e^2 = \xi_{\max}^2 \left[\sigma_q^2(D) + \sum_{l=1}^{2^D-1} P(l)\hat{A}_l(D) + b_P \sum_{l=1}^{2^M-1} P(l)A_l(M)\right], \quad (59)$$

where $\xi_{\max}^2 = L^2\sigma_\xi^2$ is related to $\sigma_x^2$ by

$$\xi_{\max}^2 = \frac{L^2\sigma_x^2}{G[1 - a_P L^2\sigma_q^2(M)]}. \quad (60)$$

The s/n, which is the principal subject of this paper, is, therefore,

$$s/n = \frac{G[1 - a_P L^2\sigma_q^2(M)]}{L^2[\sigma_q^2(D) + \sum_{l=1}^{2^D-1} P(l)\hat{A}_l(D) + b_P \sum_{l=1}^{2^M-1} P(l)A_l(M)]}. \quad (61)$$

With the exception of the two summations in the denominator, all of the quantities in (61) are properties of the input signal and the codec design parameters. These summations,

$$\sigma_{qt}^2 = \sum_{l=1}^{2^D-1} P(l)\hat{A}_l(D) + b_P \sum_{l=1}^{2^M-1} P(l)A_l(M) \quad (62)$$

comprise the effects of transmission errors on the performance of embedded DPCM. We analyze them in Section VI.

## VI. TRANSMISSION EFFECTS, BINARY-ERROR PROBABILITIES

The $2^D$ probabilities, $P(l)$, of binary-error patterns are properties of the digital transmission system, which includes a modulator, a channel, a demodulator, possibly a codec for forward error correction, and possibly a means for combining different versions (diversity branches) of the received signal. Depending on these components, the $P(l)$ exhibit properties that facilitate evaluation of the sums in (61). In the following subsections we consider three paradigms: (1) random errors with statistically independent transmission of all bits; (2) slow fading

with the bit-error probability constant over each code word, but independent from word to word; and (3) channel coding that makes all error patterns equally likely.

### 6.1 Random binary errors

Errors in all bits are statistically independent of each other and occur with probability, $P$. The probability of error pattern $l$ depends only on $w$, the Hamming weight of $l$, i.e., the number of ones in the $B$-bit binary representation of $l$. Thus,

$$P(l) = P^w(1 - P)^{B-w} = P^w \sum_{j=0}^{B-w} (-1)^j \binom{B-w}{j} P^j. \tag{63}$$

This expansion leads us to express the summations in (61) as polynomials in $P$. The coefficients of the polynomial involve the sums of all $A$ factors with a fixed weight, $w$. Let us denote these sums $S_w(B)$ where, for example,

$$\begin{aligned} S_1(B) &= A_1(B) + A_2(B) + A_4(B) + \cdots + A_{2^{B-1}}(B); \\ S_2(B) &= A_3(B) + \cdots + A_{2^{B-1}+2^{B-2}}(B), \end{aligned} \tag{64}$$

and in general,

$$S_w(B) = \sum_{l_w} A_l(B); \qquad \tilde{S}_w(B) = \sum_{l_w} \tilde{A}_l(B), \tag{65}$$

where $l_w$ is the set of all error patterns with Hamming weight, $w$. Combining (63) and (65), we can write

$$\sum_{l=1}^{2^B-1} P(l)A_l(B) = \sum_{w=1}^{B} \sum_{j=0}^{B-w} P^{j+w}(-1)^j \binom{B-w}{j} S_w(B). \tag{66}$$

The summations in (66) can be manipulated to form

$$\begin{aligned} \sum_{l=1}^{2^B-1} P(l)A_l(B) &= \sum_{w=1}^{B} P^w \sum_{j=1}^{w} \binom{B-j}{w-j} (-1)^{w-j} S_j(B) \\ &= \sum_{w=1}^{B} P^w T_w(B), \end{aligned} \tag{67}$$

where we define

$$\begin{aligned} T_w(B) &= \sum_{j=1}^{w} \binom{B-j}{w-j} (-1)^{w-j} S_j(B); \\ \tilde{T}_w(B) &= \sum_{j=1}^{w} \binom{B-j}{w-j} (-1)^{w-j} \tilde{S}_j(B). \end{aligned} \tag{68}$$

For the natural-binary and sign-magnitude representations we have

discovered and proved that for any input probability distribution, $T_w(B) = \hat{T}_w(B) = 0$ for $w \geq 3$. Thus the tranmission term in (61) is

$$\sigma_{qt}^2 = \sum_{l=1}^{2^{D}-1} P(l)\hat{A}_l(D) + b_P \sum_{l=1}^{2^{M}-1} P(l)A_l(M)$$
$$= \sum_{w=1}^{2} P^w[\hat{T}_w(D) + b_P T_w(M)] \tag{69}$$

This formula is valid for channels with random binary errors, and Table IV presents values of $T_1(B)$, $T_2(B)$, $\hat{T}_1(B)$, and $\hat{T}_2(B)$ for three quantizer load factors, $B$ = 2–6 bits and Gaussian and exponential inputs.

Table IV—Error constants for uncoded transmission

| $B$ | $T_1(B)$ | $T_2(B)$ | $\hat{T}_1(B)$ | $\hat{T}_2(B)$ | $T_1(B)$ | $T_2(B)$ | $\hat{T}_1(B)$ | $\hat{T}_2(B)$ |
|---|---|---|---|---|---|---|---|---|
| | | Sign Magnitude | | | | Natural Binary | | |
| | | | | Gaussian Inputs, Load Factor 1.78 | | | | |
| 2 | 1.146 | −0.146 | 1.085 | −0.137 | 1.250 | −0.354 | 1.194 | −0.354 |
| 3 | 1.186 | −0.186 | 1.170 | −0.183 | 1.313 | −0.439 | 1.298 | −0.439 |
| 4 | 1.196 | −0.196 | 1.192 | −0.195 | 1.328 | −0.461 | 1.325 | −0.461 |
| 5 | 1.198 | −0.198 | 1.197 | −0.198 | 1.332 | −0.466 | 1.331 | −0.466 |
| 6 | 1.199 | −0.199 | 1.198 | −0.199 | 1.333 | −0.467 | 1.333 | −0.467 |
| | | | | Gaussian Inputs, Load Factor 3.16 | | | | |
| 2 | 0.725 | 0.275 | 0.670 | 0.219 | 1.250 | −0.775 | 1.167 | −0.775 |
| 3 | 0.726 | 0.274 | 0.711 | 0.262 | 1.313 | −0.899 | 1.292 | −0.899 |
| 4 | 0.726 | 0.274 | 0.723 | 0.271 | 1.328 | −0.930 | 1.323 | −0.930 |
| 5 | 0.726 | 0.274 | 0.726 | 0.273 | 1.332 | −0.938 | 1.331 | −0.938 |
| 6 | 0.727 | 0.273 | 0.726 | 0.273 | 1.333 | −0.940 | 1.333 | −0.940 |
| | | | | Gaussian Inputs, Load Factor 5.62 | | | | |
| 2 | 0.510 | 0.490 | 0.506 | 0.273 | 1.250 | −0.990 | 1.137 | −0.990 |
| 3 | 0.460 | 0.540 | 0.467 | 0.485 | 1.313 | −1.165 | 1.292 | −1.165 |
| 4 | 0.460 | 0.540 | 0.461 | 0.527 | 1.328 | −1.196 | 1.323 | −1.196 |
| 5 | 0.460 | 0.540 | 0.460 | 0.537 | 1.332 | −1.204 | 1.331 | −1.204 |
| 6 | 0.460 | 0.540 | 0.460 | 0.539 | 1.333 | −1.206 | 1.333 | −1.206 |
| | | | | Exponential Inputs, Load Factor 1.78 | | | | |
| 2 | 0.943 | 0.057 | 0.898 | 0.000 | 1.250 | −0.557 | 1.176 | −0.557 |
| 3 | 0.958 | 0.042 | 0.950 | 0.024 | 1.313 | −0.667 | 1.296 | −0.667 |
| 4 | 0.964 | 0.036 | 0.962 | 0.031 | 1.328 | −0.692 | 1.324 | −0.692 |
| 5 | 0.966 | 0.034 | 0.965 | 0.033 | 1.332 | −0.698 | 1.331 | −0.698 |
| 6 | 0.966 | 0.034 | 0.966 | 0.033 | 1.333 | −0.700 | 1.333 | −0.700 |
| | | | | Exponential Inputs, Load Factor 3.16 | | | | |
| 2 | 0.693 | 0.307 | 0.660 | 0.168 | 1.250 | −0.807 | 1.147 | −0.807 |
| 3 | 0.667 | 0.333 | 0.670 | 0.285 | 1.313 | −0.958 | 1.291 | −0.958 |
| 4 | 0.666 | 0.334 | 0.668 | 0.321 | 1.328 | −0.990 | 1.323 | −0.990 |
| 5 | 0.666 | 0.334 | 0.667 | 0.330 | 1.332 | −0.998 | 1.331 | −0.998 |
| 6 | 0.666 | 0.334 | 0.667 | 0.333 | 1.333 | −1.000 | 1.333 | −1.000 |
| | | | | Exponential Inputs, Load Factor 5.62 | | | | |
| 2 | 0.537 | 0.463 | 0.527 | 0.205 | 1.250 | −0.963 | 1.111 | −0.963 |
| 3 | 0.465 | 0.535 | 0.492 | 0.430 | 1.313 | −1.160 | 1.287 | −1.160 |
| 4 | 0.459 | 0.541 | 0.468 | 0.512 | 1.328 | −1.198 | 1.323 | −1.198 |
| 5 | 0.458 | 0.542 | 0.461 | 0.534 | 1.332 | −1.206 | 1.331 | −1.206 |
| 6 | 0.458 | 0.542 | 0.459 | 0.540 | 1.333 | −1.208 | 1.333 | −1.208 |

### 6.2 *Slow fading*

Now the binary-error probability is a random variable that is constant over each code word but varies from word to word. In this case the effects of digital errors can be calculated as in (69) but with the average values $\overline{P^w}$ replacing $P^w$. These averages are computed over the distributions of channel s/n's that govern the random fluctuation of $P$ from one code word to the next.

### 6.3 *Error-correcting codes*

To analyze the performance of embedded DPCM protected by an error-correcting channel code, we make three simplifying approximations. The first one, which pertains to the error-correcting code, states that when there is a decoding error, all error patterns are equally likely. Thus we assume that if the $C$ most significant DPCM bits are protected by the code,

$$P(l) = \frac{1}{2^C - 1} P_W = \frac{1}{2^{C-1}} P_e, \qquad l = 1, 2, \cdots, 2^C - 1, \tag{70}$$

where $P_W$ is the word-error probability and $P_e$ is the binary-error probability of the channel code. They are related by

$$P_W = \frac{2^C - 1}{2^{C-1}} P_e. \tag{71}$$

The other two approximations apply when $C < D$, so that the $C$ most significant DPCM bits are protected and the other $D$–$C$ bits are uncoded. To simplify computations for this case, we (1) ignore simultaneous errors in the protected and unprotected parts of the $D$-bit word and (2) ignore multiple errors in the unprotected part. We consider separately three different relationships among $C$, the number of coded bits; $D$, the length of the entire DPCM code word; and $M$, the number of bits in the minimal quantizer.

#### 6.3.1 *Entire code word protected ($M \leq D = C$)*

In this case $P(l)$ may be calculated according to (70) for all $D$-bit error patterns, $l = 1, 2, \cdots, 2^{D-1}$. This value of $P(l)$ is constant throughout the first sum in (62). In the second sum we have the probability of $M$-bit error patterns. For each $M$-bit error pattern there are $2^{D-M}$ $D$-bit patterns. Hence $P(l)$ in the second sum of (62) is higher by the factor $2^{D-M}$ than $P(l)$ in the first sum. Because each sum in (61) is a constant probability times a sum of $A$ factors, we write

$$\sigma_{qt}^2 = \frac{1}{2^{C-1}} P_e[\hat{A}_{sum}^{(D)}(D) + b_P 2^{D-M} A_{sum}^{(M)}(M)], \tag{72}$$

where we define the sum of the first $2^C - 1$ $A$ factors

$$\tilde{A}^{(C)}_{\text{sum}}(D) = \sum_{l=1}^{2^C-1} \tilde{A}_l(D); \qquad A^{(C)}_{\text{sum}}(D) = \sum_{l=1}^{2^C-1} A_l(D). \tag{73}$$

Table V contains numerical values of $\tilde{A}_{\text{sum}}$ and $A_{\text{sum}}$ for the sets of conditions of interest to us here.

#### 6.3.2 *Entire minimal code word, parts of the supplemental code word protected ($M \leq C < D$)*

In this event we assume that all of the unprotected bits have the binary-error probability $P$ and that as before the protected bits have binary-error probability $P_e$. Furthermore, we set to 0 the probability of simultaneous errors in the protected and unprotected parts of the code word. (These errors occur with probability related to $P_eP$.) We also set to 0 the probability of multiple errors in the unprotected part of the code word (which occur with probability less than $P^2$). Thus we break the first sum in (61) into two parts. The first part accounts for

Table V—Error constraints $A^{(C)}_{\text{sum}}(B)$ and $\tilde{A}^{(C)}_{\text{sum}}(B)$ *for coded transmission*

| Load Factor | | $B=2$ | | $B=3$ | | | $B=4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $C=1$ | $C=2$ | $C=1$ | $C=2$ | $C=3$ | $C=1$ | $C=2$ | $C=3$ | $C=4$ |
| | | | | | Gaussian Inputs, Sign Magnitude | | | | | |
| 1.78 | | 0.90 | 2.15 | 0.87 | 2.11 | 4.37 | 0.87 | 2.10 | 4.35 | 8.78 |
| 3.16 | $A$ | 0.47 | 1.72 | 0.41 | 1.56 | 3.45 | 0.40 | 1.52 | 3.37 | 6.91 |
| 5.62 | | 0.26 | 1.51 | 0.15 | 1.11 | 2.92 | 0.13 | 1.05 | 2.77 | 5.84 |
| 1.78 | | 0.84 | 2.03 | 0.86 | 2.08 | 4.31 | 0.86 | 2.09 | 4.34 | 8.75 |
| 3.16 | $\tilde{A}$ | 0.39 | 1.56 | 0.39 | 1.52 | 3.37 | 0.39 | 1.51 | 3.35 | 6.86 |
| 5.62 | | 0.15 | 1.28 | 0.13 | 1.07 | 2.84 | 0.13 | 1.04 | 2.75 | 5.80 |
| | | | | | Gaussian Inputs, Natural Binary | | | | | |
| 1.78 | | 1.00 | 2.15 | 1.00 | 2.15 | 4.37 | 1.00 | 2.15 | 4.37 | 8.78 |
| 3.16 | $A$ | 1.00 | 1.72 | 1.00 | 1.72 | 3.45 | 1.00 | 1.72 | 3.45 | 6.91 |
| 5.62 | | 1.00 | 1.51 | 1.00 | 1.51 | 2.92 | 1.00 | 1.51 | 2.92 | 5.84 |
| 1.78 | | 0.95 | 2.03 | 0.99 | 2.12 | 4.31 | 1.00 | 2.14 | 4.36 | 8.75 |
| 3.16 | $\tilde{A}$ | 0.89 | 1.56 | 0.97 | 1.68 | 3.37 | 0.99 | 1.71 | 3.43 | 6.86 |
| 5.62 | | 0.78 | 1.28 | 0.95 | 1.46 | 2.84 | 0.99 | 1.50 | 2.90 | 5.80 |
| | | | | | Exponential Inputs, Sign Magnitude | | | | | |
| 1.78 | | 0.69 | 1.94 | 0.65 | 1.81 | 3.92 | 0.64 | 1.78 | 3.85 | 7.86 |
| 3.16 | $A$ | 0.44 | 1.69 | 0.35 | 1.41 | 3.33 | 0.34 | 1.35 | 3.20 | 6.66 |
| 5.62 | | 0.29 | 1.54 | 0.15 | 1.09 | 2.93 | 0.13 | 0.99 | 2.69 | 5.83 |
| 1.78 | | 0.62 | 1.80 | 0.63 | 1.77 | 3.85 | 0.63 | 1.77 | 3.83 | 7.82 |
| 3.16 | $\tilde{A}$ | 0.34 | 1.49 | 0.33 | 1.37 | 3.25 | 0.33 | 1.34 | 3.18 | 6.62 |
| 5.62 | | 0.15 | 1.26 | 0.13 | 1.03 | 2.83 | 0.12 | 0.97 | 2.67 | 5.79 |
| | | | | | Exponential Inputs, Natural Binary | | | | | |
| 1.78 | | 1.00 | 1.94 | 1.00 | 1.94 | 3.92 | 1.00 | 1.94 | 3.92 | 7.86 |
| 3.16 | $A$ | 1.00 | 1.69 | 1.00 | 1.69 | 3.33 | 1.00 | 1.69 | 3.33 | 6.66 |
| 5.62 | | 1.00 | 1.54 | 1.00 | 1.54 | 2.93 | 1.00 | 1.54 | 2.93 | 5.83 |
| 1.78 | | 0.90 | 1.80 | 0.97 | 1.91 | 3.85 | 0.99 | 1.93 | 3.90 | 7.82 |
| 3.16 | $\tilde{A}$ | 0.83 | 1.49 | 0.95 | 1.64 | 3.25 | 0.99 | 1.68 | 3.31 | 6.62 |
| 5.62 | | 0.73 | 1.26 | 0.92 | 1.46 | 2.83 | 0.98 | 1.52 | 2.90 | 5.79 |

errors in the first $C$ (protected) bits when the other $D$–$C$ bits are error free, $l = 1, 2, \cdots, 2^C - 1$. The second part accounts for single errors in the remaining $D$–$C$ bits when the first $C$ bits are error free, $l = 2^C, 2^{C+1}, \cdots, 2^{D-1}$. The result is

$$\sigma_{qt}^2 = \frac{P_e}{2^{C-1}} \hat{A}_{sum}^{(C)}(D) + P \sum_{i=C}^{D-1} \hat{A}_{2^i}(D) + b_P 2^{C-M} \frac{P_e}{2^{C-1}} A_{sum}^{(M)}(M). \quad (74)$$

In the second term we use the approximation

$$P \approx P(1 - P)^{D-C+1}(1 - P_W) \quad (75)$$

for the probability of a single error in the unprotected part of the code word. A further approximation $\hat{A}_{2^i}(D) \approx A_{2^i}(D)$ simplifies computation of the second term of (74) because, for natural-binary and sign-magnitude representations, (56) applies for $b > 1$. This allows us to derive

$$\sum_{i=C}^{D-1} A_{2^i}(D) = 4(4^{-C} - 4^{-D})/3 = A_{un}^{(C)}(D). \quad (76)$$

Thus for $M \leq C < D$ we have the formula

$$\sigma_{qt}^2 = \frac{1}{2^{C-1}} P_e[\hat{A}_{sum}^{(C)}(D) + b_P 2^{C-M} A_{sum}^{(M)}(M)] + P A_{un}^{(C)}(D). \quad (77)$$

#### 6.3.3 Part of the minimal code word protected ($C < M \leq D$)

Just as we decomposed the first sum in (61) into two parts in the previous case, we similarly decompose the second sum when some of the $M$ minimal bits are unprotected. The result is

$$\sigma_{qt}^2 = \frac{P_e}{2^{C-1}} [\hat{A}_{sum}^{(C)}(D) + b_P A_{sum}^{(C)}(M)] + P[A_{un}^{(C)}(D) + b_P A_{un}^{(C)}(M)]. \quad (78)$$

## VII. NUMERICAL RESULTS

The useful computational formulas (61), (62), (69), (72), (77), and (78) are summarized in Table VI. In this section we apply these formulas to illustrate some of the properties of embedded DPCM and its relationship to conventional DPCM.

### 7.1 Source characteristics

All of our numerical results pertain to a Gauss-Markov input signal with adjacent-sample correlation $r_1 = 0.85$. The codec uses single integration with coefficient $a_1 = 0.85$ and the load factor, $L = \sqrt{10}$. For this configuration the coding gain is $G = 3.6$. If the embedded codec has a minimal quantizer with $M = 2$ bits, $C_{source}$ in Table VI is 0.31. For conventional DPCM $C_{source} = 0.35$ with 3 bits/sample and

Table VI—Signal-to-noise ratio of embedded DPCM

$$s/n = \frac{C_{source}}{\sigma_q^2(D) + \sigma_{qt}^2};\ C_{source} = \frac{G[1 - a_P L^2 \sigma_q^2(M)]}{L^2}$$

(a) Notation

| Symbol | Description | General Formula | Formula for Single Integration |
|---|---|---|---|
| $G$ | Coding gain | (26) | $(1 - 2a_1 r_1 + a_1^2)^{-1}$ |
| $a_P$ | Predictor gain | (30) | $a_1^2$ |
| $L$ | Load factor | (33) | |
| $M$ | Minimal codec bits | | |
| $D$ | Transmitted bits | | |
| $b_P$ | $1 + b_P$ is integrator gain | (20), (21) | $a_1^2/(1 - a_1^2)$ |
| $r_1$ | Adjacent-sample autocorrelation | (27) | |
| $\sigma_q^2(B)$ | Quantizing noise | (49) | $2^{-2B}/3$ or Table II |
| $\sigma_{qt}^2$ | Transmission-error effects | (62) | |

(b) Error Formulas*

| Transmission Format | Error Effect $\sigma_{qt}^2$ |
|---|---|
| No channel code (Table IV) | $\sum_{w=1}^{2} P^w[\hat{T}_w(D) + b_P T_w(M)]$ |
| C bits coded (Table V) | |
| $M \leq D = C$ | $\frac{P_e}{2^{C-1}}[\tilde{A}_{sum}^{(D)}(D) + b_P 2^{D-M} A_{sum}^{(M)}(M)]$ |
| $M \leq C < D$ | $\frac{P_e}{2^{C-1}}(\tilde{A}_{sum}^{(C)}(D) + b_P 2^{C-M} A_{sum}^{(M)}(M)] + PA_{un}^{(C)}(D)$ |
| $C < M \leq D$ | $\frac{P_e}{2^{C-1}}[\tilde{A}_{sum}^{(C)}(D) + b_P A_{sum}^{(C)}(M)] + P[A_{un}^{(C)}(D) + b_P A_{un}^{(C)}(M)]$ |

* $P$: binary-error rate, uncoded bits; $P_e$: binary-error rate, coded bits; $A_{un}^{(C)}(D) = 4(4^{-C} - 4^{-D})/3$.

0.36 with 4 bits/sample. Thus the quantizing-noise penalty of the embedded codec is 0.54 dB when 3 bits are transmitted, and 0.68 dB when 4 bits are transmitted. As indicated in Ref. 8 these penalties increase for higher values of $L$ and $a_1$. They decrease rapidly as $M$ increases.

### *7.2 Modulation, channel, error-correcting codes*

In our numerical examples the modulation is coherent phase shift keying (CPSK) so that in a white-Gaussian-noise channel the binary error probability is

$$P = Q(\sqrt{2\rho}), \tag{79}$$

where $\rho$ is the channel s/n and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt. \tag{80}$$

Figures 3 and 4 depict performance with three different convolutional

codes. For all of them, we use the following truncated union bound to calculate binary-error probability:

$$P_e = \frac{1}{m} \sum_{i=d}^{d+4} w_i Q(\sqrt{2i\rho}). \tag{81}$$

where $m = 1, 2, 3$ for the rate 1/2, 2/3, and 3/4 codes, respectively, and the coefficients $w_i$ and the free distance, $d$, characterize the convolutional coder and decoder. The codes considered here are punctured codes[16] with constraint length 5 (16 states in the decoder memory). Table VII contains their coefficients and free distances. The combination of (79) and (81) with the formulas in Table VI produces the curves in Figs. 3, 4, 5, and 8.

For transmission environments other than CPSK in a white-Gaussian-noise channel, there are formulas for $P$ and $P_e$ to be used in place of (79) and (81). There are many families of modulation schemes, channel conditions, error-correcting codes, and reception techniques that are of practical interest. This paper provides the tools for studying their effects on the performance of embedded and conventional DPCM. This is a subject worthy of further investigation.

### *7.3 Binary number representation*

Without forward-error correction, the noise due to transmission errors is dominated by the effects of single errors in the most significant part of the transmitted code word. With the natural-binary representation, an error in the most significant bit always causes a noise impulse of half the peak-to-peak range of the quantizer (56). With the sign-magnitude representation, an error in the sign bit inverts the polarity of the quantized signal, thereby producing a noise impulse of approximately twice the magnitude of the quantizer input

Table VII—Source and channel code formats, convolutional code properties

| | Format 1 | Format 2 | Format 3 | Format 4 |
|---|---|---|---|---|
| Source code | | | | |
| bits/sample | 4 | 3 | 3 | 2 |
| bits/second | 32K | 24K | 24K | 16K |
| bits/sample protected | 0 | 3 | 2 | 2 |
| Channel code rate | No code | 3/4 | 2/3 | 1/2 |
| Free distance, $d$ | | 4 | 5 | 7 |
| Weight $w_d$ | | 22 | 25 | 4 |
| $w_{d+1}$ | Error | 0 | 112 | 12 |
| $w_{d+2}$ | Properties | 1687 | 357 | 20 |
| $w_{d+3}$ | | 0 | 1858 | 72 |
| $w_{d+4}$ | | 66964 | 8406 | 225 |

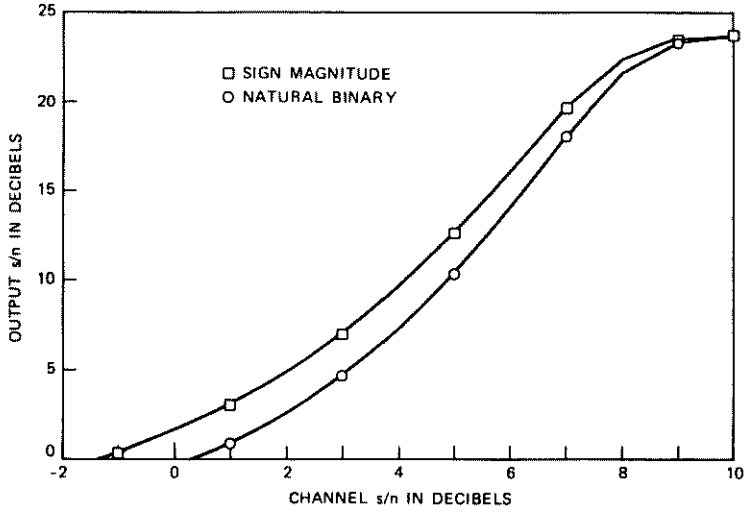



Fig. 8—Performance of embedded DPCM with sign-magnitude and natural-binary representations of quantizer outputs.

(57). Consequently, quantizers employing the sign-magnitude representation are somewhat less affected by transmission errors than quantizers with the natural-binary representation when the input probability distribution has its mode at zero. This is illustrated in Fig. 8, which pertains to uncoded 32 kb/s embedded DPCM transmission. When transmission errors are the dominant distortion, signals represented in the natural-binary format are about 2 dB noisier than signals represented by the sign-magnitude format.

With forward error correction, all error patterns are equally likely, and the two representations have essentially the same s/n.

## VIII. ACKNOWLEDGMENT

We thank Lawrence Wong for his helpful comments on this manuscript.

## REFERENCES

1. F. S. Boxall, "A Digital Carrier-Concentrator System with Elastic Traffic Capacity," IEEE Trans. Commun., *COM-22*, No. 10 (October 1974), pp. 1636–42.
2. J. A. Sciulli and S. J. Campanella, "A Speech Predictive Encoding Communication System for Multichannel Telephony," IEEE Trans. Commun., *COM-21* (July 1973), pp. 827–35.
3. T. Bially, B. Gold, and S. Seneff, "A Technique for Adaptive Voice Flow Control in Integrated Packet Networks," IEEE Trans. Commun., *COM-28*, No. 3 (March 1980), pp. 325–33.
4. D. J. Goodman and C.-E. Sundberg, "Combined Source and Channel Coding for Variable-Bit-Rate Speech Transmission," B.S.T.J., *62*, No. 7 (September 1983).
5. Proceedings of the 1982 IEEE International Conference on Acoustics Speech and

Signal Processing, Paris, May 1982. Session S8, pp. 954–83, contains four papers on adaptive DPCM at 32 kb/s.
6. J.-M. Raulin, et al., "A 60 Channel PCM-ADPCM Converter," IEEE Trans. Commun., *COM-30*, No. 4 (April 1982), pp. 567–73.
7. D. W. Petr, "32 Kbps ADPCM-DLQ Coding for Network Applications," Record of the IEEE Global Telecommunications Conference, Miami, Florida, November 1982, pp. 239–43.
8. D. J. Goodman, "Embedded DPCM for Variable Bit Rate Transmission," IEEE Trans. Commun., *COM-28*, No. 7 (July 1980), pp. 1040–6.
9. N. S. Jayant, "Variable Rate ADPCM Based on Explicit Noise Coding," B.S.T.J., *62*, No. 3 (March 1983), pp. 657–77.
10. N. Rydbeck and C.-E. Sundberg, "Analysis of Digital Errors in Non-Linear PCM Systems," IEEE Trans. Commun., *COM-24*, No. 1 (January 1976), pp. 59–65.
11. C.-E. Sundberg and N. Rydbeck, "Pulse Code Modulation with Error-Correcting Codes for TDMA Satellite Communication Systems," Ericsson Technics, *32*, No. 1 (1976), pp. 1–56.
12. N. Rydbeck and C.-E. Sundberg, "PCM/TDMA Satellite Communication Systems with Error-Correcting and Error-Detecting Codes," Ericsson Technics, *32*, No. 3 (1976), pp. 195–247.
13. D. J. Goodman, B. J. McDermott, and L. H. Nakatani, "Subjective Evaluation of PCM Coded Speech," B.S.T.J., *55*, No. 8 (October 1976), pp. 1087–109.
14. B. McDermott, C. Scagliola, and D. J. Goodman, "Perceptual and Objective Evaluation of Speech Processed by Adaptive Differential PCM," B.S.T.J., *57*, No. 5 (May–June 1978), pp. 1597–618.
15. L. J. Greenstein, "Slope Overload Noise in Linear Delta Modulators with Gaussian Inputs," B.S.T.J., *52*, No. 3 (March 1973), pp. 387–421.
16. G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*, New York: Plenum Press, 1981, pp. 237–8, and p. 403.

## AUTHORS

**David J. Goodman,** B.E.E., 1960, Rensselaer Polytechnic Institute; M.E.E., 1962, New York University; Ph.D. (Electrical Engineering), 1967, Imperial College, London; Bell Laboratories, 1967—. Mr. Goodman has studied various aspects of digital communications including analog-to-digital conversion, digital signal processing, subjective assessment of voiceband codecs, and spread spectrum modulation for mobile radio. He is Head, Communications Methods Research Department. In 1974 and 1975 he was a Senior Research Fellow, and in 1983 a Visiting Professor at Imperial College, London, England. Member, IEEE.

**Carl-Erik W. Sundberg,** M.S.E.E., 1966, and Dr. Techn., 1975, Lund Institute of Technology, University of Lund, Sweden; Bell Laboratories, 1981–1982. Mr. Sundberg is an Associate Professor in the Department of Telecommunication Theory, University of Lund, and a consultant in his field. He is Director of the consulting company SUNCOM, Lund. During 1976 he was with the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands, as an ESA Research Fellow. He has been a Consulting Scientist at LM Ericsson and SAAB-SCANIA, Sweden, and at Bell Laboratories. His research interests include source coding, channel coding (especially decoding techniques), digital modulation methods, fault-tolerant systems, digital mobile radio systems, spread spectrum systems, and digital satellite communication systems. He has published a large number of papers in these areas during the last few years. Senior Member, IEEE; member, SER, Sveriges Elektroingenjörers Riksförening.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

# CCITT Compatible Coding of Multilevel Pictures

By H. GHARAVI* and A. N. NETRAVALI*

(Manuscript received February 25, 1983)

The Comite Consultatif International Telegraphique et Telephonique (CCITT) has recently recommended a code for two-level (black and white) graphics transmission. A large number of pictures in graphics communication contain areas that cannot be represented adequately by only two shades of gray. We describe techniques by which a composite picture, containing an arbitrary mixture of two- and multilevel areas, can be coded by schemes that are compatible with the CCITT code. First, the composite picture is segmented automatically into two types of areas: one requiring only two levels (text, drawings, etc.) and the other requiring multilevels (for example, photos). A Differential Pulse Code Modulation (DPCM) scheme is then used to code the multilevel areas. Code assignment for the outputs of the DPCM quantizer are based on the local conditional statistics, and the bit stream is processed to change the statistics of the run lengths so that the CCITT run-length code becomes efficient. Results of computer simulations are presented in terms of quality of processed pictures and the required bit rate. Simulations show that our CCITT compatible scheme is as efficient as an incompatible but optimum DPCM coding scheme.

## I. INTRODUCTION

Simultaneous developments (algorithmic as well as systems) have taken place for many years in coding and transmission of two-level (black and white) document facsimile, and multilevel (many shades of gray) pictures.[1,2] The former type of pictures require very high spatial resolution to preserve the sharpness and have been coded by one-

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

dimensional run-length coding and two-dimensional edge difference coding [Comite Consultatif International Telegraphique et Telephonique (CCITT) one- and two-dimensional codes].[3] On the other hand, multilevel pictures contain gradual luminance transitions, and therefore require lower spatial resolution. They have been coded by Differential Pulse Code Modulation (DPCM) and transform methods. Most pictures used in business facsimile systems and audiographics conferencing contain a mixture of two-level and multilevel segments or subpictures. Coding such pictures using two-level techniques would not be adequate from the point of view of the picture quality, and using multilevel techniques would generate enormous data rates. Thus, it is of interest to devise schemes that automatically divide a picture into segments, each segment with a specified amplitude (gray shades) and spatial resolution and code each segment as best suited for it. Another practical requirement is that of compatibility. A coding scheme that handles a mixture of two-level and multilevel segments should be upwardly compatible with the CCITT standard schemes for two-level pictures. System cost will be reduced if the scheme for two-level multilevel pictures uses hardware blocks that are also used by the two-level picture coder. We present such a scheme below. Principal characteristics of our scheme are:

1. Compatibility with the CCITT schemes for two-level pictures
2. Automatic segmentation of pictures into two-level and multilevel segments
3. High coding efficiency by preprocessing the multilevel segments to fit the CCITT codes
4. Lower spatial resolution for gray-level segments (if desired)
5. Nonlossy (information preserving) coding of two-level segments, and lossy coding of multilevel segments.

## II. CODING ALGORITHM

The coding algorithm is explained in the following steps.

### *2.1 Segmentation*

The function of the segmentor is to classify each picture element as one of the three:

1. Black
2. White

$\Big\}$ → Two-Level

3. Gray → Multilevel

Figure 1 shows a neighborhood of the current picture element (pel) used for segmentation. We assume that each pel, obtained from the scanner, is specified by many shades of gray (e.g., 8 bits). The size of the neighborhood can be arbitrary. If it is too small, then many

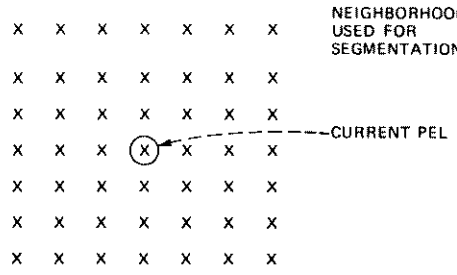



Fig. 1—Picture elements used for segmentation of the current pel. The size of the neighborhood is not necessarily 5 × 5 as suggested in the figure.

discontinuous segments of gray pixels will be generated. On the other hand, if the neighborhood is too large, then the ability to resolve small gray areas is lost. We slide this window of neighborhood over the pels along a scan line and classify each pel. We consider the boundary pels of the picture separately. Two thresholds, $t_1$ and $t_2$ ($t_1 < t_2$), are selected. It is hypothesized that most black pels will have intensity less than $t_1$, white pels will have intensity greater than $t_2$, and gray pel intensities may lie anywhere. Within the neighborhood let

$n_1$ = number of pels with intensity value $< t_1$

$n_2$ = number of pels with intensity value $> t_2$

$n_3$ = all the rest of the pels.

We define a state, $S$, consisting of three components: $S_1$, $S_2$, and $S_3$. A picture is segmented on the basis of the value of $S$. Let

$$S = S_1 + S_2 + S_3, \tag{1}$$

where

$S_1 = 1$, if $n_3 > n_1 + n_2$

$= 0$, otherwise

$S_2 = 1$, if previous pel is gray

$= 0$, otherwise

and

$S_3 = 1$, if $t_1 <$ intensity of present pel $< t_2$

$= 0$, otherwise.

The segmentation rule is then given by

$S \geq 2 \Rightarrow$ current pel gray

$S < 2$ and intensity of current pel $> T \Rightarrow$ current pel white

$S < 2$ and intensity of current pel $\leq T \Rightarrow$ current pel black.

$T$ is a threshold used to distinguish black elements from white ones, once they are known to be of the two-level type. If the range $(t_2 - t_1)$ is decreased by increasing $t_1$ and decreasing $t_2$, then more elements will be regarded as two-level and the quality of picture may suffer, but this will also decrease the bit rate.

We evaluated the performance of the segmentor, in particular its dependence on the block size and $(t_2 - t_1)$ by computer simulation. Since there are no standard mixtures of two-level and multilevel images, we created our own by taking a 512 × 512 gray-level image (shown in Fig. 2) and superimposing it on the CCITT documents four and five. Since this 512 × 512 original was scanned at low resolution (compared to 200 pels/inch used for CCITT documents), it contains significant sharp transitions that would not be present in a photograph scanned at 200 pels/inch. Also, because the original gray-level picture and the CCITT documents are rather "clean", segmentor works quite

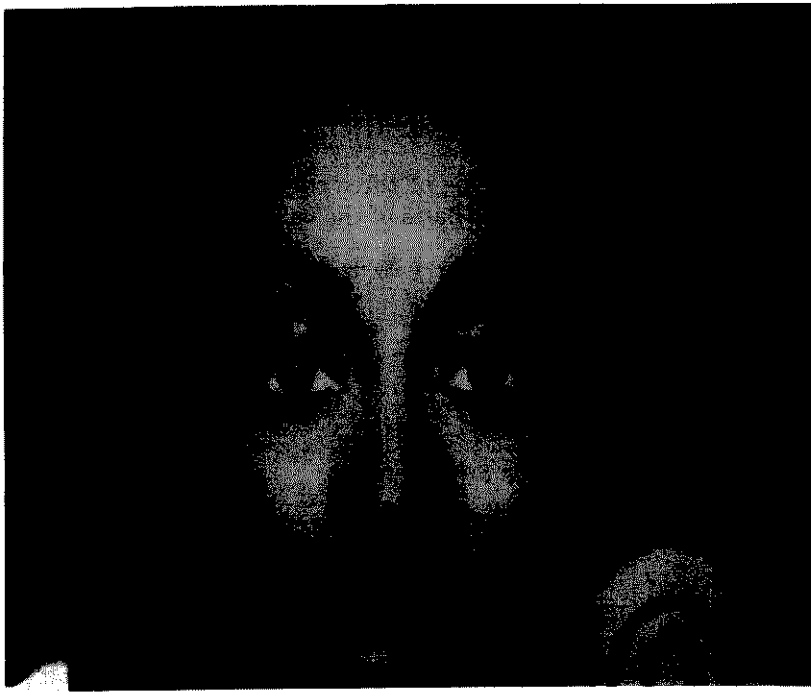


Fig. 2—A 512 × 512 multilevel (8 bits/pel) picture used for simulation.

well. However, this may not be a typical situation if a nonideal scanner was used. We, therefore, added random noise to the entire composite picture. This noise had a variance of 425 (on an 8-bit scale, 0-255). Table I shows the performance of the segmentor with respect to block size for a composite picture made from CCITT document 4. Here $t_1 = 28$, $t_2 = 195$. As we view such a segmented picture we realize that a $5 \times 5$ block may be too small. A $9 \times 9$ block appears quite adequate even when the added noise variance reaches 758. Higher block sizes result in a larger number of contiguous gray pels, thereby decreasing the number of segments. Figure 3 shows a segmented picture. Due to equipment limitations we show only a $512 \times 512$ section of the

Table I—Performance of the segmentor

| Block Size | No. of Gray Pels* | | No. of Segments | |
|---|---|---|---|---|
| | Without Noise | With Noise† | Without Noise | With Noise |
| $5 \times 5$ | 221,045 | 257,952 | 5394 | 27,910 |
| $9 \times 9$ | 222,818 | 226,691 | 3918 | 6662 |
| $15 \times 15$ | 224,595 | 224,598 | 3052 | 3062 |

* Total number of gray pels is $512 \times 512$.
† Variance of the noise = 758 (8-bit, 0-255 scale).

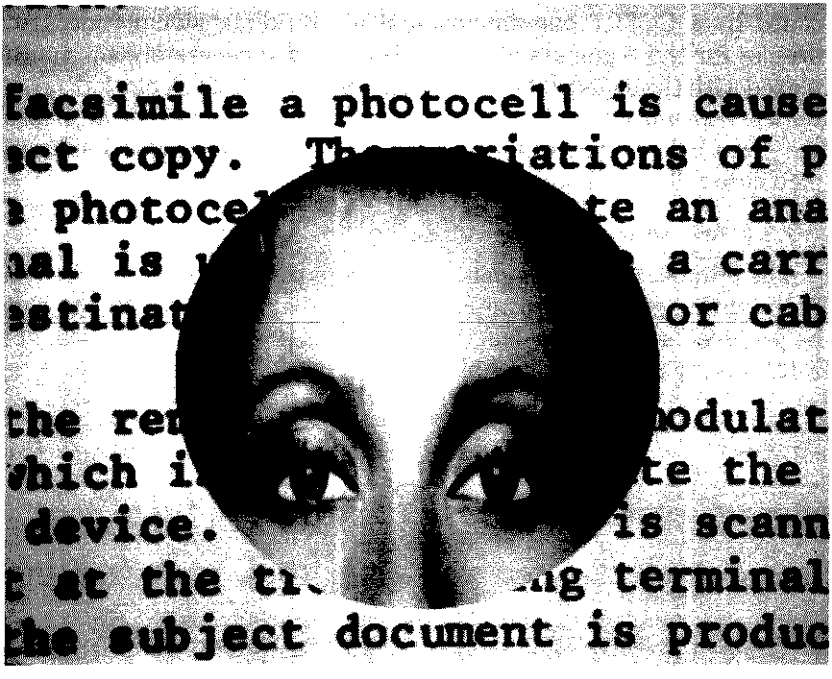



Fig. 3—A segmented picture. Pels classified black and white are reproduced with intensities 30 and 215, respectively. Gray-level pels are reproduced with 8-bit intensities.

composite picture. The segmentor has adequately separated the two-level areas from the multilevel areas.

### 2.2 Subsampling and interpolation

In most cases, areas of the picture that are segmented to be gray do not need as much spatial resolution as the two-level segments. If the two-level picture is at a very high spatial resolution (e.g., 200 pels/inch), then without any significant loss of quality, spatial resolution can be reduced in gray areas. Following is a scheme for subsampling and interpolation. A subsampling pattern is shown in Fig. 4. Interpolation is performed by averaging four surrounding pels, as in Fig. 4. Although we show only 2:1 subsampling, higher subsampling ratios may be used if the quality requirements are not very high. Also, two-dimensional subsampling may be performed, but this may increase the complexity.

### 2.3 Coding

After the pels (black, white, or gray) are classified and the gray areas to be transmitted are determined, a DPCM coder is used for gray areas. The resulting bit stream from the DPCM coder is preprocessed, multiplexed with bits from the two-level segments, and then coded by a CCITT one-dimensional or two-dimensional coder. Addresses for the segments of gray pels are coded separately and multiplexed with the coded data to transmit on the channel. A block diagram for the transmitter portion is shown in Fig. 5. Details of the algorithm are given below. Only a nonsubsampled case is illustrated; a subsampled case follows trivially.

#### 2.3.1 Grey segment coding

The purpose of gray segment coding is to convert an 8-bit/pel signal representing gray areas into a coded 3-bit/pel signal, which can then be preprocessed and run-length coded. This procedure reduces the bit rate for gray pels to about 2 bits/pel.

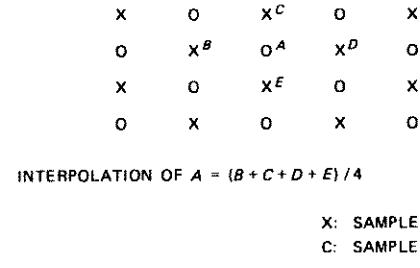



Fig. 4—Subsampling and interpolation pattern used in gray areas. Only one-dimensional subsampling is considered.

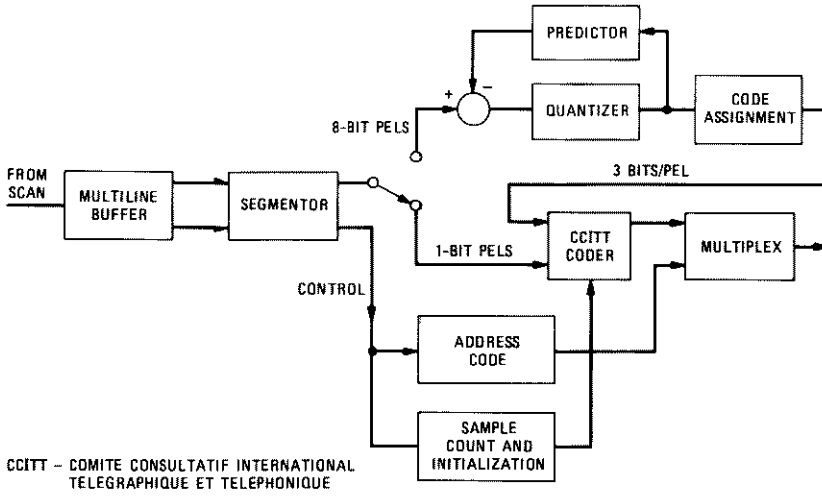



Fig. 5—Block diagram of the transmitter portion of the coder.

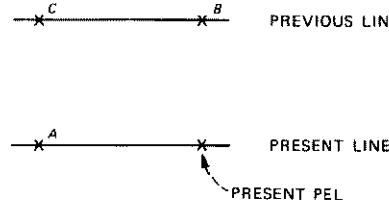



Fig. 6—Configuration of pels used for prediction. Only a nonsubsampled case is shown.

### *2.3.2 DPCM predictor*

On Fig. 6 we see that the present pel is predicted by

$$\hat{X} = \text{prediction of the present pel}$$
$$= 0.5A + 0.25(B + C).$$

It is assumed in this figure that all elements $A$, $B$, $C$ are gray elements. Appropriate modification is made if some of these are two-level elements.

### *2.3.3 DPCM quantizer*

The prediction error is quantized by a symmetric seven-level quantizer with the transfer characteristics given in Fig. 7. For most pictures with a resolution of 100 pels/inch, this appears adequate, although in some cases dynamic range may not be sufficient. Subjective studies are needed to optimize the characteristics for a given set of pictures.

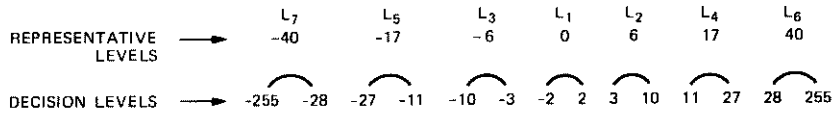



Fig. 7—Transfer characteristics of the quantizer.

Efficiency can be improved further by adapting the prediction and quantization.

### *2.3.4 Code assignment*

To reduce statistical redundancy and create a bit stream that can be coded compatibly with the CCITT code, seven levels of the quantizer output are mapped into a three-bit code. First, a table of 49 states is constructed by looking at the seven outcomes of the quantized prediction values for both elements $A$ and $B$ (in Fig. 6). Given a state, the code words for the present pel are arranged in order of conditional frequency of occurrence. Such statistics are precomputed for a set of pictures. The code word that is most frequent (for a given state) is given the code [000], the next highest is given code [001], etc. In addition, to decrease the probability of occurrence of isolated '1', if the last bit of the code word for $A$ is a '1', then the entire code word for the present pel is complemented (i.e., '0' → '1', and '1' → '0'). The table of 49 states and the corresponding code words are shown in Table II.

### *2.4 Preprocessing*

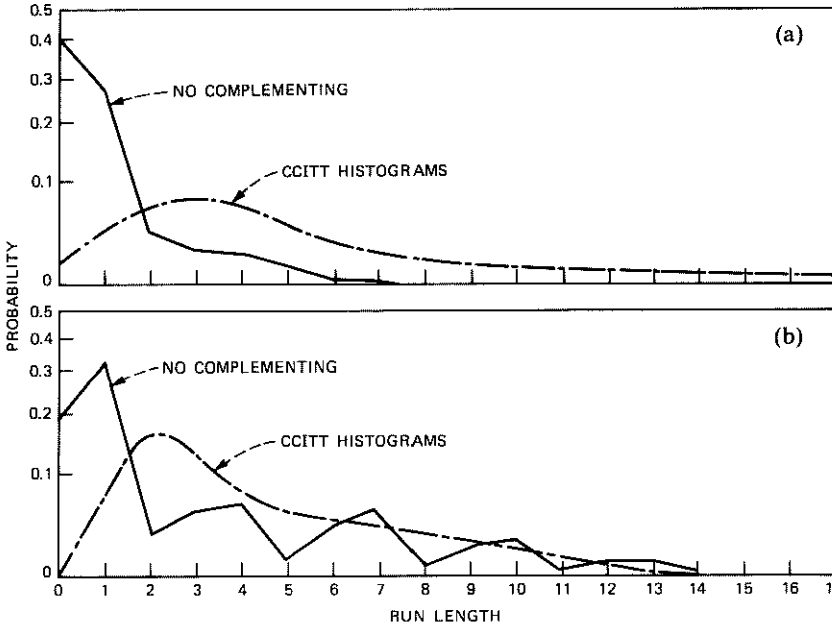
The code words for various states (e.g., runs) for the CCITT scheme are already defined based on the statistics. The statistics of the states for the gray-level segments are quite different. As an example, Fig. 8 shows histograms of the runs for black and white pels on which the one-dimensional CCITT code is based. The same figure also shows the histograms of the runs of the bits from gray-level picture (only 512 × 512) with the code assignment of the previous section but without any bit complementing. It is clear that the histograms are not similar in shape, and therefore using the CCITT code for runs of bits from gray segments would not be efficient. Since our experience shows that the two-dimensional CCITT code is not efficient for the gray segments, we give below a method of preprocessing that makes efficient use of the one-dimensional CCITT code. Let $n_c^b(i)$ and $n^b(i)$ be the histograms of the runs of black elements for the CCITT code and the gray-level segments, respectively. Also let $c(i)$ be the code assigned to the $i$th run by the CCITT coder. Let $j(i)$ be the sequence that is arranged in descending order of the histogram function $h_c^b(i)$, i.e.,

$$h_c^b(j(i)) \leq h_c^b[j(i-1)]\,.$$

Similarly, arrange the gray-level histogram $h^b(i)$ with the function $j^*(i)$. Then the code word for a length $j^*(i)$ of the gray segment is the same as $c(j(i))$. Thus, we arrange the two histograms in descending order and choose the code to be the same for entries of both the

Table II—Uncomplemented code words

| No. | Quantizer Level for $B$ | Quantizer Level for $A$ | Decreasing Frequency of Occurrence –> 000 | 001 | 011 | 111 | 110 | 100 | 010 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $L_1$ | $L_1$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ | $L_7$ |
| 2 | $L_1$ | $L_2$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ | $L_7$ |
| 3 | $L_1$ | $L_3$ | $L_1$ | $L_3$ | $L_2$ | $L_5$ | $L_4$ | $L_6$ | $L_7$ |
| 4 | $L_1$ | $L_4$ | $L_2$ | $L_1$ | $L_4$ | $L_3$ | $L_7$ | $L_5$ | $L_6$ |
| 5 | $L_1$ | $L_5$ | $L_3$ | $L_5$ | $L_1$ | $L_2$ | $L_6$ | $L_4$ | $L_7$ |
| 6 | $L_1$ | $L_6$ | $L_5$ | $L_6$ | $L_3$ | $L_2$ | $L_7$ | $L_1$ | $L_4$ |
| 7 | $L_1$ | $L_7$ | $L_7$ | $L_4$ | $L_2$ | $L_6$ | $L_5$ | $L_3$ | $L_1$ |
| 8 | $L_2$ | $L_1$ | $L_1$ | $L_3$ | $L_2$ | $L_4$ | $L_5$ | $L_7$ | $L_6$ |
| 9 | $L_2$ | $L_2$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_7$ | $L_6$ |
| 10 | $L_2$ | $L_3$ | $L_3$ | $L_1$ | $L_2$ | $L_5$ | $L_4$ | $L_6$ | $L_7$ |
| 11 | $L_2$ | $L_4$ | $L_2$ | $L_4$ | $L_1$ | $L_3$ | $L_7$ | $L_5$ | $L_6$ |
| 12 | $L_2$ | $L_5$ | $L_3$ | $L_5$ | $L_1$ | $L_2$ | $L_6$ | $L_4$ | $L_7$ |
| 13 | $L_2$ | $L_6$ | $L_6$ | $L_5$ | $L_7$ | $L_3$ | $L_4$ | $L_1$ | $L_2$ |
| 14 | $L_2$ | $L_7$ | $L_7$ | $L_5$ | $L_4$ | $L_6$ | $L_2$ | $L_3$ | $L_5$ |
| 15 | $L_3$ | $L_1$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ | $L_7$ |
| 16 | $L_3$ | $L_2$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_7$ | $L_6$ |
| 17 | $L_3$ | $L_3$ | $L_1$ | $L_3$ | $L_2$ | $L_5$ | $L_4$ | $L_6$ | $L_7$ |
| 18 | $L_3$ | $L_4$ | $L_2$ | $L_1$ | $L_4$ | $L_3$ | $L_7$ | $L_5$ | $L_6$ |
| 19 | $L_3$ | $L_5$ | $L_3$ | $L_1$ | $L_5$ | $L_2$ | $L_6$ | $L_4$ | $L_7$ |
| 20 | $L_3$ | $L_6$ | $L_6$ | $L_5$ | $L_3$ | $L_2$ | $L_1$ | $L_7$ | $L_4$ |
| 21 | $L_3$ | $L_7$ | $L_7$ | $L_6$ | $L_5$ | $L_3$ | $L_2$ | $L_4$ | $L_1$ |
| 22 | $L_4$ | $L_1$ | $L_3$ | $L_1$ | $L_2$ | $L_5$ | $L_4$ | $L_7$ | $L_6$ |
| 23 | $L_4$ | $L_2$ | $L_2$ | $L_1$ | $L_3$ | $L_4$ | $L_5$ | $L_7$ | $L_6$ |
| 24 | $L_4$ | $L_3$ | $L_3$ | $L_1$ | $L_5$ | $L_2$ | $L_4$ | $L_7$ | $L_6$ |
| 25 | $L_4$ | $L_4$ | $L_4$ | $L_2$ | $L_1$ | $L_7$ | $L_3$ | $L_5$ | $L_6$ |
| 26 | $L_4$ | $L_5$ | $L_5$ | $L_3$ | $L_1$ | $L_2$ | $L_6$ | $L_4$ | $L_7$ |
| 27 | $L_4$ | $L_6$ | $L_6$ | $L_1$ | $L_5$ | $L_4$ | $L_2$ | $L_3$ | $L_7$ |
| 28 | $L_4$ | $L_7$ | $L_7$ | $L_4$ | $L_2$ | $L_3$ | $L_6$ | $L_1$ | $L_5$ |
| 29 | $L_5$ | $L_1$ | $L_2$ | $L_1$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ | $L_7$ |
| 30 | $L_5$ | $L_2$ | $L_1$ | $L_2$ | $L_4$ | $L_3$ | $L_5$ | $L_6$ | $L_7$ |
| 31 | $L_5$ | $L_3$ | $L_3$ | $L_1$ | $L_2$ | $L_5$ | $L_4$ | $L_6$ | $L_7$ |
| 32 | $L_5$ | $L_4$ | $L_4$ | $L_2$ | $L_7$ | $L_1$ | $L_3$ | $L_5$ | $L_6$ |
| 33 | $L_5$ | $L_5$ | $L_5$ | $L_3$ | $L_6$ | $L_1$ | $L_2$ | $L_4$ | $L_7$ |
| 34 | $L_5$ | $L_6$ | $L_6$ | $L_5$ | $L_3$ | $L_1$ | $L_2$ | $L_7$ | $L_4$ |
| 35 | $L_5$ | $L_7$ | $L_6$ | $L_7$ | $L_4$ | $L_5$ | $L_2$ | $L_1$ | $L_3$ |
| 36 | $L_6$ | $L_1$ | $L_1$ | $L_6$ | $L_7$ | $L_3$ | $L_2$ | $L_5$ | $L_4$ |
| 37 | $L_6$ | $L_2$ | $L_6$ | $L_7$ | $L_2$ | $L_4$ | $L_5$ | $L_1$ | $L_3$ |
| 38 | $L_6$ | $L_3$ | $L_3$ | $L_6$ | $L_2$ | $L_4$ | $L_1$ | $L_7$ | $L_4$ |
| 39 | $L_6$ | $L_4$ | $L_7$ | $L_6$ | $L_4$ | $L_2$ | $L_1$ | $L_3$ | $L_5$ |
| 40 | $L_6$ | $L_5$ | $L_5$ | $L_6$ | $L_3$ | $L_1$ | $L_2$ | $L_4$ | $L_7$ |
| 41 | $L_6$ | $L_6$ | $L_6$ | $L_5$ | $L_3$ | $L_7$ | $L_1$ | $L_4$ | $L_2$ |
| 42 | $L_6$ | $L_7$ | $L_6$ | $L_7$ | $L_5$ | $L_4$ | $L_2$ | $L_3$ | $L_1$ |
| 43 | $L_7$ | $L_1$ | $L_7$ | $L_5$ | $L_3$ | $L_4$ | $L_6$ | $L_2$ | $L_1$ |
| 44 | $L_7$ | $L_2$ | $L_7$ | $L_2$ | $L_3$ | $L_4$ | $L_1$ | $L_5$ | $L_6$ |
| 45 | $L_7$ | $L_3$ | $L_7$ | $L_5$ | $L_6$ | $L_4$ | $L_3$ | $L_1$ | $L_2$ |
| 46 | $L_7$ | $L_4$ | $L_4$ | $L_7$ | $L_2$ | $L_3$ | $L_1$ | $L_6$ | $L_5$ |
| 47 | $L_7$ | $L_5$ | $L_7$ | $L_6$ | $L_5$ | $L_3$ | $L_4$ | $L_2$ | $L_1$ |
| 48 | $L_7$ | $L_6$ | $L_7$ | $L_6$ | $L_4$ | $L_5$ | $L_2$ | $L_3$ | $L_1$ |
| 49 | $L_7$ | $L_7$ | $L_7$ | $L_4$ | $L_1$ | $L_2$ | $L_3$ | $L_6$ | $L_5$ |



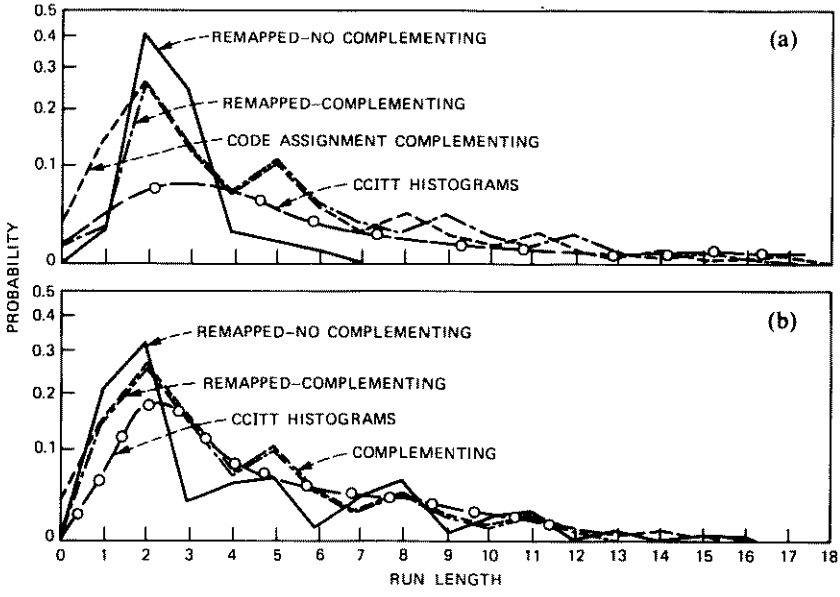

CCITT – COMITE CONSULTATIF INTERNATIONAL
TELEGRAPHIQUE ET TELEPHONIQUE

Fig. 8—Histograms of the (a) white and (b) black runs used in the 1D CCITT code and the processed gray-level pictures.

rearranged histograms. We found that in reality much of the gain in coding efficiency can be obtained by exchanging the code words for a few run lengths. This leads to simple preprocessing. Figure 9 shows the results of preprocessing on the histograms. It is clear that, if bit complementing is not used, after the preprocessing the code set is more attached to the histograms and therefore leads to more efficient code. However, if bit complementing is used, the histogram without any preprocessing is not too different from the CCITT histogram. Therefore, when bit complementing is used, the advantages of preprocessing are not large. Although this is the case for the picture we considered, more experiments are needed to evaluate statistics of typical pictures and usefulness of the preprocessing for such statistics.

### *2.5 Addressing*

To encode positional information of the boundaries of a segmented picture, each composite line is considered as a sequence of alternating black and white runs corresponding to the lengths of two-level and gray-level segments, respectively. This is then coded by the two-




CCITT – COMITE CONSULTATIF INTERNATIONAL TELEGRAPHIQUE ET TELEPHONIQUE

Fig. 9—Effect of preprocessing on the histograms of the (a) white and (b) black runs.

dimensional CCITT code and is transmitted at the beginning of each composite line.

### *2.6 Multiplexing and CCITT coding*

The bits resulting from the above procedure for gray areas are multiplexed pel by pel with those of the two-level pels. Our experiments show that while it is advantageous to encode two-level areas by two-dimensional code, most of the two-dimensional correlation in gray segments is removed by two-dimensional prediction and code assignment. Therefore, gray areas are coded by one-dimensional code. Since the number of gray-level pels may vary from line to line, in order to maintain proper registration of two-level pels (for two-dimensional coding), a sample count is maintained and is used to initialize the two-dimensional coder once it comes out of the gray segment within a line.

## III. SIMULATION RESULTS

Results of computer simulations are given in Tables III and IV. Table III shows results for 512 × 512 gray-level picture, and Table IV shows results for composite pictures with CCITT standard documents 4 and 5. It is clear from Table III that for the gray-level picture, without any preprocessing or bit complementing, coding efficiency is

Table III—Performance of coding algorithms for 512 × 512 gray picture

| No. | Coding Algorithms | Coded bits (bits/pel) |
|---|---|---|
| 1. | Entropy of the quantizer output (no subsampling) | 1.84 |
| 2. | Entropy of the quantizer output (2:1 subsampling) | 1.05 |
| 3. | Entropy with one-dimensional run-length coding (no complementing, no preprocessing, no subsampling) | 2.76 |
| 4. | Entropy with one-dimensional run-length coding (no complementing, no preprocessing, 2:1 subsampling) | 1.98 |
| 5. | One-dimensional run-length coding (no complementing, no preprocessing, CCITT code, no subsampling) | 3.68 |
| 6. | One-dimensional run-length coding (no complementing, no preprocessing, CCITT code, 2:1 subsampling | 2.86 |
| 7. | 5+ preprocessing | 3.14 |
| 8. | 6+ preprocessing | 2.24 |
| 9. | Entropy with one-dimensional run-length coding (complementing, no preprocessing) | 1.88 |
| 10. | Entropy with one-dimensional run-length coding (complementing, no preprocessing, 2:1 subsampling) | 1.17 |
| 11. | 9+ CCITT code | 2.05 |
| 12. | 10+ CCITT code | 1.31 |
| 13. | 11+ preprocessing | 1.98 |
| 14. | 12+ preprocessing | 1.19 |

Table IV—Performance of coding algorithms for composite pictures

| No. | Coding algorithms | Coded bits | |
|---|---|---|---|
| | | Document 4 | Document 5 |
| 1. | One-dimensional CCITT code on noncomposite document | 870803 | 547853 |
| 2. | Two-dimensional CCITT code on noncomposite document | 577527 | 286911 |
| 3. | Two-dimensional code for two-level, one-dimensional code for gray level (no complementing) | 1169270 | 893288 |
| 4. | Two-dimensional code for two-level, one-dimensional code for gray level (no complementing), 2:1 subsampling | 961210 | 686777 |
| 5. | 3+ preprocessing | 1086589 | 811897 |
| 6. | 4+ preprocessing | 879796 | 601189 |
| 7. | (3) + complementing | 909876 | 628626 |
| 8. | (4) + complementing | 755133 | 470539 |
| 9. | (5) + complementing | 894444 | 616956 |
| 10. | (6) + complementing | 739874 | 453229 |
| 11. | Two-dimensional code for entire document | 999125 | 714651 |
| 12. | Bits for addressing | 18147 | 18153 |

rather low. This is a result of the mismatch of the run-length statistics. Considerable improvement is obtained by preprocessing the run lengths before applying the CCITT coder. Even higher improvement is obtained by the bit-complementing technique. Much of the mis-

match between the statistics is removed by the complementing technique, and therefore additional improvement obtained by preprocessing the complemented output is marginal. The use of complementing makes it possible to achieve bit rates that are close to the entropy of the coded output. Coding of composite pictures shows similar results. Another interesting conclusion from Table IV is that the two-dimensional CCITT code is not very efficient for gray-level segments of the composite picture. This is a result of lack of line-to-line correlation among bits that are outputs of the quantizer. Much of the line-to-line correlation is already removed by the two-dimensional prediction and the bit assignment based on conditional statistics.

## IV. CONCLUSIONS

We have presented an algorithm that can automatically segment areas of a picture that require only two shades of gray from those that require many shades of gray. Gray areas are coded in a way that creates a bit stream that subsequently can be efficiently coded by a CCITT coder. We find that, for the gray areas, it is possible to achieve coding efficiencies close to the entropy of the DPCM quantizer output. Therefore, we conclude that it is possible to encode documents that contain an arbitrary mixture of two-level and multilevel areas using a CCITT coder that requires only a preprocessor at the transmitter and a postprocessor at the receiver.

## REFERENCES


1. A. N. Netravali, ed., special issue on Digital Encoding of Graphics, Proc. IEEE, *68*, No. 7 (July 1980).
2. A. N. Netravali and J. O. Limb, "Picture Coding—A Review," Proc. IEEE, *68*, No. 3 (March 1980), pp. 366–406.
3. R. Hunter and A. H. Robinson, "International Digital Facsimile Coding Standards," Proc. IEEE, *68*, No. 7 (July 1980), pp. 830–46.


## AUTHORS


**Arun N. Netravali,** B. Tech. (Honors), 1967, Indian Institute of Technology, Bombay, India; M.S., 1969, Ph.D. (Electrical Engineering), 1970, Rice University; Optimal Data Corporation, 1970–1972; Bell Laboratories, 1972—. Mr. Netravali has worked on problems related to filtering, guidance, and control for the space shuttle. At Bell Laboratories, he has worked on various aspects of digital processing and computing. He was a Visiting Professor in the Department of Electrical Engineering at Rutgers University. He is presently Director of the Computer Technology Research Laboratory. Mr. Netravali holds over 20 patents and has had more than 60 papers published. He was the recipient of the Donald Fink Prize Award for the best review paper published in the Proceedings of the IEEE. Editorial board, Proceedings of the IEEE; editor, Transactions on Communications; member, Tau Beta Pi, Sigma Xi; Senior Member, IEEE.

**Hamid Gharavi,** B.S.E.E., 1970, Tehran Polytechnic; M.S.C. (Digital Communication), Ph.D. (Electrical Engineering), 1979, University of Technology, Loughborough, England; Research fellow at University of Technology, Loughborough, England, 1979–1980; Lecturer at Auckland University, New Zealand, 1980–1981; Bell Laboratories, 1982—. Mr. Gharavi has worked on problems related to digital modulations for satellite communication. He also has worked on bandwidth compression of color television signals, source coding, graphics, and pattern recognition. Member, IEEE.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

# The Queueing Network Analyzer

By W. WHITT*

(Manuscript received March 11, 1983)

This paper describes the Queueing Network Analyzer (QNA), a software package developed at Bell Laboratories to calculate approximate congestion measures for a network of queues. The first version of QNA analyzes open networks of multiserver nodes with the first-come, first-served discipline and no capacity constraints. An important feature is that the external arrival processes need not be Poisson and the service-time distributions need not be exponential. Treating other kinds of variability is important. For example, with packet-switched communication networks we need to describe the congestion resulting from bursty traffic and the nearly constant service times of packets. The general approach in QNA is to approximately characterize the arrival processes by two or three parameters and then analyze the individual nodes separately. The first version of QNA uses two parameters to characterize the arrival processes and service times, one to describe the rate and the other to describe the variability. The nodes are then analyzed as standard GI/G/m queues partially characterized by the first two moments of the interarrival-time and service-time distributions. Congestion measures for the network as a whole are obtained by assuming as an approximation that the nodes are stochastically independent given the approximate flow parameters.

## I. INTRODUCTION AND SUMMARY

Networks of queues have proven to be useful models to analyze the performance of complex systems such as computers, switching machines, communications networks, and production job shops.[1-7] To facilitate the analysis of these models, several software packages have

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

been developed in recent years, e.g., BEST/1,[8] CADS,[9] PANACEA,[10–12] and one based on Heffes.[13] These software packages contain algorithms for Markov models that can be solved exactly. For some applications, the model assumptions are at least approximately satisfied, so that the analysis can be very helpful. For many other applications, however, the model assumptions are not even approximately satisfied, so that the analysis can be misleading.

A natural alternative to an exact analysis of an approximate model is an approximate analysis of a more exact model. This paper describes a software package called the Queueing Network Analyzer (QNA), which was recently developed at Bell Laboratories to calculate approximate congestion measures for networks of queues. QNA goes beyond existing exact methods by treating non-Markov networks: The arrival processes need not be Poisson and the service-time distributions need not be exponential. QNA treats other kinds of variability by approximately characterizing each arrival process and each service-time distribution with a variability parameter. It is also possible to analyze large networks quickly with QNA because the required calculations are minimal, the most complicated part being the solution of a system of linear equations. The current version of QNA is written in FORTRAN.

Here is a rough description of the model: There is a network of nodes and directed arcs. The nodes represent service facilities and the arcs represent flows of customers, jobs, or packets. There is also one external node, which is not a service facility, representing the outside world. Customers enter the network on directed arcs from the external node to the internal nodes, move from node to node along the internal directed arcs, and eventually leave the system on one of the directed arcs from an internal node to the external node. The flows of customers on the arcs are assumed to be random so that they can be represented as stochastic processes.

If all servers are busy at a node when a customer arrives, then the customer joins a queue and waits until a server is free. When there is a free server, that customer begins service, which is carried out without interruption. Successive service times at each node are assumed to be random variables, which may depend on the type of customer but which otherwise are independent of the history of the network and are mutually independent and identically distributed. After the customer completes service, he goes along some directed arc from that node to another node. The customer receives service in this way from several internal nodes and then eventually leaves the network. A picture of a typical network (without the external node) is given in Fig. 1.

An important feature of the model is that there may be flows from node j to node i, as well as flows from node i to node j. This is of

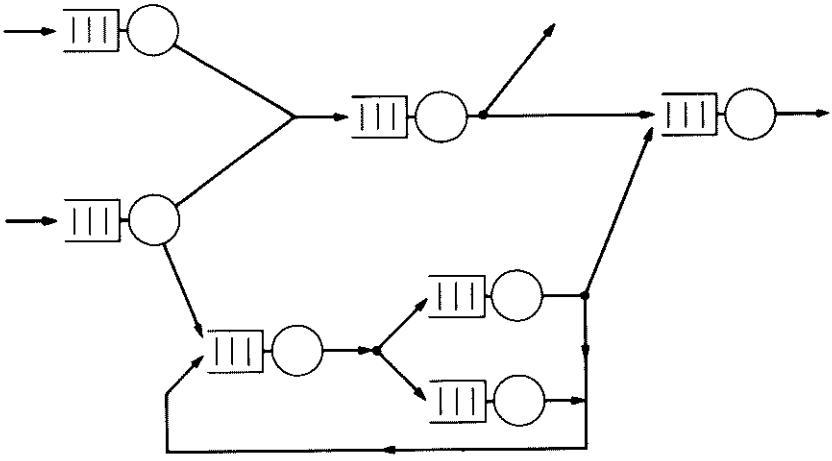


Fig. 1—An open network of queues.

course useful when customers can return to a node where they previously received service, but it is also useful when customers cannot return to a node where they previously received service. Then flows from node j to node i represent different customers than the customers that flow from node i to node j.

To be precise about the model, we give a list of basic assumptions. It is worth noting, however, that work is under way to extend QNA so that it can analyze systems in which each of the following assumptions is replaced by obvious alternatives. The general approximation technique is flexible, so that it is not difficult to modify and extend the algorithm.

*Assumption 1.* The network is *open* rather than closed. Customers come from outside, receive service at one or more nodes, and eventually leave the system.

*Assumption 2.* There are *no capacity constraints.* There is no limit on the number of customers that can be in the entire network and each service facility has unlimited waiting space.

*Assumption 3.* There can be *any number of servers* at each node. They are identical independent servers, each serving one customer at a time.

*Assumption 4.* Customers are selected for service at each facility according to the *first-come, first-served* discipline.

*Assumption 5.* There can be *any number of customer classes*, but customers cannot change classes. Moreover, much of the analysis in QNA is done for the aggregate or typical customer (see Sections 2.3 and VI).

*Assumption 6. Customers can be created* or combined at the nodes,

e.g., an arrival can cause more than one departure (see Section 2.2). (Think of messages.)

The general approach is to represent all the arrival processes and service-time distributions by a few parameters. The congestion at each facility is then described by approximate formulas that depend only on these parameters. The parameters for the internal flows are determined by applying an elementary calculus that transforms the parameters for each of the three basic network operations: superposition (merging), thinning (splitting), and flow through a queue (departure). These basic operations are depicted in Fig. 2. When the network is acyclic (e.g., queues in series), the basic transformations can be applied successively one at a time, but in general it is necessary to solve a system of equations or use an iterative method. To summarize, there are four key elements in this general approach:

1. *Parameters* characterizing the flows and nodes that will be readily available in applications and that have considerable descriptive power in approximations of the congestion at each node.
2. *Approximations for multiserver queues* based on the partial information provided by the parameters characterizing the arrival process and the service-time distribution at each node.
3. A *calculus for transforming the parameters* to represent the basic network operations: merging, splitting, and departure.
4. A *synthesis algorithm* to solve the system of equations resulting from the basic calculus applied to the network.

The current version of QNA uses two parameters to characterize the arrival processes and the service times, one to describe the rate and the other to describe the variability. (Three-parameter algorithms are being developed, however.) For the service times, the two param-

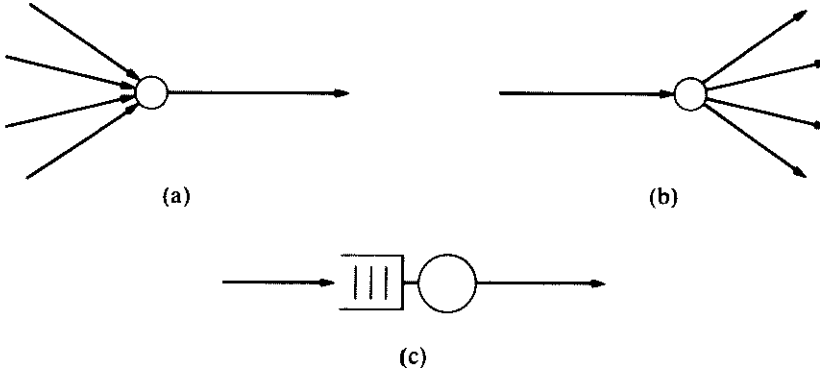



Fig. 2—Basic network operations: (a) Superposition or merging. (b) Decomposition or splitting. (c) Departure or flow through a queue.

eters are the first two moments. However, we actually work with the mean service time $\tau$ and the squared coefficient of variation $c_s^2$, which is the variance of the service time divided by the square of its mean. The user has the option of working with the service rate $\mu = \tau^{-1}$ instead of $\tau$. For the arrival processes, the parameters are associated with renewal-process approximations. The first two parameters are equivalent to the first two moments of the renewal interval (interval between successive points) in the approximating renewal process. The equivalent parameters we use are the arrival rate $\lambda$, which is the reciprocal of the renewal-interval mean, and the squared coefficient of variation $c_a^2$, which is the variance of the renewal interval divided by the square of its mean.

We obtain the approximation of the flows by applying the general framework and the basic procedures for approximating point processes in Whitt,[14] incorporating refinements such as the hybrid procedures developed for merging by Albin.[15,16] Of course, the general idea of simple two-parameter approximations for stochastic point processes goes back at least to the equivalent random method for approximating overflow streams (see Wilkinson,[17] Cooper,[18] and references there). Renewal-process approximations for such point processes were introduced by Kuczura[19] (also see Rath and Sheng[20]). Two-parameter approximations for networks of queues similar to QNA have also been developed by others, apparently first by Reiser and Kobayashi[21] (also see Kuehn,[22] Sevcik et al.,[23] Chandy and Sauer,[24] Chapter 4 of Gelenbe and Mitrani,[4] and Shanthikumar and Buzacott[6]). These two-parameter approximations for networks of queues are also similar in spirit to two-parameter approximations for networks of blocking systems with alternate routing (see Katz[25]).

Some authors have referred to these two-parameter heuristic approximations for networks of queues as diffusion approximations,[4,21] but diffusion processes are not actually used. Diffusion approximations and associated heavy-traffic limit theorems have motivated some of the heuristic approximations in the literature and in QNA, and they are closely related to the asymptotic method for approximating point processes,[14] but the heuristic approximations in QNA are not the same as the more complicated diffusion approximations for networks of queues in Iglehart and Whitt,[26] Harrison and Reiman,[27] and Reiman.[28,29]

The approximation method in QNA is perhaps best described as a parametric-decomposition method,[22] because the nodes are analyzed separately after the parameters for the internal flows are determined. Moreover, when the congestion measures are calculated for the network as a whole, the nodes are treated (approximately) as being stochastically independent. This independence can be interpreted as

a generalization of the product-form solution that is valid for Markovian networks, i.e., in the Markov models the components of the vector representing the equilibrium number of customers at each node are stochastically independent, so that the probability mass functions for the vector is the product of the probability mass functions for the components. While QNA can be thought of as a decomposition method or an extended-product-form solution, an effort is made to capture the dependence among the nodes. The idea is to represent this dependence approximately through the internal flow parameters.

To see the motivation for QNA, consider the elementary open network containing a single node with a single server, an infinite waiting room, and the first-come, first-served discipline. Suppose there is a single customer class with each customer being served only once before departing. The standard Markov model of this elementary network, which is embodied in BEST/1, CADS, and PANACEA, is the classical M/M/1 queue,[1,3,18] which has a Poisson arrival process and an exponential service-time distribution. For the M/M/1 model, the expected waiting time $EW$ (before the customer begins service) is

$$EW = \tau\rho/(1 - \rho), \tag{1}$$

where $\tau$ is the mean service time and $\rho$ is the traffic intensity, which is assumed to satisfy $0 \leqslant \rho < 1$.

On the other hand, QNA uses an approximation for the GI/G/1 model to represent this one-node network. The GI/G/1 model has a renewal arrival process and both the interarrival-time distribution and the service-time distribution are general. In QNA, the arrival process is represented by a renewal process partially characterized by two parameters: the arrival rate $\lambda$ and the variability parameter $c_a^2$. The service-time distribution is also partially characterized by two parameters: the mean service time $\tau$ and the variability parameter $c_s^2$. In contrast to (1), the formula for the expected waiting time in QNA is

$$EW = \tau\rho(c_a^2 + c_s^2)g/2(1 - \rho), \tag{2}$$

where $g \equiv g(\rho, c_a^2, c_s^2)$ is either one (when $c_a^2 \geqslant 1$) or less than one (when $c_a^2 < 1$); see (45). When $g(\rho, c_a^2, c_s^2) = 1$, (2) differs from (1) by the factor $(c_a^2 + c_s^2)/2$. When the arrival process is Poisson, $c_a^2 = 1$; when the service-time distribution is exponential, $c_s^2 = 1$. Hence, if the GI/G/1 model is actually an M/M/1 model, (2) reduces to (1). Of course, the user of QNA can set $c_a^2 = c_s^2 = 1$ and obtain (1). In fact, the values $c_a^2 = 1$ and $c_s^2 = 1$ are default values that the program uses if the user does not have variability parameters to provide. Each $c^2$ can assume any nonnegative value: $c^2 = 0$ for the degenerate deterministic distribution; $c^2 = k^{-1}$ for an erlang $E_k$, the sum of $k$ i.i.d. exponential random variables; and $c^2 > 1$ for mixtures of exponential

distributions. Obviously, the difference between (2) and (1) can be large, so that (2) often significantly reduces the error.

To obtain (2), we studied the GI/G/1 queue partially characterized by the moments of the interarrival-time and service-time distributions. Building on previous work by Holtzman,[30] Rolski,[31] and Eckberg,[32] we investigated the set of possible values of $EW$ given the partial information.[33-37] When $c_a^2 \geq 1$, formula (2) is always a possible value, i.e., there always is a GI/G/1 system with interarrival-time and service-time distributions having the specified moments in which (2) is correct. In general, (2) appears to be a reasonably typical value.

For the single-node example just considered, the arrival process was a renewal process. More generally, it is natural to think of all the non-Poisson arrival processes in the model as renewal processes, either because they are initially renewal processes or because the algorithm can be interpreted as approximating general arrival processes by renewal processes. Hence, with one customer class, it is natural to think of the model as a generalization of the open Jackson network M/M/m queues to an open Jackson network of GI/G/m queues. Each node is approximated by a GI/G/m queue having a renewal arrival process independent of service times that are independent and identically distributed with a general distribution. It is significant that QNA is consistent with the Jackson network theory: If there is a single class of customers, if all the arrival processes are Poisson, and if all the service-time distributions are exponential, then QNA is exact. However, for the general model few analytical results are available, so approximations are needed.

The software package QNA has a flexible input procedure: the model will accept more than one kind of input (see Section II). For the standard input, only limited information is required. Only two parameters are needed for each service-time distribution and each external arrival process. Also, a routing matrix is needed, which gives the proportion of those customers completing service at facility i that go next to facility j. (The algorithm is based on Markovian routing.) Hence, for $n$ nodes, the input consists of $n^2 + 4n$ numbers.

There is also an alternate input by classes and routes. In this scheme there are different classes of customers and each class enters the network at a fixed node and passes through a specified sequence of nodes. For each class, there are two parameters characterizing its external arrival process and two parameters characterizing the service-time distribution at each node on its route. With this input by routes, different classes can have different service-time distributions at a given node and the same class can have different service-time distributions during different visits to the same node. For a class with $n$ nodes on its route, the input consists of $3n + 2$ numbers. (This includes

the $n$ nodes on the route.) QNA analyzes this route input by aggregation: All the classes are aggregated by QNA to convert the route input into the standard input. Afterwards, the special parameters of each class are used to describe its sojourn times.

QNA also provides a fairly rich output. Several different congestion measures are calculated for each node: the traffic intensity (utilization), the expected number of busy servers (offered load), and the mean and variance of the equilibrium delay and number of customers present. In fact, for single-server nodes the delay distribution itself is described. Congestion measures for the entire network are also calculated, under the approximation assumption that the nodes are stochastically independent given the approximate flow parameters. Means and variances of total service times, total delays, and total sojourn times (response times) are given. When the input is by routes, these characteristics are given for each customer class. Otherwise, these characteristics are given for any route requested by the user.

A desirable feature of QNA is the structure of the calculus to transform the parameters to characterize the internal flows. The calculus is linear for each network operation, so that the parameters for the internal flows are determined simply by solving systems of linear equations. For the rates, the system of linear equations is just the familiar traffic rate equations occurring in the Jackson network of M/M/m queues. After having obtained the rates, we obtain the variability parameters of the internal flows (the squared coefficients of variations) by solving another system of linear equations. As a by-product, the existence of a unique nonnegative solution for the flow parameters is trivially guaranteed. There is no guarantee that an iterative scheme will converge, and if it does, there is typically no guarantee that a solution is unique. The linearity also guarantees that the computation required is not great. Since there is only one linear equation per node in the network, QNA can be used to analyze large networks repeatedly at minimal cost.

The linear calculus for transforming the variability parameters incorporates results of recent studies to improve the accuracy of the approximations. The general framework for approximating point processes in Whitt[14] is used. Significant improvement over previous approximation methods of this kind has been obtained by paying particular attention to the difficult superposition operation. For superposition, we use a modification of the hybrid procedure developed at Bell Laboratories by Albin.[15,16,38,39]

We emphasize that QNA is approximate. In applications it is important to validate the QNA output by comparing it with simulations and/or measurements. QNA is designed so that it is easy to incorporate improvements and it is easy to tune QNA for particular applications.

QNA also provides a useful framework for developing new approximation procedures. Moreover, it is easy to use QNA in conjunction with other special algorithms available to analyze the nodes or the flows.

The rest of this paper describes QNA in more detail. The paper is organized—just as the output is—according to the main steps in the analysis. The input is described in Section II. Section III describes the preliminary analysis to eliminate immediate feedback. The procedures to determine the internal flow parameters are contained in Section IV, and the procedures to calculate approximate congestion measures for the nodes are contained in Section V. Section VI contains the procedures to calculate approximate congestion measures for the network as a whole.

In a sequel in this issue of the *Journal*,[40] we describe the performance of QNA by comparing it with simulation and other approximations of several networks of queues. The sequel illustrates how to apply QNA and demonstrates the importance of the variability parameters.

## II. THE INPUT

In this section we describe the input options currently available for QNA. We anticipate more input options in the future. In Section 2.1 we describe the standard input, which is relatively compact. In Section 2.2 we describe a minor modification of the standard input, which allows for the creation or combination of customers at the nodes. For example, when a packet completes service at some node, it may cause several packets to be sent to other nodes. In Section 2.3 we describe an alternate input for different classes of customers having specified routes. We also describe the way QNA converts this input by classes and routes into the standard input of Section 2.1.

### *2.1 The standard input*

With the standard input, there is a single customer class and no creation of customers at the nodes. Any number of networks can be processed during a single run, so the user first specifies the number of networks. Then, for each network, the user specifies the number of nodes, and for each node the number of servers. For each node in the network, there are two parameters for the service-time distribution and two parameters for the external arrival process. Finally, there is a routing matrix, indicating the proportion of customers that go to node j from node i. Here is a list of the input data for each network with the notation we use:

$n$ = number of (internal) nodes in the network
$m_j$ = number of servers at node j
$\lambda_{0j}$ = external arrival rate to node j

$c_{0j}^2$ = variability parameter of the external arrival process to node j (squared coefficient of variation of the renewal interval in the approximating renewal process)

$\tau_j$ = mean service time at node j

$c_{sj}^2$ = squared coefficient of variation of the service-time distribution at node j

$q_{ij}$ = proportion of those customers completing service at node i that go next to node j.

In matrix notation, $Q \equiv (q_{ij})$ is an $n \times n$ matrix and $\Lambda_0 \equiv (\lambda_{0j})$ is an $1 \times n$ vector. The user has the option of inputting $\tau_j$ or its reciprocal $\mu_j$, the service rate at node j. (The same form must be used for all nodes.)

The user need not specify the variability parameters $c_{0j}^2$ and $c_{sj}^2$, in which case they are set equal to the default value one, corresponding to the M/M/1 model having a Poisson arrival proccess and an exponential service-time distribution. (Again, this option applies to all nodes.) Alternatively, the user can specify only the service-time variability parameters, $c_{sj}^2$, in which case either all the arrival-process variability parameters are automatically set equal to 1, yielding an M/G/1 approximation for each node, or the QNA algorithm is applied.

### *2.2 Creating and combining customers*

QNA has an option to allow creating or combining customers at the nodes following the completion of service. For example, a message processed at some node might cause messages to be sent to several other nodes. Alternatively, messages might be divided into packets after service at one node and then later recombined into messages after service at another node. In a job shop, the focus might shift back and forth between units and lots, e.g., at different nodes we might consider bottles, six-packs, cases, and even truckloads.

With this option, the user must specify the multiplicative factor $\gamma_j$ of customer creation or combination at node j for each $j$. There is customer creation (combination) at node j if $\gamma_j > 1$ ($\gamma_j < 1$). If customers are neither created nor combined, then $\gamma_j = 1$. If $\lambda_j$ is the overall arrival rate to node j, then the departure rate, after this modification, is $\lambda_j \gamma_j$ and the rate of departure from the network j is

$$\lambda_j \gamma_j \left(1 - \sum_{k=1}^{n} q_{jk}\right).$$

When artificial nodes are used, the creation or combination can also be placed before service.

To obtain our approximation formulas, we work with the following models of customer creation and combination. These models require integer values, but the approximation formulas and the QNA input do

not. For customer creation, we replace each departure from node j with a batch of size $\gamma_j$. For customer combination, we replace $\gamma_j^{-1}$ successive interdeparture intervals by a single one. From these models it is not difficult to calculate the impact of $\gamma_j$, e.g., the departure rate at node j is simply multiplied by $\gamma_j$.

### *2.3 Input by classes and routes*

QNA provides the option of defining different customer classes. Each class has its own route or itinerary that specifies the sequence of nodes visited. Thus, for each class the routing is deterministic. Each class has an external arrival process that goes to the first node on the route. As usual, the external arrival process is characterized by rate and variability parameters. Also, each class may have its own service-time distribution at each node on its route. The service-time distributions can be different, not only for different classes, but also for different visits to the same node by the same class. These service-time distributions are also characterized by rate and variability parameters. (Alternatively, the user can elect to input the service-time parameters for each node. Then all classes have the same service-time distribution at each visit to a particular node.)

As with the standard input, the user must specify the number of nodes and the number of servers at each node. Now we need the number of routes too. The required data are:

$n$ = number of nodes
$m_j$ = number of servers at node j
$r$ = number of routes.

Here is a list of the input data for the $k$th customer class of a network:

$n_k$ = number of nodes on route k
$\hat{\lambda}_k$ = external arrival rate of class k
$c_k^2$ = variability parameter of the external arrival process for class k
$n_{kj}$ = the $j$th node visited by customer class k
$\tau_{kj}$ = the mean service time of class k at the $j$th node of its route
$c_{skj}^2$ = the variability parameter of the service-time distribution of class k at the $j$th node of its route.

QNA converts this input by classes and routes into the standard input in Section 2.1. It then calculates the parameters of a typical or aggregate customer. Later, when computing sojourn times or response times of each customer class, QNA uses the service-time parameters of that customer class. The first version of QNA assumes as an approximation that each customer sees independent versions of the equilibrium distribution at each node. Hence, the waiting time before beginning service at each node is assumed to be the same for all classes and all visits.

We now indicate how QNA converts the input by classes and routes into the standard input of Section 2.1. For this purpose, let $1H$ be the indicator function of the set $H$, i.e., $1H(x) = 1$ if $x \in H$ and $1H(x) = 0$ otherwise.

First, we obtain the external arrival rates by

$$\lambda_{0j} = \sum_{k=1}^{r} \hat{\lambda}_k 1\{k{:}n_{k1} = j\}, \tag{3}$$

i.e., the external arrival rate at node j, $\lambda_{0j}$, is the sum of all route arrival rates $\hat{\lambda}_k$ for which the first node on the route is j. (Here the $1H$ notation is used for $H = \{k{:}n_{k1} = j\}$.) Similarly, the flow rate from i to j is

$$\lambda_{ij} = \sum_{k=1}^{r} \sum_{\ell=1}^{n_k-1} \hat{\lambda}_k 1\{(k, \ell){:}n_{k\ell} = i,\ n_{k,\ell+1} = j\} \tag{4}$$

and the flow from i out of the network is

$$\lambda_{i0} = \sum_{k=1}^{r} \hat{\lambda}_k 1\{k{:}n_{kn_k} = i\}. \tag{5}$$

From (4) and (5), we obtain the routing matrix Q. The proportion of customers that go to j from i is

$$q_{ij} = \lambda_{ij} \Big/ \left(\lambda_{i0} + \sum_{k=1}^{n} \lambda_{ik}\right). \tag{6}$$

If node i is an active part of the network, then the denominator will be strictly positive. Otherwise, QNA gives an error message.

Next, if the service-time parameters are given by routes, we obtain the service-time parameters for the nodes by averaging:

$$\tau_j = \frac{\sum_{k=1}^{r} \sum_{\ell=1}^{n_k} \hat{\lambda}_k \tau_{k\ell} 1\{(k, \ell){:}n_{k\ell} = j\}}{\sum_{k=1}^{r} \sum_{\ell=1}^{n_k} \hat{\lambda}_k 1\{(k, \ell){:}n_{k\ell} = j\}}. \tag{7}$$

The denominator in (7) will be strictly positive if node j is ever visited. Otherwise, as with (6), QNA supplies an error message.

We obtain the node variability parameters $c_{sj}^2$ using the property that the second moment of a mixture of distributions is the mixture of the second moments. Therefore, we have

$$\tau_j^2(c_{sj}^2 + 1) = \frac{\sum_{k=1}^{r} \sum_{\ell=1}^{n_k} \hat{\lambda}_k \tau_{k\ell}^2 (c_{sk\ell}^2 + 1) 1\{(k, \ell){:}n_{k\ell} = j\}}{\sum_{k=1}^{r} \sum_{\ell=1}^{n_k} \hat{\lambda}_k 1\{(k, \ell){:}n_{k\ell} = j\}}. \tag{8}$$

At this point, QNA has calculated enough information about the

standard input to compute the internal flow rates $\lambda_j$ and the traffic intensities $\rho_j$ as described in Section 4.1, i.e.,

$$\rho_j = \lambda_j \tau_j / m_j. \tag{9}$$

QNA uses this information to calculate the variability parameters $c_{0j}^2$ of the external arrival process. The hybrid approximation for superposition arrival processes in Section 4.2 is also used here because the external arrival process to node j is the superposition of the external arrival processes to node j from the different classes. If $\lambda_{0j} = 0$, then $c_{0j}^2$ does not matter and QNA sets $c_{0j}^2 = 1$. Otherwise,

$$c_{0j}^2 = (1 - \bar{w}_j) + \bar{w}_j \left[ \sum_{k=1}^{r} c_k^2 \left( \hat{\lambda}_k 1\{k{:}n_{k1} = j\} \Big/ \sum_{\ell=1}^{r} \hat{\lambda}_\ell 1\{k{:}n_{k1} = j\} \right) \right], \tag{10}$$

where

$$\bar{w}_j \equiv \bar{w}_j(\rho_j, \bar{\nu}_j) = [1 + 4(1 - \rho_j)^2(\bar{\nu}_j - 1)]^{-1}, \tag{11}$$

$\rho_j$ is the traffic intensity in (9), and

$$\bar{\nu}_j = \left[ \sum_{k=1}^{r} \left( \hat{\lambda}_k 1\{k{:}n_{k1} = j\} \Big/ \sum_{\ell=1}^{r} \hat{\lambda}_\ell 1\{\ell{:}n_{\ell 1} = j\} \right)^2 \right]^{-1}. \tag{12}$$

*Example 1:* To help fix the ideas, we consider an elementary example with $n = 2$ nodes and $r = 3$ routes. Let the number of servers at the nodes be $m_1 = 40$ and $m_2 = 10$. Let the route input be described by vectors

$$(n_k, \hat{\lambda}_k, c_k^2; n_{k1}, \tau_{k1}, c_{sk1}^2; \cdots; n_{kn_k}, \tau_{kn_k}, c_{skn_k}^2). \tag{13}$$

Here suppose that the $r$ vectors are:

$$\begin{aligned} &(2, 2, 1; 1, 1, 1; 1, 3, 3) \\ &(3, 3, 2; 1, 2, 0; 2, 1, 1; 1, 2, 1) \\ &(2, 2, 4; 2, 1, 1; 1, 2, 1). \end{aligned} \tag{14}$$

The first route corresponds to a Poisson arrival process at rate 2 to node 1, with all customers being fed back immediately for a second service before departing from the network. (Of course, the arrival process need not actually be Poisson; a Poisson process always has $c^2 = 1$ but other processes could have $c^2 = 1$ too.) The second class also enters at node 1, then goes to node 2 and back to node 1 before departing from the network, etc.

By (3), the external arrival rates are $\lambda_{01} = 5$ and $\lambda_{02} = 2$. By (4), the internal flow rates are $\lambda_{11} = 2$, $\lambda_{12} = 3$, $\lambda_{21} = 5$, and $\lambda_{22} = 0$. By (5), the flow rates out of the network are $\lambda_{10} = 7$ and $\lambda_{20} = 0$. By (6), the routing probabilities are: $q_{11} = 1/6$, $q_{12} = 1/4$, $q_{21} = 1$, and $q_{22} = 0$. By

(7), the mean service times are $\tau_1 = 2$ and $\tau_2 = 1$. By (8), $c_{s1}^2 = 1.67$ and $c_{s2}^2 = 1.00$. Note that both service times at node 2 in (14) have mean 1 and squared coefficient of variation 1, as with a common exponential distribution, so we should want $\tau_2 = c_{s2}^2 = 1$.

To obtain the internal arrival rates, we solve the traffic rate equations as in Section 4.1, i.e.,

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{n} \lambda_i q_{ij}, \tag{15}$$

to obtain $\lambda_1 = 12$, $\lambda_2 = 5$, $\rho_1 = 0.6$, and $\rho_2 = 0.5$.

Finally we obtain the variability parameters $c_{0j}^2$. First, from (12), $\bar{\nu}_1 = 25/13$ and $\bar{\nu}_2 = 1.0$. Then, from (11), $\bar{w}_1 = 0.629$ and $\bar{w}_2 = 1$, so that $c_{01}^2 = 1.38$ and $c_{02}^2 = 4$. Since there is only one external arrival process to node 2, we should have $\bar{\nu}_2 = \bar{w}_2 = 1$ and $c_{02}^2 = c_3^2 = 4$.

## III. ELIMINATING IMMEDIATE FEEDBACK

In this section we describe a function that QNA can perform before calculating the internal flow parameters and analyzing the congestion. The user can elect to reconfigure the network to eliminate immediate feedback. This procedure, which was originally suggested by Kuehn,[22] usually improves the quality of approximations. Hence, it is recommended and is performed in the standard version of QNA.

Immediate feedback occurs whenever $q_{ii} > 0$. Since QNA assumes Markovian routing, each customer completing service at node i is immediately fed back to node i to be served again with probability $q_{ii}$. Each time the customer goes to the end of the line. With the decomposition method, QNA assumes the customer finds the equilibrium number of customers at the node each time, with each visit being an independent experiment.

QNA eliminates immediate feedback by giving each customer, upon arrival from another node, his or her total service time before going to a different node. This is equivalent to putting a customer immediately fed back at the head of the line instead of at the end of the line. Transitions from node i back to node i are eliminated and the new probability of a transition to node j becomes the old conditional probability given that the customer departs from node i. In other words, each visit to node i from elsewhere plus all subsequent times immediately fed back are interpreted as a single visit. The service time is increased to compensate.

The motivation for this procedure is easy to explain. For a multiserver node with Bernoulli (Markovian) feedback and iid service times that are independent of a general arrival process (not necessarily Poisson or renewal), the distribution of the queue length process (but not the waiting times) is the same after this transformation. Hence,

we calculate the approximate values of the mean and variance of the equilibrium queue length for the transformed node without feedback and use them to derive approximate waiting time characteristics. By Little's formula,[41,42] the expected waiting time is also exact, i.e., the only error is in the approximation of the arrival process by a renewal process and the approximations for the characteristics of the GI/G/m queue; there is no additional error due to the immediate feedback.

The first step of the reconfiguring procedure is quite simple: the new service time is regarded as a geometric mixture of the $n$-fold convolution of the old service-time distribution. The parameters $\tau_i$, $c_{si}^2$, and $q_{ij}$ are changed to $\hat{\tau}_i$, $\hat{c}_{si}^2$, and $\hat{q}_{ij}$ when $q_{ii} > 0$:

$$\begin{aligned}
\hat{\tau}_i &= \tau_i/(1 - q_{ii}) \\
\hat{c}_{si}^2 &= q_{ii} + (1 - q_{ii})c_{si}^2 \\
\hat{q}_{ii} &= 0 \\
\hat{q}_{ij} &= q_{ij}/(1 - q_{ii}), \qquad j \neq i.
\end{aligned} \tag{16}$$

Afterwards, when calculating congestion measures for node i, QNA makes further adjustments. When we eliminate immediate feedback according to (16), we no longer count the times a customer is fed back immediately as separate visits. Hence, we need to adjust the congestion measures that are expressed per visit. For example, since the expected number of visits to node i per visit from outside is $(1 - q_{ii})^{-1}$, to obtain the expected waiting time per original visit to node i, we multiply the values of the expected waiting time $EW_i$ obtained from (16) by $(1 - q_{ii})$. Of course, the number of customers at each node is not affected by the feedback treatment.

Let $\bar{\lambda}_i$, $\bar{\tau}_i$, etc., represent the new adjusted values. In terms of the parameters $\lambda_i$, $\tau_i$, etc., obtained using (16), the new adjusted values are:

$$\begin{aligned}
\bar{\lambda}_i &= \lambda_i/(1 - q_{ii}) \\
\bar{\tau}_i &= (1 - q_{ii})\tau_i \\
c_{si}^2 &= (c_{si}^2 - q_{ii})/(1 - q_{ii}) \\
E\bar{W}_i &= (1 - q_{ii})EW_i \\
\mathrm{Var}(\bar{W}_i) &= (1 - q_{ii})\mathrm{Var}(T_i') - \hat{c}_{si}^2\bar{\tau}_i^2 \\
\mathrm{Var}(T_i') &= c^2(T_i')(EW_i + \tau_i)^2 \\
c^2(T_i') &= c^2(\bar{T}_i')(1 + q_{ii}) + q_{ii} \\
c^2(\bar{T}_i') &= (\mathrm{Var}\ \bar{W}_i' + \bar{c}_{si}^2\bar{\tau}_i^2)(E\bar{W}_i + \bar{\tau}_i)^{-2} \\
\mathrm{Var}(\bar{W}_i') &= EN_i\bar{c}_{si}^2\bar{\tau}_i^2 + c^2(N_i)(EN_i)^2\bar{\tau}_i^2,
\end{aligned} \tag{17}$$

where $N_i$ represents the equilibrium number of customers at node i and the $T_i$ variables represent the sojourn time per visit at node i.

We obtain the variables $\tilde{\tau}_i$ and $\tilde{c}_{si}^2$ in (17) by inverting the operation in (16), so we receive the original data again. The last five formulas in (17) involving the second-moment characteristics of $\bar{W}_i$ are based on the results of $N_i$ in the transformed system and heavy-traffic limit theorems for networks of queues by Reiman.[28,29] The main quantity desired is $\mathrm{Var}(\bar{W}_i)$; the variable $\bar{W}_i'$ is a preliminary approximation for $\bar{W}_i$.

In heavy traffic, the changes in the queue length at the nodes are negligible during a customer's sojourn in the network. Hence, if node i is visited $X_i$ times by some customer, then the total sojourn time at node i, say $T_i'$, is distributed approximately (in heavy traffic) as $X_i\bar{T}_i'$, where $X_i$ is independent of $\bar{T}_i'$ and $\bar{T}_i'$ is the sojourn time per individual visit in (17). (We use $T_i'$ and $\bar{T}_i'$ instead of $T_i$ and $\bar{T}_i$ because we do not use the description of $T_i$ obtained directly from (16) and $\bar{T}_i$ will differ from $\bar{T}_i'$.) By the independence, $ET_i'^2 = EX_i^2E\bar{T}_i'^2$. Since $X_i$ is geometrically distributed with mean $(1 - q_{ii})^{-1}$, $c^2(X_i) = q_{ii}$, and we obtain the seventh formula in (17).

The sixth and eighth formulas in (17) just express the formula for $c^2$ in terms of the mean and variance and the fact that the sojourn time is the sum of a waiting time and a service time. The final formula for $\mathrm{Var}(W_i')$ is obtained by approximating $W_i'$ by the sum of $N_i$ iid service times, using standard formulas for the variance of a random sum (e.g., compute $EW_i'^2$ by first conditioning on $N_i$). Finally, we obtain the fifth formula for $\mathrm{Var}(\bar{W}_i)$ by splitting the variance of $T_i'$ into waiting-time and service-time components and dividing by the expected number of visits to node i. As a consequence, $\mathrm{Var}(T_i')$ seems more reliable than $\mathrm{Var}(\bar{W}_i)$. This procedure makes $\mathrm{Var}(T_i)$, computed from $\mathrm{Var}(\bar{W}_i)$ by adding variance components as in Section VI, agree with the direct formula for $\mathrm{Var}(T_i')$ in (17).

The congestion measures based on (16) can be used to describe the total delays and total sojourn times of arbitrary customers in the network as in Section 6.2, but the congestion measures based on (17) are needed to describe the behavior of particular customers with specified routes as in Section 6.3. However, as stated above, $\mathrm{Var}(T_i')$ in (17) is an attractive alternative to $\mathrm{Var}(T_i)$ obtained via (16).

Experience indicates that eliminating immediate feedback often yields a better approximation (see Kuehn[22] and Sections V and VII of Whitt[40]). It is also often desirable to reconfigure the network to eliminate almost-immediate feedback, e.g., flows that return relatively quickly after passing through one or more other nodes (see Section V of Whitt[40]). Further study is needed to understand feedback phenomena and to develop improved approximations.

## IV. THE INTERNAL FLOW PARAMETERS

In this section we indicate how QNA calculates the internal flow parameters. In Section 4.1 we focus on the flow rates, which are obtained via the traffic rate equations, just as with the Markov models. In Section 4.2 we display the corresponding system of linear equations yielding the variability parameters. The remaining subsections explain how the variability parameter equations were obtained. The basic operations of superposition, splitting, and departure are discussed in Section 4.3, Section 4.4, and Section 4.5, and their synthesis in Section 4.6.

### *4.1 Traffic-rate equations*

In this step QNA calculates the total arrival rate to each node. Let $\lambda_j$ be the total arrival rate to node j, let $\gamma_j$ be the multiplicative factor of customer creation at node j as specified in Section 2.2, and let $\delta_j$ be the departure rate (to other nodes as well as out of the network) at node j. In general, $\delta_j = \lambda_j \gamma_j$. If there is no customer creation, then $\gamma_j = 1$ and the rate in equals the rate out.

The fundamental *traffic-rate equations* are just

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{n} \lambda_i \gamma_i q_{ij} \tag{18}$$

for $j = 1, 2, \cdots, n$, or in matrix notation

$$\Lambda = \Lambda_0 (I - \Gamma Q)^{-1}, \tag{19}$$

where $\Lambda_0 \equiv (\lambda_{0j})$ is the external arrival-rate vector, $Q \equiv (q_{ij})$ is the routing matrix, and $\Gamma = (\gamma_{ij})$ is the diagonal matrix with $\gamma_{ii} = \gamma_i$ and $\gamma_{ij} = 0$ for $i \neq j$. When there is no customer creation, $\gamma_i = 1$ and $\Gamma = I$. Of course, (18) is just a system of linear equations. To solve them is equivalent to inverting the matrix $(I - \Gamma Q)$ in (19). When customers can be created at the nodes as in Section 2.2, special care should be taken to be sure that (18) has a solution. We need to have $sp(\Gamma Q) < 1$ where $sp(\Gamma Q)$ is the spectral radius of $\Gamma Q$.

Given the arrival rates, it is possible to solve for the *traffic intensities* or *utilizations* at each node, defined by

$$\rho_i = \lambda_i \tau_i / m_i, \qquad 1 \leqslant i \leqslant n. \tag{20}$$

If $\rho_i \geqslant 1$, then the $i$th node is *unstable*. If any node is unstable, the algorithm gives an error message, prints out the traffic intensities, and stops. The associated *offered load* at node i, which coincides with the expected number of busy servers [see p. 400 of Heyman and Sobel[41] or (4.2.3) of Franken et al.[42]] is

$$\alpha_i = \lambda_i \tau_i, \qquad 1 \leqslant i \leqslant n. \tag{21}$$

The parameters $\alpha_i$ and $\rho_i$ coincide for a single server, with $\alpha_i$ tending to be more useful as the number of servers, $m_i$, increases (obviously when $m_i = \infty$).

After the arrival rates have been calculated for the nodes, QNA calculates related quantities for the arcs:

$$\begin{aligned} \lambda_{ij} &= \lambda_i \gamma_i q_{ij} && \text{—the arrival rate to node j from node i} \\ p_{ij} &= \lambda_{ij}/\lambda_j && \text{—the proportion of arrivals to j that came from } i,\ i \geqslant 0. \end{aligned} \tag{22}$$

Similarly, QNA calculates the following output rates:

$$\begin{aligned} d_i &= \lambda_i \gamma_i \left(1 - \sum_{j=1}^{n} q_{ij}\right) && \text{—the departure rate out of the network from node i} \\ d &= \sum_{i=1}^{n} d_i — && \text{the total departure rate out of the network.} \end{aligned} \tag{23}$$

### *4.2 Traffic variability equations*

The heart of the approximation is the system of equations yielding the variability parameters for the internal flows, i.e., the squared coefficients of variation for the arrival processes, $c_{aj}^2$. (These are derived in Sections 4.3 through 4.7.) The equations are linear, of the form

$$c_{aj}^2 = a_j + \sum_{i=1}^{n} c_{ai}^2 b_{ij}, \qquad 1 \leqslant j \leqslant n, \tag{24}$$

where $a_j$ and $b_{ij}$ are constants, depending on the input data:

$$a_j = 1 + w_j \left\{ (p_{0j} c_{0j}^2 - 1) + \sum_{i=1}^{n} p_{ij}[(1 - q_{ij}) + (1 - \nu_{ij})\gamma_i q_{ij} \rho_i^2 x_i] \right\} \tag{25}$$

and

$$b_{ij} = w_j p_{ij} q_{ij} \gamma_i [\nu_{ij} + (1 - \nu_{ij})(1 - \rho_i^2)], \tag{26}$$

where $x_i$, $\nu_{ij}$, and $w_j$ depend on the basic data determined previously, e.g., $\rho_i$, $m_i$ and $c_{si}^2$, but not on the variability parameters $c_{aj}^2$ being calculated. The parameter $\gamma_i$ is the multiplicative factor of customer creation or combination, introduced in Section 2.2. The variables $x_i$ and $\nu_{ij}$ are used to specify the departure operation; the variable $w_j$ is used to specify the superposition operation. The variables $\nu_{ij}$ and $w_j$ are weights or probabilities that are used in convex combinations

arising in hybrid approximations for departure and superpositions, respectively. The variables $x_j$, $\nu_{ij}$, and $w_j$ are included to make modification of the algorithm based on (24) easy. The specific values in this version of QNA are:

$$x_i = 1 + m_i^{-0.5}(\max\{c_{si}^2, 0.2\} - 1), \tag{27}$$

$$\nu_{ij} = 0, \tag{28}$$

and

$$w_j = [1 + 4(1 - \rho_j)^2(\nu_j - 1)]^{-1} \tag{29}$$

with

$$\nu_j = \left[\sum_{i=0}^{n} p_{ij}^2\right]^{-1} \tag{30}$$

and $p_{ij}$ in (22).

It is significant that it is easy to modify this system of equations. For example, other hybrid procedures for departures or superpositions can be introduced just by changing $\nu_{ij}$ and $w_j$, respectively. In this way, it is easy to calculate and compare the variability parameters for several different approximation procedures.

### *4.3 Superposition*

The purpose of the following sections is to explain the key approximation equations (24) through (30), which yield the variability parameters for the internal flows. The approximations are all based on the basic methods in Whitt:[14] the asymptotic method and the stationary-interval method. We consider the basic operations—superposition, splitting, and departure—in turn, and then their synthesis.

For superposition, the stationary-interval method is nonlinear so it presents difficulties.[14-16,22] Moreover, there appears to be no natural modification that makes it linear. On the other hand, the asymptotic method is linear. By the asymptotic method, the superposition squared coefficient of variation $c_A^2$ as a function of component squared coefficients of variation $c_i^2$ and the rates $\lambda_i$ is just the convex combination

$$c_A^2 = \sum_i \left(\lambda_i \Big/ \sum_k \lambda_k\right) c_i^2. \tag{31}$$

However, neither the asymptotic method nor the stationary-interval method alone works very well over a wide range of cases, e.g., see Section III of Whitt.[40] Albin[15,16] found that considerable improvement could be obtained by using a refined composite procedure, which is based on a convex combination of $c_A^2$ for the asymptotic method and $c_{SI}^2$ for the stationary-interval method. Her hybrid $c_H^2$ is of the form

$$c_H^2 = wc_A^2 + (1 - w)c_{SI}^2. \tag{32}$$

Unfortunately, since $c_{SI}^2$ is nonlinear, so is $c_H^2$. However, Albin found that a convex combination of $c_A^2$ and the exponential $c^2$ of 1 worked almost as well, having 4-percent average absolute error as opposed to 3 percent. Hence, we use such a hybrid procedure, namely,

$$\begin{aligned} c_H^2 &= wc_A^2 + (1 - w) \\ &= w \sum_i \left( \lambda_i \Big/ \sum_k \lambda_k \right) c_i^2 + 1 - w, \end{aligned} \tag{33}$$

where $w$ is a function of $\rho$ and the rates. Extensive simulation prompted Albin to suggest the weighting function

$$w = [1 + 2.1(1 - \rho)^{1.8}\nu]^{-1}, \tag{34}$$

where

$$\nu = \left[ \sum_i \left( \lambda_i \Big/ \sum_k \lambda_k \right)^2 \right]^{-1}. \tag{35}$$

Note that if there are $k$ component processes with equal rates then $\nu = k$. The parameter $\nu$ can be thought of as the number of component streams, with it being an equivalent number if the rates are unequal.

However, the weighting function (58) fails to satisfy an important consistency condition: We should have $w = 1$ when $\nu = 1$; if there is a single arrival process, the superposition operation should leave the $c^2$ parameter unchanged. Moreover, new theoretical results[39] indicate that the exponent of $(1 - \rho)$ in (34) should be 2. Hence, we use formula (33) based on the weight function $w$ in (29).

### *4.4 Splitting*

No approximation is needed for splitting because a renewal process that is split by independent probabilities (Markovian routing) is again a renewal process. However, approximation is of course indirectly associated with this step because the real process being split is typically not a renewal process and the splitting is often not according to Markovian routing.

Since a renewal process split according to Markovian routing is a renewal process, the asymptotic method and the stationary-interval method coincide. If a stream with a parameter $c^2$ is split into $k$ streams, with each being selected independently according to probabilities $p_i$, $i = 1, 2, \cdots, k$, then the $i$th process obtained from the splitting has squared coefficient of variation $c_i^2$ given by

$$c_i^2 = p_i c^2 + 1 - p_i, \tag{36}$$

which is clearly linear. Formula (36) is easy to obtain because the

renewal-interval distribution in the split stream is a geometrically distributed random sum of the original renewal intervals.

### *4.5 Departures*

For the stationary-interval method with single-server nodes, we apply Marshall's formula for the squared coefficient of variation of an interdeparture time, say $c_d^2$, in a GI/G/1 queue:[43,44]

$$c_d^2 = c_a^2 + 2\rho^2 c_s^2 - 2\rho(1 - \rho)\mu EW, \tag{37}$$

where $EW$ is the mean waiting time. Since $EW$ appears in (37), the congestion at the node affects the variability of the departure process. A stationary-interval method approximation for $c_d^2$ is obtained by inserting an approximation for $EW$ in a GI/G/1 queue. Our analysis[33–37] suggests that it suffices to use the linear approximation (2) with $g$ set equal to one. When this is combined with (37), we obtain the simple formula

$$c_d^2 = \rho^2 c_s^2 + (1 - \rho^2)c_a^2. \tag{38}$$

A simple extension of (38) for GI/G/m queues that is being used in the current version of QNA is

$$c_d^2 = 1 + (1 - \rho^2)(c_a^2 - 1) + \frac{\rho^2}{\sqrt{m}}(c_s^2 - 1). \tag{39}$$

Note that (39) agrees with (38) when $m = 1$ and (39) yields $c_d^2 = 1$ as it should for M/M/m and M/G/$\infty$ systems for which the stationary departure process is known to be Poisson. The third term in (39) approaches 0 as $m$ increases, reflecting the way multiple servers tend to act as a superposition operation. A basis for further refinements of (39) is the asymptotic analysis of departure processes in Whitt.[45] This asymptotic analysis shows that in some cases the variability of the departure process depends on the arrival and service processes in a more complicated way.

As with superposition, the asymptotic method yields a more elementary approximation than the stationary-interval method. In fact, the asymptotic-method approximation for the departure process is just the arrival process itself, i.e., the asymptotic-method approximation for $c_d^2$ is just $c_a^2$.[44] The number of departures in a long interval of time is just the number of arrivals minus the number in queue, and the number in queue fluctuates around its steady-state distribution, whereas the number of arrivals goes to infinity.

It remains to combine the basic methods to form a refined hybrid procedure. However, limited experience indicates that this refinement is not as critical as for superposition. The stationary-interval method

alone seems to perform better for departure processes than for superposition processes.[44]

The most appropriate view for the departure process—the stationary-interval method or the asymptotic method—depends on the traffic intensities at the next nodes where the departures are arrivals. As the traffic intensity of the next node increases, the asymptotic-method approximation for the departure process becomes more relevant. For example, consider the case of two queues in series with parameters $\lambda_{01}$, $c_{01}^2$, $\mu_1$, $c_{s1}^2$, $\mu_2$, and $c_{s2}^2$. If $\mu_2 \to \lambda$ while $\mu_1$ remains unchanged, then $\rho_2 \to 1$ and the second queue is in heavy traffic. Under such heavy traffic conditions, it has been shown[26] that the congestion measures at the second node are asymptotically the same as if the first facility were removed, i.e., as if the arrival process to the second node were just the arrival process to the first node. More generally, for any arrival process it has been proved that the asymptotic method is an asymptotically correct approximation for a queue in heavy traffic.[26]

Hence, it is natural to tune the departure approximation by using the traffic intensities in the following nodes. Since the departure process typically will be split and sent to different nodes with different traffic intensities, it is appropriate to do the tuning after splitting. Let $c_{di}^2$ be the departure $c^2$ at node i. Then

$$c_{ij}^2 = q_{ij}c_{di}^2 + 1 - q_{ij} \tag{40}$$

is the $c^2$ for the portion of the departures going to node j. We let $c_{ij}^2$ be a weighted combination of the approximations obtained by the asymptotic method and the stationary-interval method [using (39)]:

$$\begin{aligned} c_{ij}^2 = {} & \nu_{ij}(q_{ij}c_{ai}^2 + 1 - q_{ij}) \\ & + (1 - \nu_{ij})[q_{ij}[1 + (1 - \rho_i^2)(c_{ai}^2 - 1) \\ & + \rho_i^2 m_i^{-0.5}(c_{si}^2 - 1)] + 1 - q_{ij}], \end{aligned} \tag{41}$$

where $\nu_{ij}$ is chosen to satisfy $0 \leq \nu_{ij} \leq 1$ and be increasing in $\rho_j$ with $\nu_{ij} \to 1$ as $\rho_j \to 1$. However, we have not yet found that positive $\nu_{ij}$ helps,[44] so the current version of the QNA uses (28).

From (38) it is clear that the departure process variability, as depicted by QNA, is an appropriate weighted average of the arrival-process variability and the service-time variability. Hence, when the service time is deterministic, so that $c_{sj}^2 = 0$, the departure process is less variable than the arrival process. However, the actual reduction of variability in a network caused by deterministic service times often is not as great as predicted by (38) or (39). Hence, we have replaced (39) by

$$c_d^2 = 1 + (1 - \rho^2)(c_a^2 - 1) + \frac{\rho^2}{\sqrt{m}}(\max\{c_s^2, 0.2\} - 1). \quad (42)$$

After making this change, we get (25) through (28).

### 4.6 Customer creation or combination

We treat customer creation or combination as a modification of the departure process. When there is customer creation at node i, we replace each departure by a batch of size $\gamma_i$. When there is combination at node i, we replace each interdeparture interval by the sum of $\gamma_i^{-1}$ such intervals. These make more sense for integer values, but we do not require it. Hence, as described in Section 2.2, the departure rate from node i is $\gamma_i\lambda_i$ when the arrival rate is $\lambda_i$. We use the asymptotic method to obtain the variability parameter. Since the number of departures from node i in a large time interval is $\gamma_i$ times the number of arrivals, the asymptotic-method approximation of the variability parameter for customer creation or combination is just to multiply $c_{di}^2$ by $\gamma_i$. (By the asymptotic method, $c^2 = \lim_{t\to\infty} \text{Var } N(t)/EN(t)$; see Section 2 of Whitt.[14]) This is done before splitting.

### 4.7 Synthesis

We obtain the basic system of equations (24) through (30) by combining Sections 4.3 through 4.6 as follows:

$$\begin{aligned} c_{aj}^2 &= 1 - w_j + w_j \sum_{i=0}^{n} p_{ij} c_{ij}^2 \\ &= 1 - w_j + w_j \Bigg[ p_{0j} c_{0j}^2 + \sum_{i=1}^{n} p_{ij} (\nu_{ij}[q_{ij}\gamma_i c_{ai}^2 + (1 - q_{ij})] \\ &\quad + (1 - \nu_{ij})\{\gamma_i q_{ij}[1 + (1 - \rho_i^2)(c_{ai}^2 - 1) \\ &\quad + \rho_i^2 m_i^{-0.5}(\max\{c_{si}^2, 0.2\} - 1)] + 1 - q_{ij}\}) \Bigg]. \end{aligned} \quad (43)$$

The first line is based on superposition, Section 4.3, and the second line is based on departure, splitting and customer creation, Sections 4.4 through 4.6.

## V. CONGESTION AT THE NODES

Having calculated the rate and variability parameters associated with each internal arrival process, we are ready to calculate the approximate congestion measures for each node. At this point we have decomposed the network into separate service facilities that are analyzed in isolation. Each facility is a standard GI/G/m queue partially characterized by five parameters: the number of servers plus the first

two moments of the interarrival time and the first two moments of the service time. Instead of the moments we use the arrival rate $\lambda$, the mean service time $\tau$, and the squared coefficients of variation $c_a^2$ and $c_s^2$. Since we are focusing on a single node, we omit the subscript indexing the node throughout this section.

There are many procedures that could be applied at this point. We could fit complete distributions to the parameters,[14] and then apply any existing algorithm for solving a GI/G/m queue or a special case. Among the attractive options are procedures for analyzing the GI/G/1 queue,[46] the M/PH/m queue with phase-type service-time distributions,[47-52] the GI/$H_k$/m queue with hyperexponential service-time distributions[53,54] and the GI/$E_k$/m queue with Erlang service-time distributions.[55] Also available are approximations based on heavy-traffic and light-traffic limiting behavior.[56,57] The actual procedures used in this version of QNA, however, are quite elementary. Our study of the GI/G/1 queue[33-37] indicates that these elementary procedures are consistent with the limited information available. Since the arrival process is usually not a renewal process, and since only two moments are known for each distribution, there is little to be gained from more elaborate procedures. In fact, a user of QNA should be cautioned not to rely too heavily on detailed descriptions such as the tail of the waiting-time distribution. Such detailed descriptions may prove to be reasonably accurate, but they should certainly be checked by simulation.

We now describe the congestion measures provided by QNA. In Section 5.1 we treat the single-server node and in Section 5.2 we treat the multiserver node.

### *5.1 The GI/G/1 queue*

We begin with the steady-state waiting time (before beginning service), here denoted by $W$. The main congestion measure is the mean $EW$, but we also generate an entire probability distribution for $W$. First, the approximation formula for the mean is as in (2):

$$EW = \tau\rho(c_a^2 + c_s^2)g/2(1 - \rho), \tag{44}$$

where $g \equiv g(\rho, c_a^2, c_s^2)$ is defined as

$$g(\rho, c_a^2, c_s^2) = \begin{cases} \exp\left[-\dfrac{2(1-\rho)}{3\rho}\dfrac{(1-c_a^2)^2}{c_a^2 + c_s^2}\right], & c_a^2 < 1 \\ 1, & c_a^2 \geq 1. \end{cases} \tag{45}$$

When $c_a^2 < 1$, (44) is the Kraemer and Langenbach-Belz approximation,[58] which is known to perform well.[33-37,59] When $c_a^2 > 1$, the original Kraemer and Langenbach-Belz refinement does not seem to help, so

it is not used. Note that (44) is exact for the M/G/1 queue having $c_a^2 = 1$.

Let the number of customers in the facility, including the one in service, be denoted by $N$. The probability that the server is busy at an arbitrary time, $P(N > 0)$, and the mean $EN$ can be obtained from Little's formula (see Section 11.3 of Heyman and Sobel[41]):

$$P(N > 0) = \rho \tag{46}$$

and

$$EN = \rho + \lambda EW. \tag{47}$$

Formula (46) is exact even for stationary nonrenewal arrival processes and (47) is exact given $EW$.

For the probability of delay, $P(W > 0)$, denoted here by $\sigma$, we use the Kraemer and Langenbach-Belz approximation:[58]

$$\sigma \equiv P(W > 0) = \rho + (c_a^2 - 1)\rho(1 - \rho)h(\rho, c_a^2, c_s^2), \tag{48}$$

where

$$h(\rho, c_a^2, c_s^2) = \begin{cases} \dfrac{1 + c_a^2 + \rho c_s^2}{1 + \rho(c_s^2 - 1) + \rho^2(4c_a^2 + c_s^2)}, & c_a^2 \leqslant 1 \\ \dfrac{4\rho}{c_a^2 + \rho^2(4c_a^2 + c_s^2)}, & c_a^2 \geqslant 1. \end{cases} \tag{49}$$

Formula (48) also yields the correct value for M/G/1 systems, namely, $\rho$. Additional supporting evidence for (48) is contained in Whitt.[60]

We next focus on the conditional delay given that the server is busy, denoted by $D$. Obviously, $ED = EW/\sigma$. We first give an approximation formula for the squared coefficient of variation of $D$, $c_D^2$. This formula is the exact formula for the M/G/1 queue, with the service-time distribution being $H_2^b$ when $c_s^2 \geqslant 1$ and $E_k$ when $c_s^2 = k^{-1}$, where $H_2^b$ is the hyperexponential distribution with balanced means and $E_k$ is the Erlang distribution (see p. 256 of Cohen[61] and Section 3 of Whitt[14]). The idea underlying this approximation is that the conditional delay $D$ in a GI/G/1 queue (rather than the total delay $W$) depends more on the service-time distribution than on the interarrival-time distribution. Hence, the M/G/1 formula for $c_D^2$ is used as an approximation for all GI/G/1 systems. The M/G/1 formula for $c_D^2$ is:

$$c_D^2 = 2\rho - 1 + 4(1 - \rho)d_s^3/3(c_s^2 + 1)^2, \tag{50}$$

where $d_s^3 = E(\nu^3)/(E\nu)^3$ with $\nu$ being a service-time random variable. Even $E(\nu^3)$ is available, it can be used in (50), but since $E(\nu^3)$ is not available with two parameters, we use approximations for $d_s^3$. The approximations are based on the $H_2^b$ and $E^k$ distributions.

*Case 1*: When $c_s^2 \geqslant 1$,

$$d_s^3 = 3c_s^2(1 + c_s^2), \tag{51}$$

which comes from the $H_2^b$ formulas:

$$d_s^3 = \frac{3}{4}\left[\frac{1}{q^2} + \frac{1}{(1-q)^2}\right]$$

and

$$q = [1 + \sqrt{4(c_s^2 - 1)/(c_s^2 + 1)}]/2.$$

*Case 2*: When $c_s^2 < 1$,

$$d_s^3 = (2c_s^2 + 1)(c_s^2 + 1). \tag{52}$$

We obtain formula (52) by considering an Erlang $E_k$ variable, which can be represented as the sum of $k$ iid exponential random variables $X_i$ with mean $\tau/k$, where $\tau$ is the mean of the $E_k$ variable. In this case

$$\begin{aligned} E(X_1 + \cdots + X_k)^3 &= kE(X_1^3) + 3k(k-1)E(X_1^2)E(X_1) \\ &\quad + k(k-1)(k-2)(EX_1)^3 \\ &= \left(\frac{\tau}{k}\right)^3 [6k + 6k(k-1) + k(k-1)(k-2)] \end{aligned}$$

so that

$$d_s^3 = \frac{(k+2)(k+1)k}{k^3} = \left(1 + \frac{2}{k}\right)\left(1 + \frac{1}{k}\right),$$

which reduces to (52) because $c_s^2 = k^{-1}$ for an $E_k$ variable. Note that (51) and (52) agree at the boundary when $c_s^2 = 1$.

From (44), (48), and (50) through (52), we immediately obtain formulas for $\mathrm{Var}(D)$ and $ED^2$:

$$\begin{aligned} \mathrm{Var}(D) &= (ED)^2 c_D^2 = (EW)^2 c_D^2/\sigma^2 \\ E(D^2) &= \mathrm{Var}(D) + (ED)^2. \end{aligned} \tag{53}$$

From $D$ we then obtain second-moment characteristics for $W$:

$$c_W^2 = \frac{E(W^2)}{(EW)^2} - 1 = \frac{\sigma E(D^2)}{(\sigma ED)^2} - 1 = \frac{c_D^2 + 1 - \sigma}{\sigma},$$

$$\mathrm{Var}(W) = (EW)^2 c_W^2 \quad \text{and} \quad E(W^2) = \mathrm{Var}(W) + (EW)^2. \tag{54}$$

We now indicate how QNA calculates an approximate probability distribution for $W$. The distribution has an atom at zero as given in (48) and a density above zero. The density is chosen so that $W$ and $D$

have the first two moments already determined for them. (This is the general rule, but it is not quite followed in Cases 2 and 4 below.)

*Case 1*: $c_D^2 > 1.01$. Let $D$ have the $H_2^b$ density (hyperexponential with balanced means)

$$f_D(x) = p\gamma_1 e^{-\gamma_1 x} + (1-p)\gamma_2 e^{-\gamma_2 x}, \qquad x \geq 0, \tag{55}$$

where

$$p = [1 + \sqrt{(c_D^2 - 1)/(c_D^2 + 1)}]/2,$$

$$\gamma_1 = 2p/ED \quad \text{and} \quad \gamma_2 = 2(1-p)/ED. \tag{56}$$

*Case 2*: $0.99 \leq c_D^2 \leq 1.01$. Let $D$ have the exponential density with mean $ED$.

*Case 3*: $0.501 \leq c_D^2 < 0.99$. Let the distribution of $D$ be the convolution of two exponential distributions with parameters $\gamma_1$ and $\gamma_2$ ($\gamma_1 > \gamma_2$), i.e., let $D$ have density

$$f_D(x) = \left(\frac{\gamma_1\gamma_2}{\gamma_1 - \gamma_2}\right)(e^{-\gamma_2 x} - e^{-\gamma_1 x}), \qquad x \geq 0, \tag{57}$$

where

$$\gamma_2^{-1} = \frac{ED + \sqrt{2\ \mathrm{Var}(D) - (ED)^2}}{2}$$

and

$$\gamma_1^{-1} = ED - \gamma_2^{-1}. \tag{58}$$

The associated tail probabilities are

$$P(D > x) = (\gamma_1 e^{-\gamma_2 x} - \gamma_2 e^{-\gamma_1 x})/(\gamma_1 - \gamma_2). \tag{59}$$

*Case 4*: $c_D^2 < 0.501$. Let $D$ have an $E_2$ (Erlang) distribution with mean $ED$, which has $c^2 = 0.5$. Its density is

$$f_D(x) = \gamma^2 x e^{-\gamma x}, \qquad x \geq 0, \tag{60}$$

where $\gamma = 2/ED$. The associated tail probabilities are

$$P(D > x) = e^{-\gamma x}(1 + \gamma x), \qquad x \geq 0. \tag{61}$$

For deterministic service times, $d_s^3 = 1$, so that the smallest possible $c_D^2$ via (50) is $(1 + 2\rho)/3$. Hence, Case 4 above will not occur often.

Finally, we come to the second moment and variance of $N$, the number in system. For the M/G/1 queue, it is not difficult to compute $E(N^2)$. Since the steady-state number in the facility is equal to the number of arrivals during a customer's time in the facility, it is easy to compute the moments of $N$ from the moments of $W$; for example,

$$E(N^2) = \lambda(EW + E\nu) + \lambda^2[E(W^2) + 2EWE\nu + E(\nu^2)]$$
$$= \lambda EW + \rho + \lambda^2 E(W^2) + 2\lambda\rho EW + \rho^2(c_s^2 + 1), \quad (62)$$

and

$$\mathrm{Var}(N) = \lambda EW + \rho + \rho^2 c_s^2 + \lambda^2 \mathrm{Var}(W). \quad (63)$$

We now modify the M/G/1 formulas (62) and (63) for the GI/G/1 queue. Let $c_N^2$ be defined by

$$c_N^2 = Y_1 Y_2 / Y_3, \quad (64)$$

where $Y_1$ is the M/G/1 value of $\mathrm{Var}(N)$ in (63) using (44) for $EW$ and (54) and $\mathrm{Var}(W)$,

$$Y_2 = (1 - \rho + \sigma)/\max\{(1 - \sigma + \rho), 0.000001\}$$
$$Y_3 = \max\{(\rho + \lambda EW)^2, 0.000001\}, \quad (65)$$

and $\sigma$ is the probability of delay in (48). The maximum is used in (65) to avoid dividing by zero. For the M/G/1 queue, $Y_2 = 1$; for GI/M/1 queues, $Y_2$ in (65) provides just the right correction, so that (64) is exact given the true value of $\sigma$, $EW$ and $\mathrm{Var}(W)$. The correction $Y_2$ in (65) makes (64) too small for D/D/1 queues by a factor of $(1 + \rho)^{-1}$, but (64) is asymptotically correct in heavy traffic: $c_N^2 \to 1$ as $\rho \to 1$ if either $c_a^2 > 0$ or $c_s^2 > 0$.

From (47) and (64) we immediately obtain

$$\mathrm{Var}(N) = (EN)^2 c_N^2$$

and

$$E(N^2) = \mathrm{Var}(N) + (EN)^2. \quad (66)$$

When there is immediate feedback at the node and it is eliminated, adjustments are necessary in the formulas of this section, as indicated in Section III.

### *5.2 The GI/G/m queue*

The first congestion measures for multiserver nodes provided by QNA are exact. Even for nonrenewal stationary-arrival processes, the expected number of busy servers is just the offered load [see p. 400 of Heyman and Sobel[41] and (4.2.3) of Franken et al.[42]]:

$$E \min\{N, m\} = \alpha = \lambda\tau \quad (67)$$

and the traffic intensity or utilization is

$$\rho = \alpha/m. \quad (68)$$

By Little's formula, as in (47),

$$EN = \alpha + \lambda EW. \quad (69)$$

QNA currently provides only a few simple approximate congestion measures for multiserver nodes. These are obtained by modifying the exact formulas for the M/M/m model.[18] Let characteristics such as $EW(c_a^2, c_s^2, m)$ represent the characteristic as a function of the parameters $c_a^2$, $c_s^2$, and $m$, and let characteristics such as $EW(\mathrm{M/M/m})$ be the exact value for M/M/m system. A simple approximation for $EW$ based on heavy-traffic limit theorems[26,56,62,63] is:

$$EW(c_a^2, c_s^2, m) = \left(\frac{c_a^2 + c_s^2}{2}\right) EW(M/M/m). \tag{70}$$

Formula (70) has frequently been used for M/G/m queues and is known to perform quite well in that case.[64–67] By virtue of heavy-traffic limit theorems, we know that (70) is also asymptotically correct for GI/G/m systems as $\rho \to 1$ for fixed $m$. Limited additional study indicates that (70) is also reasonable for moderate values of $\rho$ when $c_a^2 \geq 0.9$ and $c_s^2 \geq 0.9$, or when $c_a^2 \leq 1.1$ and $c_s^2 \leq 1.1$. The actual value may be significantly smaller (larger) when $c_a^2 < 0.9$ and $c_s^2 > 1.1$ ($c_a^2 > 1.1$ and $c_s^2 < 0.9$).

The simple approximation (70) is also supplemented by simple approximations for the second moments of $W$ and $N$. They are obtained from:

$$c_W^2(c_a^2, c_s^2, m) = c_W^2(M/M/m)$$

and

$$c_N^2(c_a^2, c_s^2, m) = c_N^2(M/M/m). \tag{71}$$

Related second-moment characteristics are computed as in (54) and (66).

More detailed and sophisticated approximations for multiserver nodes are being studied. As we indicated before, a variety of methods and algorithms can be applied given the parameters of the arrival process.[47–57]

## VI. TOTAL NETWORK PERFORMANCE MEASURES

In this section we describe the approximate congestion measures calculated by QNA for the network as a whole. In Section 6.1 we discuss congestion measures representing the system view, e.g., throughput and number of customers in the network; in Sections 6.2 and 6.3 we discuss congestion measures representing the customer view, e.g., number of nodes visited and response times. In fact, there are actually two different customer views. In Section 6.2 we discuss the view of an arbitrary, typical, or aggregate customer; in Section 6.3 we discuss the view of a particular customer with a specified route through the network.

### 6.1 System congestion measures

A basic total network performance measure is the *throughput*, which we define as the *total external arrival rate* $\lambda_0$,

$$\lambda_0 = \lambda_{01} + \cdots + \lambda_{0n}. \tag{72}$$

When no customers are created at the nodes, the total external arrival rate equals the total departure rate from the network, so that there is little ambiguity about what we mean by throughput. However, when customers are created or combined at the nodes, as in Section 2.2, there is more than one possible interpretation. We might be interested in the rate at which arrivals are processed, i.e., (72). For example, the customers created at the nodes might be regarded only as extra work that must be done to serve the arrivals. On the other hand, we might be interested in the rate at which customers leave the network or in the rate of service completions. The departure *rate from the network* is

$$d = \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} \lambda_i \gamma_i \left(1 - \sum_{j=1}^{n} q_{ij}\right) \tag{73}$$

and the total *rate of service completions* is

$$s = \sum_{i=1}^{n} s_i = \sum_{i=1}^{n} \lambda_i \gamma_i. \tag{74}$$

A description of the overall congestion is provided by the mean and variance of the number $N$ of customers in the entire network. In general,

$$EN = EN_1 + \cdots + EN_n \tag{75}$$

and, as an approximation based on assuming that the nodes are independent, we have

$$\mathrm{Var}(N) = \mathrm{Var}(N_1) + \cdots + \mathrm{Var}(N_n). \tag{76}$$

Formula (76) is valid for the Markovian models as a consequence of the product-form solution, but is an approximation in general.

### 6.2 The experience of an aggregate customer

When we turn to the congestion experienced by individual customers, there are two very different approaches. The first approach keeps strict adherence to the model assumptions with the standard input in Section 2.1, and is based on interpreting the routing matrix as independent probabilities (Markovian routing). This means that each time any customer completes service at node i, that customer proceeds to node j with probability $q_{ij}$, independent of the current state

and history of the network. If the network is cyclic, this means that every customer has positive probability of visting some nodes more than once. This is the perspective of an aggregate customer. It might be that no individual customer actually ever visits the same node more than once.

If the aggregate view is desired, then the customer experience can be described by employing the basic theory of absorbing Markov chains as in Chapter III of Kemeny and Snell.[68] We can regard the external node as a single absorbing state to which all customers go when they leave the network or we can have more absorbing states, to distinguish between network departures from different nodes or different subsets of nodes. For this interpretation, the routing matrix $Q$ is the transient subchain associated with the absorbing Markov chain and the inverse $(I - Q)^{-1}$ in (19) with $\Gamma = I$ is the *fundamental matrix* of the absorbing chain (see p. 45 of Kemeny and Snell[68]). Solving the traffic-rate equations is tantamount to solving for this fundamental matrix.

From the fundamental matrix it is easy to calculate the moments of the number $n_{ij}$ of visits to any state j starting from any state i (on an external arrival process). For example, $En_{ij}$ is just the $(i, j)$-th entry of $(I - Q)^{-1}$. It is also easy to calculate the probability of absorption into each of the absorbing states starting from any initial distribution. These various congestion measures are easily obtained working with $n \times n$ matrices.[68]

Suppose that we focus on an arbitrary, typical, or aggregate customer arriving on an external arrival process. Then that customer enters node i with probability $\lambda_{0i}/\lambda_0$, where $\lambda_0$ is defined in (72) and the expected number of visits to node i for each customer is

$$EV_i = \lambda_i/\lambda_0. \tag{77}$$

(We have used the fundamental matrix to get $\lambda_i$.) The mean of the time, $T_i$, that an arbitrary customer spends in node i during his or her time in the network is thus

$$ET_i = (EV_i)(\tau_i + EW_i) \tag{78}$$

and the expected total sojourn time (time spent in the network from first arrival to final departure) for an arbitrary customer is thus

$$ET = \sum_{i=1}^{n} ET_i = \sum_{i=1}^{n} EV_i(\tau_i + EW_i). \tag{79}$$

The variance of $T_i$ is thus

$$\mathrm{Var}(T_i) = EV_i(\mathrm{Var}(W_i) + \tau_i^2 c_{si}^2) + \mathrm{Var}(V_i)(EW_i + \tau_i)^2. \tag{80}$$

The term $\mathrm{Var}(V_i)$ in (80) as well as $EV_i$ is easily obtained from the fundamental matrix. In particular, $\mathrm{Var}(V_i) = EV_i^2 - (EV_i)^2$ and

$$EV_i^2 = \sum_{j=1}^{n} (\lambda_{0j}/\lambda_0)[F(2F_{dg} - 1)]_{ji}, \tag{81}$$

where $F$ is the fundamental matrix $(I - Q)^{-1}$, $F_{dg}$ is the $n \times n$ matrix with all off-diagonal entries 0 and diagonal entries the same as $F$.

To obtain an approximation for the variance of the total sojourn time in the network, we assume that the sojourn times at the different nodes are conditionally independent, given any particular routing. (This is not valid even for all acyclic networks of M/M/1 nodes,[69] but is often approximately true.[7,70]) In particular, for a customer entering some specified node and making $V_j$ visits to node j, $1 \leq j \leq n$, before eventually leaving the network,

$$T = \left( \sum_{j=1}^{n} \sum_{k=1}^{V_j} T_{kj} \right), \tag{82}$$

where $T_{kj}$ is the sojourn time for the $k$th visit to node j. Our approximation assumption is that the variables $T_{kj}$ are mutually independent given the vector $(V_1, V_2, \cdots, V_n)$.

Hence,

$$\begin{aligned} E(T^2) &= \sum_{i=1}^{n} E\left( \sum_{k=1}^{V_i} T_{ki} \right)^2 \\ &\quad + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} E\left( \sum_{k=1}^{V_i} T_{ki} \sum_{\ell=1}^{V_j} T_{\ell j} \right) \\ &= \sum_{i=1}^{n} \{EV_i E(T_{1i}^2) + E[V_i(V_i - 1)]E(T_{1i})^2\} \\ &\quad + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} E(T_{1i})E(T_{1j})E(V_iV_j) \end{aligned} \tag{83}$$

so that

$$\mathrm{Var}(T) = \sum_{i=1}^{n} \mathrm{Var}(T_i) + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} E(T_{1i})E(T_{1j})\mathrm{Cov}(V_i, V_j). \tag{84}$$

However, the current version of QNA ignores the covariance terms in (84) in the calculation of $\mathrm{Var}(T)$.

### 6.3 *The experience of a particular customer*

Another approach is to decouple the macroscopic and microscopic interpretations. This view is common in statistical mechanics. The total network may exhibit statistical regularity not evidenced in any single particle (customer). In this view, we think of the total system evolving as if customers were routed according to independent probabilities, even though individual customers may have very different

routing probabilities, perhaps nonrandom routing or acyclic routing. For example, we may consider the cyclic network entirely appropriate for the macroscopic view even though no individual customer ever visits any node more than once. In order for this view to be realistic, each individual customer should have a relatively negligible effect on the total network.

The procedure here is to solve for the equilibrium or macroscopic behavior of the network first and then afterwards consider particular customers. The particular customers will have their own routes through the network and perhaps their own service times at the nodes along the way. There are two cases, depending on whether the input is by classes and routes, as in Section 2.3 or the standard input as in Section 2.1.

#### *6.3.1 Input by classes and routes*

First, suppose that we are using the input by classes and routes in Section 2.3. Then the particular customers correspond to the customer classes specified in the input. Hence, each customer has a deterministic route through the network and possibly special service times at the nodes on the route. In this case, as described in Section 2.3, QNA first converts the input by classes and routes into the standard input in Section 2.1. Then QNA solves for the equilibrium behavior. Finally, congestion measures are calculated for the different classes under the assumption that they follow their originally specified special routes and that upon arrival at the nodes on the route they see independent versions of the equilibrium state of the network. Hence, in the notation of Section 2.3, for a customer in class $k$, the expected total service time is

$$\sum_{j=1}^{n_k} \tau_{kj}, \tag{85}$$

the expected total waiting time is

$$\sum_{j=1}^{n_k} E(W_{n_{kj}}), \tag{86}$$

and the expected total sojourn time or response time is the sum of (85) and (86). Similarly, for a customer in class $k$ the variance of the total service time is

$$\sum_{j=1}^{n_k} \tau_{kj}^2 c_{skj}^2, \tag{87}$$

the variance of the total waiting time is

$$\sum_{j=1}^{n_k} \mathrm{Var}(W_{n_{kj}}), \tag{88}$$

and the variance of the total sojourn time is the sum of (87) and (88).

### *6.3.2 The standard input*

With the standard input in Section 2.1, the user must specify the particular customers to be analyzed. In this case, the user specifies classes with routes and possibly service times (rate and variability parameters), but these data are not used in calculating the equilibrium behavior. The decoupling principle is used with greater force here; there need not be any consistency between the microscopic and macroscopic views: This additional input does not affect the equilibrium behavior of the total network.

In the current version of QNA the individual customer routes are deterministic, so that the additional input required is just as in Section 2.3 and the congestion measures are just as in (85) through (88) in Section 6.3.1. However, it is possible to modify QNA to allow random routes. Then the additional input would be just as in Section 2.1; for each class it would consist of a routing matrix plus parameters for the arrivals process and service times.

## VII. ACKNOWLEDGMENTS

The software package QNA was written by Anne Seery. She used a subroutine for analyzing the M/M/m queue written by Shlomo Halfin. It has been a pleasure collaborating with Anne Seery on this venture. I also appreciate the help from many other colleagues and the continued support of my management: W. A. Cornell, C. S. Dawson, J. C. Lawson, C. J. McCallum, Jr., M. Segal, R. E. Thomas, and E. Wolman.

## REFERENCES

1. L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*, New York: John Wiley and Sons, 1976.
2. M. Schwartz, *Computer-Communications Network Design and Analysis*, Englewood Cliffs: Pentice-Hall, 1977.
3. F. P. Kelly, *Reversibility and Stochastic Networks*, New York: John Wiley, 1979.
4. E. Gelenbe and I. Mitrani, *Analysis and Synthesis of Computer Systems*, New York: Academic Press, 1980.
5. C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, Englewood Cliffs: Prentice-Hall, 1981.
6. J. G. Shanthikumar and J. A. Buzacott, "Open Queueing Network Models of Dynamic Job Shops," Int. J. Prod. Res., *19*, No. 3 (1981), pp. 255–66.
7. R. L. Disney, *Queueing Networks and Applications*, Baltimore: The John Hopkins University Lectures, 1982; to be published by the Johns Hopkins University Press.
8. "User's Guide for BEST/1," BGS Systems, Inc., Waltham, Massachusetts, 1980.
9. "User's Manual for CADS," Austin, TX: Information Research Associates, 1978.
10. J. McKenna, D. Mitra, and K. G. Ramakrishnan, "A Class of Closed Markovian Queueing Networks: Integral Representations, Asymptotic Expansions, and Generalizations," B.S.T.J., *60*, No. 5 (May–June 1981), pp. 599–641.

11. J. McKenna and D. Mitra, "Intergral Representations and Asymptotic Expansions for Closed Markovian Queueing Networks: Normal Usage," B.S.T.J., *61*, No. 5 (May–June 1982), pp. 661–83.
12. K. G. Ramakrishnan and D. Mitra, "An Overview of PANACEA, A Software Package for Analyzing Markovian Queueing Networks," B.S.T.J., *61*, No. 10, Part 1 (December 1982), pp. 2849–72.
13. H. Heffes, "Moment Formulae for a Class of Mixed Multi-Job-Type Queueing Networks," B.S.T.J., *61*, No. 5 (May–June 1982), pp. 709–45.
14. W. Whitt, "Approximating a Point Process by a Renewal Process, I: Two Basic Methods," Oper. Res., *30*, No. 1 (January–February 1982), pp. 125–47.
15. S. L. Albin, *Approximating Queues with Superposition Arrival Processes*, Ph.D dissertation, Department of Industrial Engineering and Operations Research, Columbia University, 1981.
16. S. L. Albin, "Approximating a Point Process by a Renewal Process, II: Superposition Arrival Processes to Queues," Department of Industrial Engineering, Rutgers University, 1982.
17. R. I. Wilkinson, "Theories for Toll Traffic Engineering in the U.S.A.," B.S.T.J., *35*, No. 2 (March 1956), pp. 421–514.
18. R. B. Cooper, *Introduction to Queueing Theory*, Second Edition, New York: North Holland, 1981.
19. A. Kuczura, "The Interrupted Poisson Process as an Overflow Process," B.S.T.J., *52*, No. 3 (March 1973), pp. 437–48.
20. J. H. Rath and D. D. Sheng, "Approximations for Overflows from Queues with a Finite Waiting Room," Oper. Res., *27*, No. 6 (November–December 1979), pp. 1208–16.
21. M. Reiser and H. Kobayashi, "Accuracy of the Diffusion Approximation for Some Queueing Systems," IBM J. Res. Dev., *18* (March 1974), pp. 110–24.
22. P. J. Kuehn, "Approximate Analysis of General Queueing Networks by Decomposition," IEEE Trans. Commun., *COM-27*, No. 1 (January 1979), pp. 113–26.
23. K. C. Sevcik, A. I. Levy, S. K. Tripathi, and J. L. Zahorjan, "Improving Approximations of Aggregated Queueing Network Subsystems," in *Computer Performance*, K. M. Chandy and M. Reiser (eds.), Amsterdam: North Holland, 1977, pp. 1–22.
24. K. M. Chandy and C. H. Sauer, "Approximate Methods for Analyzing Queueing Network Models of Computing Systems," ACM Computing Surveys, *10*, No. 3 (September 1978), pp. 281–317.
25. S. Katz, "Statistical Performance Analysis of a Switched Communications Network," Fifth Int. Teletraffic Cong., Rockefeller University, New York, 1967, pp. 566–75.
26. D. L. Iglehart and W. Whitt, "Multiple Channel Queues in Heavy Traffic, II: Sequences, Networks and Batches," Adv. Appl. Prob., *2*, No. 2 (Autumn 1970), pp. 355–69.
27. J. M. Harrison and M. I. Reiman, "On the Distribution of Multidimensional Reflected Brownian Motion," SIAM J. Appl. Math., *41*, No. 2 (October 1981), pp. 345–61.
28. M. I. Reiman, "Open Queueing Networks in Heavy Traffic," unpublished work, 1981.
29. M. I. Reiman, "The Heavy Traffic Diffusion Approximation for Sojourn Times in Jackson Networks," *Applied Probability and Computer Science—The Interface*, Volume 2, R. L. Disney and T. J. Ott (eds.), Boston: Birkhauser, 1982, pp. 409–21.
30. J. M. Holtzman, "The Accuracy of the Equivalent Random Method with Renewal Inputs," B.S.T.J., *52*, No. 9 (November 1973), pp. 1673–9.
31. T. Rolski, "Some Inequalities for GI/M/n Queues," Zast. Mat., *13*, No. 1 (1972), pp. 43–7.
32. A. E. Eckberg, Jr., "Sharp Bounds on Laplace-Stieltjes Transforms, with Applications to Various Queueing Problems," Math. Oper. Res., 2, No. 2 (May 1977), pp. 135–42.
33. W. Whitt, "On Approximations for Queues, I: Extremal Distributions," B.S.T.J., *63*, No. 1, Part 1 (January 1984), to be published.
34. J. G. Klincewicz and W. Whitt, "On Approximations for Queues, II: Shape Constraints," B.S.T.J., *63*, No. 1, Part 1 (January 1984).
35. W. Whitt, "On Approximations for Queues, III: Mixtures of Exponential Distributions," B.S.T.J., *63*, No. 1, Part 1 (January 1984).
36. W. Whitt, "The Marshall and Stoyan Bounds for IMRL/G/1 Queues are Tight," Oper. Res. Letters, *1*, No. 6 (December 1982), pp. 209–13.

37. W. Whitt, "Refining Diffusion Approximations for Queues," Oper. Res. Letters, *1*, No. 5 (November 1982), pp. 165–9.
38. S. L. Albin, "On Poisson Approximations for Superposition Arrival Processes in Queues," Management Sci., *28*, No. 2 (February 1982), 126–37.
39. W. Whitt, "Queues with Superposition Arrival Processes in Heavy Traffic," unpublished work, 1982.
40. W. Whitt, "Performance of the Queueing Network Analyzer," B.S.T.J., this issue.
41. D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research*, Vol. I, New York: McGraw-Hill, 1982.
42. P. Franken, D. König, U. Arndt, and V. Schmidt, *Queues and Point Processes*, Berlin: Akademie-Verlag, 1981.
43. K. T. Marshall, "Some Inequalities in Queueing," Oper. Res., *16*, No. 3 (May–June 1968), pp. 651–65.
44. W. Whitt, "Approximations for Departure Processes and Queues in Series," Nav. Res. Log Qtr., to be published.
45. W. Whitt, "Departures from a Queue with Many Busy Servers," Math. Oper. Res., *9* (1984).
46. A. A. Fredericks, "A Class of Approximations for the Waiting Time Distribution in a GI/G/1 Queueing System," B.S.T.J., *61*, No. 3 (March 1982), pp. 295–325.
47. Y. Takahashi and Y. Takami, "A Numerical Method for the Steady-State Probabilities of a GI/G/c Queueing System in a General Class," J. Oper. Res. Soc. Japan, *19* (1976), pp. 147–57.
48. H. C. Tijms, M. H. van Hoorn, and A. Federgruen, "Approximations for the Steady-State Probabilities in the M/G/c Queue," Adv. Appl. Prob., *13*, No. 1 (March 1981), pp. 186–206.
49. H. Groenevelt, M. H. van Hoorn, and H. C. Tijms, "Tables for M/G/c Queueing Systems with Phase-Type Service," Report No. 85, Department of Actuarial Sciences and Econometrics, The Free University, Amsterdam, The Netherlands, 1982.
50. M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models—An Algorithmic Approach*, Baltimore: The Johns Hopkins University Press, 1981.
51. M. F. Neuts, "A Program for Analyzing the M/PH/c Queue," Department of Mathematics, University of Delaware, 1981.
52. P. Hokstad, "Some Numerical Results and Approximations for the Many Server Queue with Nonexponential Service Time," Department of Mathematics, University of Trondheim, Norway, 1981.
53. J. H. A. de Smit, "The Queue GI/M/s with Customers of Different Types or the Queues $GI/H_m/s$," Adv. Appl. Prob., *15*, No. 2 (June 1983), pp. 392–419.
54. J. H. A. de Smit, "A Program for Analyzing the $GI/H_k/s$ Queue," Department of Applied Mathematics, Twente University of Technology, Enschede, The Netherlands, 1982.
55. A. Ishikawa, "On the Equilibrium Solution for the Queueing System: $GI/E_k/m$," T.R.U. Mathematics, *15*, No. 1 (1979), pp. 47–66.
56. S. Halfin and W. Whitt, "Heavy-Traffic Limits for Queues with Many Exponential Servers," Oper. Res., *29*, No. 3 (May–June 1981), pp. 567–88.
57. D. Y. Burman and D. R. Smith," A Light-Traffic Theorem for Multiserver Queues," Math. Oper. Res., *8*, No. 1 (February 1983), pp. 15–25.
58. W. Kraemer and M. Langenbach-Belz, "Approximate Formulae for the Delay in the Queueing System GI/G/1," *Congressbook*, Eighth Int. Teletraffic Cong., Melbourne, 1976, pp. 235–1/8.
59. J. G. Shanthikumar and J. A. Buzacott, "On the Approximations to the Single Server Queue," Int. J. Prod. Res. *18*, No. 6 (1980), pp. 761–73.
60. W. Whitt, "Minimizing Delays in the GI/G/1 Queue," Oper. Res., *32* (1984), to be published.
61. J. W. Cohen, *The Single Server Queue*, Amsterdam: North-Holland, 1969.
62. J. Köllerström, "Heavy Traffic Theory for Queues with Several Servers. I," J. Appl. Prob., *11*, No. 3 (September 1974), pp. 544–52.
63. J. Köllerström, "Heavy Traffic Theory for Queues with Several Servers. II," J. Appl. Prob., *16*, No. 2 (June 1979), pp. 393–401.
64. A. M. Lee and P. A. Longton, "Queueing Processes Associated with Airline Passenger Check-In," Oper. Res. Quart., *10*, No. 1 (March 1959), pp. 56–71.
65. P. Hokstad, "Approximations for the M/G/n Queue," Oper. Res., *26*, No. 3 (May–June 1978), pp. 510–23.
66. S. A. Nozaki and S. M. Ross, "Approximations in Finite Capacity Multi-Server Queues with Poisson Arrivals," J. Appl. Prob., *15* (1978), pp. 826–34.

67. F. S. Hillier and O. S. Yu, *Queueing Tables and Graphs*, New York: North-Holland, 1981.
68. J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Princeton: Van Nostrand, 1960.
69. B. Simon and R. D. Foley, "Some Results on Sojourn Times in Acyclic Jackson Networks," Management Sci., 25, No. 10 (October 1979), pp. 1027–34.
70. P. C. Kiessler, "A Simulation Analysis of Sojourn Times in a Jackson Network," Report VTR 8016, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, 1980.

## AUTHOR

**Ward Whitt,** A.B. (Mathematics), 1964, Dartmouth College; Ph.D. (Operations Research), 1968, Cornell University; Stanford University, 1968–1969; Yale University, 1969–1977; Bell Laboratories, 1977—. At Yale University, from 1973–1977, Mr. Whitt was Associate Professor in the departments of Administrative Sciences and Statistics. At Bell Laboratories he is in the Operations Research Department in the Network Analysis Center.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 9, November 1983
Printed in U.S.A.

# Performance of the Queueing Network Analyzer

By W. WHITT*

(Manuscript received March 11, 1983)

This paper describes the performance of the Queueing Network Analyzer (QNA), a software package developed at Bell Laboratories to calculate approximate congestion measures for networks of queues. QNA is compared with simulations and other approximations of several open networks of single-server queues. This paper illustrates how to apply QNA and indicates the quality that can be expected from the approximations. The examples here demonstrate the importance of the variability parameters used in QNA to describe non-Poisson arrival processes and nonexponential service-time distributions. For these examples, QNA performs much better than the standard Markovian algorithm, which does not use variability parameters. The accuracy of the QNA results (e.g., the expected delays) in these examples is satisfactory for engineering purposes.

## I. INTRODUCTION AND SUMMARY

This paper is a sequel to Whitt,[1] which described the software package called the Queueing Network Analyzer (QNA). QNA calculates approximate congestion measures for networks of queues. The first version of QNA treats open networks of multiserver queues with the first-come, first-served discipline and no capacity constraints. QNA is designed to treat non-Markovian models: The arrival processes need not be Poisson and the service-time distributions need not be exponential. QNA approximately characterizes other kinds of variability through variability parameters assigned to each arrival process and each service-time distribution. The first step in the algorithm is

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

to solve for the flow rates and the variability parameters of the internal arrival processes. The second step is to compute approximate congestion measures for each queue separately by regarding it as a standard GI/G/m queue in which the renewal arrival process and the service-time distribution are each partially characterized by their first two moments or, equivalently, the rate and variability parameters. The third and final step is to calculate congestion measures for the network as a whole.

This paper describes the performance of QNA by comparing it with simulations and other approximations of networks of queues. Even though QNA can analyze multiserver queues, only single-server queues are considered here. Among the other approximations in each case are the M/M/1 and M/G/1 approximations, which can be obtained from QNA by using default options. The M/M/1 approximation, which is embodied in the Markovian algorithms, is obtained by setting all variability parameters equal to 1. With the M/M/1 approximation, the nodes are treated as independent M/M/1 queues with the correct rates. The M/M/1 approximation yields the exact equilibrium distribution of queue lengths for the Markov model with Poisson external arrival processes, exponential service-time distributions and one customer class. The M/G/1 approximation is obtained by setting the variability parameter of each arrival process equal to 1 and using the specified service-time variability parameter $c_s^2$; then the expected waiting time at each node is $(1 + c_s^2)/2$ times the M/M/1 value.

The congestion measures we consider in the examples here are the expected waiting time (before beginning service) and the expected sojourn time (waiting time plus service time) at a node or in the entire network. Of course, QNA produces other congestion measures, but we are comparing with previously published simulation results, which are mostly limited to expected waiting times and sojourn times.

We begin in Section II with a single GI/G/1 queue and discuss the implications of previous work on approximations for the GI/G/1 queue.[2-7] In Section III we consider a single queue with a superposition arrival process and compare QNA with simulations by Albin.[8-10] In Section IV we consider a network of eight queues in series analyzed by Fraker,[11] and in Sections V and VI we consider two networks analyzed by Kuehn:[12] Section V treats a tightly coupled two-node network and Section VI treats a nine-node network. In Section VII we treat a five-node network used to model a Bell Laboratories computer system. Finally, in Section VIII we consider a model from Gelenbe and Mitrani[13] for a packet-switched communication network. The examples in Sections VII and VIII have input by classes and routes as in Section 2.3 of Whitt.[1]

These examples indicate the approximation quality that can be

expected in applications of QNA. They also demonstrate the importance of the variability parameters when the external arrival processes are not nearly Poisson or the service-time distributions are not nearly exponential. These examples also illustrate how to apply QNA, e.g., to model superposition arrival processes (Section III), to eliminate almost immediate feedback (Section V), and to conduct sensitivity analyses for the variability (Section VII).

## II. A SINGLE GI/G/1 QUEUE

We begin by considering the special network containing a single service facility, in particular, the GI/G/1 queue with service times and interarrival times each partially characterized by their first two moments or, equivalently, by the four parameters $\tau$ (the mean service time), $c_s^2$ (the squared coefficient of variation of the service time), $\lambda$ (the arrival rate), and $c_a^2$ (the squared coefficient of variation of the interarrival time).[1] The subscript indexing the node is suppressed since there is only one node.

It is useful to consider this model because it has been extensively studied and is relatively well understood. In many cases we can analytically determine the quality of the approximations for the GI/G/1 queue. Hence, we can get an idea about the quality of the approximations for more general networks. Of course, approximations for a node in a general network might be worse because the internal arrival processes usually are not actually renewal processes. On the other hand, the network operations of superposition and splitting tend to make stochastic point processes more like Poisson processes, so that larger networks may actually be better behaved.

The model specification above determines the approximate congestion measures produced by QNA, but the model specification is not complete since there are many service-time and interarrival-time distributions with the given parameters. The formulas produced by the QNA are approximations for all these systems, so it is natural to ask how the approximate congestion measures compare to the set of all possible values that are consistent with the partial specification. Fortunately, it is often possible to identify the set of all possible values.[2-5] Moreover, it is often possible to locate the more likely values by identifying the set of all possible values under various natural constraints on the distribution. When this cannot be done exactly, it can often be done approximately using bounds.[6]

We now give a brief summary of results evaluating approximations for the expected waiting time or, equivalently (by Little's formula[14]), the expected queue length in the GI/G/1 queue based on the four parameters $\lambda$, $c_a^2$, $\tau$, and $c_s^2$. First, recall that for the M/G/1 queue, with Poisson arrival process ($c_a^2 = 1$), the expected waiting time

actually depends on the service-time distribution only through the two parameters $\tau$ and $c_s^2$. Nonexponential interarrival-time distributions tend to be more difficult, however. For the GI/M/1 queue with an exponential service-time distribution ($c_s^2 = 1$), the expected waiting time depends on the interarrival-time distribution beyond the parameters $\lambda$ and $c_a^2$. Given $\lambda$ and $c_a^2$ in the GI/M/1 queue, the maximum relative error (upper bound minus lower bound divided by lower bound) in the mean queue length (number in system including anyone in service) is exactly $c_a^2$ (see Ref. 2). A similar, but somewhat less concise, result holds for the expected waiting time by virtue of Little's formula. The maximum relative error for the expected sojourn time (waiting time plus service time) is also $c_a^2$. This result suggests more generally that the reliability of the approximations might decrease when $c_a^2$ increases, which is consistent with numerical experience.

If we assume that the interarrival-time distribution is not too irregular, then the maximum relative error becomes much less. In the $H_k/M/1$ queue with a hyperexponential interarrival-time distribution (mixture of exponential distributions having $c_a^2 > 1$), the maximum relative error in the mean queue length is $(c_a^2 - 1)/2$ (see Ref. 4). It turns out that the extremal interarrival-time distributions for $H_k/M/1$ queues also are extremal for all interarrival-time distributions that have Increasing Mean Residual Life (IMRL) (also $c_a^2 > 1$) and all service-time distributions,[5] so that the maximum relative error in the mean queue length for IMRL/G/1 queues is $(c_a^2 - 1)/[2 + \rho(c_s^2 - 1)]$.

Other kinds of shape constraints for the GI/M/1 queue have been investigated by means of nonlinear programming.[3] In general, we conclude that if the distributions are not irregular, then the maximum relative error in the GI/G/1 queue might be about $0.05\, c_a^2$, e.g., about 10 percent when $c_a^2 = 2.0$.

From heavy-traffic limit theorems that describe the queue as $\rho \to 1$,[6] where $\rho = \lambda\tau$ is the traffic intensity, we know that asymptotically the queue length and waiting-time distributions depend on the interarrival-time and service-time distributions only through the four parameters $\lambda$, $c_a^2$, $\tau$, and $c_s^2$. This suggests that more generally the quality of the approximations might improve as $\rho$ increases. This is certainly consistent with experience for the GI/G/1 queue, but not necessarily for more complex networks, e.g., the tightly coupled two-node network here in Section V.

The heavy-traffic limit theorems are closely related to diffusion approximations because diffusion processes emerge as limits in the heavy-traffic limit theorems. We have recently compared various diffusion approximations for the expected waiting time in a GI/G/1 queue to known bounds.[6] We now show how QNA and other related approximations fit into this framework. Table I here compares four

Table I—Bounds and approximations for the expected waiting time, $EW$, in a GI/G/1 queue: Three cases

| | Parameter Values | | |
|---|---|---|---|
| | $c_a^2 = 0.5$<br>$c_s^2 = 4.0$<br>$\rho = 0.7$ | $c_a^2 = 2.0$<br>$c_s^2 = 4.0$<br>$\rho = 0.7$ | $c_a^2 = 0.8$<br>$c_s^2 = 4.0$<br>$\rho = 0.3$ |
| Daley's general upper bound | 5.75 | 9.00 | 1.82 |
| Monotone failure rate upper bound[5] | 5.83 | 7.50 | 1.07 |
| Kraemer and Langenbach-Belz[7] | 5.14 | 6.88 | 1.02 |
| QNA[1] | 5.14 | 7.00 | 1.02 |
| M/G/1 | 5.83 H | 5.83 L | 1.07 H |
| M/M/1 | 2.33 L | 2.33 L | 0.43 L |
| MFR lower bound[5] | 5.00 | 5.83 | 0.93 |

Notes: 1. In each case the mean service time is $\tau = 1$.
2. "H" indicates high (greater than or equal to) and "L" indicates low (less than or equal to) in comparison with the bounds.

approximations for the expected waiting time, $EW$, with various upper and lower bounds in the three cases in Table 1 of Ref. 6. The four approximations are the M/M/1, M/G/1, Kraemer and Langenbach-Belz,[7] and QNA.

The M/M/1 approximation is obtained by replacing both variability parameters $c_a^2$ and $c_s^2$ by 1. The M/M/1 approximation is produced by a direct application of the Markovian software packages. The M/G/1 approximation is obtained by replacing $c_a^2$ by 1 and using the specified value of $c_s^2$. The M/G/1 approximation $EW$ is the exact value for the approximating M/G/1 system since $EW$ depends on the service-time distribution only through its first two moments. Both the M/M/1 and M/G/1 approximations are produced by QNA using default options.

The QNA approximation is the Kraemer and Langenbach-Belz approximation when $c_a^2 \leq 1$ and is slightly greater when $c_a^2 > 1$.[1] The one case in which $c_a^2 > 1$ in Table I shows that the difference between the two approximations is small compared to the distance between the upper and lower Monotone Failure Rate (MFR) bounds.[5] The MFR bounds are for interarrival-time distributions with decreasing failure rate when $c_a^2 \geq 1$ and increasing failure rate when $c_a^2 \leq 1$. The MFR bounds are tight when $c_a^2 \geq 1$ but not when $c_a^2 < 1$ (see Ref. 5). Since the interarrival-time distribution need not have monotone failure rate, the range of all possible values is greater, but the MFR bounds indicate the more likely values.

For the three cases in Table I, the M/M/1 approximation performs very poorly, falling way outside the bounds. The M/G/1 approximation always coincides with one of the MFR bounds—the upper bound when $c_a^2 \leq 1$ and the lower bound when $c_a^2 \geq 1$—but it would be better to have an approximation somewhere in the middle between the bounds. When $c_a^2 \geq 1$, the QNA approximation is a convex combination of the

two MFR bounds.[5] Since the MFR bounds are tight when $c_a^2 \geq 1$, the QNA approximation always yields the exact value of $EW$ for some GI/G/1 system with the given parameters.[5] When $c_a^2 \leq 1$, the QNA approximation is slightly less than the convex combination of the MFR bounds. The convex combination of the MFR bounds is known to be an upper bound for $E_k/G/1$ systems,[5] so that it is appropriate to use a smaller value. We conjecture that the QNA approximation always yields an exact value of EW for some GI/G/1 system with the given parameters when $c_a^2 \leq 1$ too.

Table I provides a sample of the comparisons possible using the previous studies.[2-6] Since QNA coincides with the Kraemer and Langenbach-Belz approximation when $c_a^2 \leq 1$ and the Sakasegawa-Yu approximation when $c_a^2 \geq 1$, previous comparisons such as Tables 13 and 14 in Klincewicz and Whitt[3] also apply to QNA.

## III. A QUEUE WITH A SUPERPOSITION ARRIVAL PROCESS

In this section we consider one single-server queue with a superposition arrival process. Such a system with two component arrival processes is depicted in Fig. 1a. Since only one external arrival process at each node is allowed in QNA, this model cannot be analyzed directly, but it is easy to modify the model so that QNA does apply. We added dummy nodes with very low traffic intensity on each component arrival process, as shown in Fig. 1b. Since the new dummy nodes have low

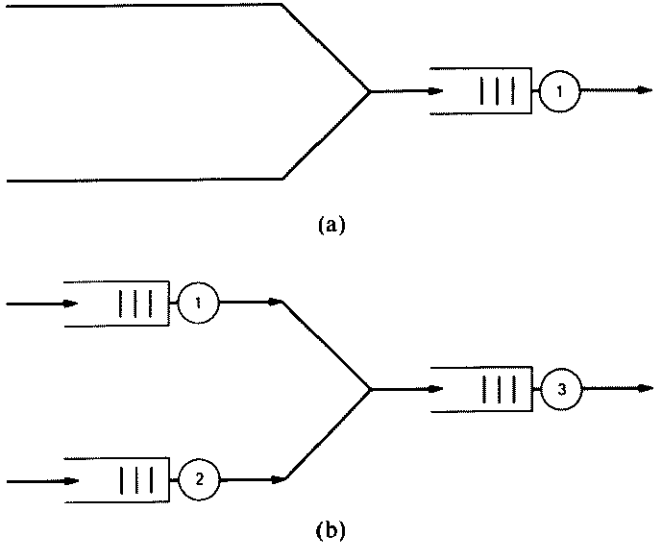



Fig. 1—(a) The original queue with a superposition arrival process. (b) The equivalent network with one external arrival process at each node.

traffic intensity, the rate and variability parameters of the departure processes from the dummy nodes will be almost identical to the corresponding parameters of the external arrival processes.

This model has recently been studied quite extensively by Albin[8-10] and is now relatively well understood. As in Section I, here we consider one illustrative example, which suggests the accuracy to expect more generally and shows the importance of the variability parameters.

The specific model we analyze is the $\Sigma GI_i/M/1$ system with an exponential service-time distribution and $n$ iid stationary renewal processes as component arrival processes. The total arrival rate is 1, so the rate of each component process is $n^{-1}$. We consider six cases involving two values of $n$, $n = 2$ and 16, and three values of the traffic intensity $\rho$, $\rho = 0.3$, 0.7, and 0.9. In each case the component renewal-interval distribution is $H_2^b$, i.e., the mixture of two exponentials with balanced means: one with mean $m_1$ realized with probability $p$ and the other with mean $m_2$ realized with probability $1 - p$, where $pm_1 = (1 - p)m_2$. In each case the squared coefficient of variation of the component renewal process interval is $c^2 = 6$. This is quite high variability, so that the component processes are not nearly Poisson. The three parameters of an $H_2$ distribution are determined by specifying the mean as $n$, $c^2 = 6$ and balanced means.

This model is taken from Chapter 3 and Appendix 6 of Albin.[8] Albin's approximations are based on the rate and variability parameters just as in QNA. In fact, the superposition approximation in QNA is a modification of Albin's procedure.[1] With Albin's procedure, the variability parameter of the superposition process is a convex combination of the variability parameters obtained from the stationary-interval method and asymptotic method described in Whitt.[15] The specific implementation of the stationary-interval method in Whitt[15] and Albin[8] is part of Kuehn's[12] algorithm for approximating networks of queues. Since Kuehn's implementation of the stationary-interval method is nonlinear, a different procedure is used in QNA. In QNA the variability parameter of the superposition process is a convex combination of the variability parameters obtained from the asymptotic method and a Poisson process.[1] Extensive experimentation has shown, however, that the approximation in QNA is very close to Albin's hybrid approximation, which performed very well in many experiments (about 3-percent average absolute relative percent error).

Several different approximations for the expected waiting time are compared with simulation in Table II. The simulation results were obtained in Albin.[8] The sample standard deviation is given in parentheses below the simulation estimate in Table II to indicate the statistical reliability of the estimate. The simulation program was written in FORTRAN using the "Super-Duper" subprogram in Mar-

Table II—A comparison of approximations and simulation of the expected waiting time, $EW$, in a $\Sigma GI/M/1$ queue with a superposition arrival process

| No. of Renewal Processes, $n$ | Traffic Intensity, $\rho$ | Simulation Estimate | Approximation Method: M/M/1 | Kuehn's Stationary-Interval Method | Asymptotic Method | Albin's Hybrid Procedure | QNA |
|---|---|---|---|---|---|---|---|
| 2 | 0.3 | 0.205 (0.007) | 0.128 (−37.6) | 0.231 (+12.7) | 0.281 (+37.1) | 0.240 (+17.1) | 0.236 (+15.1) |
| | 0.7 | 4.57 (0.14) | 1.63 (−64.3) | 3.54 (−22.5) | 5.32 (+16.4) | 4.51 (−1.3) | 4.64 (+1.5) |
| | 0.9 | 26.3 (1.2) | 8.1 (−69.2) | 18.2 (−30.8) | 28.2 (+7.2) | 27.5 (+4.6) | 27.5 (+4.6) |
| 16 | 0.3 | 0.138 (0.004) | 0.128 (−7.2) | 0.147 (+6.5) | 0.281 (+103.6) | 0.153 (+10.9) | 0.139 (+0.7) |
| | 0.7 | 2.57 (0.07) | 1.63 (−36.6) | 1.88 (−26.8) | 5.32 (+107.0) | 2.36 (−8.2) | 2.16 (−15.9) |
| | 0.9 | 20.8 (0.94) | 8.1 (−61.1) | 9.4 (−54.8) | 28.2 (+35.6) | 21.1 (+1.4) | 20.7 (−0.5) |
| Average Absolute Relative Percent Error | | | 46.0 | 25.7 | 51.2 | 7.3 | 6.4 |

Notes: 1. The total arrival rate is 1 in each case.
2. The component renewal processes have $H_2^b$ (hyperexponential) renewal-interval distributions with mean $n$, $c^2 = 6$, and balanced means.
3. The sample standard deviations appear below the simulation estimates in parentheses.
4. The relative percent error appears below the approximation values in parentheses.

saglia et al. to generate uniform random numbers.[16] A different random number seed was used for each simulation. The simulations began with an empty system, but the first 1000 customers were not counted to allow the system to approach steady-state. Each simulation consisted of 20 batches, with the number of customers per batch depending on the traffic intensity: 3,000 per batch for $\rho = 0.3$, 15,000 per batch for $\rho = 0.7$, and 50,000 per batch for $\rho = 0.9$. Even though much more simulation time was spent on the cases with higher traffic intensities, the statistical reliability was slightly less.

In Table II, in parentheses below the approximation values are the relative percent errors (RE), which are defined as

$$RE = 100(\text{Approx.} - \text{Simul.})/\text{Simul.} \tag{1}$$

At the bottom of Table II are the average absolute relative percent errors (ARE), which are defined as

$$ARE = \sum_{i=1}^{6} |RE_i|/6. \tag{2}$$

Dividing by the simulation value perhaps inflates the errors when $\rho =$

0.3 too much, but these summary measures provide a good overall comparison.

The stationary-interval method and the M/M/1 approximation are not bad for large $n$ and small $\rho$ because the superposition process converages to a Poisson process as $n \to \infty$ and the queue reflects this if $\rho$ is not too big, in particular, if $n(1 - \rho)^2$ is sufficiently large.[10,17] However, for $\rho = 0.9$, these two methods perform poorly. On the other hand, the asymptotic method performs reasonably well for $\rho = 0.9$, but not well in other cases. In particular, the asymptotic method does not reflect the convergence to a Poisson process as $n \to \infty$; it gives the same answers for $n = 2$ and 16. So, for fixed $\rho$, the asymptotic method gets worse as $n$ increases.

As Albin determined in extensive experiments,[8,9] her hybrid approximation is much better than either basic method alone. This example also illustrates how close QNA is to Albin's hybrid procedure. For queues with superposition arrival processes, we conclude that QNA usually gives reasonable results and strongly dominates the two basic methods.

In closing this section, we add a caveat. The component processes in the simulation for Table II, in Albin's hybrid procedure and in QNA, are all based on the case of balanced means. However, as discussed in Whitt,[4,5] given the first two moments, the one-parameter family of renewal processes with hyperexponential renewal-interval distributions range from a Poisson process to a batch Poisson process with geometrically distributed batch size. For a single renewal arrival process, the expected waiting time, $EW$, in an $H_2/G/1$ queue also ranges between these same extremes, i.e., the M/G/1 and the $M^B/G/1$ systems. Unfortunately, no related theory yet exists for superposition arrival processes. However, we can easily describe what happens in the two extremes. The superposition of independent Poisson processes is Poisson and the superposition of independent batch Poisson processes with geometrically distributed batches having a common mean is batch Poisson with geometrically distributed batches. Thus, we *conjecture* that the maximum and minimum values for $EW$ with a superposition of iid $H_2$-renewal processes correspond to the $M^B/G/1$ and M/G/1 systems, respectively. If the conjecture is true, then we would have the same range of possible values for $H_2$-superposition arrival processes as for $H_2$-renewal processes. However, if the component processes are not too batchy, then the superposition process will become more Poisson as $n$ increases. We should usually expect the superposition process to be more nearly Poisson as $n$ increases. To summarize, this heuristic analysis suggests that the range of possible values for $EW$ in the $\Sigma GI_i/G/1$ queue given the basic parameters $\lambda$, $c_a^2$, $\tau$, $c_s^2$ may be about the same as for the GI/G/1 queue. In fact, the

superposition operation may actually make the arrival process better behaved.

## IV. EIGHT QUEUES IN SERIES

In this section we apply QNA to a network of eight single-server queues in series previously analyzed by Fraker.[11] The external arrival process is Poisson and all service-time distributions are Erlang. Fraker considered eight cases involving four traffic intensities ($\rho$ = 0.3, 0.5, 0.7, and 0.9) and four Erlang service-time distributions ($M = E_1$, $E_4$, $E_8$, and $D = E_\infty$). Each of the traffic intensities and each of the service-time distributions are assigned randomly to two of the eight nodes. Fraker developed an approximation for these systems and compared it with simulations.

Tables III and IV describe Fraker's first two cases and the approximations for the expected waiting time at each node. The service-time squared coefficient of variation specifies the Erlang distribution since $c^2 = k^{-1}$ for $E_k$. Fraker made three simulation runs of 2500 customers, discarding the first 500 in each case to damp out the transient effects of starting the simulation. Statistics were collected for six blocks of 1000 customers each. Unfortunately, this is not enough to produce very good accuracy, especially for the nodes with higher traffic intensities. (Compare with the simulation length in Section III.) The statistical reliability can be seen from the results of the six runs displayed in Fraker.[11] (These also appear in Appendix 1 of Whitt.[18]) An idea of the variability can also be seen from node 1 because all the approximations except the M/M/1 approximation are exact for node 1. When $\rho$ = 0.9 the length of a 95-percent confidence interval approximately equals the estimated value; when $\rho$ = 0.7 the length of

Table III—A comparison of approximations and simulation of the expected waiting time at each node in Fraker's model of eight queues in series: Case 1

| Node No. | Traffic Intensity, $\rho_j$ | Squared Coefficient of Variation, $c_{aj}^2$ | Simulated Value | Approximation Methods | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | (Markovian Network) M/M/1 | (Asymptotic Method) M/G/1 | (Lag-1 Correlations) Fraker | QNA | |
| | | | | | | | $EW_j$ | $c_{aj}^2$ |
| 1 | 0.7 | 1/8 | 0.98 | 1.63 | 0.92 | 0.92 | 0.92 | 1.00 |
| 2 | 0.5 | 1 | 0.30 | 0.50 | 0.50 | 0.38 | 0.38 | 0.61 |
| 3 | 0.5 | 0 | 0.19 | 0.50 | 0.25 | 0.13 | 0.16 | 0.71 |
| 4 | 0.7 | 1/4 | 0.73 | 1.63 | 1.02 | 0.62 | 0.63 | 0.58 |
| 5 | 0.3 | 0 | 0.01 | 0.13 | 0.07 | 0.01 | 0.01 | 0.42 |
| 6 | 0.9 | 1 | 7.50 | 8.10 | 8.10 | 6.03 | 5.56 | 0.40 |
| 7 | 0.9 | 1/8 | 3.91 | 8.10 | 4.55 | 4.16 | 4.09 | 0.89 |
| 8 | 0.3 | 1/4 | 0.00 | 0.13 | 0.08 | 0.01 | 0.01 | 0.33 |

Note: The arrival process is Poisson with rate 1 and the service-time distributions are Erlang.

Table IV—A comparison of approximations and simulation of the expected waiting time at each node in Fraker's model of eight queues in series: Case 2

| Node No. | Traffic Intensity, $\rho_j$ | Squared Coefficient of Variation, $c_{aj}^2$ | Simulated Value | Approximation Methods: (Markovian Network) M/M/1 | (Asymptotic Method) M/G/1 | (Lag-1 Correlations) Fraker | QNA $EW_j$ | QNA $c_{aj}^2$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.9 | 1 | 6.25 | 8.10 | 8.10 | 8.10 | 8.10 | 1.00 |
| 2 | 0.7 | 1/8 | 0.84 | 1.63 | 0.92 | 0.92 | 0.92 | 1.00 |
| 3 | 0.3 | 1/4 | 0.01 | 0.13 | 0.08 | 0.04 | 0.04 | 0.61 |
| 4 | 0.9 | 0 | 2.61 | 8.10 | 4.05 | 2.45 | 2.28 | 0.58 |
| 5 | 0.3 | 1/8 | 0.00 | 0.13 | 0.07 | 0.00 | 0.00 | 0.27 |
| 6 | 0.5 | 1/4 | 0.02 | 0.50 | 0.31 | 0.05 | 0.06 | 0.26 |
| 7 | 0.5 | 0 | 0.02 | 0.50 | 0.25 | 0.02 | 0.02 | 0.26 |
| 8 | 0.7 | 1 | 0.78 | 1.63 | 1.63 | 0.82 | 0.89 | 0.25 |

Note: The arrival process is Poisson with rate 1 and the service-time distributions are Erlang.

a 95-percent confidence interval is about 25 percent of the estimated value.

Table V compares the approximations with simulation for the nodes with traffic intensity $\rho = 0.7$ in all eight cases. Since the approximations are exact for the first node, the first node is not included for the cases in which $\rho_1 = 0.7$ (Cases 1, 5, and 6). For the approximations, the difference between the approximation value and the simulation value is displayed.

Tables III through V show that QNA performs about the same as Fraker's approximation, which is based on lag-1 correlations and is especially designed for queues with Erlang service times. Both these approximations performed significantly better than the M/G/1 approximation, which in turn performs significantly better than the M/M/1 approximation.

Additional analysis of Fraker's models plus other queues in series is contained in Whitt.[18] The performance of QNA in these other cases is consistent with the description here.

## V. A TIGHTLY COUPLED NETWORK OF TWO NODES

In this section we consider a two-node network analyzed by Kuehn[12] and Gelenbe and Mitrani.[13] This network is depicted in Fig. 2. It has one external arrival process, which comes to node 1. Customers completing service at node 1 leave the system with probability 1/2; otherwise they go to node 2 and then back to node 1 to be served again. At node 2 customers are immediately fed back to node 2 for another service with probability $q_{22}$, but in most cases $q_{22} = 0$.

We first consider Kuehn's experiment. There are eight cases with

Table V—The expected waiting time at the nodes with $\rho_i = 0.7$ in Fraker's eight cases of eight single-server queues in series

| Case No. | Node No. | Simulated Value of $EW_i$ | Approximation Methods | | | |
|---|---|---|---|---|---|---|
| | | | M/M/1 | M/G/1 | Fraker | QNA |
| 1 | 4 | 0.73 | +0.90 | +0.29 | −0.11 | −0.10 |
| 2 | 2 | 0.84 | +0.79 | +0.08 | +0.08 | +0.08 |
| | 8 | 0.78 | +0.85 | +0.85 | +0.04 | +0.11 |
| 3 | 4 | 1.08 | +0.55 | +0.55 | −0.04 | +0.04 |
| | 8 | 0.55 | +1.08 | +0.37 | 0.00 | −0.02 |
| 4 | 3 | 1.52 | +0.11 | +0.11 | −0.04 | −0.04 |
| | 6 | 0.02 | +1.61 | +0.80 | +0.09 | +0.16 |
| 5 | 3 | 0.74 | +0.89 | +0.28 | +0.10 | +0.11 |
| 6 | 5 | 0.33 | +1.30 | +0.49 | +0.05 | −0.01 |
| 7 | 4 | 0.78 | +0.85 | +0.14 | −0.06 | −0.02 |
| | 7 | 0.17 | +1.46 | +0.65 | +0.02 | −0.01 |
| 8 | 5 | 0.50 | +1.13 | +0.52 | +0.04 | +0.02 |
| Average | | 0.67 | +0.96 | +0.43 | +0.01 | +0.03 |
| Average Absolute Difference | | | 0.96 | 0.43 | 0.05 | 0.06 |

Note: The value for the approximations is the approximate value minus the simulated value.

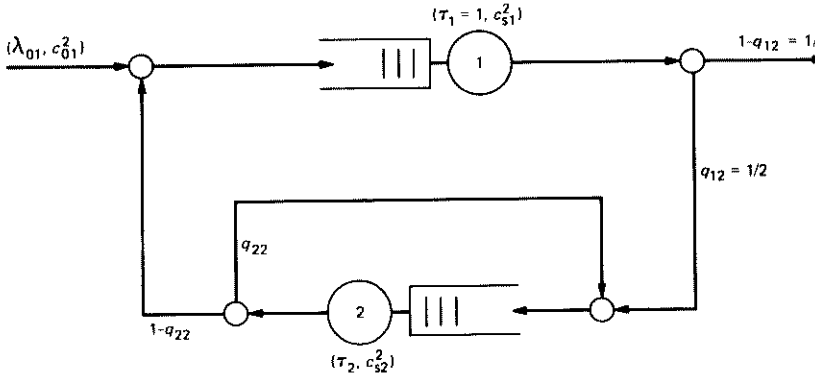



Fig. 2—Kuehn's first example: A network of two queues with one external arrival process.

three values of the external arrival rate for each case: $\lambda_{01} = 0.15$, 0.30, and 0.45. In each case the mean service time at node 1 is $\tau_1 = 1$ and the transition probability from node 1 to node 2 is $q_{12} = 1/2$, so that the traffic intensity at node 1 is $\rho_1 = 2\lambda_{01}$. For node 1 these three external arrival rates correspond roughly to light traffic ($\rho_1 = 0.3$), moderate traffic ($\rho_1 = 0.6$), and heavy traffic ($\rho_1 = 0.9$). In Cases 2 and 3 the traffic intensity at node 2 is the same as at node 1, i.e., $\rho_2 = 2\lambda_{01}$, but in all other cases it is $\rho_2 = \lambda_{01}$. In these other cases node 2 is always in relatively light traffic. The external arrival process is always a renewal process. Each interarrival-time distribution and service-time distribution is one of four distributions: deterministic ($D$ with $c^2$

= 0), Erlang of order 4 ($E_4$ with $c^2 = 0.25$), exponential ($M$ with $c^2 = 1$), or hyperexponential with balanced means ($H_2^b$ with $c^2 = 2.25$). The eight cases are indicated in Table VI. The system type is described by a triple such as $M/H_2/E_4$, which means that the interarrival-time distribution is $M$ and the service-time distributions at nodes 1 and 2 are $H_2$ and $E_4$, respectively.

The results are described in Tables VII and VIII. The simulation results and Kuehn's approximation are taken from Kuehn.[12] Two different approximation results are given for QNA in Table VII. The first column is the standard application of QNA with the network reconfigured to eliminate immediate feedback in the one case it occurs, at node 2 in Case 3. (See Section 3 of Ref. 1.) The final column of Table VII is an adjusted version of QNA to eliminate almost immediate feedback, which we discuss below.

For this network the quality of the standard QNA approximation is about the same as Kuehn's approximation. They both work well for low and moderate traffic intensities, e.g., about 10-percent average absolute relative percent error when $\rho_1 = 0.6$ (Table VIII), but not so well in heavy traffic. This two-node network presents an obvious difficulty for QNA. The network is tightly coupled so that many departures from node 1 rapidly return to node 1 for additional service. However, since these returning customers first pass through node 2, there is no immediate feedback, so that QNA does not reconfigure the network to eliminate the feedback. Nevertheless, it is evident that this almost immediate feedback for node 1 is very similar to immediate feedback and the potential exists for better results by reconfiguring the network to eliminate this feedback too.

In all cases except 2 and 3, almost immediate feedback is eliminated by applying the standard version of QNA with immediate feedback elimination twice. The first time we apply QNA to the full network and the second time we apply QNA with node 2 removed. When node

Table VI—The eight cases of Kuehn's two-node example in Fig. 2

| System Number | System Type | Defining Parameters | | | | |
|---|---|---|---|---|---|---|
| | | $c_{01}^2$ | $\tau_2$ | $c_{s1}^2$ | $c_{s2}^2$ | $q_{22}$ |
| 1 | $M/H_2/E_4$ | 1.00 | 1.0 | 2.25 | 0.25 | 0.0 |
| 2 | $M/H_2/E_4$ | 1.00 | 2.0 | 2.25 | 0.25 | 0.0 |
| 3 | $M/H_2/E_4$ | 1.00 | 1.0 | 2.25 | 0.25 | 0.5 |
| 4 | $M/H_2/H_2$ | 1.00 | 1.0 | 2.25 | 2.25 | 0.0 |
| 5 | $M/E_4/E_4$ | 1.00 | 1.0 | 0.25 | 0.25 | 0.0 |
| 6 | $M/D/D$ | 1.00 | 1.0 | 0.00 | 0.00 | 0.0 |
| 7 | $H_2/H_2/E_4$ | 2.25 | 1.0 | 2.25 | 0.25 | 0.0 |
| 8 | $E_4/H_2/E_4$ | 0.25 | 1.0 | 2.25 | 0.25 | 0.0 |

Notes: 1. In each case $\tau_1 = 1$ and $q_{12} = 1/2$.
2. In each case the arrival rate assumes one of three values: $\lambda_{01} = 0.15$, 0.30, and 0.45.

Table VII—A comparison of approximations and simulation of the expected total sojourn time (waiting time plus service time) in Kuehn's two-node network

| System No. | External Arrival Rate, $\lambda_{01}$ | Simulation (with 95-Percent Confidence Intervals) | Approximation Methods | | | | |
|---|---|---|---|---|---|---|---|
| | | | M/M/1 | M/G/1 | Kuehn | QNA | QNA Adjusted |
| 1 | 0.15 | 4.24 ± 0.08 | 4.04 | 4.50 | 4.51 | 4.51 | 4.24 |
| | 0.30 | 7.51 ± 0.29 | 6.43 | 8.13 | 8.22 | 8.26 | 7.25 |
| | 0.45 | 27.27 ± 5.63 | 21.82 | 32.67 | 33.47 | 34.10 | 27.35 |
| 2 | 0.15 | 5.95 ± 0.89 | 5.29 | 5.65 | 5.94 | 5.95 | 4.41 |
| | 0.30 | 10.91 ± 0.64 | 8.50 | 9.79 | 10.89 | 11.01 | 8.22 |
| | 0.45 | 49.49 ± 6.80 | 31.00 | 38.74 | 46.31 | 46.57 | 36.79 |
| 3 | 0.15 | 6.09 ± 0.19 | 5.29 | 5.65 | 6.11 | 6.12 | 4.48 |
| | 0.30 | 11.08 ± 0.63 | 8.50 | 9.79 | 11.60 | 11.66 | 8.25 |
| | 0.45 | 61.72 ± 17.99 | 31.00 | 38.74 | 51.27 | 51.43 | 35.07 |
| 4 | 0.15 | 4.50 ± 0.16 | 4.04 | 4.68 | 4.69 | 4.69 | 4.42 |
| | 0.30 | 7.96 ± 0.55 | 6.43 | 8.55 | 8.71 | 8.79 | 7.68 |
| | 0.45 | 29.91 ± 6.14 | 21.82 | 33.48 | 34.94 | 36.75 | 28.17 |
| 5 | 0.15 | 3.66 ± 0.05 | 4.04 | 3.63 | 3.63 | 3.64 | 3.81 |
| | 0.30 | 5.35 ± 0.16 | 6.43 | 5.13 | 4.93 | 5.00 | 5.67 |
| | 0.45 | 18.29 ± 3.27 | 21.82 | 14.67 | 12.80 | 12.80 | 18.12 |
| 6 | 0.15 | 3.43 ± 0.04 | 4.04 | 3.51 | 3.49 | 3.51 | 3.72 |
| | 0.30 | 4.83 ± 0.07 | 6.43 | 4.71 | 4.42 | 4.56 | 5.42 |
| | 0.45 | 13.59 ± 1.79 | 21.82 | 12.41 | 9.79 | 10.32 | 16.73 |
| 7 | 0.15 | 4.73 ± 0.12 | 4.04 | 4.50 | 4.60 | 4.62 | 4.78 |
| | 0.30 | 9.04 ± 0.56 | 6.43 | 8.13 | 8.59 | 8.92 | 9.14 |
| | 0.45 | 46.83 ± 10.64 | 21.82 | 32.67 | 35.88 | 39.74 | 39.50 |
| 8 | 0.15 | 3.67 ± 0.09 | 4.04 | 4.50 | 4.19 | 4.43 | 3.91 |
| | 0.30 | 5.78 ± 0.16 | 6.43 | 8.13 | 7.49 | 7.84 | 6.09 |
| | 0.45 | 17.46 ± 1.67 | 21.82 | 32.67 | 29.57 | 30.68 | 20.47 |

Notes: 1. In each case the traffic intensity at node 1 is $\rho_1 = 2\lambda_{01}$.
2. In Cases 2 and 3 the traffic intensity at node 2 is $\rho_2 = 2\lambda_{01}$; otherwise it is $\rho_2 = \lambda_{01}$.

2 is removed, the feedback to node 1 becomes immediate and the network is reconfigured by QNA to eliminate it. We use the second run with node 2 removed to determine the expected waiting time per visit at node 1. We use the first run to determine the expected number of visits to node 1 and the expected total sojourn time at node 2.

We do not treat nodes 1 and 2 symmetrically in Cases 1 and 4 through 8 because $\rho_1 = 2\rho_2$ so that $\rho_2$ is relatively small compared to $\rho_1$. Customers that return to node 1 via node 2 will not be delayed long at node 2 before coming back, but customers returning to node 2 via node 1 will be delayed relatively longer before coming back. If we had $\rho_1 < \rho_2$, we would remove node 1 in the second run of the QNA and focus instead on node 2.

In Cases 2 and 3 the traffic intensities at nodes 1 and 2 are equal, so the motivation for eliminating almost immediate feedback is less. What we have done for Table IV is first calculate the congestion

Table VIII—A comparison of approximation methods in Kuehn's two-node network with $\lambda_{01} = 0.3$ ($\rho_1 = 0.6$): The average absolute relative percent error in the expected total sojourn time compared with simulation

| System Number | Approximation Methods | | | | | |
|---|---|---|---|---|---|---|
| | M/M/1 | M/G/1 | Kuehn | QNA Standard | QNA Adjusted | QNA Refined |
| 1 | −14.3 | +8.3 | +9.5 | +10.0 | −3.5 | −3.5 |
| 2 | −22.1 | −10.3 | −0.2 | +0.9 | −24.7 | +0.9 |
| 3 | −23.3 | −11.6 | +4.7 | +5.2 | −25.5 | +5.2 |
| 4 | −19.2 | +7.4 | +9.4 | +10.4 | −3.5 | −3.5 |
| 5 | +20.2 | −4.1 | −7.9 | −6.5 | +6.0 | +6.0 |
| 6 | +33.1 | −2.5 | −8.5 | −5.6 | +12.2 | +12.2 |
| 7 | −28.9 | −10.1 | −5.0 | −1.3 | +1.1 | +1.1 |
| 8 | +11.2 | +40.7 | +29.6 | +35.6 | +5.4 | +5.4 |
| Average Percent Error | 21.5 | 11.9 | 9.4 | 9.4 | 10.2 | 4.7 |

measures for node 1 via the second run of QNA with node 2 removed as before. Then we use the results for node 1 to approximate the variability parameter of the arrival process to node 2. Finally, we analyze node 2 in isolation with the correct rates and this approximate arrival variability parameter. This works slightly better than the first procedure, but neither works well.

The results demonstrate that the standard version of QNA performs relatively well in Cases 2 and 3 when $\rho_1 = \rho_2$. The adjustment to eliminate almost immediate feedback yields a significant improvement when $\rho_1 > \rho_2$, but the results after adjustment to eliminate almost immediate feedback are much worse when $\rho_1 = \rho_2$.

As a refined procedure for this two-node network, we suggest eliminating almost immediate feedback at the node with higher traffic intensity when the traffic intensities differ significantly, and using the standard QNA algorithm otherwise. The refined procedure in Table VIII is standard QNA in Cases 2 and 3 and the adjusted QNA in all other cases.

Table VIII displays the relative percentage errors for all the approximations for the eight cases with $\lambda_{01} = 0.3$ ($\rho_1 = 0.6$). The refined procedure in the last column yields very good results. Table VIII also demonstrates that the standard version of QNA is significantly better than the M/M/1 approximation, but not uniformly better. In some cases, e.g., in Case 8, errors in opposite directions can cancel for the M/M/1 approximation.

The improvement from eliminating almost immediate feedback "by hand" suggests that it would be desirable to develop an automatic procedure for eliminating almost immediate feedback to incorporate

in QNA, and this is being investigated. It also indicates the potential for "tuning" QNA for particular applications.

We now consider Gelenbe and Mitrani's[13] experiment, which consists of five cases. The network is as depicted in Fig. 2 except the routing probability $q_{12}$ is not exactly 1/2. The parameter values are given in Table IX and the results in Table X (pp. 137, 138 of Gelenbe and Mitrani[13]). We only include the best of three approximation schemes discussed by Gelenbe and Mitrani. Unfortunately, Gelenbe and Mitrani provided no information about the statistical reliability of the simulation estimates. Since $\rho_1$ and $\rho_2$ are nearly equal in each case, we did not try to eliminate almost immediate feedback.

The M/M/1 approximation values are obviously much too large because the M/M/1 approximation does not reflect the low variability of the service times. The M/G/1 approximation is much too low at node 2 because it does not benefit from the feedback elimination procedure. The Gelenbe-Pujolle procedure is better than the M/G/1 procedure, but not uniformly so. The QNA approximation is clearly

Table IX—The parameter values for Gelenbe and Mitrani's experiment with the two-node network in Fig. 2

| Case No. | Parameter Values | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda_{01}$ | $c_{01}^2$ | $\tau_1$ | $c_{s1}^2$ | $\tau_2$ | $c_{s2}^2$ | $q_{11}$ | $q_{12}$ | $q_{21}$ | $q_{22}$ |
| 1 | 0.512 | 0.941 | 0.911 | 0.427 | 0.840 | 0 | 0 | 0.510 | 0.497 | 0.503 |
| 2 | 0.410 | 0.944 | 0.916 | 0.423 | 0.840 | 0 | 0 | 0.509 | 0.501 | 0.499 |
| 3 | 0.342 | 0.945 | 0.914 | 0.414 | 0.840 | 0 | 0 | 0.516 | 0.494 | 0.506 |
| 4 | 0.293 | 0.967 | 0.904 | 0.432 | 0.840 | 0 | 0 | 0.512 | 0.498 | 0.502 |
| 5 | 0.257 | 0.952 | 0.911 | 0.422 | 0.840 | 0 | 0 | 0.504 | 0.493 | 0.507 |

Table X—A comparison of approximations with simulations: The expected number of customers at each node in the two-node network in Gelenbe and Mitrani[13]

| Case No. | Queue No. | Arrival Rate, $\lambda_j$ | Traffic Intensity, $\rho_j$ | Simulation Values | Approximation Methods | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | M/M/1 | M/G/1 | Gelenbe-Pujolle | QNA |
| 1 | 1 | 1.04 | 0.952 | 13.82 | 19.83 | 14.14 | 11.96 | 11.98 |
| | 2 | 0.53 | 0.901 | 7.83 | 9.10 | 4.55 | 5.74 | 5.88 |
| 2 | 1 | 0.84 | 0.765 | 2.36 | 3.26 | 2.32 | 2.09 | 2.31 |
| | 2 | 0.43 | 0.713 | 1.87 | 2.48 | 1.24 | 1.69 | 1.84 |
| 3 | 1 | 0.71 | 0.646 | 1.60 | 1.82 | 1.29 | 1.20 | 1.41 |
| | 2 | 0.36 | 0.620 | 1.47 | 1.63 | 0.82 | 1.15 | 1.29 |
| 4 | 1 | 0.60 | 0.543 | 1.05 | 1.19 | 0.85 | 0.82 | 0.98 |
| | 2 | 0.31 | 0.519 | 1.01 | 1.08 | 0.54 | 0.78 | 0.90 |
| 5 | 1 | 0.52 | 0.472 | 0.76 | 0.89 | 0.63 | 0.62 | 0.76 |
| | 2 | 0.26 | 0.445 | 0.73 | 0.80 | 0.40 | 0.59 | 0.70 |

the best, but it may underestimate the congestion for high traffic intensities.

## VI. KUEHN'S NINE-NODE NETWORK

We now consider a nine-node network analyzed by Kuehn,[12] which is depicted in Fig. 3. The mean service time at node j is $\tau_j = 1$ for each j. There are three external arrival processes with $\lambda_{0j} = 0.5$ for each j; these come to nodes 1, 2, and 3. Kuehn let the three external arrival processes be Poisson processes. As in Section V, all service-time distributions are $D$, $E_k$, $M$, or $H_2^b$. Kuehn considered two cases: homogeneous servers, in which all the service-time distributions are identical, and heterogeneous servers, in which nodes 1 through 3 have one service-time distribution and nodes 4 through 9 have another.

Kuehn compared his approximation with the M/M/1 and M/G/1 approximations and simulation for service-time variability parameters ranging from $c_{sj}^2 = 0$ to 4. He focused on the expected total sojourn time (waiting time plus service time) in the network and the expected sojourn time per visit in node 4. For this network Kuehn found, first, that the service-time variability parameters are significant (the sojourn-time measures increase significantly with $c_{sj}^2$); second, that his approximation tracks the simulation well; and, third, that the M/G/1 approximation also works well, but not quite as well as his approximation.

We obtained similar results applying QNA. The QNA approximation values are indistinguishable from the approximation values displayed graphically by Kuehn, which are consistently within the sim-

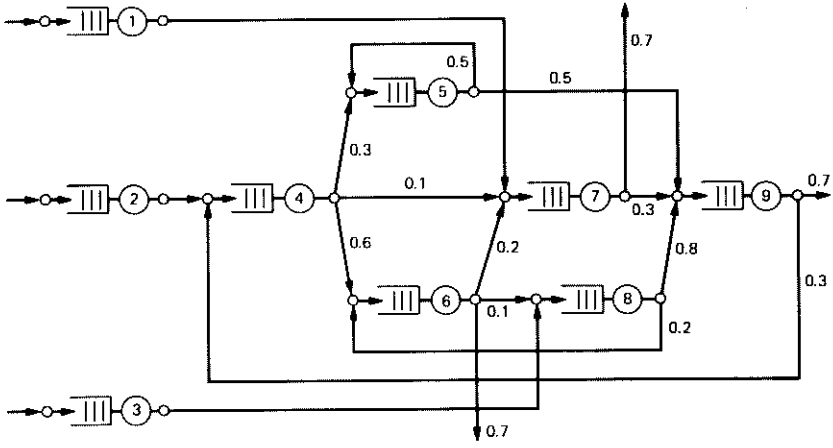



Fig. 3—Kuehn's second example: A network of nine queues with three Poisson external arrival processes.

ulation confidence intervals. In Table XI we display some of our results. Here we consider only the case of homogeneous servers in which $c_{sj}^2 = 0$, 1, or 4. However, we let the variability parameter of all external arrival processes by $c_{0j}^2 = 0.25$, 1.0, or 4.0. We thus obtain nine cases.

Kuehn did simulations only in the case $c_{0j}^2 = 1$. The different values obtained by QNA when $c_{0j}^2 \neq 1$ suggest that the congestion measures in this model are more sensitive to the variability of the service times than the variability of the interarrival times. Of course, we should expect that the M/G/1 approximation might perform well when the variability parameters of the external arrival processes are 1 or close to 1, but for relatively large and interconnected networks the M/G/1 approximation may perform well for other external arrival processes. It performs reasonably well here when $c_{0j}^2 = 0.25$ or 4.0.

## VII. A COMPUTER SYSTEM MODEL: INPUT BY ROUTES

In this section we apply QNA to a network with input by customer classes and routes as in Section 2.3 of Ref. 1. We compare QNA to a simulation model used in the development of a computer system at Bell Laboratories. In this model there are five nodes and two customer classes.

The customer classes correspond to typical functions performed by the system. The route for each class represents a typical sequence of operations performed by the system to process one of these functions.

Table XI—Approximations of expected sojourn times in Kuehn's nine-node network in Fig. 3: The case of homogeneous servers, $c_{sj}^2$ identical for all j

| Expected Total Sojourn Time in the Network | | Service-Time Variability Parameters | | |
|---|---|---|---|---|
| | | $c_{sj}^2 = 0$ | $c_{sj}^2 = 1$ | $c_{sj}^2 = 4$ |
| QNA | $c_{0j}^2 = 0.25$ | 6.1 | 11.5 | 28.3 |
| | $c_{0j}^2 = 1.00$ | 7.5 | 12.9 | 29.6 |
| | $c_{0j}^2 = 4.00$ | 12.3 | 17.7 | 34.4 |
| M/G/1 | | 8.2 | 12.9 | 27.1 |
| M/M/1 | | 12.9 | 12.9 | 12.9 |
| **Expected Sojourn Time per Visit in Node 4** | | **Service-Time Variability Parameters** | | |
| | | $c_{sj}^2 = 0$ | $c_{sj}^2 = 1$ | $c_{sj}^2 = 4$ |
| QNA | $c_{0j}^2 = 0.25$ | 1.76 | 3.71 | 9.64 |
| | $c_{0j}^2 = 1.00$ | 2.34 | 4.25 | 10.15 |
| | $c_{0j}^2 = 4.00$ | 4.38 | 6.29 | 12.19 |
| M/G/1 | | 2.62 | 4.25 | 9.12 |
| M/M/1 | | 4.25 | 4.25 | 4.25 |

In the simulation model, both the routes and the service times at the nodes are deterministic for each class. Hence, the input for QNA is just as specified in Section 2.3 of Ref. 1 with the service-time variability parameters set equal to 0, i.e., $c^2_{skj} = 0$ for each class k and j on class k's route. The two routes we consider are given in Tables XII and

Table XII—The data for the first route in the model of Section VII

| Number | Node | Mean Service Time |
|---|---|---|
| 1 | 2 | 4 |
| 2 | 1 | 10 |
| 3 | 1 | 66 |
| 4 | 1 | 60 |
| 5 | 1 | 106 |
| 6 | 1 | 65 |
| 7 | 4 | 30 |
| 8 | 1 | 16 |
| 9 | 4 | 30 |
| 10 | 1 | 36 |
| 11 | 1 | 12 |
| 12 | 1 | 40 |
| 13 | 4 | 30 |
| 14 | 1 | 16 |
| 15 | 4 | 30 |
| 16 | 1 | 36 |
| 17 | 1 | 62 |
| 18 | 1 | 63 |
| 19 | 1 | 42 |
| 20 | 1 | 14.5 |
| 21 | 3 | 550 |
| 22 | 5 | 0.01 |
| 23 | 3 | 50 |
| 24 | 1 | 10 |
| 25 | 1 | 8 |
| 26 | 1 | 20.5 |
| 27 | 1 | 55.5 |
| 28 | 1 | 42 |
| 29 | 1 | 14.5 |
| 30 | 3 | 550 |
| 31 | 5 | 0.01 |
| 32 | 3 | 50 |
| 33 | 1 | 10 |
| 34 | 1 | 8 |
| 35 | 1 | 20.5 |
| 36 | 1 | 63.5 |
| 37 | 1 | 76 |
| 38 | 4 | 30 |
| 39 | 1 | 36 |
| 40 | 1 | 15.5 |
| 41 | 1 | 45 |
| 42 | 4 | 30 |
| 43 | 1 | 16 |
| 44 | 4 | 30 |
| 45 | 1 | 16 |
| 46 | 4 | 30 |
| 47 | 1 | 16 |
| 48 | 4 | 30 |
| 49 | 1 | 36 |
| 50 | 1 | 38 |
| 51 | 1 | 45 |
| 52 | 4 | 30 |
| 53 | 1 | 16 |
| 54 | 4 | 30 |
| 55 | 1 | 36 |
| 56 | 1 | 15.5 |
| 57 | 1 | 45 |
| 58 | 4 | 30 |
| 59 | 1 | 16 |
| 60 | 4 | 30 |
| 61 | 1 | 16 |
| 62 | 4 | 30 |
| 63 | 1 | 16 |
| 64 | 4 | 30 |
| 65 | 1 | 36 |
| 66 | 1 | 38 |
| 67 | 1 | 45 |
| 68 | 4 | 30 |
| 69 | 1 | 16 |
| 70 | 4 | 30 |
| 71 | 1 | 36 |
| 72 | 1 | 27 |
| 73 | 1 | 25 |
| 74 | 1 | 40 |
| 75 | 4 | 30 |
| 76 | 1 | 16 |
| 77 | 4 | 30 |
| 78 | 1 | 36 |
| 79 | 1 | 13 |
| 80 | 1 | 58 |
| 81 | 1 | 8 |
| 82 | 2 | 16 |
| 83 | 5 | 0.01 |
| 84 | 2 | 4 |
| 85 | 1 | 10 |
| 86 | 1 | 36 |

XIII. Note that node 1 frequently appears several times in succession, so that there is immediate feedback at node 1. Also note that the service times differ at different visits to the same node.

The customer classes arrive according to independent Poisson processes. In the case we consider the arrival rates of Classes 1 and 2 are 0.00015278 and 0.00030555, respectively.

The QNA, M/G/1, and M/M/1 approximations are compared with simulation in Table XIV. The simulation values are the average of three separate runs. The values from these separate runs are displayed to give an idea of the statistical reliability. The congestion measures compared are the expected waiting times at the nodes and the expected total waiting time (excluding service time) on three route segments. The first segment is the first 25 nodes of the second route; the second segment is the first 21 nodes on the first route; and the third segment is eight nodes from node 23 to node 30 on the first route. In Table XIV the waiting times at the nodes are measured in milliseconds while the waiting times on the route segments are measured in seconds.

From Table XIV, it is apparent that QNA with immediate-feedback elimination performs reasonably well, significantly better than the M/M/1 and M/G/1 approximations. Since the approximating variability parameters of the arrival processes are very close to 1, QNA without immediate-feedback elimination is very similar to the M/G/1 approximation. Hence, again we see that eliminating immediate feed-

Table XIII—The data for the second route in the model of Section VII

| Number | Node | Mean Service Time | Number | Node | Mean Service Time | Number | Node | Mean Service Time |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 21 | 4 | 30 | 41 | 1 | 46 |
| 2 | 1 | 10 | 22 | 1 | 36 | 42 | 1 | 14.5 |
| 3 | 1 | 66 | 23 | 1 | 42 | 43 | 3 | 400 |
| 4 | 1 | 60 | 24 | 1 | 14.5 | 44 | 1 | 8 |
| 5 | 1 | 106 | 25 | 3 | 400 | 45 | 1 | 12 |
| 6 | 1 | 65 | 26 | 5 | 0.01 | 46 | 1 | 30 |
| 7 | 4 | 30 | 27 | 3 | 350 | 47 | 1 | 58 |
| 8 | 1 | 16 | 28 | 1 | 10 | 48 | 1 | 8 |
| 9 | 4 | 30 | 29 | 1 | 16 | 49 | 2 | 16 |
| 10 | 1 | 36 | 30 | 1 | 20.5 | 50 | 5 | 0.01 |
| 11 | 1 | 12 | 31 | 1 | 84 | 51 | 2 | 4 |
| 12 | 1 | 65 | 32 | 1 | 45 | 52 | 1 | 10 |
| 13 | 4 | 30 | 33 | 4 | 30 | 53 | 1 | 36 |
| 14 | 1 | 16 | 34 | 1 | 16 | | | |
| 15 | 4 | 30 | 35 | 4 | 30 | | | |
| 16 | 1 | 36 | 36 | 1 | 16 | | | |
| 17 | 1 | 62 | 37 | 4 | 30 | | | |
| 18 | 1 | 40 | 38 | 1 | 16 | | | |
| 19 | 4 | 30 | 39 | 4 | 30 | | | |
| 20 | 1 | 16 | 40 | 1 | 36 | | | |

Table XIV—A comparison of approximations and simulation of the expected waiting times for the model of Section VII

| Method | Expected Waiting Time at the Nodes | | | | Total Expected Waiting Time on a Route Segment | | |
|---|---|---|---|---|---|---|---|
| | Node 1 | Node 2 | Node 3 | Node 4 | 1 (25 nodes) | 2 (21 nodes) | 3 (8 nodes) |
| Simulation runs 1 | 51.9 | 0.058 | 224.1 | 2.11 | 1.25 | 1.12 | 0.66 |
| 2 | 58.5 | 0.058 | 245.6 | 2.27 | 1.37 | 1.27 | 0.77 |
| 3 | 57.6 | 0.055 | 236.2 | 2.17 | 1.35 | 1.23 | 0.74 |
| Simulation average | 56.0 | 0.057 | 235.3 | 2.18 | 1.32 | 1.21 | 0.72 |
| M/M/1 | 59.2 (+5.7) | 0.09 | 402.5 (+71.1) | 6.53 (+200.0) | 1.45 (+9.8) | 1.32 (+9.1) | 1.16 (+61.1) |
| M/G/1 | 43.5 (−22.3) | 0.07 | 245.5 (+4.2) | 3.26 (+49.5) | 1.01 (−23.5) | 0.91 (−24.8) | 0.75 (+4.2) |
| QNA (eliminating feedback) | 50.2 (−10.3) | 0.07 | 244.1 (+3.7) | 2.81 (+28.9) | 1.11 (−15.9) | 1.01 (−16.5) | 0.79 (+9.7) |

| | Additional Information About the Network | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| Mean service time | 33.1 | 8.0 | 350.0 | 30.0 | 1.28 | 1.31 | 0.75 |
| Traffic intensity | 0.64 | 0.11 | 0.54 | 0.18 | --- | --- | --- |
| $c^2$ of the service time from QNA | 0.47 | 0.50 | 0.22 | 0.00 | --- | --- | --- |
| $c^2$ of the arrival process from QNA | 0.92 | 1.00 | 0.99 | 0.90 | --- | --- | --- |

Notes: 1. The relative percent errors appear below the approximation in parentheses.
2. The value $c_{a1}^2 = 0.47$ at node 1 is before adjustment for feedback; after adjustment it is 0.79.
3. The units of measurement are milliseconds for the nodes and seconds for the route segments.

back helps. The M/M/1 approximations at nodes 3 and 4 evidently are too large because the service times are nearly constant. However, at node 1 the M/M/1 approximation does pretty well, apparently because two different errors cancel. (We have not displayed the relative percentage errors at node 2 since it seems of little consequence because of the low traffic intensity.)

It is interesting to know how the system would perform if the external arrival processes are not Poisson and if the service times at the nodes on the routes are not deterministic. With QNA we can easily perform such sensitivity analyses. We can simply change the variability parameters of the external arrival processes and the service times on the routes. The results of such a study are given in Tables XV and XVI. Table XV gives the approximating variability parameter of the

arrival process at each node as a function of the external arrival process $c^2$ and the service time $c^2$. It is assumed that both external arrival processes have the same $c^2$ and that all service times on both routes have the same $c^2$. From Table XV, it is evident that the variability of the external arrival process hardly has any effect. Thus, QNA predicts that this model, and perhaps the system itself, will be robust to changes in the variability of the arriving traffic. The variability from outside is evidently dissipated on the long routes through the network.

Table XVI describes the impact of changing the service-time variability at the nodes on the routes. The service-time variability at the nodes would increase significantly and, thus, QNA predicts that the expected waiting times would also increase significantly. Simulations to test these predictions are planned.

Table XV—The approximate variability parameter of the arrival process at each node determined by the QNA, as a function of the given variability parameters: The model of Section VII

| External Arrival Process $c^2$ | Node | Service Time $c^2$ at Each Node on the Route | | | | |
|---|---|---|---|---|---|---|
| | | 0.0 | 0.2 | 0.5 | 1.0 | 2.0 |
| $c^2 = 1$ | 1 | 0.9155 | 0.9506 | 1.0034 | 1.0912 | 1.2669 |
| | 2 | 0.9995 | 0.9997 | 1.0000 | 1.0001 | 1.0016 |
| | 3 | 0.9937 | 0.9966 | 1.0010 | 1.0083 | 1.0230 |
| | 4 | 0.8950 | 0.9480 | 1.0274 | 1.1598 | 1.4245 |
| $c^2 = 2$ | 1 | 0.9162 | 0.9513 | 1.0040 | 1.0918 | 1.2676 |
| | 2 | 1.0086 | 1.0088 | 1.0092 | 1.0097 | 1.0107 |
| | 3 | 0.9938 | 0.9967 | 1.0011 | 1.0084 | 1.0231 |
| | 4 | 0.8953 | 0.9483 | 1.0277 | 1.1601 | 1.4249 |
| $c^2 = 4$ | 1 | 0.9175 | 0.9527 | 1.0053 | 1.0932 | 1.2689 |
| | 2 | 1.0268 | 1.0271 | 1.0274 | 1.0279 | 1.0290 |
| | 3 | 0.9939 | 0.9968 | 1.0012 | 1.0085 | 1.0231 |
| | 4 | 0.8959 | 0.9488 | 1.0283 | 1.1607 | 1.4254 |

Table XVI—The approximate service-time variability parameter $c_{si}^2$ and mean delay $EW_j$ at node j determined by the QNA, as a function of the variability of each service time on the route: The model of Section VII

| Node Characteristic | Node | Service Time $c^2$ at Each Node on the Route | | | | |
|---|---|---|---|---|---|---|
| | | 0.0 | 0.2 | 0.5 | 1.0 | 2.0 |
| $c_{sj}^2$ | 1 | 0.787 | 0.905 | 1.082 | 1.377 | 1.968 |
| | 2 | 0.500 | 0.800 | 1.250 | 2.000 | 3.500 |
| | 3 | 0.220 | 0.464 | 0.830 | 1.441 | 2.661 |
| | 4 | 0.000 | 0.200 | 0.500 | 1.000 | 2.000 |
| $EW_j$ | 1 | 50.2 | 54.9 | 61.6 | 72.6 | 94.6 |
| | 2 | 0.07 | 0.08 | 0.10 | 0.13 | 0.20 |
| | 3 | 244.1 | 293.3 | 367.7 | 491.4 | 739.0 |
| | 4 | 2.81 | 3.72 | 4.95 | 6.87 | 10.78 |

## VIII. A PACKET-SWITCHED COMMUNICATION-NETWORK MODEL

In this section we consider a model of a packet-switched communication network analyzed in Section 4.3.1 of Gelenbe and Mitrani.[13] The basic model has 5 switching nodes and 12 one-way data links, as depicted in Fig. 4. However, in this model each data link is a server and the packets waiting for transmission on the link form the queue. Packets are assumed to arrive at the switching nodes according to independent Poisson processes. Each packet arriving from outside at node i has final destination j with probability $d_{ij}$. Each packet with destination j goes next to node $r_{ij}$ from node i. Hence, there is a fixed route for each origin-destination pair.

We analyze this network using the input by classes and routes in Section 2.3 of Ref. 1. However, unlike Section VII, here the service-time parameters are associated with the nodes rather than the routes (which is an input option in QNA). The network of queues has 12 nodes with one server at each node and 20 routes. As specified by Gelenbe and Mitrani, the service rate at nodes 1, 2, 7, 8, 11, and 12 is 4.8 (in thousands of bits per second) and the service rate at the other nodes is 48. Since packet lengths are assumed constant, $c_{ij}^2 = 0$ for all j.

For this example the matrices $D \equiv (d_{ij})$ of destination probabilities and $R = (r_{ij})$ of next-node routes are:

$$D = \begin{bmatrix} 0.00 & 0.10 & 0.20 & 0.10 & 0.60 \\ 0.40 & 0.00 & 0.40 & 0.15 & 0.05 \\ 0.10 & 0.20 & 0.00 & 0.60 & 0.10 \\ 0.30 & 0.30 & 0.30 & 0.00 & 0.10 \\ 0.10 & 0.25 & 0.30 & 0.35 & 0.00 \end{bmatrix} \quad R = \begin{bmatrix} 0 & 3 & 3 & 3 & 2 \\ 4 & 0 & 5 & 5 & 4 \\ 6 & 6 & 0 & 9 & 8 \\ 10 & 10 & 10 & 0 & 12 \\ 1 & 1 & 7 & 11 & 0 \end{bmatrix}.$$

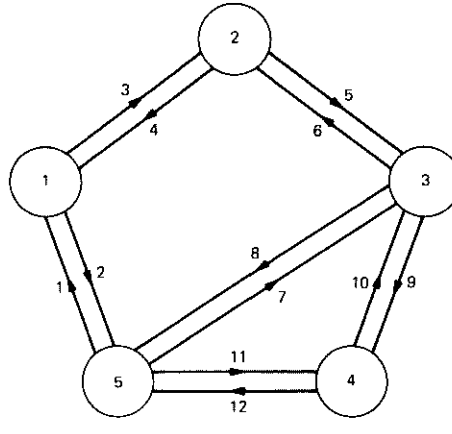



Fig. 4—Gelenbe and Mitrani's model of a packet-switched network: The 12 links are the nodes in the network of queues.

The input data for the routes are given in Table XVII. These data are obtained from the matrices $D$ and $R$ plus the external arrival rates of 6.00, 8.25, 7.50, 6.75, and 1.50 at the five switching nodes (p. 141 of Gelenbe and Mitrani[13]).

The results are compared with Gelenbe and Mitrani's approximation and simulation in Table XVIII. Since only 6000 packets reached their destination in the simulation, the statistical reliability of the simulation estimates cannot be very good, cf. Section III. Our analysis is revealing. First, Gelenbe and Mitrani describe their results as average buffer queue lengths, which might be thought to exclude the customer (packet) being served (transmitted). However, the results obviously include the customer in service. Second, the two M/M/1 approximations should agree, but they do not. Evidently, the arrival rates at switches 1 and 2 were not actually as cited in the text.[13] Hence, the numbers for the one heavily loaded link, link 2, cannot be meaningfully compared.

It is useful to consider the expected number waiting excluding the customer in service. This is easily obtained because the probability that the server is busy is exactly the traffic intensity, $\rho$ (see Section 11.3 of Heyman and Sobel[14]). We thus obtain estimates of the expected number waiting by subtracting $\rho$ from the numbers displayed in Table

Table XVII—The input data by routes for Gelenbe and Mitrani's model of a packet-switched communication network

| Route Number | Origin-Destination Pair | External Arrival Process Parameters $\hat{\lambda}_k$ | $c_k^2$ | Number of Nodes on the Route | Node Sequence |
|---|---|---|---|---|---|
| 1 | 1,2 | 0.60 | 1 | 1 | 3 |
| 2 | 1,3 | 1.20 | 1 | 2 | 3,5 |
| 3 | 1,4 | 0.60 | 1 | 3 | 3,5,9 |
| 4 | 1,5 | 3.60 | 1 | 1 | 2 |
| 5 | 2,1 | 3.30 | 1 | 1 | 4 |
| 6 | 2,3 | 3.30 | 1 | 1 | 5 |
| 7 | 2,4 | 1.24 | 1 | 2 | 5,9 |
| 8 | 2,5 | 0.41 | 1 | 2 | 4,2 |
| 9 | 3,1 | 0.75 | 1 | 2 | 6,4 |
| 10 | 3,2 | 1.50 | 1 | 1 | 6 |
| 11 | 3,4 | 4.50 | 1 | 1 | 9 |
| 12 | 3,5 | 0.75 | 1 | 1 | 8 |
| 13 | 4,1 | 2.03 | 1 | 3 | 10,6,4 |
| 14 | 4,2 | 2.03 | 1 | 2 | 10,6 |
| 15 | 4,3 | 2.03 | 1 | 1 | 10 |
| 16 | 4,5 | 0.68 | 1 | 1 | 12 |
| 17 | 5,1 | 0.15 | 1 | 1 | 1 |
| 18 | 5,2 | 0.38 | 1 | 2 | 1,3 |
| 19 | 5,3 | 0.45 | 1 | 1 | 7 |
| 20 | 5,4 | 0.53 | 1 | 1 | 11 |

XVIII. When this is done, some of the simulation estimates become negative, demonstrating that the input parameters are incorrect or the statistical reliability of the simulation is not very good. In this case, probably both problems exist.

We also observe that the traffic intensities at all link queues but the second are very small, so the numbers displayed in Table XVIII are mostly estimates of the traffic intensities themselves. Moreover, because the traffic intensities are small, the variability parameter of the departure process produced by QNA will be very close to the variability parameter of the arrival process (see Section 4.5 of Ref. 1). This would not be true for the second link with traffic intensity 0.835, but note that all departures from the second link leave the system. Since the external arrival processes are all Poisson, QNA should and does perform virtually the same as an M/G/1 approximation. In fact, since the service times are all constant ($c_{sj}^2 = 0$), the approximation reduces to the M/D/1 system. Moreover, we predict that a proper simulation of this model with the specified parameters will yield values very close to the M/D/1 approximation.

We also display in Table XIX the point-to-point (origin-destination) average total service times, delays, and sojourn times (service times plus delays) produced by QNA. No simulation values were available for comparison, however. The output is useful to indicate unacceptably high or low values. It is also useful to determine the separate contributions of service times and delays to sojourn times. Of course, in

Table XVIII—A comparison of approximations and simulation: The expected number waiting and being served on each link in the Gelenbe-and-Mitrani model of a packet-switched communication network depicted in Fig. 4

| | | | Approximation Methods | | | |
|---|---|---|---|---|---|---|
| | | | M/M/1 | | | |
| Link No. | Traffic Intensity, $\rho_j$ | Simulation Value | Gelenbe and Mitrani | via QNA | Gelenbe-Pujolle | QNA |
| 1 | 0.110 | 0.117 | 0.123 | 0.124 | 0.116 | 0.117 |
| 2 | 0.835 | 1.920 | 3.000 | 5.076 | 1.875 | 2.955 |
| 3 | 0.058 | 0.132 | 0.139 | 0.061 | 0.131 | 0.060 |
| 4 | 0.135 | 0.163 | 0.170 | 0.156 | 0.157 | 0.146 |
| 5 | 0.132 | 0.105 | 0.127 | 0.152 | 0.125 | 0.142 |
| 6 | 0.131 | 0.173 | 0.157 | 0.151 | 0.146 | 0.141 |
| 7 | 0.094 | 0.087 | 0.104 | 0.103 | 0.099 | 0.099 |
| 8 | 0.156 | 0.208 | 0.185 | 0.185 | 0.171 | 0.171 |
| 9 | 0.132 | 0.155 | 0.162 | 0.152 | 0.147 | 0.142 |
| 10 | 0.127 | 0.129 | 0.145 | 0.145 | 0.136 | 0.136 |
| 11 | 0.110 | 0.106 | 0.123 | 0.124 | 0.116 | 0.117 |
| 12 | 0.142 | 0.154 | 0.164 | 0.165 | 0.152 | 0.153 |

Table XIX—Average point-to-point service times, delays, and sojourn times for the Gelenbe-and-Mitrani model of a packet-switched communication network

| Route No. | Mean Total Service Time on Route | Poisson Arrivals $c^2 = 1.0$ | | Bursty Arrivals $c^2 = 4.0$ | |
|---|---|---|---|---|---|
| | | Mean Total Delay on Route | Mean Total Sojourn Time on Route | Mean Total Delay on Route | Mean Total Sojourn Time on Route |
| 1 | 0.021 | 0.001 | 0.021 | 0.001 | 0.022 |
| 2 | 0.042 | 0.002 | 0.044 | 0.003 | 0.044 |
| 3 | 0.063 | 0.004 | 0.066 | 0.005 | 0.068 |
| 4 | 0.208* | 0.529* | 0.737* | 1.920* | 2.128* |
| 5 | 0.021 | 0.002 | 0.022 | 0.002 | 0.023 |
| 6 | 0.021 | 0.002 | 0.022 | 0.002 | 0.023 |
| 7 | 0.042 | 0.003 | 0.045 | 0.005 | 0.046 |
| 8 | 0.229* | 0.530* | 0.759* | 1.922* | 2.151* |
| 9 | 0.042 | 0.003 | 0.045 | 0.004 | 0.046 |
| 10 | 0.021 | 0.002 | 0.022 | 0.002 | 0.046 |
| 11 | 0.021 | 0.002 | 0.022 | 0.003 | 0.024 |
| 12 | 0.208* | 0.019 | 0.228* | 0.077 | 0.285* |
| 13 | 0.063 | 0.005 | 0.067 | 0.006 | 0.068 |
| 14 | 0.042 | 0.003 | 0.045 | 0.004 | 0.046 |
| 15 | 0.021 | 0.002 | 0.022 | 0.002 | 0.023 |
| 16 | 0.208* | 0.017 | 0.226* | 0.069 | 0.277* |
| 17 | 0.208* | 0.013 | 0.221* | 0.025 | 0.233* |
| 18 | 0.229* | 0.014 | 0.243* | 0.026 | 0.255* |
| 19 | 0.208* | 0.011 | 0.219* | 0.043 | 0.251* |
| 20 | 0.208* | 0.013 | 0.221* | 0.052 | 0.260* |

Note: The larger values are marked with an asterisk.

Table XIX delays play a significant role only for routes using the second link.

We conclude by remarking that the assumption of Poisson arrivals for packets at each switch made by Gelenbe and Mitrani[13] often is not realistic. Often messages containing many packets arrive according to a Poisson process, but the packets arrive in a much more bursty manner. Hence, it is appropriate to use QNA with arrival-process variability parameters much larger than 1. The last two columns of Table XIX give the mean delays and sojourn times when the variability parameters of the external arrival processes are changed from $c^2 = 1.0$ to $c^2 = 4.0$. When this is done here, the large delays on routes 4 and 8 increase significantly. To a large extent, QNA was motivated by the need to be able to systematically study the effect of such variability.

## IX. ACKNOWLEDGMENTS

I am grateful to Anne Seery for using QNA to generate much of the data here, as well as for writing the QNA program. I am grateful to E. B. Zucker for the simulation data in Section VII.

## REFERENCES


1. W. Whitt, "The Queueing Network Analyzer," B.S.T.J., this issue.
2. W. Whitt, "On Approximations for Queues, I: Extremal Distributions," B.S.T.J., *63*, No. 1, Part 1 (January 1984).
3. J. G. Klincewicz and W. Whitt, "On Approximations for Queues, II: Shape Constraints," B.S.T.J., *63*, No. 1, Part 1 (January 1984).
4. W. Whitt, "On Approximations for Queues, III: Mixtures of Exponential Distributions," B.S.T.J., *63*, No. 1, Part 1 (January 1984).
5. W. Whitt, "The Marshall/Marshall and Stoyan Bounds for IMRL/G/1 Queues are Tight," Oper. Res. Letters, *1*, No. 6 (December 1982), pp. 209–13.
6. W. Whitt, "Refining Diffusion Approximations for Queues," Oper. Res. Letters, *1*, No. 5 (November 1982), pp. 165–9.
7. W. Kraemer and M. Langenbach-Belz, "Approximate Formulae for the Delay in the Queueing System GI/G/1," *Congressbook*, Eighth International Teletraffic Congress, Melbourne, Australia, 1976, pp. 235, 1–8.
8. S. L. Albin, *Approximating Queues with Superposition Arrival Processes*, Ph.D dissertation, Department of Industrial Engineering and Operations Research, Columbia University, 1981.
9. S. L. Albin, "Approximating a Point Process by a Renewal Process, II: Superposition Arrival Processes to Queues," Department of Industrial Engineering Rutgers University, 1982.
10. S. L. Albin, "On Poisson Approximations for Superposition Arrival Processes in Queues," Management Sci., *28*, No. 2 (February 1982), pp. 126–37.
11. J. R. Fraker, *Approximate Techniques for the Analysis of Tandem Queueing Systems*, Ph.D. dissertation, Department of Industrial Engineering, Clemson University, 1971.
12. P. J. Kuehn, "Approximate Analysis of General Queueing Networks by Decomposition," IEEE Trans. Commun., *COM-27*, No. 1 (January 1979), pp. 113–26.
13. E. Gelenbe and I. Mitrani, *Analysis and Synthesis of Computer Systems*, New York: Academic Press, 1980.
14. D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research*, Volume I, New York: McGraw-Hill, 1982.
15. W. Whitt, "Approximating a Point Process by a Renewal Process, I: Two Basic Methods," Oper. Res., *30*, No. 1 (January–February 1982), pp. 125–47.
16. G. Marsaglia, K. Ananthanarayanan, and N. Paul, "Random Number Generator Package–Super Duper," School of Computer Science, McGill University, 1973.
17. W. Whitt, "Queues with Superposition Arrival Processes in Heavy Traffic," unpublished work, 1982.
18. W. Whitt, "Approximations for Departure Processes and Queues in Series," Navy Res. Log Qtrly., to be published.


## AUTHOR


**Ward Whitt,** A. B. (Mathematics), 1964, Dartmouth College; Ph.D. (Operations Research), 1968, Cornell University; Stanford University, 1968–1969; Yale University, 1969–1977; Bell Laboratories, 1977—. At Yale University, from 1973–1977, Mr. Whitt was Associate Professor in the departments of Administrative Sciences and Statistics. At Bell Laboratories he is in the Operations Research Department in the Network Analysis Center.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Berreman D. W., **Numerical Modeling of Twisted Nematic Devices.** Phi T Roy A 309(1507):203–216, 1983.

Fishburn P. C., Pollak H. O., **Fixed-Route Cost Allocation.** Am Math Mo 90(6):366–378, 1983.

Stein A. H., An C. M., **A Mean-Value Theorem for Zeta-Functions Associated With Positive Definite Integral Forms.** Mich Math J 30(1):3–8, 1983.

## ENGINEERING

Amitay N., Saleh A. A. M., **Broad-Band Wide-Angle Quasi-Optical Polarization Rotators.** IEEE Antenn 31(1):73–76, 1983.

Bartoli P. D., **CCITT Message Handling Facilities.** Aust Tele R 16(3):53–62, 1982.

Borsuk J. A., **Light-Intensity Profiles of Surface-Emitting InGaAsP LEDs—Impact on Coupling to Optical Fibers.** IEEE Device 30(4):296–303, 1983.

Campbell J. C., Qua G. J., Dentai A. G., **Optical Comparator—A New Application for Avalanche Phototransistors.** IEEE Device 30(4):408–411, 1983.

Capasso F., Tsang W. T., Williams G. F., **Staircase Solid-State Photomultipliers and Avalanche Photo-Diodes With Enhanced Ionization Rates Ratio.** IEEE Device 30(4):381–390, 1983.

Chin A. K., Zipfel C. L., Ermanis F., Marchut L., Camlibel I., Di Giuseppe M. A., Chin B. H., **The Migration of Gold From the P-Contact as a Source of Dark Spot Defects in InP/InGaAsP LEDs.** IEEE Device 30(4):304–310, 1983.

Dutta N. K., **Current Injection in Multiquantum Well Lasers (Letter).** IEEE J Q El 19(5):794–797, 1983.

Dutta N. K., Nelson R. J., Wright P. D., Besomi P., Wilson R. B., **Optical-Properties of a 1.3-μm InGaAsP Superluminescent Diode.** IEEE Device 30(4):360–363, 1983.

Heller A., **High-Efficiency of Photo-Electrochemical Cells—Reply (Letter).** Chem Eng N 61(22):4, 1983.

Johannes V. I., **Performance Parameters for Digital-Communications (Letter).** P IEEE 71(4):539, 1983.

Lee T. P., **Special Issue on Light-Emitting Diodes and Long-Wavelength Photodetectors—Foreword (Editorial).** IEEE Device 30(4):257–258, 1983.

Levy U., Gordon E. I., Logan R. A., **Laser Cathode-Ray Tube With a Semiconductor Double-Heterostructure Screen.** IEEE Elec D 4(5):155–156, 1983.

Ma K. T., Kusic G. L., **Tie Line Contingency Studies Based Upon Partial Information.** IEEE Power 102(6):1838–1842, 1983.

Saul R. H., **Recent Advances in the Performance and Reliability of InGaAsP LEDs for Lightwave Communication-Systems.** IEEE Device 30(4):285–295, 1983.

Zipfel C. L., Chin A. K., Di Giuseppe M. A., **Reliability of InGaAsP Light-Emitting Diodes at High-Current Density.** IEEE Device 30(4):310–316, 1983.

## PHYSICAL SCIENCES

Acuna M. H. et al., **Physics of the Jovian and Saturnian Magnetospheres—Highlights of a Conference Held at the Applied Physics Laboratory, The Johns-Hopkins-University, October 22–24, 1981 (Review).** Space Sci R 35(3):269–292, 1983.

Allara D. L., Hebard A. F., Padden F. J., Nuzzo R. G., Falcone D. R., **Chemically-Induced Enhancement of Nucleation in Noble-Metal Deposition.** J Vac Sci A 1(2):376–382, 1983.

Bean J. C., **Recent Developments in Silicon Molecular-Beam Epitaxy.** J Vac Sci A 1(2):540–545, 1983.

Chabal Y. J., **Hydrogen Vibration on Si(111)7X7—Evidence for a Unique Chemisorption Site.** Phys Rev L 50(23):1850–1853, 1983.

Chang J. J., Julesz B., **Displacement Limits, Directional Anisotropy and Direction Versus Form Discrimination in Random-Dot Cinematograms.** Vision Res 23(6):639+, 1983.

Donnelly V. M., Flamm D. L., Ibbotson D. E., **Plasma-Etching of III-V-Compound Semiconductors.** J Vac Sci A 1(2):626–628, 1983.

Dubois L. H., Rowe J. E., **Surface Phonons on Clean-Covered and Adsorbate-Covered Nickel Disilicide ($NiSi_2$) Thin-Films.** J Vac Sci A 1(2):1232–1235, 1983.

Evans D., Celli V., Benedek G., Toennies J. P., Doak R. B., **Resonance-Enhanced Atom Scattering From Surface Phonons.** Phys Rev L 50(23):1854–1857, 1983.

Focht M. W., Schwartz B., **High-Resistivity in P-Type InP by Deuteron Bombardment.** Appl Phys L 42(11):970–972, 1983.

Hannay N. B., **The Information Society—Perkin Medal Address (Editorial).** Chem Ind L 1983(11):433–437, 1983.

Hart A. C., Krause J. T., **Coating Technique for High-Strength Lightguide Fusion Splices.** Appl Optics 22(11):1731–1733, 1983.

Jackel J. L., Hackwood S., Veselka J. J., Beni G., **Electrowetting Switch for Multimode Optical Fibers.** Appl Optics 22(11):1765–1770, 1983.

Lyon S. A., Chen Y. H., Lin J. F., Worlock J. M., **Optical-Transmission at 3.39 $\mu$m During Pulsed Laser Annealing of Silicon.** Appl Phys L 42(11):978–980, 1983.

Miller D. A. B., Chemla D. S., Eilenberger D. J., Smith P. W., Gossard A. C., Wiegmann W., **Degenerate 4-Wave Mixing in Room-Temperature GaAs/GaAlAs Multiple Quantum Well Structures.** Appl Phys L 42(11):925–927, 1983.

Moncton D. E., Brown G. S., **High-Resolution X-Ray-Scattering.** Nucl Instru 208(1–3):579–586, 1983.

Osinski J. S., Manzione L. T., **Characterization and Moldability Analysis of Epoxy Reaction Injection-Molding Resins (Review).** ACS Symp S 1983(221):263–282, 1983.

Ota Y., **Defect Evaluation of Si MBE Film.** J Cryst Gr 61(3):439–448, 1983.

Patel J. R., Golovchenko J. A., **X-Ray-Standing-Wave Atom Location in Heteropolar Crystals and the Problem of Extinction.** Phys Rev L 50(23):1858–1861, 1983.

Reichmanis E., Wilkins C. W., Price D. A., Chandross E. A., **The Effect of Substituents on the Photosensitivity of 2-Nitrobenzyl Ester Deep UV Resists.** J Elchem So 130(6):1433–1437, 1983.

Rousseau D. L., Ondrias M. R., **Resonance Raman-Scattering Studies of the Quaternary Structure Transition in Hemoglobin (Review).** Ann R Bioph 12:357–380, 1983.

Shapira Y., Brillson L. J., Heller A., **Investigation of InP Surface and Metal Interfaces by Surface Photo-Voltage and Auger-Electron Spectroscopies.** J Vac Sci A 1(2):766–770, 1983.

Sharma S. P., Sproles E. S., **Reaction of Palladium With Chlorine and Hydrogen-Chloride.** J Elchem So 130(6):1242–1247, 1983.

Temkin H., Logan R. A., Van Der Ziel J. P., **Integrated Arrays of 1.3-$\mu$m Buried-Crescent Lasers.** Appl Phys L 42(11):934–936, 1983.

Thiel F. A., **The Phase-Relations in the Cu-In-S-System and the Growth of $CuInS_2$ Crystals From the Melt—Reply (Discussion).** J Elchem So 130(6):1445, 1983.

Trapp K. D. C., Ermanis F., **Origin and Elimination of Crescent-Shaped Growth Defects in LPE Layers of InGaAs/InP Alloys.** J Elchem So 130(6):1381–1383, 1983.

Woodruff D. P., Johnson P. D., Smith N. V., **Inverse Photoemission.** J Vac Sci A 1(2):1104–1110, 1983.

Yao S. C., Chang Y., **Pool Boiling Heat-Transfer in a Confined Space.** Int J Heat 26(6):841–848, 1983.

## SOCIAL AND LIFE SCIENCES

Anderson P. W., **Suggested Model for Prebiotic Evolution—The Use of Chaos.** P Nas Biol 80(11):3386–3390, 1983.

Weiss A., **A Sorting-cum-Learning Model of Education.** J Polit Ec 91(3):420–442, 1983.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 62 December 1983 Number 10, Part 1

# Time-Compression Multiplexing (TCM) of Three Broadcast-Quality TV Signals on a Satellite Transponder

By K. Y. ENG,* B. G. HASKELL,* and R. L. SCHMIDT*

(Manuscript received June 15, 1983)

We describe how Time-Compression Multiplexing (TCM) might enable the transmission of three National Television System Committee (NTSC) color TV signals through a satellite transponder of 36-MHz bandwidth. The input TV signals are processed such that three fields from each TV source are compressed into an ordinary field period. This is accomplished by sending one field as is but time compressed; the other two fields are sent as differential signals, also time compressed such that all three fit into a single field period. The resultant compressed waveforms are then time multiplexed between the three sources and have a combined baseband bandwidth of 7.52 MHz for an optimal case, or 8.4 MHz for a practical version. In either case, both the transmitter-multiplexer and the receiver-demultiplexer require only three field memories for (digital) signal processing. Performance is expected to be of network broadcast quality (i.e., weighted signal-to-noise ratio, s/n $\geq$ 56 dB) for the optimal case of 7.52-MHz baseband if 12-meter receive earth stations

*Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

are employed in a system such as COMSTAR. The practical version, on the other hand, would yield an s/n ≈ 54 dB.

## I. INTRODUCTION

The problem of transmitting two or more high-quality TV signals through a satellite transponder of 36 MHz continues to be a challenge in optimizing the use of available transponders in current as well as near-future satellites. It was recently proposed[1] that by combining the concepts of Time-Compression Multiplexing (TCM)[2,3] and differential signals,[4] two or more National Television System Committee (NTSC) TV signals can be time multiplexed with bandwidth reduction for transmission with a single FM carrier in a satellite channel. This avoids crosstalk between the pictures. In fact, straightforward TCM alone would permit the transmission of two TVs in a transponder with performance close to broadcast quality [i.e., peak-to-peak video signal to weighted root-mean-square (rms) noise ratio, s/n ≥ 56 dB] if 12-meter receive earth stations were used in a satellite system such as COMSTAR. The additional application of time-companded (time-compressed or expanded) differential signals reduces the TCM signal bandwidth and thus can enhance the transmitted picture quality or enable the inclusion of a third TV signal. However, the implementation of such a system as described in Ref. 1 involves converting the input TV scan pattern from interlacing to sequential. This would mean considerable memory needed, particularly in the case of three TVs per transponder. Here, we describe an implementation that offers significant saving in memory, considerable relaxation in timing requirements, and easy adaptation to existing hardware. The technique essentially uses three field memories time shared between the three simultaneous, but *synchronized*, input TV signals to produce differential signals in a proper format for TCM. The receiver, on the other hand, also requires three field memories to reconstruct all three TV signals. Practically all the signal processing could be implemented digitally.

We will describe the details of the present system in the next section. The performance of this could be of broadcast quality if 12-meter receive earth stations were used. Finally, we will discuss the inclusion of audio, up-link synchronization for transmissions from separate earth stations and possible extensions to non-NTSC TV signals.

## II. SYSTEM DESCRIPTION

Figure 1 shows the block diagram of a transmitting earth station with three *synchronized* NTSC TV signals that are combined for transmission by a single FM carrier. The use of frame synchronizers

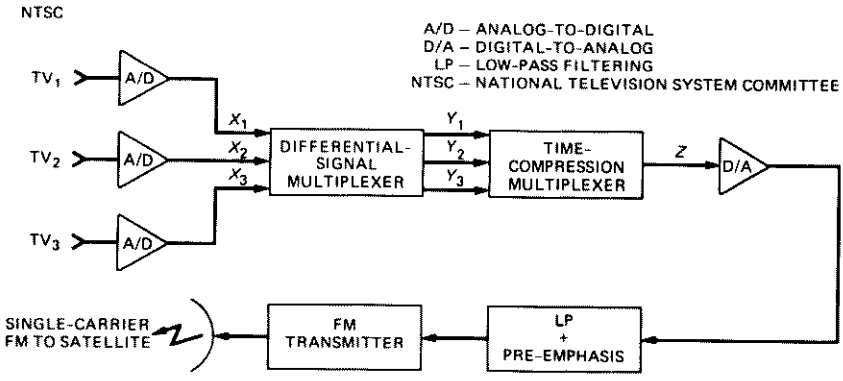



Fig. 1—Transmit earth station for the three TVs per transponder.

is therefore implicit if the three TV sources are not synchronized (the case of noncolocated TV sources is discussed later in Section IV). The TV inputs are first digitized individually. The digitized TVs, denoted by $X_i$ ($i = 1$ to 3), are processed by the differential-signal multiplexer, where various differential signals are formed and multiplexed in its three digital outputs $Y_i$ ($i = 1$ to 3). These signals ($Y_i$) are then passed through the time-compression multiplexer, which combines them into a single digital stream, $Z$. The conventional operations of digital-to-analog conversion, low-pass filtering, and preemphasis are performed before transmission to the satellite with a single FM carrier. We will describe the differential-signal multiplexer and the time-compression multiplexer in detail in the following sections.

### *2.1 The differential-signal multiplexer*

We could use three types of differential signals: line differentials, field differentials, and frame differentials.[4,5] Each type in turn can be defined in many ways. They have all been described in the cited references, and only a brief summary is provided here for the purpose of subsequent discussions.

Line differentials can be defined as a difference signal between two successive scan lines in the same field. In their digital implementation, this would mean a difference between more or less vertically adjacent picture elements (pels) from two successive lines in the *same* field, and they are chosen such that their amplitude is much smaller, on the average, than the original signal. But most importantly, the difference signal can be band limited to ≈3 MHz without degrading picture quality. Field differentials are defined essentially in the same way as line differentials except that the difference signal is derived from pels in adjacent scan lines in two successive fields. The bandwidth of field differentials can be further limited to ≈2 MHz without affecting

picture quality. These results were verified and used in a previous experiment.[5]

Frame differentials are merely an extension of the above by using pels from two temporally adjacent (or spatially coincident) lines from two successive frames. They have not been studied so far, either by computer simulation or hardware implementation. Thus, we can only speculate as to their performance. Their amplitudes may be larger than field difference amplitudes for pictures containing movement, whereas for pictures containing no movement they should be smaller. The bandwidth required for frame differentials should be comparable to or smaller than that needed for field differentials. In this regard, much depends on the relative visibility in the picture of distortions occurring at the field rate and the frame rate in detailed or moving areas of the picture. In the following discussion, we will use the field and frame differentials; the use of line differentials will only be a possible, though unlikely, extension of the system.

Our attention returns now to the differential-signal multiplexer, an illustrative implementation of which is shown in Fig. 2. The following explanation will show that the field- and frame-differential generators in this figure could just as well be replaced by two field-differential generators with some connections slightly modified. The key characteristic to note in Fig. 2 is that only three field memories are needed to produce all the differential signals required for the three input TVs.

The three input switches, $S_1$, $S_2$ and $S_3$, move in synchronism from the top position to the middle, to the bottom, and back to the top, etc.

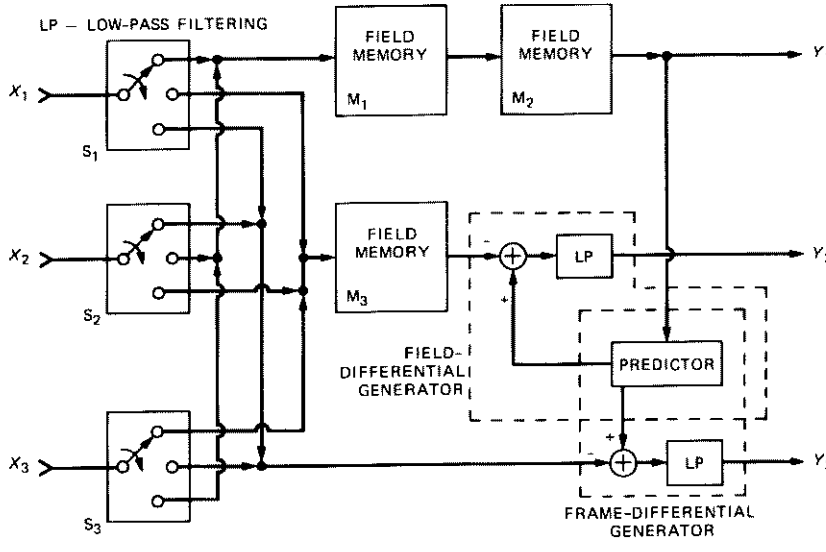



Fig. 2—The differential-signal multiplexer.

They all change position simultaneously sometime during the vertical blanking interval in such a way that complete fields of the input video are routed to either the top, middle, or bottom path.

To demonstrate how this works, we consider Fig. 3, where all the waveforms are digital. In the top of the figure, the three synchronized input TV waveforms are shown with $T$ being a field period ($\approx 1/60$ second) and $F_{ij}$ being the $j$th field from the $i$th TV source ($i = 1$ to 3). When $F_{11}$ arrives, we assume that $S_1$, $S_2$, and $S_3$ are in the top position, as shown in Fig. 2. $F_{11}$ is written onto field memory $M_1$ from time zero to $T$. The switches then change to the middle positions, and $F_{12}$ is written onto $M_3$ while $F_{11}$, in $M_1$, is being transferred to $M_2$. At the same time, $F_{21}$ is also written onto $M_1$. Consequently, at the end of $2T$, $F_{21}$ is stored in $M_1$, $F_{11}$ in $M_2$, and $F_{12}$ in $M_3$ before the switches change position again. Now with the switches in the bottom position, $F_{13}$ is routed to the bottom path. It is then used to form a frame differential with $F_{11}$, from $M_2$, denoted by $F_{11} - F_{13}$, which is the

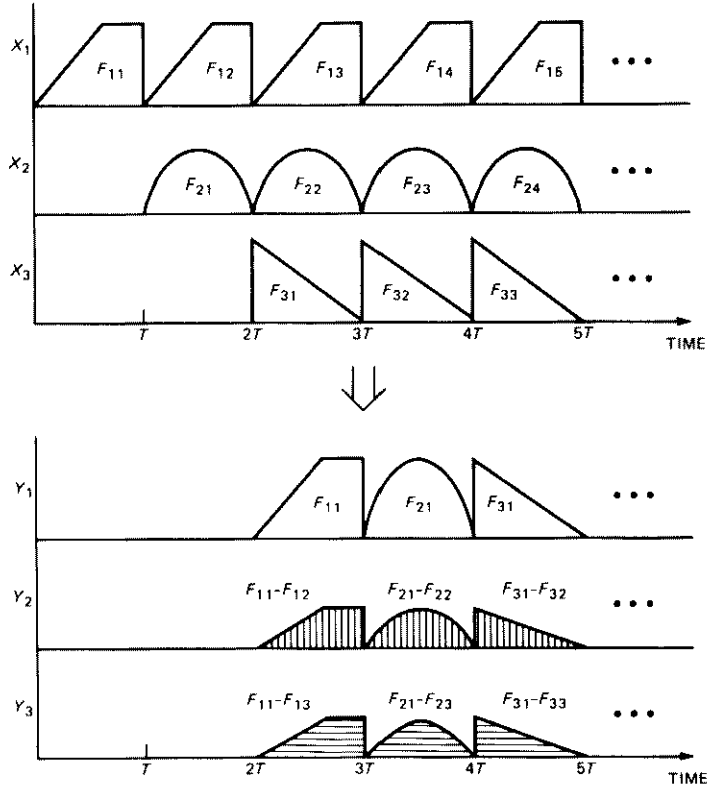



Fig. 3—Input/output waveforms for the differential-signal multiplexer.

output $Y_3$. The original unchanged signal $F_{11}$ is also read out from $M_2$ via $Y_1$. The remaining output $Y_2$, is a field differential derived from $F_{11}$ (from $M_2$) and $F_{12}$ (from $M_3$) and is denoted by $F_{11} - F_{12}$. While all these are taking place, $F_{21}$, from $M_1$, is transferred to $M_2$ with $F_{31}$ being written onto $M_1$, and $F_{22}$ is written onto $M_2$. These operations are repeated for all subsequent fields. The output waveforms are illustrated in the bottom of Fig. 3, where a processing delay of $2T$ is incurred. Such a delay enables the conversion from line-multiplexed serial inputs into time-multiplexed parallel outputs. Furthermore, there is flexibility in choosing which of the fields is to be read out as is and which type of differential signal is to be used. For instance, in the above example we could just as well send $F_{12}$ as is, send $F_{11} - F_{12}$ as a field-differential signal, and send $F_{13} - F_{12}$ as another field-differential signal. In any event, in every $T$-second output interval, there are always one original field plus two differential fields in the three outputs. The bandwidth of the original field is 4.2 MHz, and that of the differential signals is assumed to be 2 MHz.

### *2.2 The time-compression multiplexer*

The purpose of the time-compression multiplexer is to combine the three signals ($Y_1$, $Y_2$, and $Y_3$) from the differential-signal multiplexer into a single signal, $Z$. In other words, we would like to time compress the three inputs over every $T$-second interval into a single output with the same duration. This can be achieved by writing the digital words into a memory (say, a RAM) at one speed and reading them out at a faster speed (see Fig. 4). The ratio of the read clock to the write clock is the time-compression factor (>1 for time compression). Since the time compression is to be done over a $T$-second interval, we could

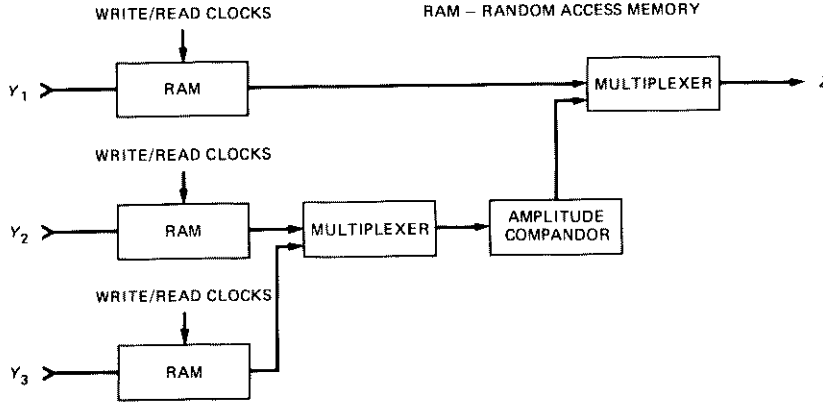



Fig. 4—The time-compression multiplexer.

write all $Y_i$'s into the RAMs for the field period before reading them out appropriately for multiplexing. But this would require the RAMs to accommodate entire fields of signals. Instead of this, we propose that the time compression be done over a line interval ($\approx$63.6 $\mu$s) so that only line memories are needed. More specifically, let us consider a line duration $T'$ within a $T$-second interval shown in Fig. 5. As before, $Y_1$ is the original 4.2-MHz TV; $Y_2$ and $Y_3$ are each a 2-MHz differential signal; and $\tau$ in the output $Z$ is the processing delay. We time compress the $T'$-second line of $Y_1$ by a factor of $\alpha(\alpha > 1)$ so that the resultant signal occupies $T_1$ seconds ($T_1 < T'$). Likewise, $Y_2$ and $Y_3$ are both compressed by $\beta(\beta > 1)$ so that each of their resultants occupies $T_2$ seconds ($T_2 < T_1 < T'$). We require that these three time-compressed signals be contained in $T'$, i.e.,

$$\frac{T'}{\alpha} + 2\,\frac{0.83T'}{\beta} = T'. \tag{1}$$

The factor 0.83 is due to the deletion of the differential-signal horizontal blanking intervals, which are identically zero and need not be sent. The above simplifies to

$$\frac{1}{\alpha} + \frac{1.66}{\beta} = 1. \tag{2}$$

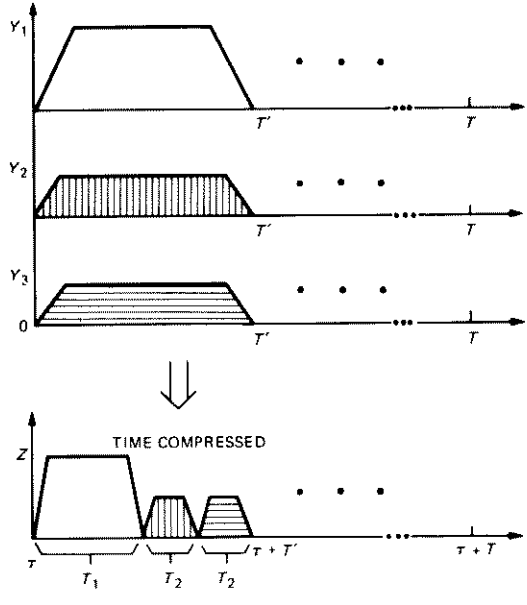



Fig. 5—Input/output waveforms for the time-compression multiplexer.

We further require that the three time-compressed signals have the same bandwidth. This can be written mathematically as[6]

$$\alpha f_1 = \beta f_2 = \beta f_3, \tag{3}$$

where $f_1$, $f_2$, and $f_3$ are the maximum frequencies of $Y_1$, $Y_2$, and $Y_3$, respectively. In this case, $f_1 = 4.2$ MHz, $f_2 = f_3 = 2$ MHz, and the solutions to (2) and (3) are

$$\alpha \approx 1.79;\ \beta = 3.76. \tag{4}$$

This yields $T_1 \approx 0.56T'$ and $T_2 \approx 0.22T'$. The maximum frequency of the combined output is given by (3) and is 7.52 MHz, as compared to 12.6 MHz obtained in a straightforward TCM of the three TVs. We call the above case *optimal* because its bandwidth has been minimized by the deletion of the horizontal blanking intervals in the differential signals. One obvious drawback, however, is that the compression factors required are noninteger, as given in (4). To circumvent this difficulty, we can simply choose $\alpha = 2$ and $\beta = 4$ exactly, i.e., compressing the original signal by two and the differential parts by four, with all their horizontal blankings retained. This *practical* case is much easier to implement with a slightly larger bandwidth of 8.4 MHz.

The last, but not the least, block in the time-compression multiplexer is the amplitude compandor (Fig. 4). As pointed out previously, the differential signals are chosen such that their amplitudes are small compared to the original signal on the average. The amplitude compandor equalizes the voltage levels for the differential signals in the combined output so that the FM link performance can be maximized. This was found to be very useful in a previous experiment[5] to suppress the effect of transmission noise on picture quality.

In summary, the present system takes in three NTSC TV signals and combines them into a 7.52-MHz (or 8.4-MHz) signal for transmission. The multiplexing technique is TCM, and the bandwidth reduction is the result of the use of differential signals. The transmission format is three fields from one TV source compressed into one ordinary field period. Thus, the transmission to the satellite is switched sequentially between the three sources at a rate equal to the field or vertical scanning frequency of ordinary NTSC TV (≈60 Hz). If the three TV sources are synchronized with one another, then the transmitter/multiplexer requires only three field memories. Otherwise, additional memory is needed for synchronization. In either case, the receiver requires only three field memories (see the appendix).

## III. PERFORMANCE

Overall performance of time-compression multiplexing of multiple TV signals in a satellite link has never been measured experimentally.

But calculations for estimating TCM performance were shown in Ref. 1. According to these calculations, the optimal case of the present system, which has a baseband-combined bandwidth of 7.52 MHz for the three TV signals, would require a receive earth station with a Gain/Temperature (G/T) of ≈33.7 dB/K to yield a receive baseband TV s/n of 56 dB. Such a G/T value is obtainable from 12-meter earth stations. The practical version (8.4-MHz baseband bandwidth), on the other hand, would require a G/T of 35.9 dB/K to yield s/n = 56 dB. Such a G/T is probably not obtainable with 12-meter stations. However, the degradation in s/n by using 12-meter receive earth stations is only about 2 dB, i.e., the received s/n would be ≈54 dB.

## IV. DISCUSSIONS

### *4.1 Audio*

With three TV sources, each producing stereo audio, we must transmit a total of six audio waveforms. We propose sending the stereo audio from each source along with its video by inserting digital audio in either the vertical or horizontal blanking periods. As for the optimal case where the horizontal blanking periods of the differential signals are deleted in transmission, the audio signals may be included in some convenient segment of the vertical blanking period. This of course will lead to a slightly more stringent timing requirement as well as some additional buffer memory.

As for the practical case where the horizontal blankings of the differential signals are retained for transmission, then the insertion of digital audio in these blanking periods can be done quite easily. Within a group of three video lines (one unchanged original plus two differentials), there are two horizontal blankings from the differential lines available. We can use one of these two blankings for one audio and the other blanking for the other audio. Within one of these time-compressed horizontal blanking intervals (≈2.7 $\mu$s), we must include the audio samples from three TV scan-line durations. Now each audio signal requires sampling at ≈32 kHz, and with nearly instantaneous companding, 10 bits per sample are sufficient.[7] Thus, we propose sampling the audio at exactly twice the TV line-scan rate, yielding a total of six samples or 60 bits from the three scan lines for transmission in the prescribed time-compressed horizontal blanking period. For this we would use twenty multilevel pulses to represent the 60-bit information. At a baud rate equal to 9/4 × color subcarrier frequency (≈8.06 MHz), the six audio samples from the three lines plus another pulse for bit timing would just fill the 2.7 $\mu$s time slot.

There are several ways of mapping the 60+ bits from the three lines into twenty multilevel pulses. More discussions in this regard are

provided in Ref. 5. Because the three TV lines are from three different fields, additional memory is needed to store their audio samples, but this requirement seems trivial compared to the video counterpart.

### *4.2 Synchronization for multiple up-links*

Use of TCM in satellite systems where up-links are not colocated requires that the three TV signals be synchronized, at least to the extent that their vertical blanking periods overlap.[8] This condition is not very stringent compared with that of some digital Time Division Multiple Access (TDMA) systems being proposed or in operation. Other than the additional synchronization hardware required for the transmitters, the only minor imposition in the system is that the receiver be able to demodulate the FM signal subject to short discontinuities in the received carrier at the vertical scanning frequency. Conventional limiter-discriminator receivers should have no problem in dealing with this. Phaselock receivers, on the other hand, might have lockup problems. But then the system is intended for high-quality transmissions with high carrier-to-noise ratios, and threshold extension is not needed.

As an aside, let us note that if the three TV sources are transmitted through noncolocated up-links, then the processing in each transmit earth station needs only two field memories (instead of three) to generate the differential signals required. The input switches in Fig. 2 are also unnecessary. A similar saving in receiver memory is possible too if only one TV is to be received in a down-link earth station.

### *4.3 Extension to non-NTSC TV signals*

Application of this technique to non-NTSC color TV signals may also be feasible. For example, with Phase Alternation Line (PAL) color television the color subcarrier phase is not the same as NTSC. However, with only a slight shift in the sampling pattern from line to line, the same differential signals can be defined and the same transmission system can be used. The same may be true of Sequential With Memory (SECAM) color television, but success is not as likely.

## V. CONCLUSION

We have described a method to transmit three NTSC TV signals in a 36-MHz satellite transponder. The technique uses differential signals to reduce the bandwidth and Time-Compression Multiplexing (TCM) to combine the three TVs into a single signal. By the use of novel circuit configurations, the memory requirements are reduced significantly compared with the more naive approach of Ref. 1. By companding the differential signals, the effect of transmission noise on

picture quality is markedly reduced. The estimated performance of the system is at or close to broadcast quality if 12-meter earth stations were to be used in a satellite system such as COMSTAR. Finally, digital audio signals can be sent without interference either to or from the video TCM signal by placing it in the horizontal blanking period. Extensions to up-links from separate earth stations and non-NTSC TVs are also possible.

## REFERENCES


1. K. Y. Eng and B. G. Haskell, "TV Bandwidth Compression Techniques Using Time Companded Differentials and Their Applications to Satellite Transmissions," B.S.T.J., *61*, No. 10, Part 1 (December 1982), pp. 2917–27.
2. J. E. Flood and D. I. Urquhart-Pullen, "Time-Compression-Multiplex Transmission," Proc. IEE, *111*, No. 4 (April 1964), pp. 647–68.
3. D. H. Morgen and E. N. Protonotarios, "Time Compression Multiplexing for Loop Transmission of Speech Signals," IEEE Trans. Commun., *COM-22*, No. 12 (December 1972), pp. 1932–9.
4. B. G. Haskell, "Time-Frequency Multiplexing of Two NTSC Color Signals—Simulation Results," B.S.T.J., *60*, No. 5 (May–June 1981), pp. 643–60.
5. R. L. Schmidt and B. G. Haskell, "Transmission of Two NTSC Color Television Signals Over a Single Satellite Transponder Via Time-Frequency Multiplexing," Globecom '82 Conf. Rec., *1*, pp. B6.3.1–.5.
6. K. Y. Eng and O. Yue, "Spectral Properties and Band-Limiting Effects of Time-Compressed TV Signals in a Time-Compression Multiplexing System," B.S.T.J., *60*, No. 9 (November 1981), pp. 2167–85.
7. M. G. Croll et al., "Nearly Instantaneous Digital Compandor for Transmitting Six Programme Signals in a 2.048 Mbit/s Multiplex," Electron. Lett. (July 12, 1973), pp. 298–300.
8. Kai Y. Eng and B. G. Haskell, "Synchronization of Noncolocated TV Signals in a Satellite Time-Compression Multiplexing System," B.S.T.J., this issue.


## APPENDIX

### *Decomposition of the TCM Signal From Three Video Sources*

As Fig. 6 shows, the received FM signal from the satellite is demodulated to recover the TCM baseband waveform. It is then digitized to produce $Z'$, which would be identical to $Z$ previously except for the transmission noise and channel distortion added. An amplitude compandor undoes the companding done to the composite waveform. Now the three segments in this waveform, namely the original field and the two differential signals, are then separated by the demultiplexer and written onto three separate memories. They are read out at slower speeds to get time expanded to the full scan-line length. The expansion factors (ratio of write clock to read clock) are precisely the compression factors used in the transmitter. Approximations to $Y_1$, $Y_2$, and $Y_3$, denoted here by $Y'_1$, $Y'_2$, and $Y'_3$, are then obtained. The same predictor as in the transmitter is used to convert the differential signals into the originals. The three output switches, $S_4$, $S_5$, and $S_6$, move in synchronism from the top position to the middle, to the bottom, and back to the top, etc. Their operations are identical to $S_1$,

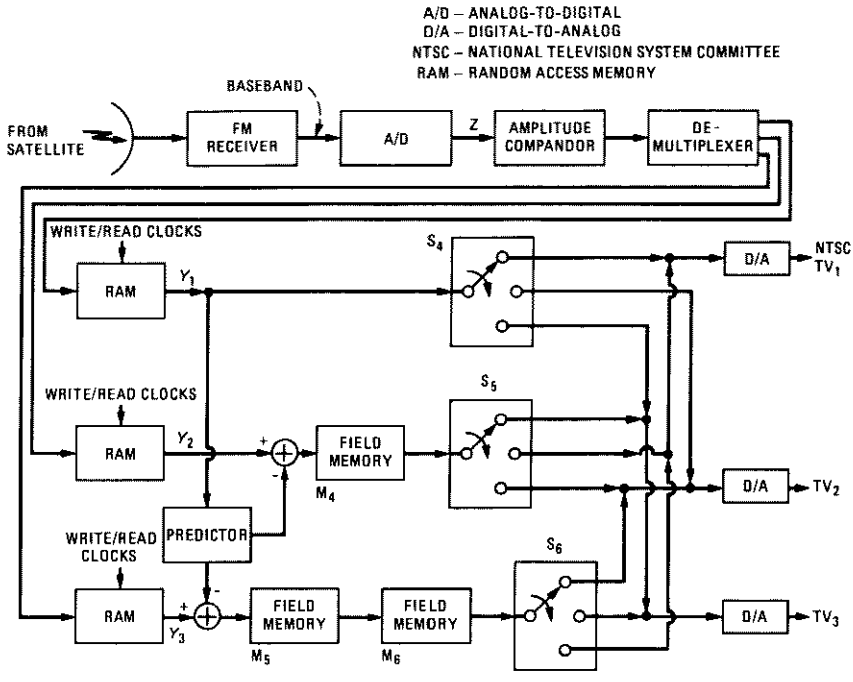



Fig. 6—Receive earth station for three TVs per transponder.

$S_2$, and $S_3$ in the transmitter, and they route the output digital signals to their appropriate outputs. The output digital signals may (or may not) then be converted to analog signals for display or local distribution.

## AUTHORS

**Kai Y. Eng,** B.S.E.E. (summa cum laude), 1974, Newark College of Engineering; M.S. (Electrical Engineering), 1976, Dr. Engr. Sc. (Electrical Engineering), 1979, Columbia University; RCA Astro-Electronics, 1974–1979; Bell Laboratories, 1979—. Mr. Eng has worked on various areas of microwave transmission, spacecraft antenna analysis, and communications satellites. He is presently a member of the Radio Research Laboratory, studying TV transmission through satellites. Member, IEEE, Sigma Xi, Tau Beta Pi, Eta Kappa Nu, Phi Eta Sigma.

**Barry G. Haskell,** B.S. (Electrical Engineering), 1964, M.S., 1965, and Ph.D., 1968, University of California, Berkeley; University of California, 1965–1968; Bell Laboratories, 1968—; Rutgers University, 1977–1979. Mr. Haskell was a Research Assistant at the University of California Electronics Research Laboratory and a part-time faculty member of the Department of Electrical Engineering at Rutgers University. At Bell Laboratories, he is presently Head of the Radio Communications Research Department, where his research

interests include television picture coding and transmission of digital and analog information via microwave radio. Member, IEEE, Phi Beta Kappa, Sigma Xi.

**Robert L. Schmidt,** B.S.E.E., 1982, Monmouth College; Bell Laboratories, 1972—. Mr. Schmidt is currently a member of the Radio Communications Research Department, where he has been exploring various techniques in combining multiple television signals onto a single analog radio channel. He has researched television signal encoding, bit rate reduction techniques, and software-controlled coding systems. Member, Eta Kappa Nu.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Synchronization of Noncolocated TV Signals in a Satellite Time-Compression Multiplexing System

By K. Y. ENG* and B. G. HASKELL*

(Manuscript received June 15, 1983)

We describe here a simple method to synchronize three TV signals originated from noncolocated up-link stations in a satellite Time-Compression Multiplexing (TCM) system. In this system, information in three fields of each TV picture is compressed into a single field time so that the compressed signals from the three sources can be time multiplexed for transmission. The up-link synchronization ensures that the Radio Frequency (RF) bursts from different sources will arrive at the satellite without collision. Our method employs a dynamic master/slave arrangement whereby the first station signing on assumes the role of a master. The other stations subsequently can synchronize their transmissions to the master's by simply monitoring the received RF bursts from the satellite, measuring their respective delays to the spacecraft, and then phase locking their local color subcarrier clocks to the master's transmitted bursts. When the master station stops transmitting, an automatic procedure is provided for the second station to take over as the new master. The worst-case jitter performance is well below 100 ns, and the initial acquisition time can be kept less than one-half second. These are more than adequate for the present TV application, although further improvements are possible if necessary.

## I. INTRODUCTION

Time-Compression Multiplexing (TCM) is a method of multiplexing various signals by time compressing their (analog) waveforms into segments in such a way that the compressed segments from different

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

sources can be sent on the same channel in separate time intervals (time-division multiplexing).[1,2] Previous published works[3–5] have discussed various properties and ways to implement TCM in the transmission of multiple high-quality TV signals through a single satellite transponder of 36-MHz bandwidth. More recently, this idea has further been refined to the transmission with practical hardware of three broadcast-quality TVs in a transponder[6] (i.e., the received peak-to-peak video to weighted rms noise ratio ≥56 dB.) As with other TCM systems, one requirement in the latter proposal is that the input three TV signals be synchronized, at least to the extent that their vertical-blanking intervals overlap. If the signals are colocated in the same up-link earth station, it merely implies that frame synchronizers be used. However, if they are to be transmitted from separate earth stations, then the up-links have to be synchronized. Of course, the up-link synchronization is needed to ensure that signal bursts from different sources would arrive at the satellite without collision. We show and discuss in this paper how this can be accomplished with simple and easy-to-implement hardware arrangements.

Synchronization techniques in communications satellite systems have been studied extensively in past years,[7–9] mostly in connection with digital Time Division Multiple Access (TDMA) applications. They could all be used in the present problem of synchronizing three TV up-links. However, these previous techniques were designed for performance far exceeding the present requirement and hence tend to be more complicated than what is needed. More importantly, they were meant for digital signals and are not suitable for analog TV where the color subcarrier and various sync pulses must bear strict phase and frequency relationships and thus cannot be advanced or retarded with respect to one another arbitrarily. We will show in the next section how a TV up-link station can synchronize its transmission by simply monitoring the Radio Frequency (RF) bursts sent by other station(s) already on the air. Such an approach enables synchronization between the three stations without a centrally controlled master station or clock, without the knowledge of one another's exact location, without the demodulation of one another's baseband video, and without the use of a separate control channel. The only assumption imposed is that the three up-link stations be within the down-link coverage of the satellite. This is true for satellites similar to Telstar III. The hardware implementation is quite simple (Section III) and can be realized by conventional equipments and digital circuits. Our timing analysis (Section IV) shows that its performance can cover all requirements under a variety of worst-case conditions, and simple procedures for failure recovery are discussed in Section V. Finally, we will make brief comparisons with other methods by showing a number

of practical advantages in using the present technique and also discuss possible extensions to further improve its performance.

## II. SYSTEM DESCRIPTION

We outline in this section the basic concept and operation of the present method. Detailed parameters and performance evaluation are left for subsequent discussions. The system configuration is illustrated in Fig. 1, where three up-link earth stations are to transmit their color TV signals to a satellite. The TV pictures are assumed to be National Television System Committee (NTSC) and are to be time compressed with processing prior to transmission so that TCM can be employed. More specifically, three fields of each TV are to be time compressed into one field period, F, ($\approx$1/60s) in a manner previously described in Ref. 6. The resulting waveform of a time-compressed TV contains successive triplets of a field with picture information followed by two blank field periods. The RF transmission of each earth station will

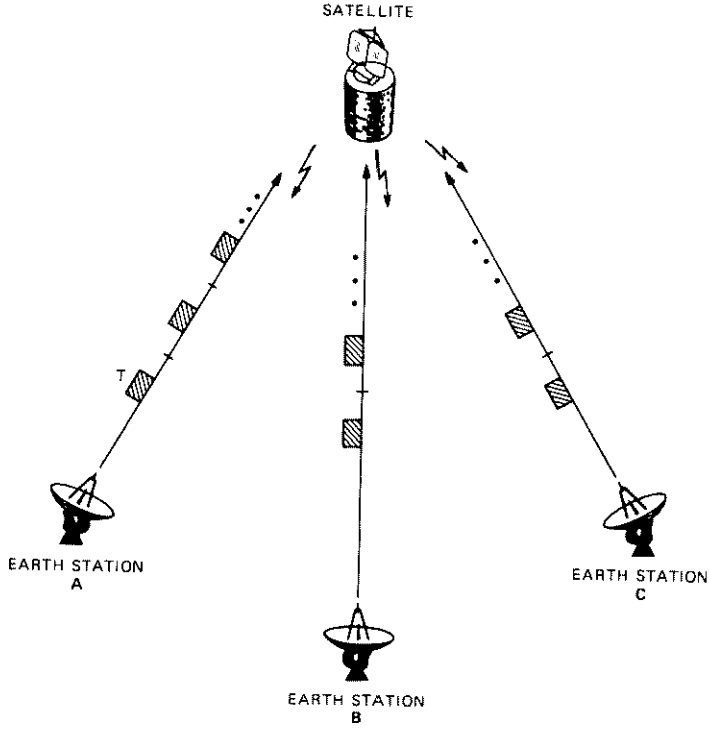



Fig. 1—A three TV/transponder TCM system.

then consist of bursts, each having approximately one field duration, with two blank field periods as separation between successive bursts (Fig. 2). The synchronization problem at hand is to align these bursts from the three stations so that they arrive at the satellite without overlap. All three stations are assumed to be within the down-link coverage of the satellite.

One could design the system, at least in principle, such that the entire portion of the vertical-blanking interval ($\approx$1.4 ms) within each TV burst is used for guard time. This would be sufficient to account for the diurnal drift of the satellite itself (maximum round-trip delay variation of about 500 $\mu$s according to Ref. 7). With the exact locations of the stations known, simple open-loop synchronization is then possible. The drawback of such an approach is twofold. First, the deletion of the entire vertical blanking is undesirable in TV transmission because a variety of test signals and nonvideo information are frequently inserted in this time period. Second, the exact known location requirement renders the scheme inflexible for the inclusion of transportable transmit earth stations.

We feel that the deletion of only a portion of a scan line (during vertical blanking, say 15 $\mu$s) for interburst guard time is reasonable and would not limit or interfere with picture performance. In addition, we do not assume that locations of the stations are known to one another. Instead, each station is assumed to know only its own approximate location, say to within $\pm$100 km. Note that the latter assumption is not imposing at all since every station needs some location information of its own for antenna pointing purposes anyway.

To illustrate the operation of the present system, the three up-link

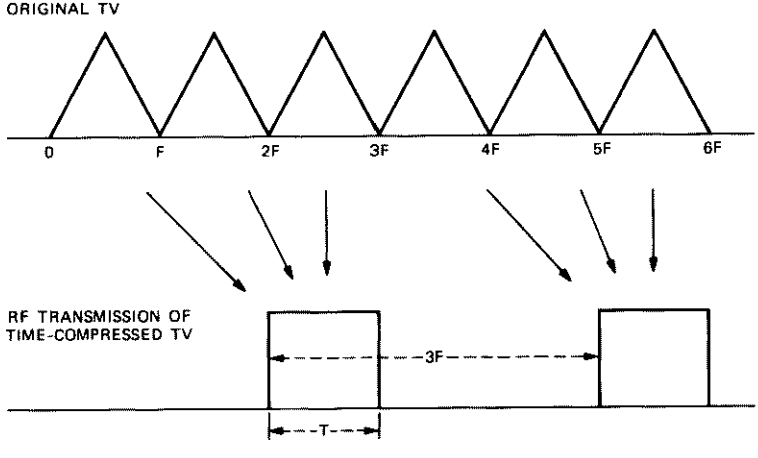



Fig. 2—Time-compression processing of a TV signal (F = Field period, T = F minus a small guard time).

earth stations are labelled A, B, and C. Station A is assumed to be the first to transmit. It can do so at will because no other transmission is taking place, and its transmission is simply synchronized to its own NTSC TV clock.

We now consider the start-up of B after A has been on the air. Station B first monitors the arrivals of the RF bursts from A and records their arrival times. Note that B does not have to demodulate A's signal; it only needs to detect the RF pulses received. (Indeed, A's baseband signal need not be video, as long as its RF timing is otherwise compatible.) The RF pulses from A occur in one out of three fields, the period is perturbed mainly by the time-varying propagation delay between A and B due to the spacecraft motion. Using these arrival times, B can extrapolate for the immediate future arrivals of A's pulses, and with the knowledge of its own approximate location (±100 km), B can compute its propagation delay to the satellite with an accuracy better than ±1.2 ms (including satellite drift). This estimated delay enables the translation of the arrival times of A's bursts from the time reference at B to that at the satellite. Using all this information, B can then position the transmission of a narrow pulse so that it arrives at the satellite in a time window adjacent to a burst from A, but not interfering with it. This narrow pulse is then received back by B, and we have an actual delay measurement, done inband, between B and the satellite. Once the actual delay is obtained, B can derive a windowing signal (frequency = one-third of the TV field rate) that denotes the proper transmission times in order to maintain collision-free synchronization with A.

The derivation of this window signal at B would mean the end of the problem if the system were for digital transmission. However, for TV applications, the picture information cannot be arbitrarily advanced or delayed without regard to the phase and frequency relationships between its color subcarrier and its sync pulses. Therefore, we propose frame (or field) synchronizing the TV picture at B to a local color subcarrier clock that is in turn phase locked to the aforementioned window signal in order to achieve proper transmission timing. This will be explained further when we discuss the hardware implementation.

Note that throughout the above procedure of synchronizing B to A, the up-link delay from A to the satellite remains unknown to B. This is possible because the timing error of B's narrow pulse (as will be shown later) is small compared with the start-up guard time allotted, i.e., a field period. Subsequent synchronization is maintained by B monitoring and updating the delay information and making adjustments accordingly. In this way, A is the master by virtue of being the first comer in the system, and B is locked onto A as a slave.

When station C wants to join in for its transmission, it has to go through the same procedure as B did, except it would lock onto B instead of A. If A drops out of transmission, B would detect that and take over as the master, using its own free-running clock, and C would stay locked to B. When A wants to resume its transmission later, it would have to join in as a slave to C. Therefore, the system assumes a dynamic master/slave arrangement where the first comer assumes the role of the master. Although this arrangement, as described, can only function properly if the three stations join the system sequentially, the time required by a station to establish itself as a slave can be designed to be well within a second, and thus for all practical purposes the initialization can be achieved almost instantaneously. We will show in the next section how all of these operations can be implemented with simple hardware.

## III. HARDWARE IMPLEMENTATION

We describe in this section the hardware implementation of the present method. The following discussion will be divided into two major parts. The first part outlines the generation of a window signal that marks the proper transmission time for the time-compressed TV bursts at the local earth station. This window signal is denoted by $r(t)$. The second part explains how $r(t)$ can be used to synchronize the incoming TV picture such that its time-compressed bursts automatically align with the transmission windows.

The window signal, $r(t)$, is a pulse train with pulse width, T, equal to a TV field period minus the guard time and with a repetition rate = 1/3F. It is generated by the window processor depicted in Fig. 3. We assume that an external clock of eight-times B's color subcarrier frequency is made available to the window processor. This (≈28-MHz)

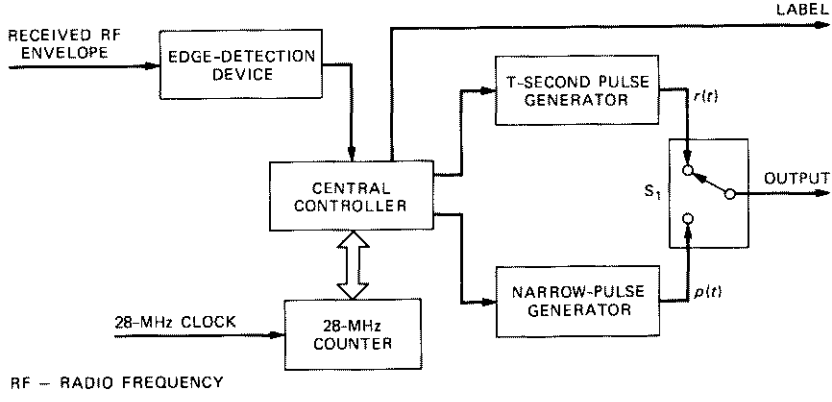



Fig. 3—Window processor for TDM/TCM synchronization.

clock is probably necessary for the time-compression operation itself, and its use here does not impose any additional burden on the system. The other input to the window processor is the received RF envelope from the satellite broadcast. In the trivial case of the first station (A) to go into the system, the window processor does very little because the transmission is free running. Let us consider the operation when the second station (B) wishes to start transmission. The received RF envelope (at B) is simply edge detected, and the arrival times of the bursts from A are recorded using the 28-MHz counter shown in Fig. 3. (Some accommodation for noise may be required, e.g., first detect envelope pulse of duration $\approx$T, then detect edges.) This information is supplied to the central controller, which could be a microprocessor and/or hardwired logic designed to carry out the windowing procedure outlined in the previous section. After acquiring the initial arrival times of the bursts from A, the central controller makes a crude estimate of the future arrival times. Furthermore, based on its location, it can compute an approximate delay to the satellite. Putting all these together, the controller produces a narrow pulse (pulse width $\ll$T) via the narrow pulse generator and sends it via the switch $S_1$ (in the lower position) to the transmitter. This narrow pulse will arrive at the satellite well within a predetermined time slot without collision with A's transmission. The return of this narrow pulse from the satellite completes a round-trip delay measurement that is then used to refine the arrival-time estimates. After a few cycles of this operation, the proper transmission time windows, $r(t)$, can be established by generating a sequence of pulses from the T-second pulse generator with $S_1$ switched to the upper position. Note that the pulse width and repetition rate of these T-second pulses are both computed using B's 28-MHz clock. A representative $r(t)$ is shown in Fig. 4a. The label output distinguishes the master from the slaves, and will be discussed later.

Before describing the rest of the hardware implementation, we show in Fig. 4 the conceptual sequence of operations needed to complete the synchronization. The transmission window is established by $r(t)$ in Fig. 4a. We use this to align (or phase lock) a composite TV sync signal, $s(t)$, such that every third, vertical sync pulse in $s(t)$ straddles the beginning of a transmission window (Fig. 4b). The sync signal, $s(t)$, is then used to synchronize an incoming video, resulting in $x_s(t)$, as shown in Fig. 4c. Finally, the video, $x_s(t)$, can be time compressed to obtain $x_c(t)$ (Fig. 4d), which is in synchronism with the transmission windows. The complete hardware to do all these is shown in Fig. 5.

Referring now to Fig. 5, the TV signal, $x(t)$, is passed through a frame synchronizer (and/or time base corrector) whose reference sync signal, $s(t)$, is derived from the TCM synchronizer. The frame synchronizer aligns $x(t)$ to $x_s(t)$ (Fig. 4c). The subsequent time compres-

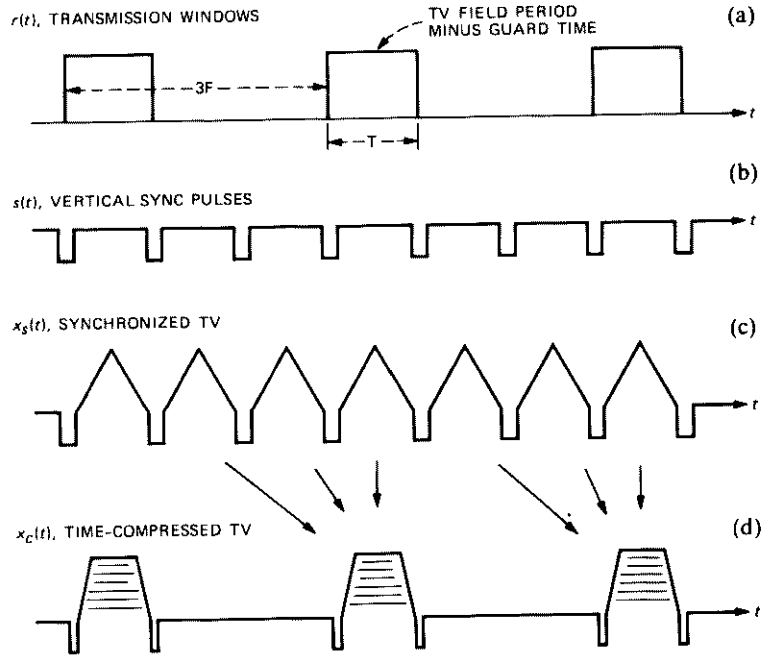



Fig. 4—Illustration of synchronization procedure. Horizontal sync, color burst, etc., are now shown.

sion on $x_s(t)$ is done in the time-compression processor previously described in Ref. 6. In this example, we assume that the time-compression processor requires three clock inputs in addition to the incoming video: a four-times color subcarrier clock ($\approx$14 MHz), an eight-times color subcarrier clock ($\approx$28 MHz), and the transmission window signal, $r(t)$. The time-compressed video, $x_c(t)$, is ready for immediate transmission through the FM modulator and the rest of the system. The *pin* modulator shown after the FM modulator is included to ensure the proper transmission timing as well as to enable the transmission of the narrow pulses at start-up.

As for the TCM synchronizer, its output is $s(t)$, as mentioned previously, and its input is the received RF envelope from the satellite broadcast. From the detected RF pulses, the window processor (Fig. 3) generates either $r(t)$ or the narrow pulses, depending on its state. When it is in the delay measurement mode, i.e., narrow pulses are being generated, the rest of the TCM synchronizer is free running. After $r(t)$ is generated, an internal 3.58-MHz color subcarrier is phase locked onto $r(t)$ via a TV sync generator and appropriate dividers as shown in Fig 3. This simple scheme ensures that the composite sync, $s(t)$, is synchronized with the transmission windows, $r(t)$. The label

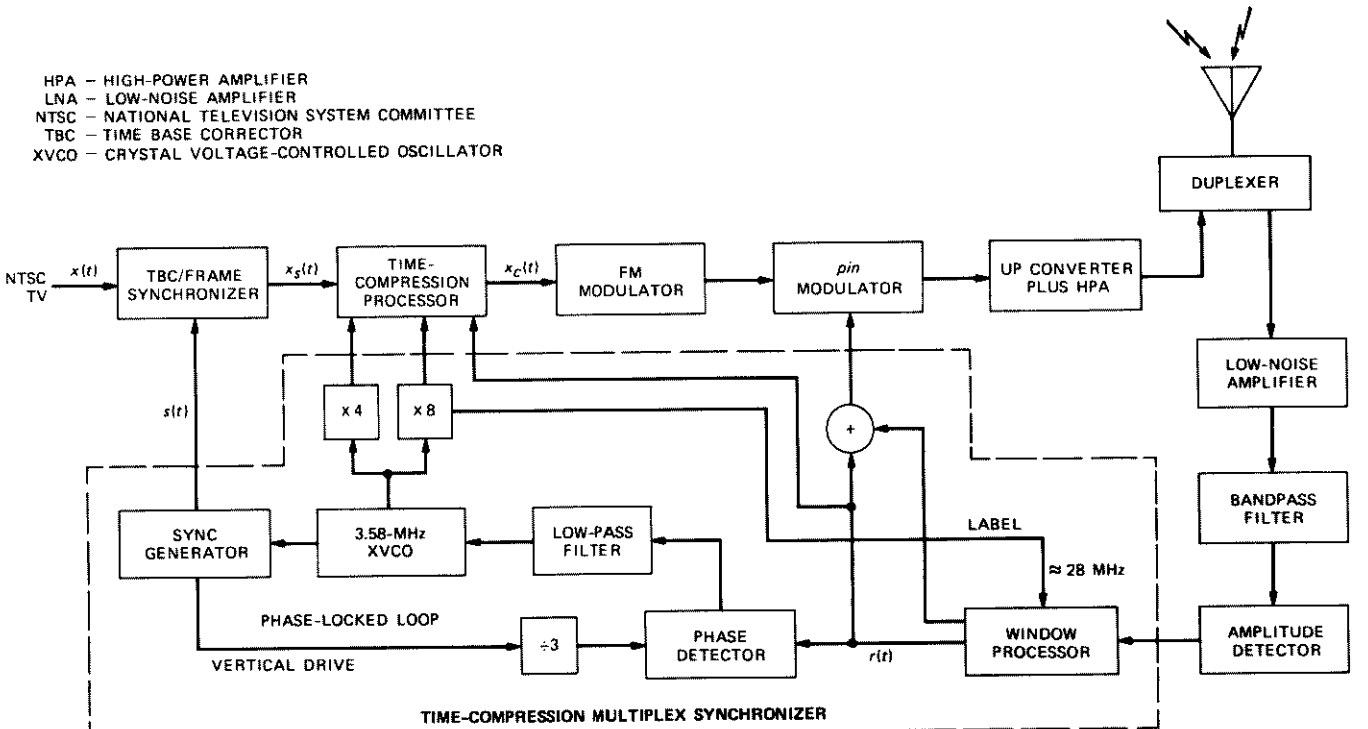



Fig. 5—Block diagram for earth station using present method for synchronization.

output of the window processor causes short RF pulses to be generated in the guard time in order to distinguish master from slaves. More discussion of labels will follow.

## IV. TIMING ANALYSIS

Two important timing parameters reflecting the performance of the synchronization method are considered here: the initial acquisition time and the subsequent timing jitter in steady state. In our case of TV broadcasting, transmission is usually planned ahead of time and thus an initial acquisition time of, say, a few seconds should be adequate. However, faster acquisition is probably desirable in the case of failure recovery, as will be discussed later.

The guard time needed between bursts from different users is obviously determined by the timing jitter of the synchronization method and is a rather critical parameter. In the present system, each RF burst has the duration of a TV field time minus guard time, and the TCM synchronizer at the transmit earth station has to detect these bursts individually in order to start, as well as to maintain, lock-up. Therefore, we must ensure that some detectable gap always exists between successive bursts. Since we have the freedom to choose how to segment the original TV into three-field groups before compression, we may as well do it in a way that creates a small gap, and therefore we propose that the segmentation be done during a line of the vertical-blanking interval, which contains no information. Furthermore, we deliberately delete from transmission a portion of that line, thus generating a gap between bursts that could amount to, say, 15 $\mu$s. This deletion during vertical blanking does not affect the video quality because it is done where there is no information. The resulting benefits of this are twofold: we have created the necessary time gap between bursts from different stations; and we have a sizable guard time of 15 $\mu$s to accommodate the timing jitter (and to include labels to be described in Section V).

The major causes and their effects on the steady-state timing jitter in our system are summarized in Table I. We now discuss briefly the

Table I—Summary of timing jitter performance

| Parameter | Jitter (ns) |
|---|---|
| Delay measurement uncertainty | ±22 |
| Up-link delay drift | ±11 |
| Down-link delay drift | ±1 |
| Clock resolution | ±17 |
| Field-rate jitter | ±0 |
| Total | ±51 |

meaning of each entry in the table, while the detailed derivation is left to the appendix:

1. Delay measurement uncertainty—Delay measurement is made either via the narrow pulses or by the monitoring of the up-link's own returned TV bursts. In either case, a local 28-MHz clock is used to record the time elapsed, and the clock resolution is limited to half a cycle. It is implicit that fixed delays through the satellite and earth station hardware can be calibrated out from the raw measurement. Since this measurement is done in the communication band, propagation effects are automatically minimized.

2. Up-link delay drift—This refers to the up-link delay variation from station A to the satellite, which is not known to station B. It is time varying because of the spacecraft motion. This cannot be eliminated because we assume station B does not know A's location or have any ranging information on the propagation from A to B.

3. Down-link delay drift—This refers to the down-link delay variation between the satellite and station B. It is also time varying because of the spacecraft motion, but it is trackable via the delay measurement at B. A simple linear prediction should almost eliminate this.

4. Clock resolution—This is the limitation in the TCM synchronizer to time itself for the exact instant to start transmission due to the finite clock resolution (half a cycle in the 28-MHz clock).

5. Field-rate jitter—B and C are trying to lock to the inherent jitter in the RF bursts from A. However, if A's TV source conforms to the NTSC standard, this is so small that it can be dropped for all practical purposes. Otherwise, this item must be included in the table.

As we saw in Table I, the steady-state jitter is so small compared to the 15 $\mu$s guard time that under normal circumstances the system can be regarded as jitter free.

We now make a worst-case estimate of the initial acquisition time. It is convenient to make the simplifying assumptions that the system is jitter free and the satellite is truly stationary. The resulting error due to these assumptions is only in the order of less than 100 ns, while the acquisition time, as will be shown below, is in the order of a second. Again we will treat the case of station B trying a cold start after station A has already been on the air.

After turn-on at station B, the window processor needs to monitor a few received bursts from A before it can position its narrow pulses for delay measurements. Since the bursts from A are arriving every 50 ms, the monitoring takes $\approx$150 ms. After this 150-ms listening period, the narrow pulses are sent for delay measurements, and in order to allow for two delay measurements, we need a maximum of $\approx$500 ms. Therefore, after $\approx$650 ms have elapsed since turn-on, the synchronizer

has completed the delay measurements and can compute the near-past, current, and near-future arrival times of A's bursts at the satellite. At this point, transmission can commence in the next available time slot, which in the worst case involves a delay of three field periods (≈50 ms). Putting everything together, we have a worst-case total of ≈700 ms between initial turn-on and the first TV transmission. Such an acquisition time certainly meets our objective of keeping it below a second. In fact, a potential saving of ≈250 ms exists if we do a single narrow pulse delay measurement instead of two. Therefore, we conclude that our acquisition time is less than one-half second with a single delay measurement and less than one second with delay verification.

## V. FAILURE RECOVERY

In any prudent system design the possibility of failure of certain components must be taken into account. Here, we desire that the failure of one channel does not disrupt the transmissions of the remaining channels. In order to facilitate this, we provide for a labeling mechanism, in which the window processor causes short RF pulses to be transmitted immediately following the video RF burst, i.e., at the beginning of the guard time. These pulses are then used to distinguish the master from the slaves, as well as to detect anomalies.

For example, station A (being the master) could transmit three pulses. Station B, the next in command, would send two pulses, and station C, one pulse. Additional pulses could identify the up-link station or, alternatively, this information could be embedded in the baseband video.

The window processor keeps track of time and labeling of all received RF bursts, and is ready to accommodate to any change in operating conditions. For example, if A finishes its transmission and goes off the air, B becomes the new master transmitting three pulses, and C becomes second in command transmitting two pulses.

It is never possible to predict all failure modes. The best we can do is accommodate the most likely ones. For example, a brief up-link failure will not be detected at any earth station (including, possibly, the faulty one) for about 240 ms, and during that time it is possible for transmission to resume. Moreover, corrective action by the faulty earth station will not be known to the remaining ones for another 240 ms. Thus, in the case of an up-link failure at master station A, station B should not try to take over as master immediately. Otherwise, there would be the possibility of two masters existing at the same time. In any event, as soon as station A determines that its up-link is unreliable, it should resign as master. This could be done by not transmitting any pulses following its video RF burst. The other stations would recognize

this condition and assume their proper responsibilities, after which station A would begin transmitting a single pulse designating itself as last station aboard.

In the case of a down-link failure, continued operation is not possible unless the faulty station is master. If it were not already the master, it could take over this role by sending, say, four pulses following the video RF burst. The other stations would then recognize this condition and assume their proper responsibilities.

In the case of an earth station power glitch, transmission would have to cease immediately and the start-up procedure would be reinvoked, since the window processor would, in all probability, lose its timing information. Such a restart could be speeded up considerably if nonvolatile memory were provided, however.

## VI. COMPARISONS AND DISCUSSIONS

As mentioned previously, a number of synchronization methods are applicable to solve the present problem. The most obvious one is probably that of a centrally controlled station broadcasting a master sync to all three up-link stations. Within this broad class of techniques, a large variety of alternatives are possible. As an example, one fixed station may be assigned as the master and the other stations must lock their transmissions to the master; a master sync marker may be broadcast to all stations by a centrally controlled station, and the marker could contain sufficient information to TV field and color subcarrier synchronizations, as well as ranging data for extremely fast open loop acquisition. In fact, only one such master is needed for the whole satellite system. Its main advantages are that fast acquisition is possible, and the various up-link stations do not have to monitor one another's transmissions, although the hardware implementation at each up-link station is certainly not simpler than our method. The key concern, though, is the reliability of the master station—its maintenance and hardware complexity. A single up-link failure at the master station would immobilize the whole system. In contrast, our method would tolerate quite a combination of different failures because an automatic takeover procedure exists for the master assignment. Any single up-link or down-link failure at a station can interrupt service only at that station and has no bearing on the rest of the system.

It is possible to use a separate channel to perform interstation ranging as proposed in Ref. 9. The bandwidth requirement for this ranging channel is critically determined by the rise time of the ranging pulses, which, in turn, affects the resulting synchronization accuracy. Therefore, the addition of this ranging channel could be an imposing requirement in the system.

Improvements in the jitter performance and the acquisition time in our system are both possible. The up-link drift could be removed if the up-link delay information from each station were inserted into one of the vertical-blanking pulses, and the stations could then demodulate for these data. Higher clock frequencies could be used in the delay measurement, thereby decreasing its uncertainty. This would also increase the time resolution of system and thus enable the synchronizer to time the transmissions more accurately. As for the acquisition time, if an accurate site location plus its up-link delay were provided by the first (or the master) station in one of its vertical-blanking pulses, then the other stations could compute their respective delays to the spacecraft without performing the narrow pulse measurements, resulting in a significant reduction in the acquisition time.

## VII. CONCLUSIONS

We have described a method of synchronizing up-link earth stations in a TCM system where the stations take turns transmitting TV information in bursts, each lasting for a field duration. The technique is simple and requires only that the stations receive their own as well as others' transmissions. It has a dynamic master/slave arrangement whereby the first station signing on assumes the role of a master. The other stations subsequently can synchronize their transmissions to the master's by simply monitoring the received RF bursts from the satellite, measuring their respective delays to the spacecraft, and then phase locking their local color subcarrier clocks to the master's transmitted bursts. When the master station stops transmitting, an automatic procedure exists for the second station to take over as the new master. As a result, any single up-link or down-link failure can only affect the station involved, and there is no need to have centralized control. Most of the hardware in the synchronizer can be implemented digitally. The worst-case jitter performance in the system is well below 100 ns, while the initial acquisition time can be kept to less than one-half second. These are more than adequate for the TV application, and we conclude that the proposed method offers a practical means to synchronize the three up-links in our TCM system.

## REFERENCES


1. J. E. Flood and D. I. Urquhart-Pullen, "Time-Compression-Multiplex Transmission," Proc. IEE, *111*, No. 4 (April 1964), pp. 647–68.
2. D. H. Morgen and E. N. Protonotarios, "Time Compression Multiplexing for Loop Transmission of Speech Signals," IEEE Trans. Comm., *COM-22*, No. 12 (December 1972), pp. 1932–9.
3. K. Y. Eng and O.-C. Yue, "Spectral Properties and Band-Limiting Effects of Time-Compressed TV Signals in a Time-Compression Multiplexing System," B.S.T.J., *60*, No. 9 (November 1981), pp. 2167–85.
4. K. Y. Eng and O.-C. Yue, "Time Compression Multiplexing of Multiple Television

Signals in Satellite Channels Using Chirp Transform Processors," IEEE Trans. Commun., *COM-29*, No. 12 (December 1981), pp. 1932–40.
5. K. Y. Eng and B. G. Haskell, "TV Bandwidth Compression Techniques Using Time Companded Differentials and Their Applications to Satellite Transmission," B.S.T.J., *61*, No. 10, Part 1 (December 1982), pp. 2917–27.
6. K. Y. Eng, B. G. Haskell, and R. L. Schmidt, "Time-Compression Multiplexing (TCM) of Three Broadcast-Quality TV Signals on a Satellite Transponder," B.S.T.J., this issue.
7. P. P. Nuspl et al., "Synchronization Methods for TDMA," Proc. IEEE, *65*, No. 3 (March 1977), pp. 434–44.
8. J. J. Spilker, Jr., *Digital Communications by Satellite*, Englewood Cliffs, NJ: Prentice-Hall, 1977.
9. A. Acampora, "Synchronizing Transmissions From Two Earth Stations to Satellite," U.S. Patent 4,320,503, applied for August, 1979, issued March 1982.
10. D. D. Slavinskas, private communication.

## APPENDIX

### *Parameters for Timing Jitter*

We show briefly here the derivations for the various contributions to the steady-state timing jitter (Table I). The following estimates are, by and large, worst-case and very conservative.

#### *A.1 Delay measurement uncertainty*

The slave stations have to measure their respective delays to the satellite in order to start, as well as to keep, synchronized with the master. This is done in the beginning via the narrow pulses, and then it is updated continuously via the monitoring of its own returned bursts. The delay is, of course, measured from edge to edge in the transmitted and received RF bursts. Given a 36-MHz RF channel bandwidth, the fastest RF pulse rise time is in the order of 30 ns. If we have to measure delay from one edge to another, an accuracy of ±5 ns seems reasonable. In addition, the clock used for the measurement is resolution limited due to its finite frequency (≈28 MHz, or eight-times color subcarrier frequency). The uncertainty due to this clock is about ±17 ns, yielding a total uncertainty of ±22 ns.

#### *A.2 Up-link delay drift*

In the absence of any knowledge of the master's (or A's) location, a slave station (or B) cannot predict the up-link delay from the master to the satellite. Furthermore, this up-link delay is time varying due to the motion of the spacecraft. The net result is that B's prediction of the near-future burst arrivals from A can never be exact, even though the down-link delay between B and the satellite can be predicted exactly. To illustrate this point, let us consider a burst transmitted from A to B at $t = 0$. The up-link delay (from A to the satellite) is $u_0$; the down-link delay (from the satellite to B) is $d_0$; and the delay through the satellite is conveniently chosen to be zero. The arrival time of this burst at B is simply

$$T_0 = u_0 + d_0.$$

Now, at a later instant $t = t_1$, A transmits another burst to B. The corresponding up-link and down-link delays are $u_1$ and $d_1$, respectively. Again $u_0 \neq u_1$ and $d_0 \neq d_1$ because of the spacecraft motion. The arrival time at B is then

$$T_1 = u_1 + d_1 + t_1.$$

In order to predict $T_1$ at B at the time $T_0$, B has to compute

$$T_1 - T_0 = (u_1 - u_0) + (d_1 - d_0) + t_1,$$

where $t_1$ is known to B because A is transmitting at a fixed rate; $(d_1 - d_0)$ can be extrapolated based on B's delay measurements; but the quantity $(u_1 - u_0)$ cannot be estimated without knowing A's location. In this example, $(u_1 - u_0)$ is simply the up-link delay variation for A due to the spacecraft displacement in the time interval $t_1$. As such, an easy upper can be written as

$$|u_1 - u_0| \leq cvt_1,$$

where $c$ is the velocity of light; $v$ is the radial velocity of the spacecraft toward or away from an earth station; and $t_1$, the time interval, is understood to be small compared to a day. If we replace $v$ by the highest radial velocity of the spacecraft, and $t_1$ by the round-trip satellite propagation delay ($\approx$300 ms), we have a worst-case estimate on the up-link delay drift. According to an example given in page 149 of Ref. 8 and comparisons to data from more recent communications satellites,[10] a convenient upper bound on the spacecraft radial velocity in geostationary orbit is 10 m/s. Using this, the worst-case up-link delay drift is $\pm$10 ns.

### *A.3 Down-link delay drift*

With the spacecraft radial velocity limited to 10 m/s and continuous updates on the delay measurement, we feel that the down-link delay drift can easily be computed with an accuracy an order of magnitude lower than the up-link delay drift, or about $\pm$1 ns.

### *A.4 Clock resolution*

Using a 28-MHz clock, the resolution is about half a cycle or $\pm$17 ns.

### *A.5 Field-rate jitter*

This refers to the jitter in the rate at which the master station is transmitting its RF bursts. The burst transmission is of course governed by the TV field rate, and only one burst is sent in every three fields. The NTSC standard specifies that the color subcarrier fre-

quency (3,579,545 Hz) must be stable within ±10 Hz and cannot vary more than 0.1 Hz/s. For a worst-case situation, we assume that the color subcarrier is at the lowest value, i.e., 3,579,535 Hz. It then drifts at the maximum rate of 0.1 Hz/s. Thus, at the end of a second, the new frequency is 3579535.1 Hz. The difference in TV field period derived from these two frequencies is only $7.8 \times 10^{-15}$ s. The net result is that the TV field rate is jitter free over a short period of time, say a few seconds. Moreover, this implies that a much less stable color subcarrier frequency is still quite compatible with our synchronization system.

## AUTHORS

**Kai Y. Eng,** B.S.E.E. (summa cum laude), 1974, Newark College of Engineering; M.S. (Electrical Engineering), 1976, Dr. Engr. Sc. (Electrical Engineering), 1979, Columbia University; RCA Astro-Electronics, 1974–1979; Bell Laboratories, 1979—. Mr. Eng has worked on various areas of microwave transmission, spacecraft antenna analysis, and communications satellites. He is presently a member of the Radio Research Laboratory, studying TV transmission through satellites. Member, IEEE, Sigma Xi, Tau Beta Pi, Eta Kappa Nu, Phi Eta Sigma.

**Barry G. Haskell,** B.S. (Electrical Engineering), 1964, M.S., 1965, and Ph.D., 1968, University of California, Berkeley; University of California, 1965–1968; Bell Laboratories, 1968—; Rutgers University, 1977–1979. Mr. Haskell was a Research Assistant at the University of California Electronics Research Laboratory and a part-time faculty member of the Department of Electrical Engineering at Rutgers University. At Bell Laboratories, he is presently Head of the Radio Communications Research Department, where his research interests include television picture coding and transmission of digital and analog information via microwave radio. Member, IEEE, Phi Beta Kappa, Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Theory of Reflection From Antireflection Coatings

By R. H. CLARKE*

(Manuscript received November 24, 1982)

The reflection that occurs when a beam, rather than a plane wave, is incident normally on a quarter-wavelength matching layer can be of vital importance in semiconductor laser design. An analysis in three dimensions is given for the general case of a field of arbitrary form and polarization incident on the matching layer. The field is represented as an angular spectrum of plane waves, each component plane wave being modified by the appropriate Fresnel reflection coefficient to give the field reflected back onto the diode structure. Brown's antenna reciprocity theorem is used to determine the amplitude of the corresponding mode traveling back down the diode.

## I. INTRODUCTION

Antireflection coatings are used on one face of superluminescent diodes[1] and on both faces of diode-laser amplifiers.[2] The theoretical performance of such coatings has been analyzed by Clarke[3] using the technique of representing the emerging laser beam as an angular spectrum of plane waves, as originally applied by Reinhart et al.[4] and Gordon[5] to determine the reflectivity of an uncoated facet. Each plane wave was modified by the appropriate reflection coefficient of the uniform coating,[6] and Brown's antenna reciprocity theorem[7] used to calculate the amplitude of the wave coupled back into the device. The previous analysis[3] was restricted to two dimensions, on the grounds

* Work done while at Bell Laboratories. Now at Imperial College of Science and Technology, London, England.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

that the active region in the device would be a wide flat stripe, so that the emerging radiation would be a thin fan-shaped beam. Many important laser diode structures, particularly of the refractive index guided type, have relatively narrow active regions, hence the previous restriction is limiting. This restriction is removed in the present work, and the full three-dimensional analysis is presented.

## II. FIELDS IN THE DIODE

The transverse electric field of a single mode traveling in the positive-$z$ direction (see Fig. 1) along the length of the active-region stripe in a diode laser can be written in general as

$$\mathbf{E}_t^+(x, y, z) = [\mathbf{u}_x E_{tx}(x, y) + \mathbf{u}_y E_{ty}(x, y)]e^{-j\beta_m z}, \tag{1}$$

where $\beta_m$ is the phase constant of the mode and the time variation $\exp(j\omega t)$ has been suppressed. The field in this mode reflected back into the diode by the coating is

$$\mathbf{E}_t^-(x, y, z) = \rho[\mathbf{u}_x E_{tx}(x, y) + \mathbf{u}_y E_{ty}(x, y)]e^{+j\beta_m z}. \tag{2}$$

The objective of this paper is to calculate the reflection coefficient $\rho$ for arbitrary thickness $h$ and refractive index $n_2$ of the coating. (Coupling to other modes is ignored here for the sake of simplicity.) It will be assumed that the beam eventually emerges into air, so that $n_3 = 1$, and that the refractive index of the diode has the effective value $n_1$, which is that of the active region in which the field is largely confined. (The surrounding bulk material has a refractive index that is some 10 percent below $n_1$.[1] A better choice of effective refractive index might therefore be a weighted average, as suggested by Kaplan's analysis.[8])

The field incident at the plane $z = 0$ can be represented as an

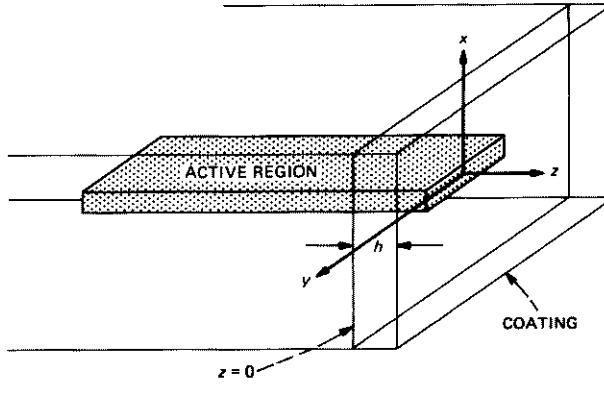



Fig. 1—Diode laser with coating.

angular spectrum of plane waves by the two spectrum functions $F_x(\alpha, \beta)$ and $F_y(\alpha, \beta)$, where $(\alpha, \beta, \gamma)$ are the direction cosines in the $x$-, $y$-, and $z$-directions.[9,10] Thus, the elemental plane wave incident in the direction $(\alpha, \beta)$ is $\mathbf{e}_{\text{inc}}(\alpha, \beta)d\alpha d\beta$, where

$$\mathbf{e}_{\text{inc}}(\alpha, \beta) = F_x(\alpha, \beta)\left(\mathbf{u}_x - \mathbf{u}_z\frac{\alpha}{\gamma}\right) + F_y(\alpha, \beta)\left(\mathbf{u}_y - \mathbf{u}_z\frac{\beta}{\gamma}\right) \quad (3)$$

with

$$F_x(\alpha, \beta) \leftrightarrow E_{tx}(x, y)$$

and

$$F_y(\alpha, \beta) \leftrightarrow E_{ty}(x, y), \quad (4)$$

in which $\leftrightarrow$ symbolizes a Fourier transform, such as

$$F_x(\alpha, \beta) = \frac{1}{\lambda_1^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E_{tx}(x, y)\exp\{jk_1(\alpha x + \beta y)\}dxdy. \quad (5)$$

The phase constant in the diode is $k_1 = 2\pi/\lambda_1 = n_1 k_0$, where $k_0$ is the phase constant of free space.

It should be noted for later reference that the above angular spectrum corresponds to a radiation far field (assuming that the subscript 1 region continues indefinitely but the active region stops in the plane $z = 0$), as $k_1 r \to \infty$, of[10]

$$\mathbf{E}(r, \theta, \phi) \simeq \frac{\exp(-jk_1 r)}{k_1 r}\mathbf{e}_{\text{rad}}(\alpha, \beta), \quad (6)$$

where the far-field vector pattern function is given in this case by

$$\mathbf{e}_{\text{rad}}(\alpha, \beta) = j2\pi\gamma\mathbf{e}_{\text{inc}}(\alpha, \beta) \quad (7)$$

and

$$\begin{aligned}\alpha &= \sin\theta\cos\phi\\ \beta &= \sin\theta\sin\phi\\ \gamma &= \cos\theta,\end{aligned} \quad (8)$$

where $\theta$ is the polar angle to the $z$-axis, $\phi$ is the azimuth angle in the $x$-$y$ plane, and $r$ is the distance to the point of observation.

## III. REFLECTION AT THE COATING

The incident plane wave given by eq. (3) will be reflected by the coating. The amplitude reflection coefficient for a plane wave incident on such a uniform layer with its electric vector perpendicular to its

plane of incidence (see Fig. 2) is[6]

$$R_\perp = \frac{P_1 \cos B + jP_3 \sin B}{P_2 \cos B + jP_4 \sin B} \tag{9}$$

and, with its electric vector parallel to its plane of incidence, the reflection coefficient is

$$R_\parallel = \frac{Q_1 \cos B + jQ_3 \sin B}{Q_2 \cos B + jQ_4 \sin B}, \tag{10}$$

where

$$P_{1,2} = n_2(1 - n_1^2 s^2/n_2^2)^{1/2}[n_1\gamma \mp (1 - n_1^2 s^2)^{1/2}]$$

$$P_{3,4} = n_1\gamma(1 - n_1^2 s^2)^{1/2} \mp n_2^2(1 - n_1^2 s^2/n_2^2) \tag{11}$$

$$Q_{1,2} = n_2(1 - n_1^2 s^2/n_2^2)^{1/2}[n_1(1 - n_1^2 s^2)^{1/2} \mp \gamma]$$

$$Q_{3,4} = n_1(1 - n_1^2 s^2/n_2^2) \mp n_2^2\gamma(1 - n_1^2 s^2)^{1/2} \tag{12}$$

with

$$s^2 = \alpha^2 + \beta^2 = 1 - \gamma^2 \tag{13}$$

and

$$B = (2\pi h/\lambda_2)(1 - n_1^2 s^2/n_2^2)^{1/2}, \tag{14}$$

where $\lambda_2 = \lambda_0/n_2$.

Note that when the magnitude of the sine of the angle of incidence $|s| > (n_1)^{-1}$, the wave will be totally internally reflected. In that case, the magnitude of the reflection coefficient will always be unity, but its phase will vary with the angle of incidence. But note also that the $\exp(j\omega t)$ sign convention adopted here means taking the negative

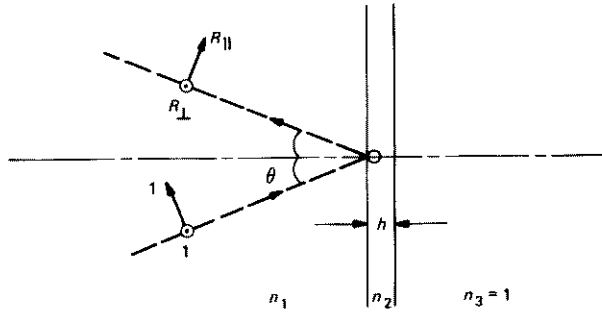



Fig. 2—Definition of the amplitude reflection coefficients $R_\perp$ and $R_\parallel$. Their phases are defined at O.

square root when the round-bracketed quantities in eqs. (11), (12), and (14) become negative.

The elemental plane wave given by the angular spectrum of eq. (3) consists, in general, of the sum of perpendicular and parallel polarized components, such that

$$\mathbf{e}_{\text{inc}} = \mathbf{e}_{\perp} + e_{\parallel}. \tag{15}$$

This resolution can be achieved by noting that the unit vector $\mathbf{u}_n$, which is both normal to the plane of incidence of the plane wave travelling in the direction $(\alpha, \beta)$ and also parallel to the bounding plane surface $x$O$y$, is

$$\mathbf{u}_n = \frac{1}{\sqrt{\alpha^2 + \beta^2}} [\mathbf{u}_x\beta - \mathbf{u}_y\alpha]. \tag{16}$$

Hence we may calculate

$$\mathbf{e}_{\perp} = \mathbf{u}_n[\mathbf{u}_n \cdot \mathbf{e}_{\text{inc}}) \tag{17}$$

and

$$\mathbf{e}_{\parallel} = \mathbf{e}_{\text{inc}} - \mathbf{u}_n(\mathbf{u}_n \cdot \mathbf{e}_{\text{inc}}). \tag{18}$$

The elemental plane wave $\mathbf{e}_{\text{refl}}(\alpha, \beta)d\alpha d\beta$ reflected by the coating is thus given by

$$\mathbf{e}_{\text{refl}}(\alpha, \beta) = R_{\perp}\mathbf{e}_{\perp}(\alpha, \beta) + R_{\parallel}\mathbf{u}_{\parallel}(\alpha, \beta) \tag{19}$$

and comes *from* the direction $(-\alpha, -\beta)$. (To avoid possible confusion it should be noted that the argument of $\mathbf{e}_{\text{refl}}(\alpha, \beta)$ denotes the direction of the incident wave.)

## IV. COUPLING BACK INTO THE DIODE

Brown's antenna reciprocity theorem states that if a plane wave of vector amplitude $\mathbf{e}_p$ is incident from the direction $\mathbf{u}_p$ on a linear, reciprocal device, which when radiating has the far-field vector pattern function (see eq. 6) of $\mathbf{e}_{\text{rad}}(\mathbf{u})$, then the coupling ratio

$$c = \frac{\lambda^2}{j4\pi ZP_0} \mathbf{e}_p \cdot \mathbf{e}_{\text{rad}}(\mathbf{u}_p) \tag{20}$$

gives the complex ratio of the single-mode amplitude when receiving to that when transmitting a total power $P_0$.[7,11] $Z$ and $\lambda$ are the characteristic impedance and wavelength in the radiating medium. Equation (20) is a precise result, being a consequence ultimately of the Lorentz reciprocity theorem.

In the present instance the incident plane wave $\mathbf{e}_p$ is the elemental reflected plane wave $\mathbf{e}_{\text{ref}}d\alpha d\beta$ given by eq. (19), and so, integrating

over all directions in the forward hemisphere, the reflection coefficient describing the returning mode amplitude is

$$\rho = \frac{\lambda^2}{j4\pi Z P_0} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{e}_{\text{refl}}(\alpha, \beta) \cdot \mathbf{e}_{\text{rad}}(-\alpha, -\beta) d\alpha d\beta \tag{21}$$

or

$$\rho = \frac{\lambda^2}{2ZP_0} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \{R_{\parallel} \mathbf{e}_{\text{inc}}(\alpha, \beta) + \mathbf{u}_n (R_{\perp} - R_{\parallel}) [\mathbf{u}_n \cdot \mathbf{e}_{\text{inc}}(\alpha, \beta)]\} \cdot \mathbf{e}_{\text{inc}}(-\alpha, -\beta) \gamma d\alpha d\beta \tag{22}$$

with $\mathbf{e}_{\text{inc}}(\alpha, \beta)$ given by eq. (3) and $\mathbf{u}_n$ by eq. (16). The total radiated power $P_0$, when the radiation is specified by the two spectrum functions $F_x(\alpha, \beta)$ and $F_y(\alpha, \beta)$, is given by[12]

$$P_0 = \frac{\lambda^2}{2Z} \int_D \int \left[ \frac{1 - \beta^2}{\gamma} |F_x(\alpha, \beta)|^2 + \frac{1 - \alpha^2}{\gamma} |F_y(\alpha, \beta)|^2 \right] d\alpha d\beta, \tag{23}$$

where $D$ is the domain of $(\alpha, \beta)$ such that $\alpha^2 + \beta^2 = \leq 1$.

## V. APPLICATION TO A Y-POLARIZED LASER MODE

In order to see what this result means, consider a guided mode in the laser whose tangential electric field is wholly $y$-directed, for which therefore $F_x \equiv 0$. Then

$$\mathbf{e}_{\text{inc}}(\alpha, \beta) = F_y(\alpha, \beta) \left( \mathbf{u}_y - \mathbf{u}_z \frac{\beta}{\gamma} \right) \tag{24}$$

and

$$\mathbf{u}_n = \frac{1}{\sqrt{\alpha^2 + \beta^2}} (\mathbf{u}_x \beta - \mathbf{u}_y \alpha). \tag{25}$$

Consequently, the reflection coefficient in this case is

$$\rho = \frac{\lambda^2}{2ZP_0} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ R_{\parallel} \left( 1 + \frac{\beta^2}{\gamma^2} - \frac{\alpha^2}{\alpha + \beta^2} \right) + R_{\perp} \frac{\alpha^2}{\alpha^2 + \beta^2} \right] \gamma F_y(\alpha, \beta) F_y(-\alpha, -\beta) d\alpha d\beta. \tag{26}$$

Then finally, assuming that the beam spread is vanishingly narrow in the $y$–$z$ plane compared to the $x$–$y$ plane,

$$\rho = \frac{\lambda^2 K}{2ZP_0} \int_{-\infty}^{\infty} \gamma R_{\perp} F_y(\alpha, 0) F_y(-\alpha, 0) d\alpha, \tag{27}$$

where $K$ depends on the $\beta$-dependence of $F_y$. Equation (27) is the two-dimensional result used previously.[3]

## VI. CONCLUSIONS

A complete three-dimensional analysis has been presented for the calculation of reflection from antireflection coatings. It reduces to the two-dimensional result given previously where the incident beam was assumed to be narrowly confined in one of the principal planes. The form and polarization of the field incident on the coating can be arbitrarily specified.

## REFERENCES


1. I. P. Kaminow et al., "Lateral Confinement InGaAsP Superluminescent Diode at 1.3 $\mu m$," IEEE J. Quantum Electron., *QE-19*, No. 1 (January 1983), pp. 78–81.
2. D. Marcuse, "Computer Model of an Injection Laser Amplifier," IEEE J. Quantum Electron., *QE-19*, No. 1 (January 1983), pp. 63–73.
3. R. H. Clarke, "Theoretical Performance of an Anti-Reflection Coating for a Diode Laser Amplifier," Int. J. Electron., *53*, No. 5 (November 1982), pp. 495–9.
4. F. K. Reinhart, I. Hayashi, and M. B. Panish, "Mode Reflectivity and Waveguide Properties of Double-Heterostructure Injection Lasers," J. Appl. Phys., *42*, No. 11 (October 1971), pp. 4466–79.
5. E. I. Gordon, "Mode Selection in GaAs Injection Lasers Resulting from Fresnel Reflection," IEEE J. Quantum Electron., *QE-9* (July 1973), pp. 772–6.
6. M. Born and E. Wolf, *Principles of Optics*, Fifth Edition, Elmsford, N.Y.: Pergamon, 1975, p. 61.
7. J. Brown, "A Generalized Form of the Aerial Reciprocity Theorem," Proc. IEE, *105*, Part C (1958), pp. 472–5.
8. D. R. Kaplan, private communication.
9. P. C. Clemmow, *The Plane Wave Spectrum Representation of Electromagnetic Fields*, Elmsford, N.Y.: Pergamon, 1966.
10. R. H. Clarke and J. Brown, *Diffraction Theory and Antennas*, N.Y.: Halsted, 1980, p. 85.
11. Reference 10, p. 109.
12. Reference 10, pp. 83–4.


## AUTHOR


**Richard H. Clarke,** B.Sc., 1956, Ph.D., 1960 (Electrical Engineering), University College, London; Assistant Professor, University of California, Berkeley, 1962–1964; Bell Laboratories, 1964–1968, and visiting member of technical staff, summers, 1981 and 1982. Mr. Clarke worked for the NATO ASW Research Centre, La Spezia, Italy, from 1969–1974 in their theoretical studies group. Since 1974 he has been a lecturer in electrical engineering at Imperial College, London. While at Bell Laboratories he worked on the theory of mobile radio and of propagation of random optical fields, and on the design of antireflection coatings for semiconductor laser diodes.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Equivalent Queueing Networks and Their Use in Approximate Equilibrium Analysis

By A. KUMAR*

(Manuscript received March 14, 1983)

Most Markovian queueing networks that arise as models of stochastic congestion systems (e.g., communication networks and multiprogrammed computer systems) do not have a product form in their stationary probability distributions, and hence are not amenable to the simplicity of product-form analysis. In this paper we suggest an approach for systematically examining the validity of a class of approximation schemes that is based on the idea of equivalent networks and is used for the approximate equilibrium analysis of nonproduct-form networks. We study equivalent networks, and prove a generalization of the so-called "Norton's" Theorem for closed product-form networks in order to study and generalize the equivalent flow method for the approximate analysis of nonproduct-form queueing networks. We then present the results of a study of the approximation scheme as applied to a type of network model (called a central-server model) that arises frequently in modeling multiprogrammed computer systems. In this model the central server uses a priority discipline, so the resulting network is nonproduct form. This study demonstrates the situations under which the approximation can be expected to do well or poorly and the kinds of errors it introduces.

## I. INTRODUCTION

Mathematical modeling of stochastic systems frequently gives rise to models in a class referred to as Markovian queueing networks—specifically, queueing networks whose time evolution can be described by a discrete-state, regular Markov stochastic process. Markovian

*Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

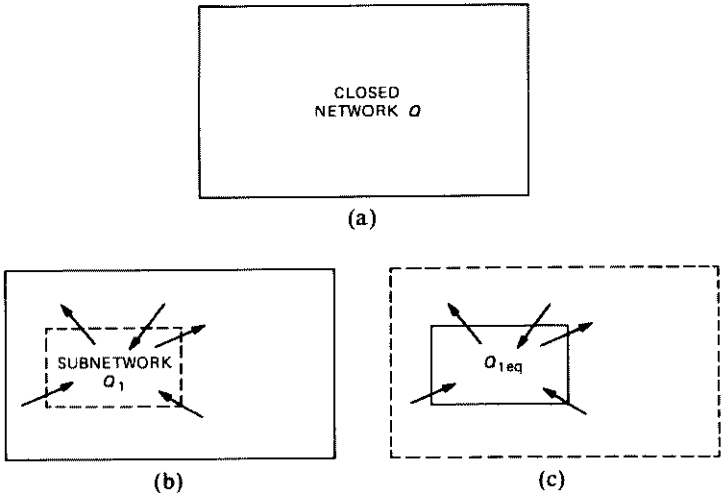



Fig. 1—Notion of an equivalent network; (a) original network, (b) with arrows indicating flows between $Q_1$ and its complements, and (c) with arrows indicating models of flows between $Q_1$ and its complement.

queueing network models, known as product-form networks, have been widely studied, owing primarily to their well-understood stochastic behavior, and the simplicity of their analysis in equilibrium. However, the class of product-form queueing network models is far from adequate for modeling many simple real-world congestion systems. The exact equilibrium analysis of nonproduct-form queueing networks is, in most cases, computationally, and often fundamentally, intractable. Much effort has, therefore, been directed towards devising approximation schemes that attempt to reconcile the conflicting requirements of modeling fidelity and the simplicity of product-form analysis. One such class of approximation schemes is based on the idea of equivalent networks. In this paper we systematically study this approximation.

By an equivalent network we mean the following (cf. Fig. 1). Consider a closed queueing network $Q$ constructed from the set of nodes M and the subnetwork $Q_1$ consisting of nodes $M_1 \subset M$. Let $Q_{1eq}$ be a network constructed from $M_1$ such that the joint equilibrium (probability) distribution of $Q_{1eq}$ is the same as the marginal joint equilibrium distribution of $Q_1$ in $Q$. The network $Q_{1eq}$ is then said to be equivalent to $Q_1$.* Clearly, to study $Q_1$ in isolation, one needs to account for the

* This notion of equivalence may appear unduly restrictive. Why not establish a more detailed stochastic equivalence? For the calculation of many performance analysis criteria, the present notion is adequate. However, it is easy to see that the equivalent networks we later identify yield equality in distribution for the entire process in equilibrium.

influence of the nodes in $M - M_1$ on the nodes in $Q_1$. When $Q$ is product form, the influence of the complementary network on $Q_1$ takes an especially simple form, and can be determined by analyzing a modified version of the complementary network in isolation! For the case where $M_1$ consists of a single node, this fact was first recognized by Chandy et al.,[1] who called the equivalent network so obtained a "Norton" equivalent, because of the similarity of this equivalence to Norton equivalence in electrical circuits.

In Section II we study equivalent networks and demonstrate the simplifications that arise for product-form networks. The development yields a generalization of Norton's Theorem to multinode subnetworks of closed product-form networks. Essentially, the same extension to the entire class of closed product-form networks has been obtained independently and concurrently by Kritzinger et al.[2] and Balsamo et al.,[3] through an approach based on verification via detailed computations from the product-form solution. Our approach is substantially different, in that it derives Norton's Theorem directly as a special case of a general result for stochastically equivalent networks. This approach is concise, conceptually and intuitively appealing, gives the result a probabilistic interpretation, and shows up clearly the role played by the product-form solution. It also seems to be the natural approach for the purposes of this study.

This generalization of Norton's Theorem motivates the following approximation scheme. Suppose now that $Q$ is a nonproduct-form network, but for the purposes of studying the subnetwork $Q_1$, we follow the equivalence procedure for product-form networks. Suppose also that in doing so we find that the version of the complementary network we have to analyze, in order to determine the latter's influence on $Q_1$, is product form. Let the equivalent network thus obtained be $\hat{Q}_{1eq}$. The approximation scheme, referred to above, approximates the equilibrium distribution of $Q_1$ with that of $\hat{Q}_{1eq}$ (i.e., approximates $Q_{1eq}$ by $\hat{Q}_{1eq}$). The effort to determine and analyze $\hat{Q}_{1eq}$ will, in general, be considerably less than the effort to exactly analyze $Q_1$ in $Q$.

This approximation scheme is an extension of one (often referred to in the literature as an equivalent flow approximation) that has been utilized by several workers, in the field of network performance analysis, with remarkably accurate results. Sauer and Chandy,[4] and Chow and Yu[5] use this idea as the basic step in iterative schemes for approximating central-server models in which the central server is not of product-form type. Schwartz[6] uses the basic scheme directly to approximately analyze a model for a multiple-access communication system. In Section III we draw upon the theoretical development in Section II to study the validity of the approximation scheme when it is applied to a simple test-bed model.

## II. EQUIVALENT NETWORKS

Consider a closed Markovian queueing network $Q$ consisting of $M$ congestion nodes. In this section we study the problem of the equilibrium analysis of a subnetwork $Q_1$ (embedded in $Q$). To simplify the discussion we shall limit our considerations to networks of First In, First Out (FIFO) nodes. It is easily recognized that the ideas in this section can be extended to apply to more general networks. In Section 2.3 we establish Theorem 1, which explicates the structure of equivalent subnetworks of the networks described in Section 2.1. By combining this result with Theorem 2, we get a generalization of Norton's Theorem.

### *2.1 Network specifications*

$Q$ is a closed queueing network consisting of $M$ FIFO nodes (indexed by $i \in \{1, \cdots, M\}$). There are $R$ classes/types of customers (indexed by $r \in \mathbf{R} = \{1, \cdots, R\}$) with $N_r$ customers in the $r$th class. The $M$ x $M$ matrix $P^{(r)} = [p_{ij}^{(r)}]$ is the routing probability matrix of type $r$ customers; customers do not change class as they move from node to node. For each $r$ in $\{1, \cdots, R\}$, $P^{(r)}$ is a stochastic matrix which, when considered as a transition probability matrix, leads to a Markov chain, on the state space $\{1, \cdots, M\}$, with a single positive, communicating class.

Throughout the following discussion, the network state process is assumed to be in equilibrium. The state of the $i$th node (denoted by $S^i$) is a finite string drawn from the set $\mathbf{R}$. Given a state vector $S^i$, $r \in \mathbf{R}$ appearing in the $k$th position in the string $S^i$ denotes that a customer of type $r$ is in the $k$th position, in FIFO order, at the node $i$. Thus, by definition, the customer in service is in the first position. $S = (S^1, \cdots, S^M)$ denotes the state of the entire network $Q$. The $i$th node is equipped with an exponential server which, when the state of the network is $S$, serves a customer of class $r$ at the rate $\nu_{ir}(S)$.

$Q_1$ is a subnetwork of $Q$, consisting of $M_1$ ($<N$) nodes (indexed by $i \in \{1, \cdots, M_1\}$). $Q_2$ is the complementary network consisting of $M_2 = M - M_1$ nodes (indexed by $i \in \{M_1 + 1, \cdots, M\}$).

Some additional notation is inevitable; this we proceed to describe in the next subsection.

### *2.2 Notation*

$\mathbf{N} = (N_1, \cdots, N_R)$ is the population vector of the network $Q$, where $N_r$ is the number of customers of class $r$, $r \in \mathbf{R}$.

Let $\mathbf{R}^* = (\cup_{n \geq 1} \{1, \cdots, R\}^n) \cup \Theta$ where $\Theta$ denotes the empty string. For any $s \in \mathbf{R}^*$, denote by $N_r(s)$ the population of class $r$ in the string $s$, and let

$$N(s) = (N_1(s), \cdots, N_r(s), \cdots, N_R(s)).$$

For $K$, a positive integer, let

$$S_{\mathbf{N}}^{K} = \{(s^1, \cdots, s^K): (\text{for every } i,\ 1 \leqslant i \leqslant K,\ s^i \in \mathbf{R}^*)$$

$$\text{and} \quad \sum_{i=1}^{K} N(s^i) = \mathbf{N}\}.$$

As stated in Section 2.1, $S = (S^1, \cdots, S^M)$ denotes the $Q$-network state. Let $S_1 = (S^1, \cdots, S^{M_1})$ denote the $Q_1$-network state and $S_2 = (S^{M_1+1}, \cdots, S^M)$ denote the $Q_2$-network state. Let $\mathbf{F}_{\mathbf{N}}^{Q}$, and $\mathbf{F}_{\mathbf{N}}^{Q_1}$ and $\mathbf{F}_{\mathbf{N}}^{Q_2}$ denote respectively the sets of feasible states, in equilibrium, of the state process of the networks $Q$, $Q_1$, and $Q_2$, respectively.

A network $Q_{1\text{eq}}$ constructed from the nodes $\{1, \cdots, M_1\}$ is said to be *equivalent* to $Q_1$ if the joint equilibrium (probability) distribution of the state processes of $Q_{1\text{eq}}$ is the same as the marginal joint equilibrium distribution of the state process of $Q_1$ in $Q$.

### *2.3 Construction of $Q_{1\text{eq}}$*

Let $\pi$: $\mathbf{F}_{\mathbf{N}}^{Q} \to (0, 1)$ be the equilibrium distribution of the state process of the network $Q$. Let (for every $(1 \leqslant i \leqslant M)(1 \leqslant r \leqslant R)$) (for every $S \in \mathbf{F}_{\mathbf{N}}^{Q}$) $\nu_{ir}(S) = \nu_{ir}(S_1)$ and (for every $S_1 \in \mathbf{F}_{\mathbf{N}}^{Q_1}$)

$$\rho_{ir}^{S_1} \triangleq \pi\{\text{A customer of type } r \text{ is in service at node}$$

$$i/Q_1 \ i \text{ is in state } S_1\}.$$

Construct a network $Q_{1\text{eq}}$ from the nodes $\{1, \cdots, M_1\}$ as follows:

1. The routing between the nodes in $Q_{1\text{eq}}$ is the same as in $Q_1$ (self loops around nodes in $Q_1$ are included in $Q_{1\text{eq}}$).

2. When the state of $Q_{1\text{eq}}$ is $S_1$, node $j (1 \leqslant j \leqslant M_1)$ receives an exogenous arrival stream of class $r$ customers with (state dependent) rate $\sum_{i=M_1+1}^{M} \rho_{ir}^{S_1} \nu_{ir}(S_1) p_{ij}^{(r)}$.

3. A customer of class $r$, after completing service at node $i$ $(1 \leqslant i \leqslant M_1)$, leaves the network $Q_{1\text{eq}}$ with probability $\sum_{j=M_1+1}^{M} p_{ij}^{(r)}$.

*Theorem 1: $Q_{1eq}$ as constructed above is equivalent to the subnetwork $Q_1$ of $Q$.*

*Proof:* The intuitive appeal of the construction is manifest. In Step 2 of the construction, for every $i(M_1 + 1 \leqslant i \leqslant M)$, $\rho_{ir}^{S_1} \nu_{ir}(S_1)$ is the conditional throughput of type $r$ customers through node $i$, when $Q_1$ is in the state $S_1$. A fraction $p_{ij}^{(r)}$ of this flow through $i$ finds its way into node $j$ of $Q_1$.

A simple detailed proof can be obtained by summing the Kolmogorov equilibrium equations for $Q$ over the set $\{S \in \mathbf{F}_{\mathbf{N}}^{Q} : S_1 \text{ fixed}\}$, and observing that the resulting equations are exactly the equilibrium equations for $Q_{1\text{eq}}$ described above (cf. Ref. 7). □

*Remarks:* But for the explosion in notation that occurs in setting up

a detailed proof, it is clear that the construction of $Q_{1eq}$ described above extends easily to networks other than those described in Section 2.1. In this work, however, we continue to restrict our attention to networks of the latter type.

We now turn to the subclass of product-form networks of the class of networks described in Section 2.1. Since we are concerned here with FIFO nodes, the service rates cannot be class dependent. We further assume that the service rates are not state dependent in any way, i.e., we now have

$$(\text{for every } (1 \leq i \leq M)(1 \leq r \leq R) \quad \text{and} \quad S \in \mathbf{F}_{\mathbf{N}}^{Q}) \nu_{ir}(S) = \nu_i.$$

Let (for every $r \in \mathbf{R}$) $\mathrm{C}^{(r)} \subset \{1, \cdots, \mathrm{M}\}$ be the subset of nodes of $Q$ that communicate under $P^{(r)}$ (i.e., in queueing-network terminology, the chain corresponding to class $r$). Let $\mathbf{R}_2 = \{r \in \mathbf{R} : \mathrm{C}^{(r)} \cap \{M_1 + 1, \cdots, M\} \neq \emptyset\}$ be the set of customer types that visit $Q_2$. Let $\|\mathbf{R}_2\| = R_2$, $\|\mathbf{R} - \mathbf{R}_2\| = R_1$ (where $\| \ \|$ denotes set cardinality), and reindex $\mathbf{R}$ so that the elements of $\mathbf{R}_2$ receive the highest indices. Let $\mathbf{N}^2 = (N_{R_1+1}, \cdots, N_R)$ and if $s$ is a string in $\mathbf{R}^*$, let $N^2(s) = (N_{R_1+1}(s), \cdots, N_R(s))$, i.e., $N^2(s)$ is the population vector, of the string $s$, restricted to the classes in $\mathbf{R}_2$.

For every $\mathbf{N}' = (N'_{R_1+1}, \cdots, \mathrm{N}'_R) \leq \mathbf{N}^2$, consider the network $Q'_2(\mathbf{N}')$ obtained from $Q$ by replacing all servers in $Q_1$ with infinite speed servers (i.e., by short-circuiting the nodes in $Q_1$), and placing $\mathbf{N}'$ customers in the resulting network. Let $\pi_{\mathbf{N}'}$ be the equilibrium distribution of the state process of the network $Q'_2(\mathbf{N}')$. Define for every $(M_1 + 1 \leq i \leq M)$, $r \in \mathbf{R}_2$,

$$\xi_{ir}^{\mathbf{N}'} \triangleq \pi_{\mathbf{N}'}\{\text{A customer of type } r \text{ is in service at node } i \text{ in } Q'_2(\mathbf{N}')\}.$$

### 2.4 *The product-form case*

*Theorem 2: If $Q$ is a product-form network then*

$$\text{for every } (M_1 + 1 \leq i \leq M)(r \in \mathbf{R}_2) \text{ and every } S_1 \in \mathbf{F}_{\mathbf{N}}^{Q_1}$$
$$\rho_{ir}^{S_1} = \xi_{ir}^{\mathbf{N}^2 - N^2(S_1)} \qquad (\rho_{ir}^{S_1} \text{ as defined earlier}).$$

(Note: it is obvious that (for every $(M_1 + 1 \leq i \leq M)$, $r \notin \mathbf{R}_2$ and $S_1 \in \mathbf{F}_{\mathbf{N}}^{Q_1}$) $\rho_{ir}^{S_1} = 0$.)

*Proof:* The proof utilizes a simple lemma and is outlined in the appendix. □

*Remarks:* Theorem 2, when combined with Theorem 1, yields a generalization of Norton's Theorem[1] to multinode subnetworks. Even though the previous development is specific to the class of networks described in Section 2.1, it is clear that the same approach can be used to extend Norton's Theorem to the entire class of closed product-form networks. The product-form solution continues to play the same role

as it does in Theorem 2, i.e., it allows the rates of the external arrival streams in $Q_{1eq}$ to be computed from an analysis of $Q_2'$ for all possible customer populations in $Q_2'$.

## III. AN APPROXIMATION SCHEME

In an IBM Research Report, Chow and Yu[5] suggest a somewhat ad hoc, iterative approximation scheme for a class of central-server models, with a priority discipline at the central server. As mentioned earlier, the scheme relies on an inexact application of Norton's Theorem to such networks. In Section I we described a natural generalization of this so-called equivalent flow approximation scheme to more general nonproduct-form networks. In this section, we present the results of a detailed study of the application of this approximation to a simple, test-bed, central-server network.

### *3.1 The test-bed model*

Consider the two-node network $Q$ shown in Fig. 2. There are two customer classes, namely 1 and 2, with $N_1$ and $N_2$ customers, respectively (i.e., $\mathbf{N} = (N_1,N_2)$). At node 1, the customers of class 1 (high-priority) have preemptive priority over class 2 (low-priority) customers; after being preempted by a class 1 customer, when a class 2 customer reaches the service station again, it resumes service where it left off; class 1 and 2 customers have exponential service times with rates $\nu_{11}$ and $\nu_{12}$, respectively. Such a service discipline is commonly referred to as a *preemptive resume* discipline. At node 2, there is no priority; customers are served in the order in which they arrive (irrespective of class), at the class independent exponential service rate $\nu_2$. Customers alternately seek service at nodes 1 and 2 and stay in the network forever. This model belongs to a class of central-server networks that arise as models of computer systems.

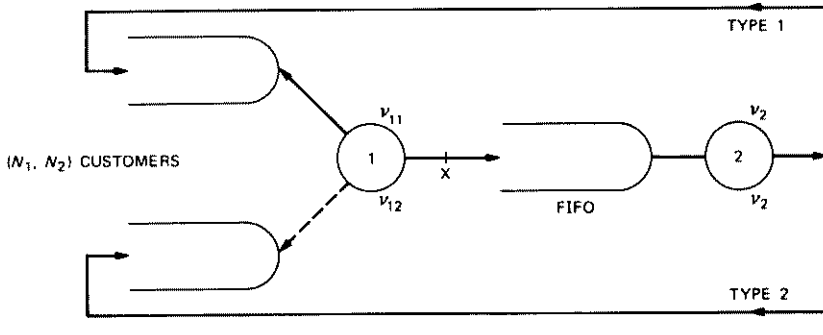



Fig. 2—$Q$.

### 3.2 *Approximating the test-bed network*

The network described in Section 3.1 is nonproduct form because of the preemptive resume discipline at node 1. In order to approximate the equilibrium behavior of node 1, we first increase the service rates at node 1 to infinity, thus effectively short circuiting the node. Denote the resulting network by $Q_2'$ (Fig. 3). Then for each $(k_1,k_2) \leq (N_1,N_2)$ analyze $Q_2'$ with $k_1$ and $k_2$ customers of types 1 and 2, respectively, in the network. Let (cf. Thm. 2) (for every $(k_1,k_2) \leq (N_1,N_2)$) (for every $r \in \{1, 2\}$) $\xi_{2r}^{(k_1,k_2)}$ = Prob {A customer of type $r$ is in service at node 2 when $(k_1,k_2)$ customers are in $Q_2'$}.

This probability will not depend on the sequence in which the $(k_1,k_2)$ customers are placed in $Q_2'$. Since the service rate at node 2 is class independent, it is clear that, in equilibrium, all possible states, for any arrangement of the customers, are equally likely. From this we can directly conclude that

$$\text{(for every } (0, 0) < (k_1,k_2) \leq (N_1,N_2)\text{) (for every } r \in \{1, 2\}\text{)},$$

$$\xi_{2r}^{(k_1,k_2)} = \frac{k_r}{k_1 + k_2}.$$

Now consider the open network $\hat{Q}_{1\text{eq}}$ consisting of the node 1 in isolation. The service rates and discipline remain the same as in $Q$. When there are $n_1$ customers of type 1 and $n_2$ customers of type 2 in $\hat{Q}_{1\text{eq}}$ then customers of type $r$ ($\in \{1, 2\}$) enter the network at the rate $\hat{\lambda}_r^{(n_1,n_2)}$ where

$$\text{(for every } (n_1,n_2) \leq (N_1,N_2)\text{)} \hat{\lambda}_r^{(n_1,n_2)} = \xi_{2r}^{\mathbf{N}-(n_1,n_2)} \nu_2.$$

When a customer finishes service in $\hat{Q}_{1\text{eq}}$, it leaves the network (Fig. 4).

The evolution of the network $\hat{Q}_{1eq}$ can be described by a regular Markov process on the state space $\{(n_1,n_2) : (n_1,n_2) \leq (N_1,N_2)\}$. The idea is to approximate the equilibrium distribution of customers at node 1 in $Q$ with the equilibrium distribution of customers in $\hat{Q}_{1\text{eq}}$.

At first glance, the approximation technique described above may seem rather ad hoc. However, we can draw upon the development in

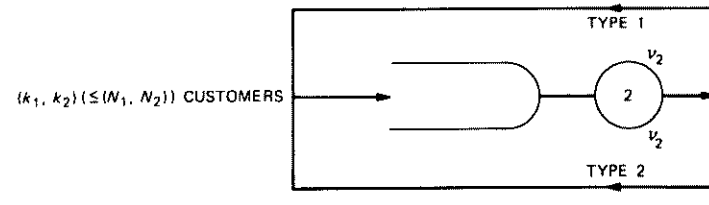



Fig. 3—$Q_2$.

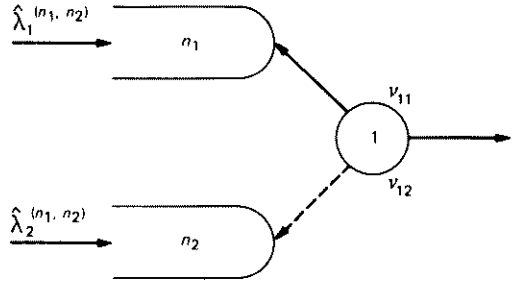



Fig. 4—$\hat{Q}_{1\,eq}$.

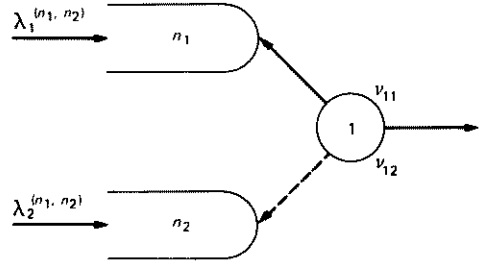



Fig. 5—$Q_{1\,eq}$.

Section II to understand the inner workings of the test-bed model, and to show that, at least in principle, the approximation scheme is not altogether unreasonable.

It is clear that we can think of node 1, in the test-bed network $Q$, as comprising two FIFO nodes with service rates that depend just on the joint state of these two nodes. Consider the subnetwork $Q_1$ of $Q$ consisting only of node 1. Theorem 1 can now be invoked to determine the exact equivalent network $Q_{1eq}$. Let

(for every $(n_1,n_2) \leqslant (N_1,N_2)$) (for every $r \in \{1, 2\}$),

$$\rho_{2r}^{(n_1,n_2)} = \pi\{\text{A customer of type } r \text{ is in service at node } 2/(n_1,n_2) \text{ customers in } Q_1\}.$$

$Q_{1eq}$ is then an open network consisting of node 1. When there are $n_1$ customers of type 1 and $n_2$ customers of type 2 in $Q_{1eq}$, then customers of type $r$ ($\in \{1, 2\}$) enter the network at the rate $\lambda_r^{(n_1,n_2)}$ where

$$(\text{for every } (n_1,n_2) \leqslant (N_1,N_2))\lambda_r^{(n_1,n_2)} = \rho_{2r}^{(n_1,n_2)}\nu_2.$$

When a customer finishes service in $Q_{1eq}$ it leaves the network (Fig. 5).

The equilibrium distribution of customers in $Q_{1eq}$ is exactly the same

as the equilibrium distribution of customers in $Q_1$. Observe, though, that the form of $Q_{1eq}$ is the same as that of $\hat{Q}_{1eq}$, the difference lying in the state-dependent input rates. It is in this sense that the approximation scheme is reasonable. The idea now is to compare the exact state-dependent input rates, $\rho_{2r}^{(n_1,n_2)}\nu_2$, with the approximate state-dependent rates,

$$\xi_{2r}^{\mathbf{N}-(n_1,n_2)}\nu_2 = \left(\frac{N_r - n_r}{N_1 - n_1 + N_2 - n_2}\,\nu_2\right),$$

i.e., to compare $\rho_{2r}^{(n_1,n_2)}$ with

$$\frac{N_r - n_r}{N_1 - n_1 + N_2 - n_2}$$

for all $(n_1,n_2) \lneq (N_1,N_2)$ and for $r \in \{1, 2\}$.

### 3.3 *Qualitative evaluation of the approximation*

In this section, we present a qualitative evaluation of the approximation scheme as applied to the test-bed model.

Observe that if the service rates for the two FIFO queues comprising node 1, in $Q_1$, were not state dependent (in the priority scheme they are state dependent), then $Q_1$ would, in fact, be a product-form network. Theorem 2 would then lead us to conclude that $\hat{Q}_{1eq}$ and $Q_{1eq}$ were the same. Consider what happens if, in $Q_1$, $\nu_{11}$ is allowed to go to infinity. Then, effectively, the high-priority customers do not interfere with the low-priority customers at node 1. With $\nu_{11} = \infty$, the network becomes the one shown in Fig. 6, which is a product-form network. Thus according to our observation above, for values of $\nu_{11}$ that are large, compared to $\nu_{12}$ and $\nu_2$, the approximation can be expected to yield very good results.

In order to discover the situations in which the approximation can be expected to behave poorly, one needs to understand what aspects

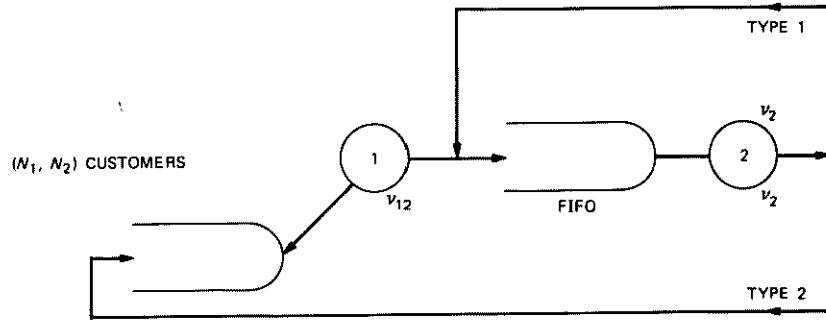



Fig. 6—"Lim" $Q$ '$11^{-\infty}$.

of the exact network the approximation fails to capture. If $Q_1$ were a product-form network, then, given that $(k_1,k_2)(\leqslant(N_1,N_2))$ customers were in node 2, all arrangements of customers within the node would be equally likely. As it stands, however, at node 1, priority 1 customers can preempt customers of priority 2. This suggests that given $(k_1,k_2)(\leqslant N_1,N_2)$ customers in node 2, some arrangements of customers would be more likely than others. In fact, we conjecture that priority 1 customers are more likely to be ahead of priority 2 customers, leading to the (conjectured) conclusion that

$$(\text{for every } (0,0) < (k_1,k_2) \leqslant (N_1,N_2))\rho_{21}^{\mathbf{N}-(k_1,k_2)} \geq \frac{k_1}{k_1+k_2},$$

and

$$\rho_{22}^{\mathbf{N}-(k_1,k_2)} \leqslant \frac{k_2}{k_1+k_2}.$$

Thus $\hat{Q}_{1eq}$ uses smaller (resp. larger) state-dependent input rates for type 1 (resp. type 2) customers than the exact equivalent $Q_{1eq}$. This idea is suggestive, but it is difficult to draw any immediate conclusions from this conjecture as to the relationship between exact and approximate performance measures of the network.

Another approach to discovering the direction in which the approximation can be expected to err is to observe that if node 2 in $Q$ is replaced by a processor-sharing node, with class-independent service rate $\nu_2$, then $\hat{Q}_{1eq}$ becomes the exact equivalent of $Q_1$ (cf. Fig. 7). (This follows because when node 2 is processor sharing, if $(k_1,k_2)(\leqslant(N_1,N_2))$ customers are present at node 2, then the rate of flow of class $r(\in\{1,2\})$ customers into node 1 is $(k_r/(k_1+k_2))\nu_2$.) To fix ideas consider the case $N_1 = n(\geqslant 1)$ and $N_2 = 1$. The throughput of the class 2 customer is simply the reciprocal of the mean successive passage times of the (single) class 2 customer through the point X (cf. Figs. 2 and

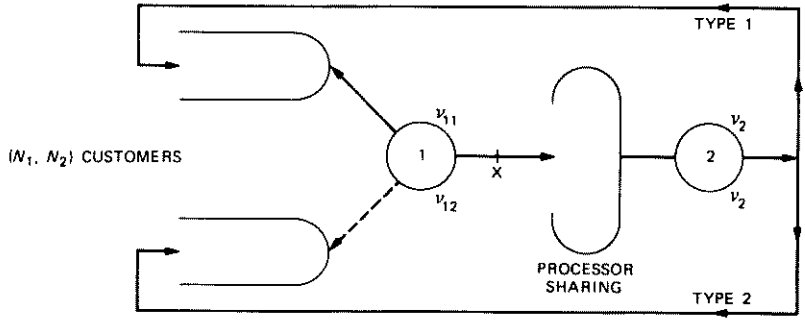



Fig. 7—$Q$ with node 2 processor sharing.

7). In either network, when the class 2 customer crosses the point X to enter node 2, it finds all the class 1 customers receiving service at this node. In the original network, since node 2 is FIFO, the class 2 customer will have to wait for full service completion of the $n$ class 1 customers before it can leave node 2 (and subsequently, at some future time instant, cycle back through X). Thus, the mean sojourn time of the type 2 customer, in node 2 of the original network, is $(n + 1)/\nu_2$. However, if node 2 is processor sharing, then on entering node 2, the customer of type 2 starts receiving service immediately at the rate $\nu_2/(n + 1)$, and continues to receive service at a rate $\nu_2/(k + 1)$ $(0 \leq k \leq n)$ until it finally leaves. Thus, in this case, the sojourn time of the class 2 customer at node 2 is stochastically dominated by an exponentially distributed random variable with mean $n + 1/\nu_2$, and hence has a mean smaller than $(n + 1)/\nu_2$. We further expect, intuitively, that, after completing service at node 2, when the type 2 customer returns to node 1, it expects to find more type 1 customers at node 1 when node 2 is FIFO than when it is processor sharing. Given that the type 2 customer finds $k(0 \leq k \leq n)$ type 1 customers on its arrival at node 1, its sojourn time at node 1 does not depend on whether node 2 is FIFO or processor sharing, and increases with increasing $k$. Thus, we expect that the mean sojourn time of the type 2 customer at node 1 will be larger if node 2 is FIFO than when it is processor sharing. The conclusion is that the mean passage time of the type 2 customer, through the point X, is larger in the original network than in the approximating network.

To see the magnitude of the error this effect could cause, let $N_1 = 1$ and allow $\nu_{12} \to \infty$, $\nu_2 \to \infty$, and $\nu_2/\nu_{12} \to 0$. Under these assumptions, in the original network, the class 2 customer will be blocked once (and only once) at node 1 each time it cycles through the point X. The average time it spends in the blocked condition is $1/\nu_{11}$. The rest of the time in each cycle tends to 0. Hence, the mean response time for the class 2 customer in the original network is $1/\nu_{11}$. If node 2 is replaced by a processor sharing node, then, each time the class 2 customer cycles through X, it is blocked once (and only once) with probability 1/2. Hence, the mean response time for the class 2 customer in the approximating network is $1/2\nu_{11}$, thus yielding an error of 100 percent.

We do not yet have a simple but rigorous argument that would allow us to say conclusively that the approximation yields higher throughputs for low-priority customers. However, the arguments presented above do make the conclusion plausible.

### *3.4 Numerical examples*

To examine how the approximation works with specific examples,

we wrote a FORTRAN program to solve the equilibrium equations for $\hat{Q}_{1eq}$ using a simple recursive technique.[8] The program was somewhat more general, in that it could accept arbitrary state-dependent input rates and output rates. Thus, the same program could be used to solve the network exactly, if it were given the exact values of $\rho_{12}^{(n_1,n_2)}$ and $\rho_{22}^{(n_1,n_2)}$ for the various feasible $(n_1,n_2)$.

It is not hard to calculate exactly the probabilities $\rho_{21}^{(n_1,n_2)}$ and $\rho_{22}^{(n_1,n_2)}$ for some simple cases. Of course for $(n_1,n_2) \neq (N_1,N_2)$, $\rho_{22}^{(n_1,n_2)} = 1 - \rho_{21}^{(n_1,n_2)}$. Consider, for the purpose of illustration, the case $N_1 = 1$, $N_2 = 1$. Note that the state of the network $Q$ is completely described by the state of node 2. The epochs of entry into the state $S^2 = (12)$ are renewal epochs. The next state is, inevitably, $S^2 = (2)$. The next state is $S^2 = (21)$ with probability $\nu_{11}/(\nu_{11} + \nu_2)$ and $S^2 = (\Theta)$ with probability $\nu_2/(\nu_{11} + \nu_2)$. Because of the preemptive discipline, the next state to be entered in the set $\{(12), (21)\}$ will be $S^2 = (12)$, thus completing a renewal cycle. Since the expected holding time in each state in $\{(12), (21)\}$ is $1/\nu_2$, therefore

$$\rho_{21}^{(0,0)} = \frac{\dfrac{1}{\nu_2}}{\dfrac{1}{\nu_2} + \dfrac{\nu_{11}}{\nu_{11} + \nu_2} \cdot \dfrac{1}{\nu_2}} = \frac{1}{1 + \dfrac{\nu_{11}}{\nu_{11} + \nu_2}},$$

and, of course,

$$\rho_{21}^{(1,0)} = 0 \quad \text{and} \quad \rho_{21}^{(0,1)} = 1.$$

In Table I we list the exact expressions for $\rho_{21}^{(n_1,n_2)}$, for all $(n_1,n_2) \leq (N_1,N_2)$, for some values of $(N_1,N_2)$. These were computed in the same fashion as in the above example.

In Tables II(a), (b), and (c), we give several numerical examples of exact and approximate solutions of the test-bed network. The exact solutions and the approximate solutions were obtained using the FORTRAN program described above. The program yields the equilibrium joint-probability distribution of queue lengths at node 1. In Tables II(a), (b), and (c), we display these joint probabilities and the node 1 utilizations.

The following observations are immediate and summarize our conclusions regarding the performance of the approximation scheme when applied to the test-bed network.

1. The numerical computations support our earlier observations that if $\nu_{11}$ is large, then the approximation can be expected to yield excellent results [cf. case (1) in each of Tables II(a), (b), and (c)].

2. The low-priority utilizations are consistently higher, again supporting our earlier observations regarding the direction in which the approximation can be expected to err.

Table I—Exact expressions for $\rho_{21}^{(n_1,n_2)}$ in the test-bed network $Q$. (cf. Thm. 1 and Fig. 2)

| $N_1$ | $N_2$ | $(n_1,n_2)$ | $\rho_{21}^{(n_1,n_2)*}$ | Comparison with $\hat{\rho}_{21}^{(\cdot,\cdot)}$ |
|---|---|---|---|---|
| 1 | 1 | (1,0) | $0$ | |
| | | (0,1) | $1$ | |
| | | (0,0) | $\dfrac{1}{1+\dfrac{\nu_{11}}{\nu_{11}+\nu_2}}$ | $\geq\frac{1}{2}$ |
| 2 | 1 | (2,0) | $0$ | |
| | | (0,1), (1,1) | $1$ | |
| | | (1,0) | $\dfrac{1}{1+\dfrac{\nu_{11}}{\nu_{11}+\nu_2}+\dfrac{\nu_2}{\nu_{11}+\nu_2}\cdot\dfrac{\nu_{11}}{\nu_{11}+\nu_2}}$ | $\geq\frac{1}{2}$ |
| | | (0,0) | $\dfrac{1+\dfrac{\nu_{11}}{\nu_{11}+\nu_2}}{1+\dfrac{\nu_{11}}{\nu_{11}+\nu_2}+\left(\dfrac{\nu_{11}}{\nu_{11}+\nu_2}\right)^2+\dfrac{\nu_2}{\nu_{11}+\nu_2}\cdot\left(\dfrac{\nu_{11}}{\nu_{11}+\nu_2}\right)^2}$ | $\geq\frac{2}{3}$ |
| 1 | 2 | (1,0), (1,1) | $0$ | |
| | | (0,2) | $1$ | |
| | | (0,1) | $\dfrac{1}{1+\dfrac{\nu_2}{\nu_{12}+\nu_2}\cdot\dfrac{\nu_{11}}{\nu_{11}+\nu_2}+\dfrac{\nu_{12}}{\nu_{12}+\nu_2}\cdot\left[\dfrac{\nu_{11}}{\nu_{11}+\nu_2}+\dfrac{\nu_2}{\nu_{11}+\nu_2}\cdot\dfrac{\nu_{11}}{\nu_{11}+\nu_2}\right]}$ | $\geq\frac{1}{2}$ |
| | | (0,0) | $\dfrac{1}{1+\dfrac{\nu_{11}}{\nu_{11}+\nu_2}+\dfrac{\nu_{11}}{\nu_{11}+\nu_2}\cdot\dfrac{\nu_{12}}{\nu_{12}+\nu_2}\left[1+\dfrac{\nu_2}{\nu_{12}}+\dfrac{\nu_2}{\nu_2+\nu_{11}}\right]}$ | $\geq\frac{1}{3}$ |

*$(\rho_{22}^{(n_1,n_2)} = 1 - \rho_{21}^{(n_1,n_2)}$ if $(n_1,n_2) \neq (N_1,N_2)$; $\rho_{21}^{(N_1,N_2)} = \rho_{22}^{(N_1,N_2)} = 0)$.

3. When $\nu_{11}$, $\nu_{12}$ and $\nu_2$ are comparable, then the approximation yields good results with errors in the utilizations in the neighborhood of 10 percent.

4. Considerable errors in the low-priority utilizations can arise, however. Witness case 3 in each of the Tables II(a), (b), and (c). With a very large low-priority service rate at node 1, the approximate low-priority utilization suffers from an error of 20 to 50 percent.

5. For the range of examples studied, the equilibrium probabilities

Table II—Numerical comparisons of exact and approximate solutions of the test-bed network

| Case No. | $\nu_{11}$ | $\nu_{12}$ | $\nu_2$ | Equilibrium State Probabilities at Node 1 | | | Node 1 Utilizations | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | State $(n_1,n_2)$ | Exact Probability | Approximate* Probability | Class | Exact Utilization | Approximate* Utilization |
| | | | | (a) | $(N_1 = 1, N_2 = 1)$ | | | | |
| 1 | 10 | 1 | 1 | (0,0) | 0.614 | 0.606 | | | |
| | | | | (1,0) | 0.029 | 0.028 | 1 | 0.064 | 0.063 |
| | | | | (0,1) | 0.322 | 0.331 | 2 | 0.322 | 0.331 |
| | | | | (1,1) | 0.035 | 0.036 | | | |
| 2 | 1 | .5 | 1 | (0,0) | 0.231 | 0.222 | | | |
| | | | | (1,0) | 0.077 | 0.056 | 1 | 0.462 | 0.444 |
| | | | | (0,1) | 0.308 | 0.333 | 2 | 0.308 | 0.333 |
| | | | | (1,1) | 0.385 | 0.389 | | | |
| 3 | 1 | 100 | 2 | (0,0) | 0.393 | 0.488 | | | |
| | | | | (1,0) | 0.196 | 0.163 | 1 | 0.601 | 0.504 |
| | | | | (0,1) | 0.0059 | 0.0081 | 2 | 0.0059 | 0.0081 |
| | | | | (1,1) | 0.405 | 0.341 | | | |
| | | | | (b) | $(N_1 = 2, N_2 = 1)$ | | | | |
| 1 | 10 | 1 | 1 | (0,0) | 0.681 | 0.676 | | | |
| | | | | (1,0) | 0.044 | 0.043 | | | |
| | | | | (2,0) | 0.002 | 0.002 | 1 | 0.0754 | 0.0748 |
| | | | | (0,1) | 0.244 | 0.249 | 2 | 0.244 | 0.249 |
| | | | | (1,1) | 0.027 | 0.027 | | | |
| | | | | (2,1) | 0.003 | 0.003 | | | |
| 2 | 1 | .5 | 1 | (0,0) | 0.179 | 0.171 | | | |
| | | | | (1,0) | 0.083 | 0.065 | | | |
| | | | | (2,0) | 0.024 | 0.016 | 1 | 0.631 | 0.618 |
| | | | | (0,1) | 0.191 | 0.211 | 2 | 0.191 | 0.211 |
| | | | | (1,1) | 0.250 | 0.260 | | | |
| | | | | (2,1) | 0.274 | 0.276 | | | |
| 3 | 1 | 100 | 2 | (0,0) | 0.170 | 0.217 | | | |
| | | | | (1,0) | 0.116 | 0.109 | | | |
| | | | | (2,0) | 0.050 | 0.036 | 1 | 0.827 | 0.780 |
| | | | | (0,1) | 0.0022 | 0.0033 | 2 | 0.0022 | 0.0033 |
| | | | | (1,1) | 0.187 | 0.188 | | | |
| | | | | (2,1) | 0.474 | 0.447 | | | |
| | | | | (c) | $(N_1 = 1, N_2 = 2)$ | | | | |
| 1 | 10 | 1 | 1 | (0,0) | 0.464 | 0.457 | | | |
| | | | | (1,0) | 0.015 | 0.014 | | | |
| | | | | (0,1) | 0.316 | 0.319 | 1 | 0.0495 | 0.0487 |
| | | | | (1,1) | 0.016 | 0.016 | 2 | 0.487 | 0.494 |
| | | | | (0,2) | 0.170 | 0.175 | | | |
| | | | | (1,2) | 0.019 | 0.019 | | | |
| 2 | 1 | .5 | 1 | (0,0) | 0.116 | 0.109 | | | |
| | | | | (1,0) | 0.028 | 0.108 | | | |
| | | | | (0,1) | 0.177 | 0.182 | 1 | 0.437 | 0.418 |
| | | | | (1,1) | 0.070 | 0.055 | 2 | 0.447 | 0.473 |
| | | | | (0,2) | 0.270 | 0.291 | | | |
| | | | | (1,2) | 0.340 | 0.346 | | | |
| 3 | 1 | 100 | 2 | (0,0) | 0.473 | 0.584 | | | |
| | | | | (1,0) | 0.167 | 0.130 | | | |
| | | | | (0,1) | 0.008 | 0.010 | 1 | 0.517 | 0.404 |
| | | | | (1,1) | 0.115 | 0.090 | 2 | 0.0101 | 0.0123 |
| | | | | (0,2) | 0.002 | 0.002 | | | |
| | | | | (1,2) | 0.235 | 0.184 | | | |

*Based on Norton's Equivalent.

are never drastically wrong, and follow trends similar to the exact values.

The test-bed model is hard to analyze exactly for population sizes larger than the ones considered. We have run detailed simulations of the test-bed model for larger population sizes, and the results of these simulations continue to support the qualitative observations we have made above.

## IV. CONCLUSIONS

We have demonstrated an approach for systematically analyzing the equivalent flow approximation. Our investigations have (1) revealed the conceptual basis for the approximation scheme, and (2) led to an understanding of the reasons for, and directions of, the errors that such an approximation scheme could introduce when applied to a class of prioritized central-server models. The approximation as described in the paper is of more general applicability, and much work remains to be done to discover its validity (accuracy and computational tractability) for more complicated, nonproduct-form networks. Our work, we think, provides the theoretical understanding and motivation for pursuing more detailed investigations.

## V. ACKNOWLEDGMENTS

We are indebted to J. S. Kaufman for introducing us to the problem. We are grateful to him and to B. T. Doshi, B. Melamed, and B. Sengupta for making themselves available for several helpful discussions.

## REFERENCES

1. K. M. Chandy, U. Herzog, L. Woo, "Parametric Analysis of Queueing Networks," IBM J. Res. and Develop. *19*, No. 1 (January 1975), pp. 36–42.
2. P. S. Kritzinger, S. Van Wyck, A. E. Krzesinski, "A Generalization of Norton's Theorem for Multiclass Queueing Networks," Performance Evaluation, *2* (July 1982), pp. 98–107.
3. S. Balsamo and G. Iazeolla, "An Extension of Norton's Theorem for Queueing Networks," IEEE Trans. Software Eng., *SE-8*, No. 4 (July 1982), pp. 298–305.
4. C. H. Sauer and K. M. Chandy, "Approximate Analysis of Central Server Models," IBM J. Res. Develop., *19*, No. 3 (May 1975), pp. 301–13.
5. W. Chow and P. S. Yu, "An Approximation Technique for Central Server Queueing Models with a Priority Dispatching Rule," IBM Res. Rpt. (1980), RC8163 (No. 35479).
6. M. Schwartz, "Performance Analysis of the SNA Virtual Route Pacing Control," IEEE Trans. Commun., *COM-30*, No. 1, Part II (January 1982), pp. 172–84.
7. A. Kumar, unpublished work.
8. J. S. Kaufman, unpublished work.
9. F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," J. ACM, *22*, No. 2 (April 1975), pp. 248–60.

## APPENDIX

*Proof of Theorem 2:* Index the nodes of $Q_2'$ in the same order in which they were indexed in $Q$. We need the following lemma.

*Lemma*: *Let* $T^{(r)}$ *denote the class* $r$ *routing probability matrix for the network* $Q_2'$. *Partition* $P^{(r)}$ *as follows*:

$$P^{(r)} = \begin{array}{c|c|c} & \overbrace{\phantom{M_1}}^{M_1} & \overbrace{\phantom{M_2}}^{M_2} \\ \hline M_1 & P_{11}^{(r)} & P_{12}^{(r)} \\ \hline M_2 & P_{21}^{(r)} & P_{22}^{(r)} \end{array}.$$

*Then*

(1) $r \in \mathbf{R}_2 \Rightarrow T^{(r)} = P_{22}^{(r)} + P_{21}^{(r)}[I - P_{11}^{(r)}]^{-1}P_{12}^{(r)}$

(2) *If* $\Lambda^{(r)}$ *solves* $\Lambda^{(r)}P^{(r)} = \Lambda^{(r)}$ *then, partitioning* $\Lambda^{(r)}$ *as*

$$\Lambda^{(r)} = \left[\begin{array}{c:c} \Lambda_1^{(r)} & \Lambda_2^{(r)} \\ \hline \overbrace{\phantom{M_1}}^{M_1} & \overbrace{\phantom{M_2}}^{M_2} \end{array}\right]$$

(*a*) *if* $r \notin \mathbf{R}_2$ *then* $\Lambda_2^{(r)} = 0$

(*b*) *if* $r \in \mathbf{R}_2$ *then* $\Lambda_2^{(r)} = \Lambda_2^{(r)}T^{(r)}$.

*Proof of Lemma:* Conclusion 1 follows readily from the fact that, for each $r \in \mathbf{R}$, $P^{(r)}$ is the transition probability matrix of a finite Markov chain with a single, positive communication class that has a nonempty intersection with $\{M_1 + 1, \cdots, M\}$. For details see Ref. 7.

Conclusion 2 follows directly from Conclusion 1. □

Returning to the proof of Theorem 2, we let $\pi$: $\mathbf{F}_N^Q \to (0, 1)$ be the equilibrium distribution of the state process of network $Q$.* It is now well known (cf. Ref. 9) that $\pi(\cdot)$ is of the form

$$\pi(S) = \frac{1}{G}\prod_{i=1}^{M} f_i(S^i),$$

where $G$ is a normalization constant and, for each $i \in \{1, \cdots, M\}$, $f_i$ depends only $N(S^i)$, $\nu_i$ and (for every $r$, $(1 \leqslant r \leqslant R)$)$\lambda_i^{(r)}$, where $\Lambda^{(r)} = (\lambda_1^{(r)}, \cdots, \lambda_M^{(r)})$ is any solution of $\Lambda^{(r)}P^{(r)} = \Lambda^{(r)}$.

Hence, (for every $S_1 \in \mathbf{F}_N^{Q_1}$) (for every $i$, $r$, $(M_1 + 1 \leqslant i \leqslant M)$, $r \in \mathbf{R}_2$)

$$\rho_{ir}^{S_1} = \frac{\sum_{\{S:S\in\mathbf{F}_N^Q,(S^1,\cdots,S^{M_1})=S_1,S^i(1)=r\}}\prod_{j=1}^{M} f_j(S^j)}{\sum_{\{S:S\in\mathbf{F}_N^Q,(S^1,\cdots,S^{M_1})=S_1\}}\prod_{j=1}^{M} f_j(S^j)}$$

$$= \frac{\sum_{\{S_2:S_2\in\mathbf{F}_N^{Q_2}\cap S_{M_2-M_2(S_1)}^{M_2},S_2^i(1)=r\}}\prod_{j=M_1+1}^{M} f_j(S_2^j)}{\sum_{\{S_2:S_2\in\mathbf{F}_N^{Q_2}\cap S_{M_2-M_2(S_1)}^{M_2}\}}\prod_{j=M_1+1}^{M} f_j(S_2^j)},$$

*For notation see Section 2.2.

(where $\mathbf{F}_{\mathbf{N}}^{Q_2} \cap S_{\mathbf{N}^2-N^2(S_1)}^{M_2}$ is the set of feasible states of $Q_2$ when

$$\mathbf{N}^2 - N^2(S_1) \text{ customers are in } Q_1),$$

which, using the above lemma and the fact that the equilibrium distribution of the $Q_2'$ network state process is still product form,

$$= \xi_{ir}^{\mathbf{N}^2-N^2(S_1)}.$$

*Remarks:* Some care is needed in asserting the last equality in the case where there are classes $r_1$ and $r_2$, such that the submatrices of the communicating classes under $T^{(r_1)}$ and $T^{(r_2)}$ are the same permutation matrices (i.e., members of classes $r_1$ and $r_2$ cannot overtake each other). In this case the equality follows because for each $\mathbf{N}'$, $\xi_{ir_1}^{\mathbf{N}'}$ and $\xi_{ir_2}^{\mathbf{N}'}$ are independent of the order in which members of these classes circulate in the network $Q_2'$. □

## AUTHOR

**Anurag Kumar,** B.Tech (Electrical Engineering), 1977, Indian Institute of Technology, Kanpur, India; Ph.D. (Major: Electrical Engineering, Minor: Operations Research/Math), 1981, Cornell University; Bell Laboratories, 1981—. Mr. Kumar joined the Performance Analysis Department in December 1981, and has since been involved in two projects: a study of a class of approximation schemes for queueing-network analysis; and the identification, analysis, and evaluation of possibilities for giving priority to critical user traffic in the Public Switched Network.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# A Model for Special-Service Circuit Activity

By D. R. SMITH*

(Manuscript received April 20, 1983)

We describe a model for special-service circuit activity to assist in forecasting, provisioning, and "churn" studies. We assume that customers order a random number of circuits for an exponentially distributed period of time and that the rate of new connect orders grows exponentially with time. These assumptions yield simple formulae giving the means and variances of the number of active circuits at a future time and the total number of connected and disconnected circuits during a future period. Distributions of these variables can, in principle, also be computed. There are three important parameters characterizing the model: growth rate, disconnect rate, and batchiness; we describe their physical meaning and discuss methods to estimate them. This document describes the analytical portion of an effort to develop a model based on the physics of special-service circuit activity.

## I. INTRODUCTION

The purpose of this paper is to describe a model for special-service circuit activity to assist in forecasting, provisioning, and "churn" studies, which can be summarized by a few parameters that have a physical interpretation. The calibration and measurement of the fit of this model to data in a New Jersey Bell database is being pursued simultaneously and will be reported elsewhere.

The model treated here is derived from a priori consideration of the physical behavior of customers. It is based on the assumption that the number of active circuits, although growing, is in some sense in

*Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

equilibrium as well; that is, certain characteristics of the system are not changing. This is to be contrasted with a model proposed by Nucho in which transient analysis is fundamental.[1] The primary difference between these models is that the demand rate for new circuits is a function of the number of active circuits in the Nucho model, whereas it is considered to be an exogenous variable here. In the Nucho model, the variance to mean ratio of the number of active circuits increases indefinitely with time (since fluctuations tend to feed on themselves); in the model considered here this ratio remains constant. Another difference between the models is that the model described here allows an order to be for more than one circuit.

Here, we assume that (1) the arrivals of special-service circuit orders are given by a nonhomogeneous Poisson process with exponentially growing intensity, (2) each order is for a random number of circuits (a batch) with arbitrary distribution, and (3) the lifetime of an order is an exponentially distributed random variable, during which time the number of held circuits per order remains constant. Note that the last assumption implies that an order lifetime and a circuit lifetime have the same distribution.

We use three important parameters in special-services modeling, each with its own physical interpretation. These parameters may be described as growth rate, disconnect rate (per circuit), and batchiness.

The growth rate summarizes the rate at which the mean number of active circuits increases with time. It may be expressed in terms of proportion increase per unit of time; we denote it $\beta$. Thus the mean number of circuits at time $t$ is proportional to $e^{\beta t}$. We actually assume that connect activity grows at rate $\beta$, but it turns out that the number of active circuits, the total connect rate, and the total disconnect rate are all proportional to $e^{\beta t}$ in this model. Of course, for small growth rates or short periods of time, exponential growth is very close to linear growth.

The disconnect rate, denoted $\mu$, is the ratio of the number of disconnects per unit time (i.e., the total disconnect rate) to the number of active circuits. The mean circuit lifetime is then $1/\mu$. The distributions of circuit lifetimes have been shown to be well approximated by negative exponential distributions;[2] thus the disconnect rate does not vary with the age of a circuit.

The batchiness of the arrival process is related to the tendency of special service circuits to be ordered in multiples greater than one. We call the batchiness parameter $\nu$ and define it to be the ratio of the second moment to the first moment of the number of circuits in an order.

The ultimate goal of this modeling process is to provide a tool that can be used to predict special-services needs in the future. The model

contained herein should be very useful in this regard. One should remember that the underlying process is stochastic so that there is a fundamental uncertainty even if one has exact specification of the parameters of the model. The standard deviation of future requirements can be quite large compared to the mean for small circuit groupings, and this presents a major problem for provisioning at the most detailed level. This problem cannot be surmounted with a better model and/or additional data collection. The present analysis allows quantification of the fundamental uncertainty of forecasting, an insight which is difficult to obtain purely by statistical methods. The only possible method to further decrease relative uncertainty is to aggregate demand, or to obtain advance knowledge of connect or disconnect activity (sometimes called "deterministic events").

The rest of the paper is organized as follows: Section II summarizes the important results of the paper, giving formulae for the means and variances of the number of active circuits in the future, the total number of connects in a future interval, and the total number of disconnects in a future interval; and giving statistical methods to estimate the fundamental parameters of the model such as growth rate, disconnect rate, and batchiness. The reader not interested in the derivation of these results may stop at this point.

The predictions (summarized in Section II) of the model are derived in Section IV. These derivations are primarily substitutions into formulae given in Section III. Section III describes and analyzes a much more general model than the one described in this introduction (we refer to the latter simply as "the model"). We have chosen to introduce this generalized model for two reasons. First, the analysis required for the treatment of the generalized model is little different in complexity from that required for treatment of the specific model. Second, the general results of Section III allow rapid exploration of the consequences of changes in assumptions of the model. For example, one can explore the effects of linear growth of demand, or the super-exponential growth in demand which follows introduction of a new service. However, we do feel that the original assumptions are appropriate in most circumstances. Thus, the consequences of this model are the only ones summarized in Section II, and it is this specific model which is being verified with respect to the New Jersey Bell Telephone Co. database. Thus, Section III is provided for reference in case of non-typical special service applications.

Section V derives the statistical methods (summarized in Section II) for estimation of the fundamental parameters of the model. Section VI is a summary.

Appendix A gives background information on the compound Poisson random variable, and Appendix B gives background information on

the non-homogeneous Poisson process. These results are needed in Sections III and IV.

Table I presents values of a function useful in estimating growth (see Section II) and Table II lists the notation used in the paper.

## II. SUMMARY OF KEY RESULTS

This section provides a summary of the important results of the paper derived in Sections IV and V.

### *2.1 Churn*

Our model depends on three physical parameters: growth rate ($\beta$), disconnect rate ($\mu$), and batchiness ($\nu$). The meaning of these parameters is described in Section I. Another physical parameter is "churn," which has been defined in many different ways. For any reasonable definition, the churn is determined by the growth and disconnect rates of the model. We define the churn to be the minimum of the disconnect rate per circuit and the connect rate per circuit, and denote it by $\gamma$. With this definition, it can be shown [see (75)] that

$$\gamma = \min(\mu, \mu + \beta). \tag{1}$$

The values of churn under other definitions are also readily available. For example, if one defines churn to be the ratio of the average total connect rate to the average rate of change of net active circuits, then this value of churn is $(1 - \gamma)^{-1}$. Under still another definition, the churn equals $\mu/(\mu + \beta)$.

### *2.2 Mean and variance of total active circuits at a future time*

Here we give the mean $M(t)$ and the approximate variance $V(t)$ of the number of circuits in service at a given time $t$ in the future. The mean and the variance depend on the present (at time $t = 0$) number $k$ of circuits in service, the present instantaneous rate $D_o$ of circuit demand due to new orders, and the three key parameters described previously: $\beta$, $\mu$, and $\nu$. We give these two relationships below:

$$M(t) = ke^{-\mu t} + \frac{D_o}{\mu + \beta}(e^{\beta t} - e^{-\mu t}), \tag{2}$$

and

$$V(t) = \nu\left[ke^{-\mu t}(1 - e^{-\mu t}) + \frac{D_o}{\mu + \beta}(e^{\beta t} - e^{-\mu t})\right]. \tag{3}$$

It is interesting that (2) and (3) together imply the relationship

$$V(t) = \nu(M(t) - ke^{-2\mu t}), \tag{4}$$

which relates the variance of a forecast to the mean of the forecast, the number of circuits currently active, $k$, and the parameters $\nu$ and $\mu$.

### 2.3 Mean and variance of the total number of connected or disconnected circuits in the future

Similar results are available for the mean and variance of the total number of connected circuits (variables subscripted with a $C$) and the mean and variance of the total number of disconnected circuits (subscripted with a $D$) in an interval of length $t$ beginning immediately:

$$M_C(t) = \frac{D_o}{\beta}(e^{\beta t} - 1), \tag{5}$$

$$V_C(t) = \nu \frac{D_o}{\beta}(e^{\beta t} - 1) = \nu M_C(t) \tag{6}$$

$$M_D(t) = k(1 - e^{-\mu t}) + \frac{D_o \mu}{\beta(\mu + \beta)} e^{\beta t} + \frac{D_o}{\mu + \beta} e^{-\mu t} - \frac{D_o}{\beta}, \tag{7}$$

and

$$V_D(t) = \nu \left\{ k e^{-\mu t}(1 - e^{-\mu t}) + \frac{D_o \mu}{\beta(\mu + \beta)} e^{\beta t} + \frac{D_o}{\mu + \beta} e^{-\mu t} - \frac{D_o}{\beta} \right\}$$
$$= \nu(M_D(t) - k e^{-2\mu t}). \tag{8}$$

In this case, the total numbers of connected and disconnected circuits are dependent random variables.

We may also obtain the coefficient of correlation $\rho$ between the number of active circuits at different times

$$\rho[Y(t),\ Y(t + \tau)] = e^{-(\mu + \beta/2)\tau}, \tag{9}$$

where $Y(t)$ is the number of active circuits at time $t$.

### 2.4 Estimation of the model parameters

To use results such as (2) through (9), we must be able to estimate the parameters $\beta$, $D_o$, $\nu$, and $\mu$. These questions are addressed in Section V; we provide a brief summary here. Suppose that the system has been observed over the interval $[-\Theta, 0]$ and $n$ connect orders are observed at times $t_1, \ldots, t_n$. Form the statistic

$$S = \sum_{i=1}^{n} t_i / n\Theta + 1, \tag{10}$$

and then the maximum likelihood estimator $\hat{\beta}$ for the growth rate $\beta$ is

$$\hat{\beta} = \frac{1}{\Theta} f^{-1}(S), \tag{11}$$

where $f$ is the function given in (79). Values of $f^{-1}$ are available in Table I. Once $\hat{\beta}$ has been obtained from (11), the estimator $\hat{D}_o$ for the *instantaneous* present demand $D_o$ (assumed to be at the end of the interval of observation $[-\Theta, 0]$) is

$$\hat{D}_o = \frac{n\hat{\beta}}{1 - e^{-\hat{\beta}\Theta}} \hat{N}, \tag{12}$$

where $\hat{N}$ is an estimator for the average number of circuits per order and is equal to the average number of circuits actually observed per order. The estimator $\hat{\nu}$ for the batchiness $\nu$ of the order size is

$$\hat{\nu} = \frac{\sum_{k=1}^{\infty} k^2 i_k}{\sum_{k=1}^{\infty} k i_k}, \tag{13}$$

where $i_k$ is the observed number of existing orders of size $k$. The estimator $\hat{\mu}$ for the parameter $\mu$ can be obtained as the average disconnect rate for observed circuits

$$\hat{\mu} = \frac{m}{\tau}, \tag{14}$$

where $m$ is the total number of disconnects observed, and $\tau$ is the sum of the observed connection times for all circuits; $\mu$ can also be obtained from estimators of the churn and growth rate through the use of (1).

Estimation of these parameters from data supplied by New Jersey Bell Telephone Co. is being investigated. Estimates of the disconnect rate $\hat{\mu}$ by service family are available in the Reed and Smith paper,[2] in which it is shown that the lifetimes of special-service circuits are well approximated by exponential functions with means dependent on the service families.

## III. A GENERALIZED MODEL

This section treats a model that is more general than that which we propose for special-service activity in most cases. The analysis presented here will be applied to the specific model in Section IV.

### *3.1 Description of the generalized model*

We examine an arbitrarily defined category of special-service circuits (for example, circuits of a particular service family in a given

wire center) and divide the active circuits into independent groups. Possibly, each group is the demand from a single user, since it is reasonable that the activity of one user does not affect another. To facilitate this method of thinking we shall refer to the groups as "orders." Each order becomes nonzero for the first time at some point in time (referred to as the arrival or connect time of the order) and then has some history of changing size in some arbitrary manner before possibly becoming zero again indefinitely at some time (the departure or disconnect time of the order). The length of the interval between the arrival and departure of an order will be called the lifetime of the order. Obviously, the number of active circuits at any time equals the sum of the sizes of the existing orders at that time.

We assume that there is a large pool of customers (or potential orders) so that the arrival of an order has little effect on the potential arrival of others. Thus, the arrival of orders can be modeled by a nonhomogeneous Poisson process, whose intensity at time $t$ is given by some function $\lambda(t)$. For background on this process see Ross[3] or Karlin and Taylor.[4] Denote the probability that an arriving order at time $t$ is initially of size $m$ as $q_m(t)$, and let $P^*_{mn}(t, x)$ be the probability that an order arriving at time $t$ of initial size $m$ as becomes size $n$ at time $x \geq t$.

### 3.2 Distribution of the number of active circuits at a given time

Since the orders are noninterfering it can easily be seen (see Appendix B) that the number of orders of size $n$ at time $x$ is Poisson distributed with mean $\alpha_n(x)$, where

$$\alpha_n(x) = \sum_m \int_{-\infty}^{x} \lambda(t) q_m(t) P^*_{mn}(t, x) dt, \tag{15}$$

and that the numbers of orders of different sizes at time $x$ are independent of each other. If $Y(x)$ is the total number of active special-services circuits in the category of interest at time $x$, then $Y(x)$ has a compound Poisson distribution (see Appendix A), and

$$E[Y(x)] = \sum_{n=1}^{\infty} n\alpha_n(x), \tag{16}$$

and

$$\text{var}[Y(x)] = \sum_{n=1}^{\infty} n^2\alpha_n(x). \tag{17}$$

### 3.3 Distribution of future active circuits due to present orders

The transient behavior of this model is easily derived if one has knowledge of the distribution of *order sizes* at a given time. We treat

this case first and then consider the more difficult case where only the total number of *active circuits* at a given time is known. In either case, we will find the distribution of the number of active circuits at time $y$ resulting from the orders observed to be active at time $x$. The total circuits active at time $y$ is the sum of this with the number of circuits at time $y$ resulting from orders arriving between $x$ and $y$.

### *Case 1: Order sizes known*

Given an order is of size $n$ at time $x$, the conditional density that it arrived as an order of size $m$ at time $t$ is $\rho_{mn}(t, x)$, where

$$\rho_{mn}(t, x) = \frac{\lambda(t)q_m(t)P^*_{mn}(t, x)}{\alpha_n(x)} \tag{18}$$

and if several orders of size $n$ are present at $x$ their arriving times and sizes may be considered to be conditionally independent (see Appendix B). Thus an order of size $n$ at time $x$ becomes an order of size $l$ at time $y \geq x$ with probability $r_{nl}(x, y)$ where

$$r_{nl}(x, y) = \alpha_n^{-1}(x) \sum_{m=1}^{\infty} \int_{-\infty}^{x} \lambda(t)q_m(t)P^*_{mn}(t, x)q_{mnl}(t, x, y)dt, \tag{19}$$

and $q_{mnl}(t, x, y)$ is the conditional probability that an order arriving as size $m$ at time $t$ which is of size $n$ at time $x$ becomes size $l$ at time $y$. Note that $q_{mnl}$ is not available solely from $P^*$.

Equation (19) allows us to compute the distribution of the total number of circuits at time $y$ that were due to orders *observed* at time $x$, since all orders behave independently. Evaluating these distributions explicitly can be quite difficult. We can, however, easily evaluate the moments. Let $M_n(x, y)$ be the mean order size at time $y$ for an order observed to be size $n$ at time $x$, and let $V_n(x, y)$ be the mean order size at time $y$ for an order observed to be size $n$ at time $x$, and let $V_n(x, y)$ be the analogously defined variance. Then

$$M_n(x, y) = \sum_{l=1}^{\infty} l r_{nl}(x, y), \tag{20}$$

$$V_n(x, y) = \sum_{l=1}^{\infty} l^2 r_{nl}(x, y) - M_n^2(x, y). \tag{21}$$

If $i_n(x)$ is the number of orders of size $n$ observed at time $x$, and $M(x, y)$ and $V(x, y)$ denote the mean and variance of the number of circuits at time $y$ due to orders observed at time $x$, then

$$M(x, y) = \sum_n i_n(x)M_n(x, y), \tag{22}$$

and

$$V(x, y) = \sum_n i_n(x) V_n(x, y). \tag{23}$$

Note that there is a potential problem if orders can become size zero and then become nonzero later, since determination of $i_0$, the number of active orders of size 0, may be an impossible task.

***Case 2: Order sizes unknown***

We now examine the more difficult case where we observe the total number of active circuits at time $x$ (call this $k$), without observing the distribution of the order sizes. The conditional probability that there are $j_1$ orders of size 1, $j_2$ orders of size 2, etc., given that $k$ total circuits are observed at time $x$, written $\delta_{k,x}(j_1, j_2, \ldots)$, is easily found to be

$$\delta_{k,x}(j_1, j_2, \ldots) = \frac{\prod_i [\alpha_i(x)^{j_i}/j_i!]}{\sum_{j_1+2j_2+\cdots=k} \prod_i [\alpha_i(x)^{j_i}/j_i!]}, \tag{24}$$

provided that $j_1 + 2j_2 + \cdots = k$. Let the conditional first and second moment of the number of circuits at time $y$ due to orders observed at time $x$, given that a total of $k$ circuits were observed at time $x$, be $M_{k,x}(y)$ and $M^{(2)}{}_{k,x}(y)$ respectively. Then

$$M_{k,x}(y) = \sum E(J_i) M_i(x, y), \tag{25}$$

where $J_i$ is a random variable with the same distribution as the conditional number of orders of size $i$ at time $x$, so that the expectation is the expectation with respect to the probability distribution given in eq. (24). Also,

$$M^{(2)}_{k,x}(y) = \sum E(J_i) V_i(x, y) + E((\sum J_i M_i(x, y))^2), \tag{26}$$

where the expectation is in the same sense as before. Needless to say, these expectations with respect to the distribution in (24) are very difficult to evaluate for substantial $k$.

Things simplify somewhat if

$$M_i(x, y) = i\theta(x, y), \tag{27}$$

that is, if the conditional means are proportional to the size of the order. In this case, (25) and (26) give

$$M_{k,x}(y) = k\theta(x, y), \tag{28}$$

and

$$M^{(2)}_{k,x}(y) = \sum E(J_i) V_i(x, y) + k^2\theta^2(x, y), \tag{29}$$

so that

$$V_{k,x}(y) = \sum E(J_i) V_i(x, y), \tag{30}$$

where $V_{k,x}(y)$ denotes the conditional variance. Equation (30) can be approximated by using the following approximation which is intuitively reasonable for $k$ near $\sum i\alpha_i(x)$,

$$E(J_i) \approx k \frac{\alpha_i(x)}{\sum i\alpha_i(x)}. \tag{31}$$

In this case, (30) and (31) give the following useful approximation:

$$V_{k,x}(y) \approx k \frac{\sum \alpha_i(x) V_i(x, y)}{\sum i\alpha_i(x)}. \tag{32}$$

### 3.4 Distribution of future active circuits due to future orders

In section 3.3, we found the mean and the variance of the number of circuits at time $y$ from orders observed at time $x$. To obtain the total number of circuits at time $y$, we need to add to this the (independent) number of circuits due to orders arriving between time $x$ and time $y$. The number of orders of size $n$ at time $y$ that arrived between times $x$ and $y$ is easily seen to be Poisson with mean $\alpha_n(x, y)$, where

$$\alpha_n(x, y) = \sum_{m=1}^{\infty} \int_x^y \lambda(t) q_m(t) P^*_{mn}(t, y) dt, \tag{33}$$

and the number of orders of different sizes are independent of each other (see Appendix B). Thus, the number of circuits at time $y$ due to arrivals occurring between $x$ and $y$ has a compound Poisson distribution (see Appendix A) with mean and variance denoted $M^*(x, y)$ and $V^*(x, y)$, where

$$M^*(x, y) = \sum_n n\alpha_n(x, y), \tag{34}$$

and

$$V^*(x, y) = \sum_n n^2\alpha_n(x, y). \tag{35}$$

### 3.5 Mean and variance of future active circuits

To find expressions for the mean or variance of the total number of active circuits at time $y$, we merely add together the appropriate means or variances from the circuits active at time $y$ due to orders observed at time $x$ and from the circuits active at time $y$ due to arrivals between $x$ and $y$, since these are independent. For example, eqs. (28) and (34) give

$$M^T_{k,x}(y) = k\theta(x, y) + \sum_n n\alpha_n(x, y), \tag{36}$$

where $M^T_{k,x}(y)$ is the total mean number of circuits observed at time $y$ given $k$ circuits are observed at time $x$ [and assuming relationship (27)]. Also, eqs. (32) and (35) give the following approximation:

$$V^T_{k,x}(y) \approx k \frac{\sum \alpha_i(x) V_i(x, y)}{\sum i\alpha_i(x)} + \sum n^2 \alpha_n(x, y), \tag{37}$$

where $V^T_{k,x}(y)$ is the similarly defined variance.

### 3.6 Churn

We have previously defined churn as the minimum of the disconnect rate per circuit and the connect rate per circuit. Values of churn from other definitions are also easily obtained. We will here derive the churn, which happens to be a function of time in this case. To compute churn we need to know the probability measure for the individual order histories. Let $U_m(t, x)$ be the expected number of connects for an order of size $m$ arriving at time $t$ in the interval $[t, x]$ (thus $U_m(t, t) = m$). The expected total connect rate at time $x$, denoted $U(x)$, is then found to be:

$$U(x) = \frac{d}{dx}\left(\sum_m \int_{-\infty}^{x} \lambda(t) q_m(t) U_m(t, x) dt\right), \tag{38}$$

and similarly for the disconnects using the variable $D$,

$$D(x) = \frac{d}{dx}\left(\sum_m \int_{-\infty}^{x} \lambda(t) q_m(t) D_m(t, x) dt\right), \tag{39}$$

and thus we obtain the churn at time $x$, $\gamma(x)$:

$$\gamma(x) = \min\{D(x)/E[Y(x)],\ U(x)/E[Y(x)]\}, \tag{40}$$

where $E[Y(x)]$ is given by (16).

## IV. THE MODEL FOR SPECIAL-SERVICE CIRCUIT ACTIVITY

Here, we assume that the demand rate grows exponentially and that the behavior of orders is not dependent on the time of arrival. Specifically, we assume,

$$\lambda(t) = \lambda_o e^{\beta t}, \tag{41}$$

$$q_m(t) = q_m, \tag{42}$$

and

$$P^*_{mn}(t, x) = P_{mn}(x - t). \tag{43}$$

Later we will assume a specific form for $P_{mn}$.

Assumptions (41) through (43) are equivalent to:

1. exponential growth in the rate of new orders at rate $\beta$ (new orders occur as a nonhomogeneous Poisson process),
2. the probability that a new order is for $m$ circuits is $q_m$, and
3. an order initially for $m$ circuits requires a total of $n$ circuits after $z$ units of time with probability $P_{mn}(z)$. We will shortly further specify $P_{mn}$ to represent unchanging orders of exponential lifetime.

We now explore the consequences of (41) to (43) in the analysis presented in Section III. Substituting into (15) we find that the number of orders of size $n$ at time $x$ is Poisson distributed with mean $\alpha_n(x)$, where

$$\alpha_n(x) = \alpha_n e^{\beta x}, \tag{44}$$

and

$$\alpha_n = \lambda_o \sum_m q_m \int_o^\infty e^{-\beta y} P_{mn}(y) dy. \tag{45}$$

(The total number of circuits required at any time has a compound Poisson distribution, see Appendix A.) Thus, the mean and variance of the number of circuits at time $x$, $Y(x)$, are growing exponentially at the same rate, and the ratio remains fixed:

$$E[Y(x)] = e^{\beta x} \sum_{n=1}^{\infty} n\alpha_n, \tag{46}$$

$$\text{var}[Y(x)] = e^{\beta x} \sum_{n=1}^{\infty} n^2\alpha_n, \tag{47}$$

or

$$\text{var}[Y(x)] = \nu E[Y(x)], \tag{48}$$

where

$$\nu = \frac{\sum_{n=1}^{\infty} n^2\alpha_n}{\sum_{n=1}^{\infty} n\alpha_n}. \tag{49}$$

Further results are possible if the behavior for orders over time is specified. We assume that the order size does not change over its lifetime, which has a common distribution with c.d.f. $F$ independent of size. Later we will assume that $F$ is an exponential distribution. Although in practice the number of circuits per order does change with time, it is conceivable that this movement is relatively unimportant; or even if important, that the general form of eqs. (2) and (3)

will hold, although the parameter $\mu$ may then have a different physical meaning than we will associate here. In the model suggested here, $P_{mo}(y) = F(y)$; $P_{mm}(y) = 1 - F(y)$; $P_{mn}(y) = 0$, $n \neq 0$, $n \neq m$. In this case we may compute $\alpha_n$ more explicitly. Substituting into (45), we obtain

$$\alpha_n = q_n \left( \frac{\lambda_o}{\beta} [1 - \tilde{F}(\beta)] \right), \tag{50}$$

where

$$\tilde{F}(\beta) = \int_o^\infty e^{-\beta y} dF(y). \tag{51}$$

When the lifetimes are exponentially distributed with mean $1/\mu$, i.e. $F(x) = 1 - e^{-\mu x}$,

$$\alpha_n = q_n \lambda_o (\mu + \beta)^{-1}. \tag{52}$$

Also, the batchiness $\nu$ is related to the order-size distribution; substitution into (49) yields

$$\nu = \frac{\sum n^2 q_n}{\sum n q_n}. \tag{53}$$

The assumption that the order size does not change throughout its lifetime also allows more explicit representation of the mean and variance of the future requirements for circuits. Our development here parallels that of Section III. We first compute the probability that an observed order will change size during the period of observation. Recall that $q_{mnl}(t, x, y)$ is the conditional probability that an order is of size $l$ at time $y$ given that it was of size $n$ at time $x$ and arrived as size $m$ at time $t$. We easily obtain:

$$q_{mmm}(t, x, y) = \frac{\bar{F}(y - t)}{\bar{F}(x - t)}, \tag{54}$$

and

$$q_{mmo}(t, x, y) = 1 - q_{mmm}(t, x, y),$$

where

$$\bar{F}(x) = 1 - F(x),$$

and $q_{mml}(t, x, y) = 0$, $l \neq 0$, $l \neq m$. The value of $q$ is irrelevant for $n \neq m$.

We next find the probability that an order of size $n$ at time $x$ becomes of size $l$ at time $y$, which we denote $r_{nl}(x, y)$. Substitution into

(19) gives

$$r_{nl}(x, y) = 0, \qquad n \neq l, \qquad n \neq 0; \tag{55}$$

$$r_{nn}(x, y) = \bar{G}(y - x), \tag{56}$$

where

$$\bar{G}(t) = \frac{\int_o^\infty e^{-\beta z}\bar{F}(z + t)dz}{\int_o^\infty e^{-\beta z}\bar{F}(z)dz}; \tag{57}$$

and

$$r_{n0}(x, y) = 1 - \bar{G}(y - x).$$

Note that in the exponential-lifetime case, where $\bar{F}(z) = e^{-\mu z}$,

$$\bar{G}(y) = e^{-\mu y}. \tag{58}$$

We next find the mean and variance of the number of circuits in an order at time $y$, which was observed to be of size $n$ at time $x$, denoted $M_n(x, y)$ and $V_n(x, y)$, respectively. Substitutions of (55) and (56) into (20) and (21) give:

$$M_n(x, y) = n\bar{G}(y - x), \qquad \text{and} \tag{59}$$

$$V_n(x, y) = n^2\bar{G}(y - x)[1 - \bar{G}(y - x)]. \tag{60}$$

Notice that the conditional means are proportional to the size of the order, i.e., (59) implies (27).

We now focus on the mean and variance of the number of circuits at time $y$ due to orders which were observed at time $x$, given that $k$ circuits were observed at time $x$. These quantities are denoted $M_{k,x}(y)$ and $V_{k,x}(y)$ respectively. Equation (28) gives

$$M_{k,x}(y) = k\bar{G}(y - x). \tag{61}$$

We also conclude that, given the approximation in (32),

$$V_{k,x}(y) \approx k\bar{G}(y - x)[1 - \bar{G}(y - x)]\frac{\sum n^2 q_n}{\sum nq_n}, \tag{62}$$

thus

$$V_{k,x} \approx \nu[1 - \bar{G}(y - x)]M_{k,x}(y), \tag{63}$$

where use has been made of (53) and (61).

We next find the expected number of orders of size $n$ at time $y$ that arrived during the interval $(x, y)$ denoted $\alpha_n(x, y)$. Use of (3.3) yields

$$\alpha_n(x, y) = \lambda_o e^{\beta y} q_n \int_o^{y-x} e^{-\beta z} \bar{F}(z) dz, \tag{64}$$

where use is made of the fact $P^*_{mn}(t, x) = q_{mmn}(t, t, x)$, which follows from (54). When the lifetime distribution for orders is exponential, (64) becomes

$$\alpha_n(x, y) = \lambda_o e^{\beta y} q_n \left( \frac{1 - e^{-(\beta+\mu)(y-x)}}{\beta + \mu} \right). \tag{65}$$

The mean and variance of all circuits at time $y$ due to orders arriving in the interval $(x, y)$, $M^*(x, y)$ and $V^*(x, y)$, respectively, can be obtained by substitution of (64) into (34) and (35) yielding

$$M^*(x, y) = \lambda_o \sum n q_n e^{\beta y} \int_o^{y-x} e^{-\beta z} \bar{F}(z) dz, \tag{66}$$

and

$$V^*(x, y) = \nu M^*(x, y), \tag{67}$$

while for exponential lifetimes,

$$M^*(x, y) = \frac{D_o}{\beta + \mu} (e^{\beta t} - e^{-\mu t}), \tag{68}$$

where

$$D_o = \lambda_o e^{\beta x} \sum n q_n, \tag{69}$$

and

$$t = y - x.$$

Note that (61) and (68) [or (36)] give eq. (2), and (63) and (67) [or (37)] give (3), since the total number of active circuits at time $y$ is the sum of the number of active circuits due to orders present at time $x$ and the number of active circuits due to order arrivals between times $x$ and $y$, and these random variables are independent. Equations (5) through (9) can easily be derived by the methods described in the paper, although we omit the details here.

Next, turning our attention to churn for the specific model of this section, we find that the expected number of connects in the interval $[t, x]$ for an order arriving at time $t$ of size $m$, denoted $U_m(t, x)$, is given by

$$U_m(t, x) = m, \tag{70}$$

and similarly,

$$D_m(t, x) = mF(x - t), \tag{71}$$

where the variable $D$ represents disconnects. The total connect rate, total disconnect rate, and churn rate at time $x$, $U(x)$, $D(x)$, and $\gamma(x)$, respectively, can be obtained from (38) through (40), yielding

$$U(x) = \lambda_o e^{\beta x} \sum mq_m, \tag{72}$$

$$D(x) = \lambda_o e^{\beta x} \sum mq_m \hat{F}(\beta), \tag{73}$$

and

$$\gamma(x) = \gamma = \beta\hat{F}(\beta)/(1 - \hat{F}(\beta)), \qquad \beta \geqslant 0;$$

$$\gamma(x) = \gamma = \beta/(1 - \hat{F}(\beta)), \qquad \beta < 0. \tag{74}$$

In the special case where lifetimes are exponentially distributed,

$$\gamma = \mu, \qquad \beta \geqslant 0;$$

$$\gamma = \mu + \beta, \qquad \beta < 0. \tag{75}$$

## V. ESTIMATION OF THE PARAMETERS OF INTEREST

In this section, we describe the methodology that can be used to estimate the three key parameters of the model; $\beta$, the growth rate; $\mu$, the disconnect rate; and $\nu$, the batchiness.

### *5.1 Estimation of $\beta$*

Suppose that we wish to estimate $\beta$ on the basis of observed arrivals of orders, which by assumption occur according to a nonhomogeneous Poisson process with intensity $\lambda_o e^{\beta t}$. Suppose that the system is observed over the interval $[-\Theta, 0]$ and arrivals have been noted at times $t_1, \ldots, t_n$. We show how to obtain the maximum-likelihood estimator for $\beta$. (For a discussion of maximum-likelihood estimation, see any elementary book on statistics such as Mood & Graybill.)[5] The log-likelihood function, $\ln L(n, t_1, \ldots, t_n)$, is easily seen to be

$$\ln L(n, t_1, \ldots, t_n) = n \ln\lambda_o + \beta \sum_{i=1}^{n} t_i - \lambda_o \left(\frac{1 - e^{\beta\Theta}}{\beta}\right). \tag{76}$$

Differentiating with respect to $\lambda_0$ and $\beta$ we find the necessary conditions for a maximum:

$$n/\lambda_o = \frac{1 - e^{-\beta\Theta}}{\beta}, \tag{77}$$

$$\sum_{i=1}^{n} t_i - \frac{\lambda_o\Theta}{\beta} e^{-\beta\Theta} + \lambda_o \left(\frac{1 - e^{-\beta\Theta}}{\beta^2}\right) = 0. \tag{78}$$

Using (77) to eliminate $\lambda_o$ in (78), we obtain

$$S = \frac{xe^x - e^x + 1}{x(e^x - 1)} \equiv f(x), \tag{79}$$

where

$$S = \frac{\sum t_i}{n\Theta} + 1 \tag{80}$$

and

$$x = \beta\Theta. \tag{81}$$

The function $f$ defined in (79) can be seen to be strictly monotonic with range between 0 and 1. Therefore, eq. (79) allows us to solve for $\beta\Theta$ as $f^{-1}(S)$, where $S$ is the statistic defined in (80) equal to the proportion of the interval (*after* $-$ $\Theta$) at which the average time of arrival occurs. Thus, the maximum-likelihood estimator for $\beta$, written $\hat{\beta}$, is given by

$$\hat{\beta} = \frac{f^{-1}(S)}{\Theta}. \tag{82}$$

The function $f$ has the properties:

$$f(-\infty) = 0, f(0) = 1/2, f(\infty) = 1, \quad \text{and} \quad f(x) + f(-x) = 1.$$

Thus,

$$f^{-1}(0) = -\infty,$$
$$f^{-1}(1/2) = 0,$$
$$f^{-1}(1) = \infty,$$

and

$$f^{-1}(1/2 - x) = -f^{-1}(1/2 + x).$$

The function $f^{-1}$ is tabulated in Table I.

For small $x$, $f(x)$ may readily be expanded in the power series:

$$f(x) = 1/2\left[1 + (1/6)x - \frac{1}{360}x^3 \cdots\right]$$

so that

$$f^{-1}(1/2 + y) = 12y + 28.8y^3 \cdots. \tag{83}$$

Similarly, a large $f$ expansion yields

$$f^{-1}(1 - 1/y) \approx y - y^2e^{-y}. \tag{84}$$

We may also determine the mean and variance of the statistic $S$ given the correct parameter $\beta$ and the number of observed arrivals. It is well known that the distribution of the arrival times for a nonhomogeneous Poisson process, conditioned on a given number of arrivals

Table I—Values of the function $f^{-1}$ useful in estimating the growth rate $\beta$, and several approximations for the function [see eq. (79) and following]

| X | $f^{-1}(X)$ | 12X-6 | Approximation in (83) | Approximation in (84) |
|---|---|---|---|---|
| 0.50 | 0.0000 | 0.0000 | 0.0000 | |
| 0.52 | 0.2402 | 0.2400 | 0.2402 | |
| 0.54 | 0.4819 | 0.4800 | 0.4818 | |
| 0.56 | 0.7263 | 0.7200 | 0.7262 | |
| 0.58 | 0.9751 | 0.9600 | 0.9747 | |
| 0.60 | 1.2299 | 1.2000 | 1.2288 | |
| 0.62 | 1.4926 | 1.4400 | 1.4898 | |
| 0.64 | 1.7654 | 1.6800 | 1.7590 | |
| 0.66 | 2.0507 | 1.9200 | 2.0380 | |
| 0.68 | 2.3517 | | 2.3280 | |
| 0.70 | 2.6721 | | 2.6308 | |
| 0.72 | 3.0168 | | 2.9467 | |
| 0.74 | 3.3920 | | | |
| 0.76 | 3.8060 | | | |
| 0.78 | 4.2703 | | | |
| 0.80 | 4.8010 | | | 4.8316 |
| 0.82 | 5.4219 | | | 5.4362 |
| 0.84 | 6.1691 | | | 6.1746 |
| 0.86 | 7.1010 | | | 7.1025 |
| 0.88 | 8.3164 | | | 8.3166 |
| 0.90 | 9.9954 | | | 9.9955 |
| 0.92 | 12.4994 | | | 12.4994 |
| 0.94 | 16.6667 | | | 16.6667 |
| 0.96 | 25.0000 | | | 25.0000 |
| 0.98 | 50.0000 | | | 50.0000 |
| 1.00 | $\infty$ | | | $\infty$ |

in the interval, is the same as the order statistics from $n$ i.i.d. random variables with probability density proportional to the arrival rate. Thus $S$ has the distribution of the average of $n$ i.i.d. random variables, $Y_i$ on [0, 1] with density $g(\rho)$, where

$$g(\rho) = \frac{xe^{\rho x}}{e^x - 1}$$

and

$$x = \beta\Theta.$$

It is easily seen that

$$E(Y) = f(x) \tag{85}$$

and

$$\text{var}(Y) = \frac{1}{x^2} - \frac{1}{e^x + e^{-x} - 2}. \tag{86}$$

Equation (86) is valid if $x \neq 0$; when $x = 0$, $\text{var}(Y) = 1/12$, the limit

of (86) as $x$ goes to 0. The expression for the variance is symmetric in $x$ and takes its maximum value at $x = 0$.

Thus, for a given value of the growth rate $\beta$, and a (large) given number of observations $n$, the observed statistic $S$ is approximately normally distributed with mean $f(x)$ and variance less than or equal to $1/12n$. This observation can be readily translated into confidence intervals through the use of elementary statistical theory. For example, a 95-percent confidence interval for $f(x)$ (assuming normality of the statistic) is

$$S - 1.96 \sqrt{\frac{1}{12n}} \leq f(x) \leq S + 1.96 \sqrt{\frac{1}{12n}}, \tag{87}$$

which translates to

$$f^{-1}\left(S - 1.96 \sqrt{\frac{1}{12n}}\right) \leq \beta\Theta \leq f^{-1}\left(S + 1.96 \sqrt{\frac{1}{12n}}\right). \tag{88}$$

If $S$ is close to 0.5, then we can use $f^{-1}(x) \approx 12x - 6$ [see (83)] to obtain for the 95-percent confidence interval for $\beta\Theta$

$$\beta\Theta = 12S - 6 \pm \frac{6.79}{\sqrt{n}}. \tag{89}$$

### *5.2 Estimation of $\nu$*

There are several possible statistics for the measurement of the batchiness $\nu$. We shall take as our starting point eq. (49) which defines the batchiness in terms of the distribution of the order size at (any) point in time. This is preferable and is more robust than using the distribution of the order size on arrival, although the two happen to equal when order sizes do not change with time. Thus, if $\nu$ is to be estimated based on observation of the system at a given point in time at which $i_n$ orders of size $n$ are observed, then a reasonable estimator for $\nu$, which we write $\hat{\nu}$, is:

$$\hat{\nu} = \frac{\sum n^2 i_n}{\sum n i_n}. \tag{90}$$

When the number of circuits does change during the lifetime of an order, it is possible that the form of (3) still holds. In this case, it is likely that the parameter $\nu$ which multiplies each of the two terms in (3) is different. Equation (90) is a reasonable estimator for the multiplier of the second term. The multiplier of the first term should be estimated by other methods.

Table II—Notation

| | |
|---|---|
| $D_o$ | Present ($t = 0$) circuit demand rate due to new orders. |
| $D(x)$ | Expected total disconnect rate at time $x$. |
| $D_m(t, x)$ | Expected number of disconnects in the interval $[t, x]$ for an order of size $m$ arriving at time $t$. |
| $i_n(x)$ | Observed number of orders of size $n$ at time $x$. |
| $J_i$ | Conditional number of orders of size $i$ due to orders observed at time $x$ given that $k$ circuits were observed at time $x$. |
| $M(x, y)$ | Mean total number of circuits at time $y$ due to orders. |
| $M^*(x, y)$ | Mean of the total number of circuits at time $y$ due to orders arriving between $x$ and $y$. |
| $M^T_{k,x}(y)$ | Mean of the number of circuits at time $y$ given $k$ are observed at time $x$. |
| $M_{k,x}(y)$ | Conditional expectation of the number of circuits at time $y$ due to orders observed at time $x$ given that $k$ circuits are observed at time $x$. |
| $M^{(2)}_{k,x}(y)$ | Conditional second moment of the number of circuits at time $y$ due to orders observed. |
| $M_n(x, y)$ | Mean order size at time $y$ for an order observed to be of size $n$ at time $x$. |
| $P_{mn}(t)$ | Probability that an order of initial size $m$ becomes of size $n$, $t$ time units after arrival. |
| $P^*_{mn}(t, x)$ | Probability that an order arriving at time $t$ of initial size $m$ becomes of size $n$ at time $x$. |
| $q_m$ | $q_m(t)$ when there is no dependence on $t$. |
| $q_m(t)$ | Probability that an order at time $t$ is initially for $m$ circuits. |
| $q_{mnl}(t, x, y)$ | Conditional probability that an order arriving as size $m$ at time $t$ and of size $n$ at time $x$ becomes of size $l$ at time $y$. |
| $r_{nl}(x, y)$ | Probability that an order of size $n$ at time $x$ becomes an order of size $l$ at time $y$. |
| $U(x)$ | Expected total connect rate at time $x$. |
| $U_m(t, x)$ | Expected number of connects for an order of size $m$ arriving at time $t$ in the interval $[t, x]$. |
| $V(x, y)$ | Variance of the total number of circuits at time $y$ due to orders observed active at time $x$. |
| $V^*(x, y)$ | Variance of the total number of circuits at time $y$ due to orders arriving between $x$ and $y$. |
| $V^T_{k,x}(y)$ | Variance of the number of circuits at time $y$ given that $k$ are observed at time $x$. |
| $V_{k,x}(y)$ | Conditional variance of the total number of circuits at time $y$ due to orders observed at $x$ given that $k$ circuits are observed at time $x$. |
| $V_n(x, y)$ | Variance of order size at time $y$ for an order observed to be of size $n$ at time $x$. |
| $Y(x)$ | Total number of active special service circuits at time $x$. |
| $\alpha_n$ | Constant of proportionality for the exponential growth of $\alpha_n(x)$. |
| $\alpha_n(x)$ | Mean number of orders of size $n$ at time $x$. |
| $\alpha_n(x, y)$ | The number of orders of size $n$ at time $y$ which arrived between times $x$ and $y$. |
| $\beta$ | Growth rate. |
| $\gamma$ | Churn. |
| $\delta_{k,x}(j_1, j_2, \ldots)$ | Conditional probability that there are $j_1$ orders of size 1, $j_2$ orders of size 2, etc., at time $x$ given that $k$ total circuits are observed at time $x$. |
| $\theta(x, y)$ | Defined in (27). |
| $\Theta$ | Length of observation period. |
| $\lambda_o$ | Present ($t = 0$) arrival rate of orders. |
| $\lambda(t)$ | Instantaneous arrival rate of orders at time $t$. |
| $\mu$ | Disconnect rate (per circuit). |
| $\nu$ | Batchiness. |
| $\rho_{mn}(t, x)$ | The conditional probability density that an order arrived at time $t$ of initial size $m$ given that it is of size $n$ at time $x$. |

### 5.3 Estimation of $\mu$

The estimation of $\mu$ is relatively straightforward. The maximum-likelihood estimator is given in (14) and further details including estimated values by service family are given in Reed and Smith.[2]

### 5.4 Estimation of $D_o$

Equation (77) allows the MLE estimator of $\lambda_o$,

$$\hat{\lambda}_o = \frac{n\hat{\beta}}{1 - e^{-\hat{\beta}\Theta}}, \tag{91}$$

where the estimator $\hat{\beta}$ has been previously described in (82). The estimator of $D_o$ (the instantaneous demand at the end of the observation interval of length $\Theta$), $\hat{D}_o$ then is

$$\hat{D}_o = \lambda_o \hat{N}, \tag{92}$$

where $\hat{N}$ is an estimate of the average batch size. The previous expectation can be estimated from the order sizes at arrival epochs, or more crudely from the general distribution of order sizes at an arbitrary point of time.

Interestingly enough, $D_o/(\mu + \beta)$ can be estimated solely from the number of active circuits at a point of time. For simplicity, we assume that the orders are solely of size one, although the analysis could be repeated for other distributions. In this case, the following analysis is applicable.

Suppose that the number of active circuits at time $t$ is a Poisson random variable with mean $\lambda e^{\beta t}$. The time that the mean is between $x$ and $x + dx$ is $dx/x\beta$. The expected time that the mean is between $x$ and $x + dx$ and a total of $k$ active circuits are observed is $\frac{dx}{x\beta}\frac{x^k}{k!}e^{-x}$. Thus, if $k$ active circuits are observed, the conditional distribution of the mean (in our case this is $D_o/(\mu + \beta)$) has density proportional to $x^{k-1}e^{-x}$, or is a standard gamma random variable with $k$ degrees of freedom. This random variable has mean and variance equal to $k$. Thus, if $k$ circuits are observed, the conditional distribution of $D_o/(\mu + \beta)$ has mean $k$ and variance $k$, and is approximately normally distributed if $k$ is large. This information can be used to modify eqs. (2) and (3), to take into account the variability of $D_o/(\mu + \beta)$ to obtain:

$$M_k(t) = ke^{\beta t}, \tag{93}$$

and

$$V_k(t) = \nu k[e^{\beta t} - e^{-2\mu t}] + k[e^{\beta t} - e^{-\mu t}]^2, \tag{94}$$

where the subscript $k$ on the variables on the left-hand side indicates

conditional means and variances knowing $\beta$, $\mu$, and $\nu$ but not knowing $D_0$. Note that the variance-to-mean ratio is unbounded for increasing $t$, since errors in estimation of $D_0$ accumulate indefinitely.

## VI. SUMMARY

We have described a model for special-services activity useful in forecasting special-services requirements. It requires three physical characterizations of the process (growth rate, disconnect rate, and batchiness) and two instantaneous measurements (the current number of active circuits and the instantaneous rate of new connects). We give means and variances for the numbers of active circuits at a given point in the future and for the total number of connects or disconnects during a future period. The distribution of these variables can be computed by the methodology described in the paper. We also describe general techniques for estimation of the required parameters.

Work is being undertaken to verify and calibrate the model with the New Jersey Bell Telephone Co. database and will be reported elsewhere.

## VI. ACKNOWLEDGMENTS

I would like to thank Moshe Segal and Werner Stahel for many helpful suggestions.

## REFERENCES

1. R. Nucho, "Transient Behavior of the Kendall Birth-Death Process—Applications to Capacity Expansion for Special Services," B.S.T.J. *60*, No. 1 (January 1981), pp. 57–87.
2. F. F. Reed and D. R. Smith, unpublished work.
3. S. Ross, *Applied Probability Models with Optimization Applications*, San Francisco: Holden-Day, 1970.
4. S. Karlin and H. Taylor, *A First Course in Stochastic Processes*, second edition, New York: Academic Press, 1975.
5. A. Mood and F. Graybill, *Introduction to the Theory of Statistics*, second edition, New York: McGraw-Hill, 1963.

## APPENDIX A

### *The Compound Poisson Random Variable*

Briefly, a random variable is said to be a compound Poisson random variable if it can be thought of as the sum of a Poisson number of independent identically distributed positive integer-valued random variables. Thus $Y$ is a compound Poisson random variable if

$$Y = \sum_{i=1}^{N} X_i,$$

where $N$ is a Poisson random variable with mean $\alpha$, $N$ and the $X_i$ are

independent and

$$P\{X_i = n\} = p_n.$$

We have rather easily

$$E(Y) = \alpha E(X) = \alpha \sum np_n$$

$$\text{Var}(Y) = \alpha E(X^2) = \alpha \sum n^2 p_n.$$

An alternative (and equivalent) way of specifying a compound Poisson random variable is

$$Y = \sum_{i=1}^{\infty} iZ_i,$$

where

$Z_i$ are independent Poisson random variables with $E(Z_i) = \alpha p_i \equiv \alpha_i$. In this case it is convenient to think of $Z_i$ as the number of batches or orders of size $n$ that are aggregated to give the total number denoted $Y$.

## APPENDIX B

### *The Nonhomogeneous Poisson Process*

A process which counts events is a nonhomogeneous Poisson process (see, for example, Ross, Ref. 5) with intensity $\lambda(t) \geqslant 0$ if the number of events in the interval $[x, y]$ is a Poisson random variable with mean $\int_x^y \lambda(t)dt$, and the number of events in disjoint intervals are independent.

*Fact*: If events from a nonhomogeneous Poisson process are of two types, and an event at time $t$ is of Type 1 with probability $p(t)$, then the process which counts Type 1 events is a nonhomogeneous Poisson process with intensity $\lambda(t)p(t)$, and it is independent of the counting process which counts Type 2 events [which is a nonhomogeneous Poisson process intensity $\lambda(t)(1 - p(t))$].

*Fact*: Suppose that a nonhomogeneous Poisson process is observed over the interval $[x, y]$ and $n$ events are observed. If the times of these events are arranged in random order, their distribution is identical to that of $n$ independent identically distributed random variables whose density at $t$ is

$$\frac{\lambda(t)}{\int_x^y \lambda(z)dZ}$$

if $t \in [x, y]$ and is zero otherwise.

## AUTHOR

**Donald R. Smith,** A.B. (Physics), 1969, Cornell University; M.S. (Operations Research), 1974, Columbia University; Ph.D. (Operations Research), 1975, University of California, Berkeley; Bell Laboratories, 1980—. Before joining Bell Laboratories, Mr. Smith was employed at Adaptive Technology, Inc., 1970–1974, and was Assistant Professor in the Department of Industrial Engineering and Operations Research, Columbia University, 1975–1979. At Adaptive Technology, Mr. Smith developed mathematical models for new techniques in statistical multiplexing. At Bell Laboratories he is in the Operations Research Department working on the application of stochastic processes to special service and DACS modeling.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# TELBECC—A Computational Method and Computer Program for Analyzing Telephone Building Energy Consumption and Control

By P. B. GRIMADO*

(Manuscript received February 2, 1983)

Telephone Building Energy Consumption and Control (TELBECC) program has been developed to accurately and efficiently analyze environmental control and energy use in telephone company buildings. The program simulates various operational plans to determine the relative energy and cost savings. By analyzing the operation of the heating, ventilation, and air conditioning system as it regulates a changing environment, TELBECC calculates the heating and cooling load, dry-bulb temperature, and relative humidity in the building. The user specifies the building's dry-bulb temperature limits, which are the control variables for the program analysis. The simplified computational procedure of the program incorporates a recursive scheme using time series to perform the necessary calculations. The results of the computations can be obtained for different periods: the quarter hour, hour, day, or month. Energy consumption and control in several equipment buildings located in three different geographical areas have been analyzed by TELBECC. Analysis and comparison of the resulting data demonstrate the advantages of the program.

## I. INTRODUCTION

An ambitious energy-cost savings program has been instituted to reduce energy use in telephone company buildings. In recent years, telephone companies have saved energy mainly by redesigning and

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

retrofitting buildings to operate and maintain environmental control equipment at peak performance, to turn out unneeded lights, to reduce heating and cooling losses, and the like. Further energy and cost savings, although requiring additional capital investment, could be achieved through modification of environmental control systems, purchase of sophisticated microprocessors for more efficient control of building Heating, Ventilating, and Air conditioning (HVAC) systems, and installation of alternative energy sources, such as solar power and wind power. Before adopting any conservation plans that require appreciable capital investment, however, we should make a thorough economic evaluation. Such an evaluation can be carried out by correlating the changing operating characteristics of a building with selected energy conservation plans. This procedure would enable us to pinpoint the most economical operating strategies.

There are several commercially available computer programs to perform this type of analysis, such as DOE II, ESP, and BLAST;[1,2] most, however, are proprietary. These complex programs can analyze any of a broad spectrum of commercial, industrial, and residential buildings. But, because of their versatility, they require large computer systems, extensive data preparation, and high costs. The use of energy in the majority of telephone company equipment buildings, which are small, single-story structures varying in area from 1500 to 10,000 square feet, can be best evaluated by a more focussed computer program.

This paper describes a new computational method and computer program called Telephone Building Energy Consumption and Control (TELBECC). This program simulates building operations and quickly evaluates numerous energy conservation plans and cost-saving strategies under variable weather conditions [according to standard hourly Test Reference Year (TRY) weather data[3]]. The program can evaluate energy consumption for intermittent or proportional HVAC plant operation, economy cycle operation, enthalpy cycle operation, and wideband temperature operation with no heating or cooling between preset room temperature limits. Also, the program can calculate the optimum building orientation and U factor (heat transmission characteristics) of the outside walls and roof, chiller and heater plant size, dry-bulb temperature and Relative Humidity (RH) variations, and quantity of water required to maintain 20-percent minimum RH during economy cycle operations.

## II. PROGRAM DESCRIPTION

We can derive the heating or cooling load in an enclosed building from the following considerations:

1. Conduction of heat through the building walls and roof.

2. Permeation of outside air through the building envelope.
3. Internal heat generation from equipment, lights, and people.
4. Direct solar radiation through windows and skylights (fenestration).

Since most operating company equipment buildings have few windows, the program does not consider item 4 in the present version.

A Constant-Air-Volume (CAV) supply fan system typically controls the air temperature of a building. The building engineer normally sizes the fan system using the elementary steady-state heat balance, which takes into account the internal heat loads, outside air temperatures, and solar radiation in conjunction with the U factor of the building envelope. In general, this conservative approach produces fan systems that are oversized and therefore inefficient. To find a smaller, and perhaps more efficient, fan capacity, a TELBECC user selects different fan capacities for analysis by the program. The program generates data on the space temperatures, relative humidity, peak heating and chiller loads, and the hours of system operation for the different fan capacities that can be used to find an optimum air supply fan system.

For comfort, a limit is imposed on the difference between HVAC supply and return air temperatures. For cooling, this temperature difference is −20 degrees F; for heating, +40 degrees F. These default values may be overridden by the user. With an environmental dry-bulb temperature standard specified, the program computes the required operation of the HVAC.

The user can specify one of two basic ways to operate the HVAC: intermittent operation or proportional control. With intermittent operation, the HVAC does not supply any heating or cooling when the dry-bulb air temperature is within the wideband temperature range. Reaching or exceeding either wideband temperature limit activates the HVAC. The HVAC stays on and does not deactivate until the dry-bulb air temperature reaches 3 degrees F above the lower limit of the wideband temperature range for heating and 3 degrees F below the upper limit of the wideband temperature range for cooling. The TELBECC user can reset the numerical values of the throttling range if a different range is appropriate. The proportional control plan operates by continuously adjusting the supply and return air temperature difference in increments of 1 degree F to satisfy the instantaneous building heating or cooling load. This plan follows the building load to closely track the lower and upper limits of the wideband temperature range with essentially no throttling. When selecting a dual or extended wideband temperature standard (that is, one with different wideband limits for occupied and unoccupied times), the HVAC activates before occupancy in order to reach the preset temperature standard.

TELBECC calculates the heat added or removed by the HVAC

system in controlling the dry-bulb air temperature every quarter hour. In particular, when cooling is required, the sensible and latent loads on the chiller plant are simultaneously computed by incorporating any of three standard methods of fan system operation:

1. Conventional operation, which is chiller operation with no economizer.
2. Chiller operation with a dry-bulb economy cycle.
3. Enthalpy cycle.

In conventional operation, the minimum quantity of outside air needed for ventilation is circulated. This mode is also used as a benchmark for the program. The dry-bulb economy cycle uses outside air for cooling whenever the outside dry-bulb air temperature falls below the maximum value. The default value is 55 degrees F, but the user can reset the value. The enthalpy cycle checks the enthalpy of the inside air and the outside air. If the outside air enthalpy is lower, 100 percent outside air is circulated to reduce the load on the chiller regardless of the relative humidity. Otherwise, only the minimum quantity of outside air required for ventilation is circulated.

System control is based on dry-bulb air temperature and is not predicated on maintaining a particular value of relative humidity. Nevertheless, the program computes changes in relative humidity for the three methods of fan-system operation discussed above. The program summarizes the variation in relative humidity for the time period chosen by giving the number of hours the relative humidity is less than 10 percent, between 10 and 15 percent, between 15 and 20 percent, between 20 and 55 percent, between 55 and 60 percent, and greater than 60 percent.

In addition, since dry-bulb economy cycle operation generally calls for bringing in winter air with low humidity, the program calculates the quantity of water required for humidification. The operating company minimum standard of 20 percent RH in the inside air for dry-bulb economy cycle operation in winter is the basis for calculating the amount of water added to the air.

## III. TRANSIENT HEAT CONDUCTION THROUGH THE BUILDING ENVELOPE

Weather conditions influence the heating and cooling load of a building by heat conduction through the structural and decorative materials of the exterior walls and roof, as well as by permeation of outside air and direct absorption of solar radiation through window areas. Since, as previously mentioned, most operating company equipment buildings have few windows, only heat conduction and permeation are treated in the computer program. The program must account for the heat storage effects of the structure, as well as the daily and

seasonal variation of the outside air temperature and solar radiation. We can account for these influences on the building by considering the building envelope elements as one-dimensional flat slabs or plates. We then obtain a solution to a partial differential equation with time-dependent boundary conditions. A classical analytical solution of this equation[4] produces a set of equations that require an inordinate quantity of computational effort and time, rendering the whole idea of performing the analysis impractical and uneconomical. However, the analytical solution can be recast into a simpler, though effective, computational scheme with a method first introduced by Mitalas and Stephenson,[5] which is ideally suited to calculation by computer.

The inside-wall and roof-surface temperature $T_{BE}(t)$ and the air temperature of the building $T_a(t)$, which are dependent on time, determine the environmental load due to convection.* $T_{BE}(t)$ is represented in the form of the following time series:

$$T_{BE_t} = -\sum_{i=1}^{m-1} b_i T_{BE_{t-i}} + \sum_{i=1}^{m} a_i T_{O_{t-i-1}} + \sum_{i=1}^{m} a_i' T_{a_{t-i-1}}, \qquad (1)$$

where

$t$ = current time,
$T_{BE_t}$ = inside-wall temperature of the building at the current time,
$T_{BE_{t-i}}$ = inside-wall temperature of the building $i$ time units prior to $t$,
$T_{O_{t-i}}$ = outside sol-air temperature[6] $i$ time units prior to $t$, and
$T_{a_{t-i}}$ = air temperture of the building $i$ time units prior to $t$.

For hourly temperature calculations, the number of terms, $m$, will rarely exceed 5, and for quarter-hour calculations, $m$ will generally be less than 15. The recursive properties of the calculation make it extremely efficient and economical, especially when the operating characteristics of the building may need to be tracked for an entire calendar year. The coefficients $b_i$, $a_i$, and $a_i'$ in eq. (1) are determined from the thermophysical properties of the structure. Only six values are needed to uniquely specify these coefficients: wall thickness, wall U factor, wall-weight density, effective heat-transfer coefficient of the inside- and outside-wall surface, and the time interval between successive calculations. The appendix presents the mathematical procedure to evaluate these coefficients. Figures 1 and 2 show the mathematical and physical models for deriving the coefficients.

We can validate this simplified computational approach by compar-

* Radiative interchange between inside building-wall surfaces is not included in the present version of the program.

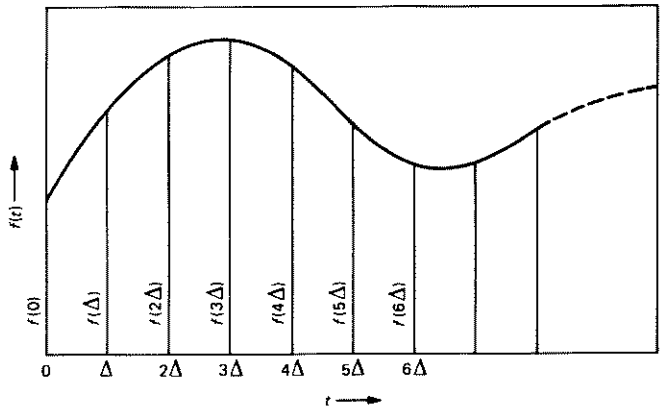



Fig. 1—Discrete and continuous functions.

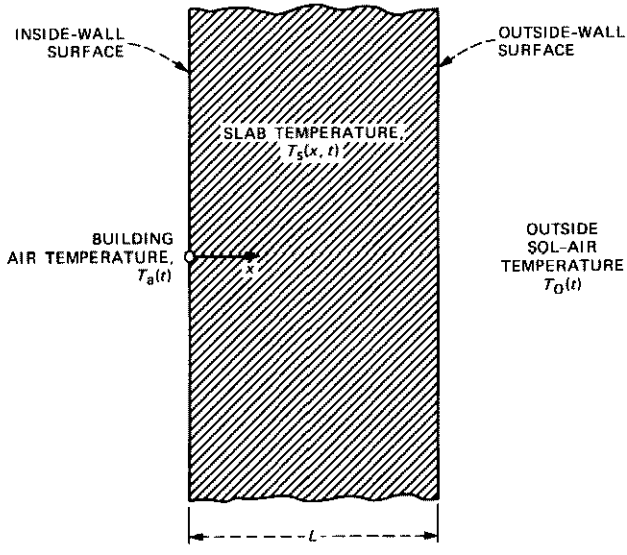



Fig. 2—Homogeneous flat slab.

ing it with an exact solution given in the literature.[7] We can see this in Figs. 3 and 4 for several values of the inside-wall dimensionless convective heat-transfer parameter $B_i \equiv h_i L/k$ and two limiting values of the outside-wall convective heat-transfer parameter $B_O \equiv h_O L/k$. In Fig. 3 the outside-wall convective heat transfer parameter $B_O = 0$; i.e., the surface $x = L$ is insulated. In Fig. 4 the solution corresponds to $B_O$ approaching $\infty$; i.e., the surface $x = L$ is maintained at a constant temperature. The initial and boundary conditions are indicated in the figures. The solid curves represent the exact condition given in Ref. 7,

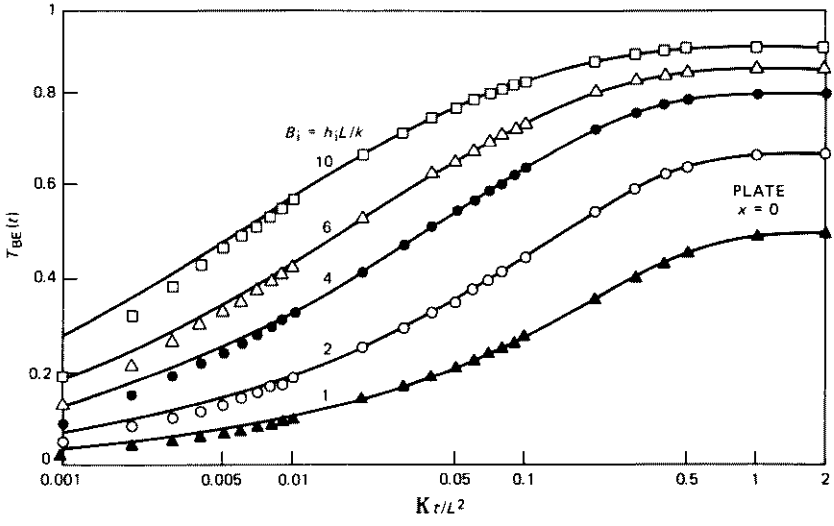



Fig. 3—Temperature response of front face of plate, $0 \leq x \leq L$, with back face $0 \leq x \leq L$, with back face $x = L$ maintained at $T_0 = 0$ degree F after sudden exposure to uniform-temperature convective environment $T_a = 1$ degree F at $x = 0$. Sampling interval $\Delta$ is one min.

and the dots represent the numerical values computed by the time series in eq. (1). The sampling interval for this example is $\Delta = 1$ min. or, in terms of dimensionless time, $k\Delta/L^2 = 0.001$. We can see that after some time has elapsed the exact solution and the time-series solution match identically. For the problem considered here at $t = 0$, the ambient convected temperature $T_a$ is suddenly raised from $T_a = 0$ to $T_a = 1$; i.e., the boundary condition is a step function. However, since the development of the time series assumes a linear variation between time intervals, as stated in the appendix, the solution resembles an initial ramp followed by a constant value, as shown in Fig. 5. Once the effect of the initial ramp input diminishes after about 10 sampling intervals, the solution coincides with the exact solution. This characteristic of the time series is not a problem here, since instantaneous changes of air temperatures inside and outside the building do not occur.

## IV. CALCULATION OF BUILDING AIR TEMPERATURE AND ENERGY USE

The air temperature of the building is obtained through the following equation for the heat balance within the building:

$$q_{air}(t) = q_{equipt}(t) + q_{lights}(t) + q_{people}(t) + q_{infiltration}(t) + q_{walls}(t) + q_{HVAC}(t), \quad (2)$$

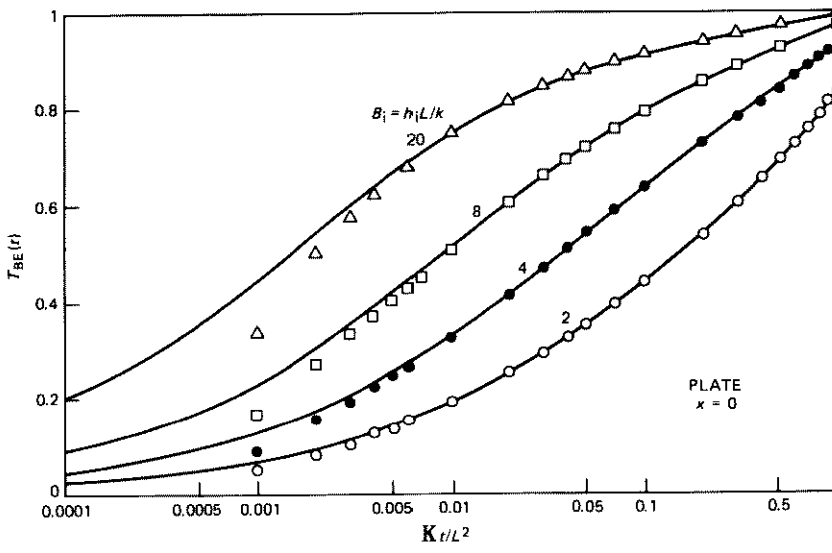



Fig. 4—Temperature response of front face of plate $0 \leq x \leq L$, with insulated back face $x = L$ after sudden exposure to uniform-temperature convective environment $T_a =$ 1 degree F at $x = 0$. Sampling interval $\Delta$ in one min.

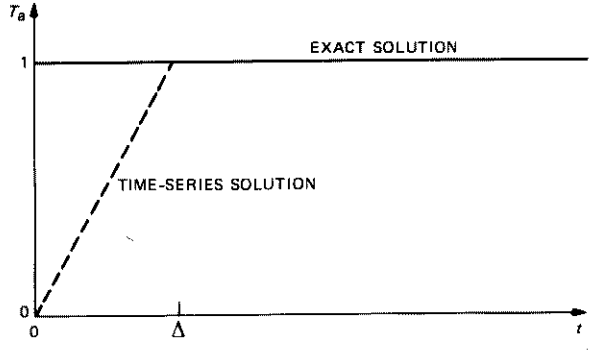



Fig. 5—Convection temperature environment at inside boundary surface $x = 0$.

where $q_{air}(t)$ (Btu/hr) represents the sensible thermal-energy convection rate of the inside air, resulting in a change in the overall dry-bulb air temperature. The other terms on the right represent the rate at which heat is convected to the air from the following: equipment heat dissipation; lighting; people; inadvertent infiltration of outside air; the exterior walls, floor, and roof of the building; and the building's HVAC control system.

To evaluate the air temperature $T_a(t)$ from eq. (2), each term is expressed as the difference in temperatures between the air and the

heat-convecting medium. We can determine $T_a(t)$ from the following differential equation:

$$\rho c v \frac{dT_a(t)}{dt} = H_{EL}A_{EL}[T_{EL}(t) - T_a(t)] + \rho cm[T_O(t) - T_a(t)] + H_{BE}A_{BE}[T_{BE}(t) - T_a(t)] + \rho cQ[DT_{SP}(t)], \quad (3)$$

where

$T_a(t)$ = average building dry-bulb air temperature;
$T_{EL}(t)$ = combined average temperature of the equipment, lights, and people;
$T_O(t)$ = outside dry-bulb air temperature;
$T_{BE}(t)$ = inside surface temperature of the building envelope;
$DT_{SP}(t)$ = difference between the air-supply temperature and the air temperature of the building;
$\rho$ = air density;
$c$ = air specific heat;
$v$ = volume of the building space;
$H_{EL}$ = heat-transfer coefficient between the air and equipment;
$A_{EL}$ = average surface area of equipment;
$m$ = rate at which outside air infiltrates the building;
$H_{BE}$ = heat-transfer coefficient between the building envelope and air;
$A_{BE}$ = surface area of the building envelope; and
$Q$ = air-supply rate of the HVAC system fan units.

Only one of the temperatures in eq. (3) is presumed known: the outside-air dry-bulb temperature, $T_O(t)$. The other four temperatures—$T_a(t)$, $T_{EL}(t)$, $T_{BE}(t)$, and $DT_{SP}(t)$—are coupled; therefore, additional equations are needed for their resolution.

We can consolidate our terms into two other equations. When we combined the heat gain generated by the equipment, lighting, and people into a single term for heat dissipation per unit of building floor area $W(t)$(W/ft$^2$), one additional equation can be written as

$$C_{EL} \frac{dT_{EL}(t)}{dt} = 3.41\ A_f W(t) + H_{EL}A_{EL}[T_a(t) - T_{EL}(t)], \quad (4)$$

where

$C_{EL}$ = heat capacity of the equipment, lights, and people;
$A_f$ = building floor area; and
$W(t)$ = combined heat dissipation of the equipment, lights, and people per unit of building floor area.

A third equation coupling the building envelope temperature $T_{BE}(t)$ and the building air temperature $T_a(t)$ is needed. A likely equation

would be the time series given by eq. (1). But before this can be applied, both eqs. (3) and (4) must also be recast in the form of time series. This is easily done by using the properties of the $z$ transform and the procedure already delineated in the appendix. The time-series solution of eqs. (3) and (4) assumes the form

$$T_{a_t} = sT_{a_{t-1}} + \left(\frac{\Delta\tau + s - 1}{\Delta t^2}\right)[\tau_{EL}T_{EL} + \tau_O T_O + \tau_{BE}T_{BE} + \tau_{SP}DT_{SP}]_t$$

$$+ \left(\frac{1 - s(\Delta\tau + 1)}{\Delta\tau^2}\right)[\tau_{EL}T_{EL} + \tau_O T_O + \tau_{BE}T_{BE} + \tau_{SP}DT_{SP}]_{t-1}, \quad (5)$$

and

$$T_{EL_t} = \hat{s}T_{EL_{t-1}} + \frac{(\Delta\hat{\tau} + \hat{s} - 1)}{\Delta\hat{\tau}^2}[\hat{\tau}_w W + \hat{\tau}_a T_a]_t$$

$$+ \frac{[1 - \hat{s}(\Delta\hat{\tau} + 1)]}{\Delta\hat{\tau}^2}[\hat{\tau}_w W + \hat{\tau}_a T_a]_{t-1}, \quad (6)$$

where

$\Delta$ = sampling time interval,
$\tau_{EL} = H_{EL}A_{EL}/\rho cv$,
$\tau_O = m/v$,
$\tau_{BE} = H_{BE}A_{BE}/\rho cv$,
$\tau_{SP} = Q/v$,
$\tau = \tau_{EL} + \tau_O + \tau_{BE} + \tau_{SP}$,
$s = \mathrm{EXP}(-\Delta\tau)$,
$\hat{\tau}_w = A_f/C_{EL}$,
$\hat{\tau}_a = H_{EL}A_{EL}/C_{EL}$,
$\hat{\tau} = \hat{\tau}_w + \hat{t}_a$, and
$\hat{s} = \mathrm{EXP}(-\Delta\hat{\tau})$,

and the subscripts $t$ and $t-1$ indicate that the temperature is evaluated at times $t = n\Delta$ and $t = (n-1)\Delta$.

The calculation of the humidity ratio in the space is also formulated in terms of a first-order linear differential equation in time similar to eqs. (3) and (4). This equation is also recast in the form of a time series [see eqs. (5) and (6)]. Knowing the humidity ratio and the dry-bulb air temperature, we can find the relative humidity by employing standard psychometric formulae.[6]

As previously noted in Section II, the system of eqs. (1), (5), and (6) permits controlling the dry-bulb air temperature in two basically different ways, intermittent operation and proportional control, and

determines the hours of operation. We use these equations for intermittent HVAC operation where we assume that the HVAC system holds the supply and return air-temperature difference constant at $DT_{SP} = -20$ degrees F whenever cooling is required and at $DT_{SP} = 40$ degrees F whenever heating is required. The hours of operation, and hence the total quantity of heat removed or added to satisfy the imposed dry-bulb temperature standard, can then be determined. For the proportional control plan, eqs. (1), (5), and (6) are also used to calculate not only the hours of HVAC operation but also the numerical value of the supply and return air-temperature difference $DT_{SP}(t)$, which in general varies continuously for this mode of control. The variation in time of the numerical value of $DT_{SP}(t)$ is determined by just satisfying the instantaneous building environmental load. When the HVAC system is activated, the proportional control plan closely follows the lower limit (for heating) or the upper limit (for cooling) of the building wideband temperature range.

Once $DT_{SP}(t)$ and the hours of operation are known, the program calculates the heat added or removed from the building by the HVAC system during every quarter hour and for whatever other period is of interest, e.g., monthly. As a corollary, we can estimate the environmental control system energy use, assuming the following HVAC system characteristics:

1. For chiller operation, a constant Coefficient Of Performance (COP) supplied by the user, together with the quantity of heat removed from the building air, characterizes its energy requirements.

Table I—Equipment buildings analyzed

| Energy Consumption and Control | | | | |
|---|---|---|---|---|
| | Wideband Temperature (°F) | | | |
| Case | Occupied Times | Unoccupied Times | Control | Geographic Location |
| 1 | 65–80 | 65–80 | Intermittent | New York City |
| 2 | 65–80 | 65–80 | Proportional | New York City |
| 3 | 65–80 | 60–85 | Intermittent | New York City |
| 4 | 65–80 | 65–80 | Intermittent | New Orleans |
| 5 | 65–80 | 65–80 | Intermittent | Phoenix |

| Building Parameters | |
|---|---|
| Factor | Parameter |
| Size (L × W × H) | 60 ft × 40 ft × 13 ft |
| Average heat transmission | U = 0.25 Btu/hr − ft$^2$ |
| Occupancy time | 8 a.m. to 6 p.m. |
| Fan support rate | 7400 CFM |
| Ventilation capacity | 150 CFM |
| Static fan pressure | 2 in. of water |
| Internal heat load | 15W/ft$^2$ |
| Economy cycle temperature limit | ≤65 degrees F |

Table IIa—Intermittent control of building space air temperature—New York City, case 1 in Table I

| Month | Degree-Days | | Space Temp | | Max Load (tons) | | Total Load (MBtu) | | Number of Hours | | | Heating (kWh Elect) | Cooling (kWh) | | Water (gal) to Maintain 20% Min RH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Heat | Cool | Min | Max | Heat | Cool | Heat | Cool | Heat | Cool | Econ | | No Econ | Econ | |
| Jan | 858 | 0 | 76.1 | 80.5 | 0.0 | 13.3 | 0.0 | 49.2 | 0 | 319 | 314 | 0.0 | 5291.2 | 1229.3 | 1352.7 |
| Feb | 797 | 0 | 76.3 | 80.5 | 0.0 | 13.3 | 0.0 | 44.2 | 0 | 256 | 281 | 0.0 | 4754.4 | 1114.7 | 1143.5 |
| Mar | 703 | 0 | 76.5 | 80.5 | 0.0 | 13.3 | 0.0 | 56.1 | 0 | 361 | 356 | 0.0 | 6021.7 | 1391.1 | 1026.2 |
| Apr | 358 | 0 | 76.6 | 80.6 | 0.0 | 13.6 | 0.0 | 65.1 | 0 | 414 | 334 | 0.0 | 6967.9 | 2582.9 | 321.7 |
| May | 118 | 38 | 76.6 | 80.6 | 0.0 | 13.9 | 0.0 | 77.5 | 0 | 486 | 179 | 0.0 | 8271.2 | 5901.2 | 101.2 |
| Jun | 23 | 147 | 76.6 | 80.9 | 0.0 | 14.2 | 0.0 | 83.1 | 0 | 512 | 36 | 0.0 | 8833.2 | 8348.4 | 0.0 |
| Jul | 0 | 325 | 76.7 | 80.8 | 0.0 | 14.2 | 0.0 | 92.6 | 0 | 563 | 0 | 0.0 | 9819.7 | 9819.7 | 0.0 |
| Aug | 0 | 268 | 76.7 | 81.1 | 0.0 | 14.2 | 0.0 | 90.4 | 0 | 550 | 0 | 0.0 | 9590.8 | 9587.5 | 0.0 |
| Sep | 32 | 116 | 76.8 | 80.9 | 0.0 | 14.1 | 0.0 | 80.1 | 0 | 495 | 42 | 0.0 | 8521.6 | 7960.9 | 20.2 |
| Oct | 192 | 21 | 76.4 | 80.6 | 0.0 | 13.9 | 0.0 | 71.7 | 0 | 450 | 207 | 0.0 | 7659.2 | 4928.8 | 79.3 |
| Nov | 625 | 0 | 76.1 | 80.6 | 0.0 | 13.6 | 0.0 | 53.7 | 0 | 344 | 306 | 0.0 | 5755.8 | 1764.3 | 1124.1 |
| Dec | 793 | 0 | 76.4 | 80.5 | 0.0 | 13.4 | 0.0 | 50.2 | 0 | 323 | 292 | 0.0 | 5385.9 | 1593.2 | 1245.5 |
| Totals | 4499 | 915 | 76.1 | 81.1 | 0.0 | 14.2 | 0.0 | 814.0 | 0 | 5107 | 2351 | 0.0 | 86873.0 | 56222.0 | 6414.0 |

Notes: Fan supply rate = 7400 CFM, ventilation = 150 CFM, wideband temperature limits for occupied and unoccupied times = 65° to 80°F, economizer temperature limit = 65°F, time period = 1 to 365 days, total hours = 8760.

2. For fan operation, fan power is calculated from the following equation:

$$\text{Fan power (kW)} = 2.487 \times 10^{-4} \times Q \times \Delta P, \tag{7}$$

where

$Q(CFM)$ = air flow delivered by the fan; and
$\Delta P$ = static pressure head of the fan in inches of water.

By multiplying the fan power by the total hours of fan operation, we can obtain the total energy use (kWh).

3. Humidification costs are based on supplying energy at the rate of 1000 Btu per pound of water added to the supply air stream. Costs are derived from the unit cost of energy, such as electricity ($/kWh), natural gas ($/1000 ft$^3$), and fuel oil ($/gal), which is supplied by the user. An 80 percent efficiency rate is assumed for these energy sources.

4. Heating costs are similarly calculated by the unit cost. An 80 percent efficiency rate is also used in these calculations.

## V. ILLUSTRATIVE EXAMPLES OF ENERGY CONSUMPTION AND CONTROL

Different geographic loctions of equipment buildings, dual or extended wideband temperature limits, and the method of HVAC control (intermittent or porportional) are considered in several variations. Table I gives this information along with some of the more salient building parameters. The results of the calculation are summarized by month in Tables II through VI for the cases specified in Table I. We assume here that a conventional cooling system, consisting of a chiller

Table IIb—Intermittent control of building space air temperature—New York City, case 1 in Table I

| | Number of Hours at Specified Relative Humidity (No Humidity Control) | | | | | |
|---|---|---|---|---|---|---|
| | <10% | 10–15% | 15–20% | 20–55% | 55–60% | >60% |
| Conv (no econ) | 697.00 | 1090.00 | 1086.75 | 5886.25 | 0.0 | 0.0 |
| Economy | 815.25 | 1048.75 | 1029.75 | 5866.25 | 0.0 | 0.0 |
| Enthalpy | 697.50 | 1101.00 | 1144.00 | 5817.50 | 0.0 | 0.0 |

| | Estimated Operating Cost for Cooling at $0.10/kWh for Electricity (Chiller COP = 3.50) | |
|---|---|---|
| Conv (no econ) | $9687 for 86870 kWh | (Fans = 18669 kWh, chiller = 68201 kWh) |
| Economy | $8622 for 56221 kWh | (Fans = 18669 kWh, chiller = 37552 kWh) |
| Enthalpy | $7534 for 75342 kWh | (Fans = 18669 kWh, chiller = 56673 kWh) |

| | Estimated Operating Cost for Humidification (20% min) and Heating at $0.10/kWh for Electricity |
|---|---|
| Humidification | $1959 for 19580 kWh |
| Heating | $0 for 0 kWh |

Notes: Min space temp occurred on day 300, max space temp occurred on day 213, max cooling load occurred on day 224.

Table IIIa—Proportional control of building space air temperature—New York City, case 2 in Table I

| Month | Degree-Days | | Space Temp | | Max Load (tons) | | Total Load (MBtu) | | Number of Hours | | | Heating (kWh Elect) | Cooling (kWh) | | Water (gal) to Maintain 20% Min RH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Heat | Cool | Min | Max | Heat | Cool | Heat | Cool | Heat | Cool | Econ | | No Econ | Econ | |
| Jan | 8.58 | 0 | 78.9 | 80.0 | 0.0 | 7.8 | 0.0 | 44.7 | 0 | 744 | 741 | 0.0 | 6465.4 | 2734.1 | 4261.7 |
| Feb | 797 | 0 | 79.9 | 80.0 | 0.0 | 8.3 | 0.0 | 40.4 | 0 | 672 | 667 | 0.0 | 5839.7 | 2487.0 | 3737.9 |
| Mar | 703 | 0 | 79.9 | 80.0 | 0.0 | 8.5 | 0.0 | 52.6 | 0 | 744 | 744 | 0.0 | 7126.1 | 2719.8 | 2885.7 |
| Apr | 358 | 0 | 79.9 | 80.0 | 0.0 | 10.0 | 0.0 | 62.3 | 0 | 720 | 652 | 0.0 | 7855.6 | 3182.1 | 837.9 |
| May | 118 | 38 | 79.9 | 80.0 | 0.0 | 11.4 | 0.0 | 75.4 | 0 | 744 | 434 | 0.0 | 9034.2 | 5461.4 | 262.5 |
| Jun | 23 | 147 | 79.9 | 80.0 | 0.0 | 12.2 | 0.0 | 81.4 | 0 | 720 | 139 | 0.0 | 9451.8 | 8252.0 | 0.0 |
| Jul | 0 | 325 | 79.9 | 80.0 | 0.0 | 13.0 | 0.0 | 91.0 | 0 | 744 | 0 | 0.0 | 10342.3 | 10342.3 | 0.0 |
| Aug | 0 | 268 | 79.9 | 80.0 | 0.0 | 12.3 | 0.0 | 88.9 | 0 | 744 | 10 | 0.0 | 10172.1 | 10080.5 | 0.0 |
| Sep | 32 | 116 | 79.9 | 80.0 | 0.0 | 11.4 | 0.0 | 78.2 | 0 | 720 | 135 | 0.0 | 9187.3 | 8078.6 | 76.3 |
| Oct | 192 | 21 | 79.9 | 80.0 | 0.0 | 11.2 | 0.0 | 69.2 | 0 | 744 | 514 | 0.0 | 8520.7 | 4672.8 | 241.7 |
| Nov | 625 | 0 | 79.9 | 80.0 | 0.0 | 8.6 | 0.0 | 50.1 | 0 | 720 | 696 | 0.0 | 6828.1 | 2797.4 | 3312.4 |
| Dec | 793 | 0 | 79.9 | 80.0 | 0.0 | 8.5 | 0.0 | 46.1 | 0 | 744 | 731 | 0.0 | 6584.7 | 2806.8 | 4105.5 |
| Totals | 4499 | 915 | 79.9 | 80.0 | 0.0 | 13.0 | 0.0 | 780.3 | 0 | 8960 | 5466 | 0.0 | 97408.0 | 63615.0 | 19721.0 |

Notes: Fan supply rate = 7400 CFM, ventilation = 150 CFM, wideband temperature limits for occupied and unoccupied times = 65° to 80°F, economizer temperature limit = 65°F, time period = 1 to 365 days, total hours = 8760.

and air-handling unit, provides cooling. By comparing the different cases, we find some interesting results.

Case 1 differs from case 2 in that the HVAC is intermittently controlled in case 1, but proportionally controlled in case 2. We see from Tables IIb and IIIb, for example, that the maximum cooling loads for both cases occur close in time [August 12 (day 224) and July 31 (day 212)]. However, the maximum cooling load of 13 tons for the proportional control plan (Table IIIa), which compares favorably with the load of 14.2 tons for the intermittent control plan (Table IIa), reduces the required size of the chiller plant by 9 percent. We would expect such a reduction from using a control sequence that follows the load closely and minimally overshoots the dry-bulb air temperature. Also, a control plan that matches the fan capacity to the load would compare favorably in energy use with on-off fan operation.

We can see in Table IIIa (in the column labeled "Space Temp") that the proportional control plan regulates the temperature to within one-tenth of a degree of the wideband temperature limit for the entire calendar year. For the economy cycle operation, the yearly electrical use of case 1 in Table IIb is 56,221 kWh, and that of case 2 in Table IIIb is 63,614 kWh. The chiller energy consumption for case 1, 37,522 kWh, is larger than that for use 2, 31,591 kWh. The proportional control plan, which modulates the air-supply temperature, requires the fan to run continuously at maximum power for the entire year. This maximum use of the fan creates larger overall energy requirements in spite of lower chiller energy use. However, a variable-air-volume system that modulates the fan supply rate to match loads should decrease the required fan power and significantly reduce total energy use.

Table IIIb—Proportional control of building space air temperature—New York City, case 2 in Table I

| | Number of Hours at Specified Relative Humidity (No Humidity Control) | | | | | |
|---|---|---|---|---|---|---|
| | <10% | 10–15% | 15–20% | 20–55% | 55–60% | >60% |
| Conv (No Econ) | 869.75 | 1090.75 | 1118.75 | 4393.25 | 1011.25 | 276.50 |
| Economy | 911.25 | 1092.50 | 1110.50 | 4354.00 | 1015.00 | 276.75 |
| Enthalpy | 869.75 | 1095.50 | 1134.50 | 4337.75 | 938.75 | 383.75 |

| Estimated Operating Cost for Cooling at $0.10/kWh for Electricity (Chiller COP = 3.50) | | |
|---|---|---|
| Conv (no econ) | $9741 for 97400 kWh | (Fans = 32023 kWh, chiller = 65383 kWh) |
| Economy | $6361 for 63614 kWh | (Fans = 32023 kWh, chiller = 31591 kWh |
| Enthalpy | $8930 for 88296 kWh | (Fans = 32023 kWh, chiller = 56273 kWh) |

| Estimated Operating Cost for Humidification (20% min) and Heating at $0.10/kWh for Electricity | |
|---|---|
| Humidification | $6022 for 60219 kWh |
| Heating | $0 for 0 kWh |

Notes: Min space temp occurred on day 100, max space temp occurred on day 10, max cooling load occurred on day 212.

Table IVa—Intermittent control of building space air temperature—New York City, case 3 in Table I

| Month | Degree-Days | | Space Temp | | Max Load (tons) | | Total Load (MBtu) | | Number of Hours | | | Heating (kWh Elect) | Cooling (kWh) | | Water (gal) to Maintian 20% Min RH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Heat | Cool | Min | Max | Heat | Cool | Heat | Cool | Heat | Cool | Econ | | No Econ | Econ | |
| Jan | 858 | 0 | 74.7 | 85.4 | 0.0 | 13.2 | 0.0 | 46.8 | 0 | 304 | 299 | 0.0 | 5033.1 | 1171.1 | 1590.2 |
| Feb | 797 | 0 | 75.1 | 85.4 | 0.0 | 13.3 | 0.0 | 42.2 | 0 | 274 | 268 | 0.0 | 4534.1 | 1074.7 | 1341.3 |
| Mar | 703 | 0 | 76.0 | 85.4 | 0.0 | 13.3 | 0.0 | 53.8 | 0 | 347 | 343 | 0.0 | 5776.8 | 1326.7 | 1302.5 |
| Apr | 358 | 0 | 76.3 | 85.6 | 0.0 | 13.6 | 0.0 | 62.8 | 0 | 400 | 326 | 0.0 | 6725.0 | 2452.0 | 446.6 |
| May | 118 | 38 | 76.3 | 85.6 | 0.0 | 13.9 | 0.0 | 75.4 | 0 | 473 | 225 | 0.0 | 8047.1 | 5073.1 | 158.9 |
| Jun | 23 | 147 | 76.3 | 85.6 | 0.0 | 14.1 | 0.0 | 81.3 | 0 | 502 | 70 | 0.0 | 8643.7 | 7711.2 | 0.0 |
| Jul | 0 | 325 | 76.3 | 85.7 | 0.0 | 14.2 | 0.0 | 90.7 | 0 | 553 | 0 | 0.0 | 9625.4 | 9625.4 | 0.0 |
| Aug | 0 | 268 | 76.2 | 85.8 | 0.0 | 14.2 | 0.0 | 88.6 | 0 | 541 | 3 | 0.0 | 9406.7 | 9356.9 | 0.0 |
| Sep | 32 | 116 | 75.7 | 85.6 | 0.0 | 14.1 | 0.0 | 78.3 | 0 | 485 | 72 | 0.0 | 8336.3 | 7384.6 | 45.6 |
| Oct | 192 | 21 | 76.0 | 85.5 | 0.0 | 13.9 | 0.0 | 69.5 | 0 | 437 | 245 | 0.0 | 7422.9 | 4189.6 | 148.0 |
| Nov | 625 | 0 | 75.1 | 85.5 | 0.0 | 13.6 | 0.0 | 51.7 | 0 | 332 | 307 | 0.0 | 5545.3 | 1556.7 | 1332.1 |
| Dec | 793 | 0 | 75.5 | 85.4 | 0.0 | 13.4 | 0.0 | 48.0 | 0 | 310 | 285 | 0.0 | 5152.6 | 1457.1 | 1429.6 |
| Totals | 4499 | 915 | 74.7 | 85.8 | 0.0 | 14.2 | 0.0 | 789.0 | 0 | 4962 | 2447 | 0.0 | 84249.0 | 52379.0 | 7795.0 |

Notes: Fan supply rate = 7400 CFM, ventilation = 150 CFM, wideband temperature limits for occupied times = 65° to 80°F, wideband temperature limits for unoccupied times = 60° to 85°F, economizer temperature limit = 65°F, time period = 1 to 365 days, total hours = 8760.

Tables IVa and IVb display monthly energy use for case 3 of Table I. Differing from case 1, this plan imposes dual wideband temperature limits. The wideband temperature limits for unoccupied times increase to 60 degrees F and 85 degrees F. Tables IVa and IVb show that this simple change with economy cycle operation reduces annual cooling energy use by 7 percent, from 56,222 kWh to 52,379 kWh. We can attribute this saving mainly to the lower chiller energy requirement, from 37,552 kWh to 34,237 kWh, and to a lesser extent to the smaller fan energy requirements, from 18,669 kWh to 18,141 kWh.

Tables V and VI show the results of the simulation for the buildings located in New Orleans and Phoenix, cases 4 and 5 of Table I. We can see that the maximum chiller load, 14.4 tons, is the same for these diverse locations. The total energy required in these buildings for economy cycle cooling is also nearly equal, 85,027 kWh and 85,123 kWh. Since the cooling load includes both sensible and latent energy, we can surmise from the degree-day totals that the dominant load on the system for Phoenix is sensible heat, and on that for New Orleans latent heat. The distribution of the relative humidity and the costs for humidification shown in Tables Vb and VIb tends to support these observations.

We can see the advantages in running the TELBECC program to compare different control plans. For example, although the intermittent control plan appears to use less energy overall, the proportional control plan actually reduces the size of the chiller plant by 9 percent in the same locale under similar conditions. The higher overall costs can be attributed to continuous operation of the fan, which, if operated to match the load, would consume much less power and make the

Table IVb—Intermittent control of building space air temperature—New York City, case 3 in Table I

| Number of Hours at Specified Relative Humidity (No Humidity Control) | | | | | | |
|---|---|---|---|---|---|---|
| | <10% | 10–15% | 15–20% | 20–55% | 55–60% | >60% |
| Conv (No Econ) | 857.50 | 1205.75 | 1213.00 | 5483.75 | 0.0 | 0.0 |
| Economy | 976.75 | 1115.75 | 1138.25 | 5529.25 | 0.0 | 0.0 |
| Enthalpy | 857.50 | 1241.00 | 1212.00 | 5449.50 | 0.0 | 0.0 |

| Estimated Operating Cost for Cooling at $0.10/kWh for Electricity (Chiller COP = 3.50) | | |
|---|---|---|
| Conv (no econ) | $8425 for 84247 kWh | (Fans = 18141 kWh, chiller = 66106 kWh) |
| Economy | $5238 for 52378 kWh | (Fans = 18141 kWh, chiller = 34237 kWh) |
| Enthalpy | $7278 for 72776 kWh | (Fans = 18141 kWh, chiller = 54635 kWh) |

| Estimated Operating Cost for Humidification (20% min) and Heating at $0.10/kWh for Electricity | |
|---|---|
| Humidification | $2380 for 23802 kWh |
| Heating | $0 for 0 kWh |

Notes: Min space temp occurred on day 22, max space temp occurred on day 224, max cooling load occurred on day 224.

Table Va—Intermittent control of building space air temperature—New Orleans, case 4 in Table I

| Month | Degree-Days | | Space Temp | | Max Load (tons) | | Total Load (MBtu) | | Number of Hours | | | Heating (kWh Elect) | Cooling (kWh) | | Water (gal) to Maintain 20% Min RH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Heat | Cool | Min | Max | Heat | Cool | Heat | Cool | Heat | Cool | Econ | | No Econ | Econ | |
| Jan | 502 | 0 | 76.4 | 80.6 | 0.0 | 13.5 | 0.0 | 60.8 | 0 | 388 | 341 | 0.0 | 6518.0 | 2044.9 | 386.5 |
| Feb | 465 | 0 | 76.4 | 80.6 | 0.0 | 13.8 | 0.0 | 55.9 | 0 | 356 | 287 | 0.0 | 5984.1 | 2231.2 | 865.2 |
| Mar | 204 | 8 | 76.6 | 80.7 | 0.0 | 14.0 | 0.0 | 72.8 | 0 | 458 | 223 | 0.0 | 7776.9 | 4836.3 | 74.3 |
| Apr | 21 | 133 | 76.6 | 80.9 | 0.0 | 14.2 | 0.0 | 81.1 | 0 | 499 | 36 | 0.0 | 8624.8 | 8136.8 | 0.2 |
| May | 0 | 296 | 76.7 | 81.1 | 0.0 | 14.3 | 0.0 | 91.7 | 0 | 555 | 2 | 0.0 | 9716.7 | 9690.2 | 0.0 |
| Jun | 0 | 459 | 76.8 | 81.1 | 0.0 | 14.4 | 0.0 | 95.7 | 0 | 570 | 0 | 0.0 | 10103.2 | 10103.2 | 0.0 |
| Jul | 0 | 457 | 76.8 | 81.1 | 0.0 | 14.4 | 0.0 | 98.3 | 0 | 583 | 0 | 0.0 | 10369.4 | 10369.4 | 0.0 |
| Aug | 0 | 477 | 76.7 | 80.8 | 0.0 | 14.4 | 0.0 | 98.6 | 0 | 586 | 0 | 0.0 | 10402.5 | 10402.5 | 0.0 |
| Sep | 0 | 397 | 76.8 | 81.0 | 0.0 | 14.4 | 0.0 | 92.8 | 0 | 553 | 0 | 0.0 | 9797.1 | 9797.1 | 0.0 |
| Oct | 25 | 147 | 76.7 | 80.9 | 0.0 | 14.1 | 0.0 | 83.9 | 0 | 516 | 31 | 0.0 | 8915.4 | 8503.2 | 0.0 |
| Nov | 123 | 62 | 76.6 | 80.9 | 0.0 | 14.2 | 0.0 | 73.5 | 0 | 456 | 141 | 0.0 | 7824.2 | 5953.5 | 83.7 |
| Dec | 383 | 1 | 76.1 | 80.6 | 0.0 | 13.9 | 0.0 | 64.6 | | 409 | 300 | 0.0 | 6909.0 | 2960.9 | 228.1 |
| Totals | 1723 | 2437 | 76.1 | 81.1 | 0.0 | 14.4 | 0.0 | 969.7 | 0 | 5934 | 1364 | 0.0 | 102941.0 | 85029.0 | 1638.0 |

Notes: Fan supply rate = 7400 CFM, ventilation = 150 CFM, wideband temperature limits for occupied and unoccupied times = 65° to 80°F, economizer temperature limit = 65°F, time period = 1 to 365 days, total hours = 8760.

Table Vb—Intermittent control of building space air temperature—New Orleans, case 4 in Table I

| Number of Hours at Specified Relative Humidity (No Humidity Control) | | | | | | |
|---|---|---|---|---|---|---|
| | <10% | 10–15% | 15–20% | 20–55% | 55–60% | >60% |
| Conv (No Econ) | 14.25 | 395.50 | 441.25 | 7909.00 | 0.0 | 0.0 |
| Economy | 60.0 | 383.00 | 474.75 | 7842.25 | 0.0 | 0.0 |
| Enthalpy | 14.25 | 419.00 | 455.25 | 7871.50 | 0.0 | 0.0 |

| Estimated Operating Cost for Cooling at $0.10/kWh for Electricity (Chiller COP = 3.50) | | |
|---|---|---|
| Conv (no econ) | $10294 for 102939 kWh | (Fans = 21693 kWh, chiller = 81246 kWh) |
| Economy | $8503 for 85027 kWh | (Fans = 21693 kWh, chiller = 63334 kWh) |
| Enthalpy | $9295 for 92949 kWh | (Fans = 21693 kWh, chiller = 71256 kWh) |

| Estimated Operating Cost for Humidification (20% min) and Heating at $0.10/kWh for Electricity | |
|---|---|
| Humidification | $500 for 5002 kWh |
| Heating | $ 0 for 0 kWh |

Notes: Min space temp occurred on day 345, max space temp occurred on day 147, max cooling load occurred on day 165.

proportional control plan much more attractive. We can conclude that TELBECC has great potential for pinpointing significant energy reductions and cost savings before a building's HVAC system is purchased.

## VI. CONCLUSIONS

The TELBECC program analyzes more efficiently and quickly than any method used heretofore the possible telephone building environmental energy use and control options. To pinpoint the most economical energy-conservation plan, the program analyzes multiple plans at minimal cost and with minimal expenditure of time. The program calculates the energy consumed every quarter hour by the HVAC in regulating the environment under changing weather conditions. It computes the required energy from the physical characteristics of the building envelope, such as the U factor, internal heat generation, geographic location, orientation of the building, and the dry-bulb temperature standard. In order to make it feasible to calculate by computer, we employ a simplified recursive computation procedure using time series. For each of the illustrative problems in Tables II through VI, the procedure produced monthly projections; yet it took less than 40 seconds to calculate results on an IBM/3033 computer. From the examples, we see the advantages and disadvantages of both the intermittent and proportional control plans, as well as the significant savings obtained from increasing the range of the dual or

Table VIa—Intermittent control of building space air temperature—Phoenix, Ariz., case 5 in Table I

| Month | Degree-Days | | Space Temp | | Max Load (tons) | | Total Load (MBtu) | | Number of Hours | | | Heating (kWh Elect) | Cooling (kWh) | | Water (gal) to Maintain 20% Min RH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Heat | Cool | Min | Max | Heat | Cool | Heat | Cool | Heat | Cool | Econ | | No Econ | Econ | |
| Jan | 432 | 0 | 76.3 | 80.6 | 0.0 | 13.5 | 0.0 | 62.8 | 0 | 401 | 300 | 0.0 | 6724.2 | 2812.0 | 1132.0 |
| Feb | 257 | 1 | 76.5 | 80.7 | 0.0 | 13.6 | 0.0 | 61.5 | 0 | 390 | 249 | 0.0 | 6578.6 | 3313.1 | 455.0 |
| Mar | 139 | 12 | 76.6 | 80.7 | 0.0 | 13.5 | 0.0 | 74.0 | 0 | 468 | 212 | 0.0 | 7908.0 | 5125.0 | 1705.1 |
| Apr | 38 | 122 | 76.6 | 81.1 | 0.0 | 13.7 | 0.0 | 79.6 | 0 | 497 | 104 | 0.0 | 8486.9 | 7105.4 | 336.4 |
| May | 16 | 403 | 76.7 | 81.2 | 0.0 | 13.9 | 0.0 | 92.1 | 0 | 570 | 38 | 0.0 | 9807.0 | 9301.8 | 64.9 |
| Jun | 0 | 560 | 76.7 | 81.1 | 0.0 | 14.1 | 0.0 | 95.8 | 0 | 589 | 2 | 0.0 | 10181.0 | 10148.0 | 256.0 |
| Jul | 0 | 847 | 76.9 | 81.2 | 0.0 | 14.4 | 0.0 | 109.5 | 0 | 657 | 0 | 0.0 | 11574.5 | 11574.5 | 0.0 |
| Aug | 0 | 679 | 76.7 | 81.2 | 0.0 | 14.4 | 0.0 | 104.1 | 0 | 626 | 0 | 0.0 | 11016.5 | 11016.5 | 0.0 |
| Sep | 0 | 521 | 76.8 | 80.8 | 0.0 | 14.2 | 0.0 | 94.4 | 0 | 574 | 0 | 0.0 | 10010.3 | 10010.3 | 0.0 |
| Oct | 14 | 201 | 76.6 | 81.0 | 0.0 | 14.0 | 0.0 | 84.5 | 0 | 525 | 86 | 0.0 | 9003.7 | 7860.2 | 12.0 |
| Nov | 196 | 0 | 76.5 | 80.6 | 0.0 | 13.6 | 0.0 | 67.8 | 0 | 428 | 236 | 0.0 | 7243.1 | 4138.2 | 161.3 |
| Dec | 413 | 0 | 76.5 | 80.6 | 0.0 | 13.4 | 0.0 | 63.0 | 0 | 401 | 307 | 0.0 | 6742.1 | 2719.4 | 458.0 |
| Totals | 1505 | 3346 | 76.3 | 81.2 | 0.0 | 14.4 | 0.0 | 989.0 | 0 | 6130 | 1536 | 0.0 | 105276.0 | 85124.0 | 4582.0 |

Notes: Fan supply rate = 7400 CFM, ventilation = 150 CFM, wideband temperature limits for occupied and unoccupied times = 65° to 80°F, economizer temperature limit = 65°F, time period = 1 to 365 days, total hours = 8760.

Table VIb—Intermittent control of building space air temperature—Phoenix, Ariz., case 5 in Table I

| | Number of Hours at Specified Relative Humidity (No Humidity Control) | | | | | |
|---|---|---|---|---|---|---|
| | <10% | 10–15% | 15–20% | 20–55% | 55–60% | >60% |
| Conv (No Econ) | 168.50 | 611.50 | 1430.00 | 6550.00 | 0.0 | 0.0 |
| Economy | 199.00 | 627.00 | 1432.25 | 6501.75 | 0.0 | 0.0 |
| Enthalpy | 185.50 | 648.50 | 1375.00 | 6551.00 | 0.0 | 0.0 |

| Estimated Operating Cost for Cooling at $0.10/kWh for Electricity (Chiller COP = 3.50) | | |
|---|---|---|
| Conv (no econ) | $10527 for 105274 kWh | (Fans = 22412 kWh, chiller = 82862 kWh) |
| Economy | $8512 for 85123 kWh | (Fans = 22412 kWh, chiller = 62711 kWh) |
| Enthalpy | $9205 for 92054 kWh | (Fans = 22412 kWh, chiller = 69642 kWh) |

| Estimated Operating Cost for Humdification (20% min) and Heating at $0.10/kWh for Electricity | |
|---|---|
| Humidification | $1399 for 13991 kWh |
| Heating | $0 for 0 kWh |

Notes: Min space temp occurred on day 18, max space temp occurred on day 225, max cooling load occurred on day 204.

extended wideband temperature limits for unoccupied times. In the larger view, we can understand how TELBECC can significantly contribute toward the operating companies' energy-conservation plan for future savings.

## ACKNOWLEDGMENT

The author wishes to express his gratitude to Mr. P. P. Gwozdz for the many fruitful and informative discussions during the development of the computer program.

## REFERENCES

1. C. Kalasinsky and F. Ferreira, "Energy Simulation with ESP-II," ASHRAE J., *23*, No. 8 (August 1981), pp. 26–9.
2. T. Kusuda, "A Comparison of Energy Calculation Procedures," ASHRAE J., *23*, No. 8 (August 1981), pp. 21–4.
3. *Test Reference Year (TRY) Manual*, National Climatic Center (Dept. of Commerce, National Oceanic and Atmospheric Admin., National Environmental Satellite, Data, and Information Service), Asheville, NC.
4. H. S. Carslaw and J. C. Jaeger, *Conduction of Heat in Solids*. 2d ed. London: Clarendon Press, 1959.
5. G. P. Mitalas and D. G. Stephenson, *Calculation of Heat Conduction Transfer Functions for Multi-Layer Slabs*," ASHRAE Publications, No. 2203, August 1971.
6. *ASHRAE Handbook & Product Directory, 1977 Fundamentals*, New York: American Society of Heating, Refrigerating, and Air-Conditioning Engineers, 1977.
7. P. J. Schneider, *Temperature Response Charts*, New York: John Wiley, 1963.
8. E. I. Jury, *Theory and Application of the Z-Transform Method*. Reprint. Melbourne, FL: R. E. Krieger, 1973.

## APPENDIX

### *Calculating Inside-Wall Temperature Time Series Coefficients $b_i$, $a_i$, $a_i'$ of Eq. (1)*

The basis for computing the time-series coefficients is the $z$ transform,[8] a discrete function transformation. This transformation is applied to time functions sampled at regular intervals of time. The $z$ transform has the same role in discrete systems that the Laplace transform has in continuous systems analysis.

Let us consider a continuous function of time $f(t)$. When the function is sampled at regular intervals $\Delta$, the output consists of a train of pulses, as illustrated in Fig. 1. We defined the $z$ transform of this output as a polynomial in powers of $z^{-1}$ in the following:

$$f(0) + f(\Delta)z^{-1} + f(2\Delta)z^{-2} + f(3\Delta)z^{-3} + \cdots . \tag{8}$$

The successive outputs of the sampler are the coefficients of the successive powers of $z^{-1}$ in the $z$ transform.

A linear system is characterized when its response to an elementary input (such as a pulse, a unit step, or, as will be adopted here, a unit ramp) is ascertained. This is nothing more than obtaining a transfer function of the system. If both input and output of the system are expressed in terms of their $z$ transforms, the ratio of output/input is the $z$ transform of the system. If we assume that such a transfer function, $G(z)$, can be found and that it can be expressed as the quotient of two polynomials in $z^{-1}$, then

$$G(z) = \frac{N(z)}{D(z)} = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + \cdots a_jz^{-j}}{b_0 + b_1z^{-1} + b_2z^{-2} + \cdots b_pz^{-p}}. \tag{9}$$

It follows that the $z$-transform of the output $O(z)$ resulting from an arbitrary input $I(z)$ is represented by

$$O(z) = G(z)I(z) \quad \text{or} \tag{10}$$

$$O(z)D(z) = N(z)I(z). \tag{11}$$

Since both sides of (11) are polynomials, the coefficients of the various powers of $z^{-1}$ must be the same on both sides of the equation. If, say, the coefficients of $z^{-n}$ are equated, eq. (11) yields

$$b_0O_n = a_0I_n + a_1I_{n-1} + a_2I_{n-2} + \cdots + a_jI_{n-j} - [b_1O_{n-1} + b_2O_{n-2} + \cdots + b_pO_{n-p}]. \tag{12}$$

The subscript $n$ on $O$ and $I$ indicates the value of the function at $t = n\Delta$; i.e., $O_n \equiv O(n\Delta)$, the coefficient of $z^{-n}$ in the $z$ transform of $O(z)$. This expression relates the output at any time $t = n\Delta$ to the input at that time and the input and output at earlier times. The coefficients $a_0, \cdots a_j$ and $b_0, \cdots b_p$ contain all the characteristics of

the system. With the properties of the $z$ transform described above, a method for determining the $z$ transform or time-series coefficients for the inside building wall temperature follows.

If we consider the outside building walls and roof structure as homogeneous flat slabs (Fig. 2), the temperature in the slab adheres to the following equations:

$$\kappa \frac{\delta^2 T_s}{\delta x^2} = \frac{\delta T_s}{\delta t},$$

$$T_s(x, 0) = 0,$$

$$k \frac{\delta T_s}{\delta x}(L, t) = -h_o[T_s(L, t) - T_o(t)],$$

$$k \frac{\delta T_s}{\delta x}(0, t) = h_i[T_s(0, t) - T_a(t)], \tag{13}$$

where

$T_s(x, t)$ (°F) = temperature in the slab,
$L$(ft) = slab thickness,
$k$(Btu/hr − ft − °F) = thermal conductivity,
$k \equiv \kappa/\rho c$(ft$^2$/hr) = diffusivity,
$\rho c$(Btu/ft$^3$) = volumetric heat capacity,
$h_o$(Btu/hr − ft$^2$ − °F) = outside-wall heat transfer coefficient,
$h_i$(Btu/hr − ft$^2$ − °F) = inside-wall heat transfer coefficient,
$T_o(t)$ = outside sol-air temperature, and
$T_a(t)$ = inside building air temperature.

It is convenient to use the Laplace transform

$$\bar{T}_s(x, p) = \int_0^\infty T_s(x, t) e^{-pt} dt$$

to eliminate the independent time variable $t$ in eq. (13). Then the solution for the inside wall surface ($x = 0$) temperature in terms of the transform parameter $p$ assumes the form:

$$\bar{T}_s(0, p) = \frac{h_i[kq \cosh(qL) + h_o \sinh(qL)]\bar{T}_a(p) + h_o kq \bar{T}_o(p)}{h_i[kq \cosh(qL) + h_o \sinh(qL)] + kq[kq \sinh(qL) + h_o \cosh(qL)]}, \tag{14}$$

where $q = (p/\kappa)^{1/2}$.

Letting $T_a(t)$ and $T_o(t)$ be unit ramp functions and inverting eq. (14) back to the real-time domain by using standard residue theory in the complex plane, the solution for $T_s(0, t)$, the temperature of the inside surface, is expressed as

$$T_s(0, t) = T_s^{(1)}(0, t) + T_s^{(2)}(0, t), \tag{15}$$

where $T_s^{(1)}(0, t)$ is the portion of the solution dependent on the outside sol-air temperature, and $T_s^{(2)}(0, t)$ the part dependent on the building space-air temperature. These temperatures are explicitly:

$$T_s^{(1)}(0, t) = B_O \left[ \frac{1}{B_i + B_O B_i + B_O} \left( t - \frac{L^2}{6\kappa} \frac{(3B_i + B_O B_i + 3B_O + 6)}{(B_i + B_O B_i + B_O)} \right) \right.$$

$$\left. - \frac{2L^2}{\kappa} \sum_{n=1}^{\infty} \frac{e^{-\alpha_n^2 \kappa t / L^2}}{\alpha_n^2 \{[(B_i + B_O B_i + B_i) - \alpha_n^2] \cdot \cos \alpha_n - \alpha_n (2 + B_O + B_i) \sin \alpha_n\}} \right], \tag{16}$$

and

$$T_s^{(2)}(0, t) = B_i \left[ \frac{1}{B_i + B_O B_i + B_O} \left( \frac{L^2}{6\kappa} (3 + B_O) \right. \right.$$

$$\left. \left. - \frac{L^2}{6\kappa} \frac{(1 + B_O)(3B_i + B_i B_O + 3B_O + 6)}{(B_i + B_O B_i + B_O)} + (1 + B_O)t \right) \right.$$

$$\left. - \frac{2L^2}{\kappa} \sum_{n=1}^{\infty} \frac{[\alpha_n \cos \alpha_n + B_O \sin \alpha_n] e^{-\alpha_n^2 \kappa t / L^2}}{\sigma_n^3 [(B_i + B_i B_O + B_O - \alpha_n^2] \cdot \cos \alpha_n - \alpha_n [2 + B_O + B_i] \sin \alpha_n} \right], \tag{17}$$

where $B_O = \dfrac{h_O L}{k}$, $B_i = \dfrac{h_i L}{k}$, and $\alpha_n$ are roots of the transcendental equation

$$\cot \alpha_n = \frac{\alpha_n^2 - B_O B_i}{\alpha_n (B_O + B_i)}, \qquad n = 1, 2, \cdots.$$

Equations (15) through (17) contain all the ingredients for forming the $z$-transform transfer functions for the inside-wall surface temperature. These are obtained by forming the ratio of output/input $z$ transforms as per eq. (9):

$$G^{(1)}(z) = \frac{T_s^{(1)}(0, z)}{T_O(z)} \quad \text{and} \quad G^{(2)}(z) = \frac{T_s^{(2)}(0, z)}{T_a(z)}. \tag{18}$$

$T_O(t)$ and $T_a(t)$ were taken as unit ramp functions, and therefore their $z$ transform from Ref. 8 is given as

$$T_O(z) = T_a(z) = \frac{\Delta}{z(1 - z^{-1})^2}. \tag{19}$$

The sampling interval is $\Delta$. The use of the input ramp function amounts to linear interpolation between the discrete values given by the $z$-transform coefficients.

The $z$ transforms of both $T_s^{(1)}(0, z)$ and $T_s^{(2)}(0, z)$ are similar in form and, with the aid of the table of $z$ transforms given in Ref. 8, can be expressed as

$$T_s^{(1,2)}(0, z) = \frac{A^{(1,2)}}{1 - z^{-1}} + \frac{B^{(1,2)}\Delta}{z(1 - z^{-1})^2} + \sum_{j=1}^{\infty} \frac{C_j^{(1,2)}}{1 - s_j z^{-1}}, \tag{20}$$

where

$$A^{(1)} = \frac{-B_O L^2}{6\kappa(B_i + B_O B_i + B_O)^2}(3B_i + B_O B_i + 3B_O + 6),$$

$$B^{(1)} = \frac{B_O}{B_i + B_O B_i + B_O},$$

$$C_j^{(1)} = -\frac{2L^2 B_O}{\kappa}\left(\frac{1}{\alpha_j^2\{[(B_i + B_O B_i + B_O) - \alpha_j^2] \cdot \cos\alpha_j - \alpha_j(2 + B_O + B_i)\sin\alpha_j\}}\right),$$

$$A^{(2)} = \frac{B_i}{B_i + B_O B_i + B_O}\left(\frac{L^2}{6\kappa}(3 + B_O) - \frac{L^2}{6\kappa}\frac{(1 + B_O)(3B_i + B_i B_O + 3B_O + 6)}{(B_i + B_O B_i + B_O)}\right),$$

$$B^{(2)} = \frac{B_i(1 + B_O)}{B_i + B_O B_i + B_O},$$

$$C_j^{(2)} = -\frac{2B_i L^2}{\kappa}\left(\frac{\alpha_j\cos\alpha_j + B_O\sin\alpha_j}{\alpha_j^3[(B_i + B_i B_O + B_O - \alpha_j^2)] \cdot \cos\alpha_j - \alpha_j(2 + B_O + B_i)\sin\alpha_j}\right), \quad \text{and}$$

$$s_j = e^{-\alpha_j^2 \Delta\kappa/L^2}$$

Equation (18) can now be expressed in the form of a ratio of polynomials in $z^{-1}$:

$$G^{(1,2)}(z) = \frac{N^{(1,2)}(z)}{D(z)}, \tag{21}$$

where, from the results of eqs. (19) and (20),

$$N^{(1,2)}(z) = \left(A^{(1,2)}\frac{z(1 - z^{-1})}{\Delta} + B^{(1,2)}\right)\prod_{j=1}^{\infty}(1 - s_j z^{-1}) + \frac{z(1 - z^{-1})^2}{\Delta}\sum_{n=1}^{\infty} C_n^{(1,2)}\prod_{j=n}^{\infty}(1 - s_j s^{-1}),$$

$$D(z) = \prod_{j=1}^{\infty}(1 - s_j z^{-1}).$$

Equation (21) is the form of eq. (9); consequently, the $b_i$ coefficients that are derived, as in eq. (12), from the coefficients of the polynomial $D(z)$ can be generated by a recursive scheme given as

$$b_0 = 1,\ b_i^{(1)} = 0,\ b_i^{(n+1)} = b_i^{(n)} - s_{i-n+1} b_{i-1}^{(n)}\ n = 1, \cdots N \qquad (22)$$

The number of $s_j$ terms needed to obtain the desired degree of accuracy for the $b_i$ coefficients is indicated by the index $n = N$, which in most instances should not exceed 20.

The $a_i$ and $a_i'$ coefficients in eq. (1) came from $N^{(1)}(z)$ and $N^{(2)}(z)$, respectively, by expanding these functions into polynomials in powers of $z^{-1}$; i.e.,

$$N^{(1)}(z) = a_1 z^{-1} + a_2 z^{-2} + \cdots$$

$$N^{(2)}(z) = a_1' z^{-1} + a_2' z^{-2} + \cdots$$

The desired coefficients are sorted out.

## AUTHOR

**Philip B. Grimado**, B.S. (Civil Engineering), 1961, City University of New York; M.S. (Applied Mechanics), 1962, Columbia University; Ph.D. (Applied Mechanics), 1968, Columbia University; Bell Laboratories, 1968—. Mr. Grimado's responsibilities include vulnerability studies of antiballistic missile systems, fire protection studies involving fire risk analyses, heat-transfer calculations, development of standard fire testing methods for operating company equipment, and development of algorithms for optimum control of building environmental equipment. He is currently engaged in developing computer software for automatic generation of optimum layout configurations of administrative office space.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Recursive Fixed-Order Covariance Least-Squares Algorithms

By M. L. HONIG*

(Manuscript received May 6, 1983)

This paper derives fixed-order recursive Least-Squares (LS) algorithms that can be used in system identification and adaptive filtering applications such as spectral estimation, and speech analysis and synthesis. These algorithms solve the sliding-window and growing-memory covariance LS estimation problems, and require less computation than both unnormalized and normalized versions of the computationally efficient order-recursive (lattice) covariance algorithms previously presented. The geometric or Hilbert space approach, originally introduced by Lee and Morf to solve the prewindowed LS problem, is used to systematically generate least-squares recursions. We show that combining subsets of these recursions results in prewindowed LS lattice and fixed-order (transversal) algorithms, and in sliding-window and growing-memory covariance lattice and transversal algorithms. The paper discusses both least-squares prediction and joint-process estimation.

## I. INTRODUCTION

Computationally efficient recursive Least-Squares (LS) algorithms have recently attracted attention in applications such as adaptive equalization,[1-4] echo cancellation,[5] and speech analysis and synthesis[6,7] because of their fast convergence properties when compared to older least-mean-square or gradient adaptation techniques.[8-10] Since the work on computationally efficient LS algorithms by Morf and others first appeared in Refs. 11 and 12, numerous papers have followed that produce computationally efficient algorithms that solve different types

* AT&T Information Systems.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

of autoregressive LS estimation problems.[13-17] In general, these algorithms fall into four categories: (1) prewindowed recursive LS, (2) sliding-window recursive LS, (3) growing-memory covariance recursive LS, and (4) nonrecursive LS algorithms. Each of the first three categories has two subcategories: fixed-order, or transversal, algorithms; and order-recursive, or lattice, algorithms.

References 1 and 12 present a prewindowed LS algorithm that satisfies a transversal filter structure (fast Kalman algorithm). Subsequent Refs. 7, 18, and 19 describe prewindowed and growing-memory covariance LS algorithms that satisfy a lattice structure. Normalized prewindowed LS lattice algorithms that involve fewer recursions than the original unnormalized versions, and which have the important advantage that all internal variables are less than or equal to unity in magnitude are presented in the more recent Ref. 13. Reference 14 extends the normalized lattice algorithms to solve the sliding-window and growing-memory covariance LS problems. The recursive algorithms mentioned so far require order $N$ arithmetic operations per iteration to update the filter parameters, where $N$ is the order of the filter. A computationally efficient order-recursive algorithm that solves the set of linear equations for the covariance LS prediction problem has been presented in Ref. 11, and extended to the joint-process-estimation case in Ref. 17. These algorithms require order $N^2$ operations to compute the LS prediction coefficients and are nonrecursive in the sense that the solution generated at time interval $i$ is not used to generate the solution at time interval $i + 1$.

This paper attempts to unify and extend the previous work by (1) systematically generating all of the recursions needed to derive all of the previously mentioned algorithms, and (2) using these recursions to derive new recursive fixed-order sliding-window and growing-memory covariance LS algorithms. These new algorithms solve directly for the prediction- or autogressive-model coefficients, and involve significantly less computation than both the unnormalized and normalized versions of the order-recursive or covariance lattice algorithms presented in Ref. 14. In addition, in some applications it may be advantageous to work directly with the prediction- or autogressive-model coefficients, rather than the set of reflection coefficients produced by lattice algorithms. The algorithms mentioned in the previous paragraph, along with the new ones derived here, are obtained by appropriately arranging subsets of least-squares recursions. The geometric or Hilbert space approach originally used by Lee and Morf[20] to derive the prewindowed LS lattice algorithm is used to derive all of the basic least-squares recursions in a cohesive manner. In this paper, however, only scalar-valued data are considered.

The next section defines the sliding-window and growing-memory

covariance LS problems to be solved. Then Section III reviews the geometric approach to LS estimation. Fundamental order and time updates for the least-squares projection operator are given in Section IV, with derivations in Appendix A. In Section V these projection updates systematically derive least-squares recursions. Section VI gives fixed-order covariance algorithms and Section VII extends the preceding discussion to the joint-process-estimation case. Appendix B lists subsets of recursions in Sections V and VII that constitute other LS algorithms.

## II. PROBLEM STATEMENT

We start by defining a sequence of data values $y_0, y_1, \cdots, y_i$, where $i$ is the current time index. A linear least-squares *forward predictor* of order $n$ chooses the coefficients $f_{j|n}$ to minimize

$$\epsilon_f(i|n) = \sum_{m=i'}^{i} \left( y_m - \sum_{j=1}^{n} f_{j|n} y_{m-j} \right)^2, \tag{1}$$

where $i'$ to $i$ is the time window of interest. The coefficients $f_{j|n}$, $1 \leq j \leq n$, are called the forward-prediction coefficients. A linear least-squares *backward predictor* of order $n$ chooses the backward-prediction coefficients $b_{j|n}$, $1 \leq j \leq n$, to minimize

$$\epsilon_b(i|n) = \sum_{m=i'}^{i} \left( y_{m-n} - \sum_{j=1}^{n} b_{j|n} y_{m-j+1} \right)^2. \tag{2}$$

Minimization of (1) rather than (2) is generally desired for a given application. The forward and backward prediction problems stated above are closely related, however, and the LS algorithms to be presented use the backward prediction coefficients to solve for the forward prediction coefficients in a computationally efficient manner.

If, instead of estimating future values of the *same* process, we wish to estimate another related process $\{x_i\}$, the least-squares cost function becomes

$$\epsilon_x(i|n) = \sum_{m=i'}^{i} \left( x_m - \sum_{j=1}^{n} c_{j|n} y_{m-j+1} \right)^2, \tag{3}$$

where tap coefficients $c_{j|n}$ replace the prediction coefficients $f_{j|n}$ and $b_{j|n}$. The cost function (3) is relevant to joint-process-estimation problems such as channel equalization and echo cancellation. In the case of channel equalization, $y_i$ is the $i$th sample of the channel output, and $x_i$ is the $i$th channel symbol.

Setting the derivatives of the cost functions (1), (2), and (3) with respect to the prediction (tap) coefficients equal to zero results in the

following linear equations:

$$\Phi_{i'-1,i-1|n}\mathbf{f}(i|n) = \sum_{j=i'}^{i} y_j \mathbf{y}_{j-1|n}, \tag{4a}$$

$$\Phi_{i',i|n}\mathbf{b}(i|n) = \sum_{j=i'}^{i} y_{j-n}\mathbf{y}_{j|n}, \tag{4b}$$

and

$$\Phi_{i',i|n}\mathbf{c}(i|n) = \sum_{j=i'}^{i} x_j \mathbf{y}_{j|n}, \tag{4c}$$

where

$$\mathbf{f}^T(i|n) \equiv [f_{1|n} f_{2|n} \cdots f_{n|n}], \tag{5a}$$

$$\mathbf{b}^T(i|n) \equiv [b_{1|n} b_{2|n} \cdots b_{n|n}], \tag{5b}$$

$$\mathbf{c}^T(i|n) \equiv [c_{1|n} c_{2|n} \cdots c_{n|n}], \tag{5c}$$

$$\mathbf{y}_{j|n}^T \equiv [y_j y_{j-1} \cdots y_{j-n+1}], \tag{6}$$

and the covariance matrix

$$\Phi_{i',i|n} \equiv \sum_{j=i'}^{i} \mathbf{y}_{j|n}\mathbf{y}_{j|n}^T. \tag{7}$$

Suppose now that $i' = 0$, and that $y_0$ is the first available data sample. The least-squares solutions for **f**, **b**, and **c**, obtained by solving (4), are undefined since they depend on the unspecified data values $y - 1$, $y - 2$, $\cdots$, $y_{-n}$. The simplest, and perhaps most popular, technique for circumventing this problem is to assume all data values $y_j$, $j < 0$, are zero. The least-squares solutions resulting from this so-called prewindowed estimation are then well defined as long as the covariance matrix is nonsingular. In applications such as speech modelling, however, where estimates of the prediction coefficient vector $\mathbf{f}(i|n)$ are desired given relatively few data samples, prewindowed estimation may result in undesirable edge effects from assuming data is zero outside a given window. For these types of applications, it is desirable to estimate the prediction coefficients without any assumptions concerning the data outside the time window of interest.

Covariance least-squares estimation replaces the lower time limit $i'$ in (1), (2), and (3) by $n$, so that only known data values are used to compute the LS prediction (tap) coefficients. The improved estimates so obtained are not without cost, however. The resulting covariance LS algorithms derived in this paper and elsewhere[7] involve more computation than prewindowed LS algorithms. Notice that at each

iteration $i$, the prediction coefficients are computed from $i + 1$ data values. Because the number of data samples entering the least-squares computation grows with time, this type of estimation has been called growing-memory covariance estimation.[16]

Finally, another windowing technique that has attracted attention recently is the sliding-window technique, in which the lower time limit $i'$ in (1) and (2) is replaced by $i - M + n + 1$, and in (3) by $i - N + n$, where $M$ is a predetermined constant. At each iteration the least-squares prediction coefficients are therefore computed from a fixed number ($M$) of data samples. Notice that data samples outside the time window $i - M + 1$ to $i$ have no effect on the least-squares solution for $\mathbf{f}$, $\mathbf{b}$, and $\mathbf{c}$ at time $i$, i.e., they are totally forgotten. This is in contrast to more conventional exponential forgetting techniques that reduce the effects of past data samples in a more continuous fashion.[16] The sliding window is therefore useful in applications where the autoregressive model changes abruptly with time, or where undesirable transients periodically affect the data samples. In the former case, when the model parameters change, the sliding window eventually discards data values corresponding to previous model parameters. In the latter case, the sliding window eventually discards corrupted data values.

Computationally efficient recursive algorithms that solve the growing-memory covariance and sliding-window LS estimation problems will be derived in Sections V through VII. The next section develops the necessary mathematical background by reviewing the geometric interpretation of linear least-squares estimation.

## III. MATHEMATICAL BACKGROUND

Given two vectors $\mathbf{X}$ and $\mathbf{Y}$ having the same dimension $i$, the inner product of $\mathbf{X}$ and $\mathbf{Y}$ is defined to be

$$\langle \mathbf{X}, \mathbf{Y} \rangle \equiv \mathbf{X}^T \mathbf{W} \mathbf{Y}, \tag{8}$$

where $\mathbf{W}$ is some prespecified $i \times i$ weighting matrix. As an example, a typical weighting matrix is the exponential weighting matrix

$$\mathbf{W}_i = [1 \; w \; w^2 \; \cdots \; w^{i-1}]\mathbf{I}, \tag{9}$$

where $\mathbf{I}$ is the $i \times i$ identity matrix. For convenience, we will assume that $\mathbf{W}$ is the identity matrix. Modification of the results in this paper to the case where $\mathbf{W}$ is arbitrary is straightforward. The distance between two vectors $\mathbf{X}$ and $\mathbf{Y}$ with the same dimension is therefore the regular Euclidean distance,

$$d(\mathbf{X}, \mathbf{Y}) = \| \mathbf{Y} - \mathbf{X} \| \equiv \langle \mathbf{Y} - \mathbf{X}, \mathbf{Y} - \mathbf{X} \rangle^{1/2}. \tag{10}$$

The ($n$th order) projection of a vector $\mathbf{Y}$ onto a subspace (or

manifold) $M$, which is spanned by the $n$ vectors $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n\}$, is denoted as $P_M\mathbf{Y}$. The orthogonal projection of $\mathbf{Y}$ onto $M$ is defined as

$$P_M^{\perp}\mathbf{Y} \equiv \mathbf{Y} - P_M\mathbf{Y}, \tag{11}$$

and is orthogonal to the subspace $M$. This implies that

$$\langle \mathbf{X}_j, \mathbf{Y} - P_M\mathbf{Y} \rangle = 0, \qquad \text{for} \quad j = 1, \cdots, n. \tag{12}$$

Since $P_M\mathbf{Y}$ lies in $M$, there exist constants, or regression coefficients $f_1, f_2, \cdots, f_n$ such that

$$P_M\mathbf{Y} = \sum_{j=1}^{n} f_j\mathbf{X}_j = \mathbf{S}\mathbf{f}, \tag{13}$$

where $\mathbf{S} = [\mathbf{X}_1 \cdots \mathbf{X}_n]$ and $\mathbf{f}^T = [f_1 \cdots f_n]$. Using (12) and (13), it is easy to show that

$$\mathbf{f} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{X} \tag{14}$$

and

$$P_M\mathbf{Y} = \mathbf{S}(\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{Y}, \tag{15}$$

assuming $\mathbf{S}^T\mathbf{S}$ is nonsingular.

The linear least-squares estimate of $\mathbf{Y}$, based upon the vectors $\mathbf{X}_1, \cdots, \mathbf{X}_n$, is formed by choosing $f_1, \cdots, f_n$ such that

$$\| \underline{\epsilon} \|^2 \equiv \| \mathbf{Y} - \sum_{j=1}^{n} f_j\mathbf{X}_j \|^2 \tag{16}$$

is minimized. Differentiating this quantity with respect to $f_j$ and setting the result equal to zero gives

$$\hat{\mathbf{Y}} \equiv \sum_{j=1}^{n} f_j\mathbf{X}_j = P_M\mathbf{Y}, \tag{17}$$

and the vector of estimation errors,

$$\underline{\epsilon} = \mathbf{Y} - \sum_{j=1}^{n} f_j\mathbf{X}_j = P_M^{\perp}\mathbf{Y}. \tag{18}$$

We have identified the operator $P$ as a least-squares projection.

## IV. PROJECTION-OPERATOR UPDATE FORMULAS

In this section some fundamental relationships satisfied by the least-squares projection operator are presented. These projection updates fall into two main categories: order updates and time updates. Under time updates are two subcategories, forward and backward time updates. We point out in advance that a total of three projection-operator updates will be used throughout this paper: one order update, one

forward time update, and one backward time update. In addition, one forward and one backward time update for inner products will be needed.

### *4.1 Order updates*

Given two vectors, $\mathbf{Y}$ and $\mathbf{X}$, and a linear space $M$ spanned by the vectors $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n$, all in $\mathbf{R}^i$, suppose we wish to calculate the least-squares estimate of $\mathbf{Y}$ based upon the vectors $\mathbf{X}_1, \cdots, \mathbf{X}_n$ and $\mathbf{X}$. In particular, we wish to find coefficients $a_j$, $j = 1, \cdots, n$, and $b$ such that $\| \mathbf{Y} - (\sum_{j=1}^{n} a_j \mathbf{X}_j + b\mathbf{X}) \|^2$ is minimized. From the discussion in the last section, we know that the least-squares estimate of $\mathbf{Y}$ is

$$\sum_{j=1}^{n} a_j \mathbf{X}_j + b\mathbf{X} = P_{\{M+\mathbf{X}\}}\mathbf{Y}, \tag{19}$$

where $\{M + \mathbf{X}\}$ denotes the space spanned by $M$ and $\mathbf{X}$. We can write the following orthogonal decomposition of the space $\{M + \mathbf{X}\}$,[21]

$$\{M + \mathbf{X}\} = M \oplus \{P_M^{\perp}\mathbf{X}\}. \tag{20}$$

By the Hilbert space projection theorem,[21] we have that for any vector $\mathbf{Y} \in \mathbf{R}^i$,

$$P_{\{M+\mathbf{X}\}}\mathbf{Y} = P_M\mathbf{Y} + P_{\{P_M^{\perp}\mathbf{X}\}}\mathbf{Y}. \tag{21}$$

Figure 1 illustrates this equation for the special case $n = 1$. The projection of $\mathbf{Y}$ onto the space spanned by two vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ is shown as the sum of the two projections $P_{X_1}\mathbf{Y}$ and $P_{\{P_{X_1}^{\perp}\mathbf{X}_2\}}\mathbf{Y}$.

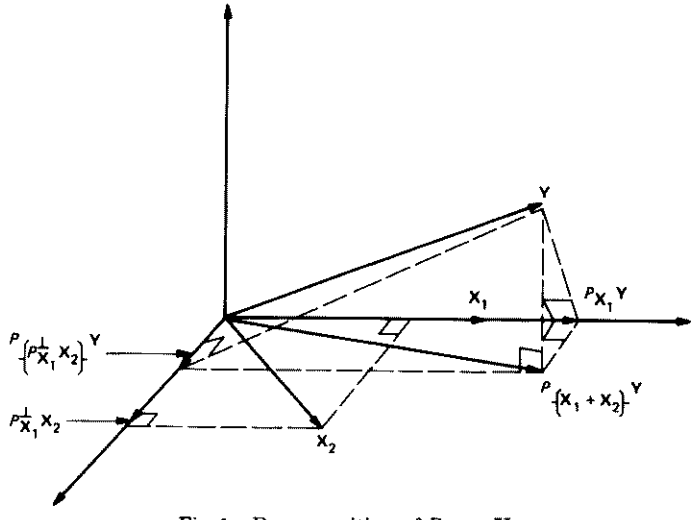



Fig. 1—Decomposition of $P_{\{\mathbf{X}_1+\mathbf{X}_2\}}\mathbf{Y}$.

Equation (21) constitutes a fundamental order update for the least-squares projection operator. The $(n+1)$st-order projection $P_{[M+\mathbf{X}]}$ is expressed as the sum of the $n$th order projection $P_M$ and the first order projection $P_{[P_M^{\perp}\mathbf{X}]}$. By subtracting both sides of (21) from $\mathbf{Y}$, we obtain the following order update for the *orthogonal* projection operator $P^{\perp}$,

$$P^{\perp}_{[M+\mathbf{X}]}\mathbf{Y} = P^{\perp}_{M}\mathbf{Y} - P_{[P^{\perp}_{M}\mathbf{X}]}\mathbf{Y}. \tag{22}$$

### 4.2 Forward time updates

The forward time updates derived in this section compute a least-squares projection at time $i$ given the same least-squares projection at time $i-1$. These recursions, when combined with the order recursions in the last subsection, can be used to derive prewindowed LS algorithms. We first consider the following vectors $\mathbf{X}_{i_0,i}$ and $\mathbf{Y}_{i_0,i}$, which are composed of data samples from time $i_0$ to $i$, i.e.,

$$\mathbf{X}^T_{i_0,i} = [x_i \; x_{i-1} \; \cdots \; x_{i_0}], \tag{23a}$$

and

$$\mathbf{Y}^T_{i_0,i} = [y_i \; y_{i-1} \; \cdots \; y_{i_0}]. \tag{23b}$$

For notational convenience, in this section only we will omit the lower time subscript on the data vectors and assume it to be $i_0$. Our objective is to compute the linear least-squares estimate of $\mathbf{Y}_i$, given $\mathbf{X}_i$ in terms of a least-squares estimate that does not use the most recent value $y_i$. With this in mind we define the unit vector

$$\mathbf{u}_i^T = [1 \; 0 \; \cdots \; 0 \; 0], \tag{24}$$

which has the same dimension as $\mathbf{Y}_i$, i.e., $\mathbf{u}_i \in \mathbf{R}^{i-i_0+1}$. Associated with $\mathbf{u}_i$ is the space spanned by $\mathbf{u}_i$, or the space of most recent data values, denoted as $U_i$. Note that $P_{U_i}\mathbf{Y}_i = y_i\mathbf{u}_i$. For notational convenience we define a tilde operator as follows,

$$\tilde{\mathbf{Y}}_i \equiv P^{\perp}_{U_i}\mathbf{Y}_i = [0 \; y_{i-1} \; y_{i-2} \; \cdots \; y_{i_0+1} \; y_{i_0}], \tag{25}$$

i.e., $\tilde{\mathbf{Y}}_i$ is the projection of $\mathbf{Y}_i$ onto the subspace of past data values.

The basic prediction problem is illustrated in Fig. 2, where $\mathbf{Y}_i$ is a vector having its endpoint in back of the plane of the paper and $\mathbf{X}_i$ has its endpoint in front of the plane of the paper. We are given the vector $\mathbf{X}_i$, from which the least-squares estimate of $\mathbf{Y}_i$, $P_{\mathbf{X}_i}\mathbf{Y}_i$, is to be recursively obtained. At time $i$ we therefore assume a regression coefficient $a$ computed at time $i-1$ (i.e., $P_{\mathbf{X}_{i-1}}\mathbf{Y}_{i-1} = a\mathbf{X}_{i-1}$, or equivalently, $P_{\tilde{\mathbf{X}}_i}\tilde{\mathbf{Y}}_i = a\tilde{\mathbf{X}}_i$), which we wish to modify using the most recent data values $y_i$ and $x_i$. Figure 2 therefore shows $P_{\mathbf{X}_i}\mathbf{Y}_i$ decomposed into the two vectors $a\mathbf{X}_i$ and $P_{\mathbf{X}_i}(\mathbf{Y}_i - a\mathbf{X}_i)$. Figure 3 illustrates the plane spanned by $\mathbf{X}_i$, $\tilde{\mathbf{X}}_i$, and $U_i$. Since ABC and ADE are similar

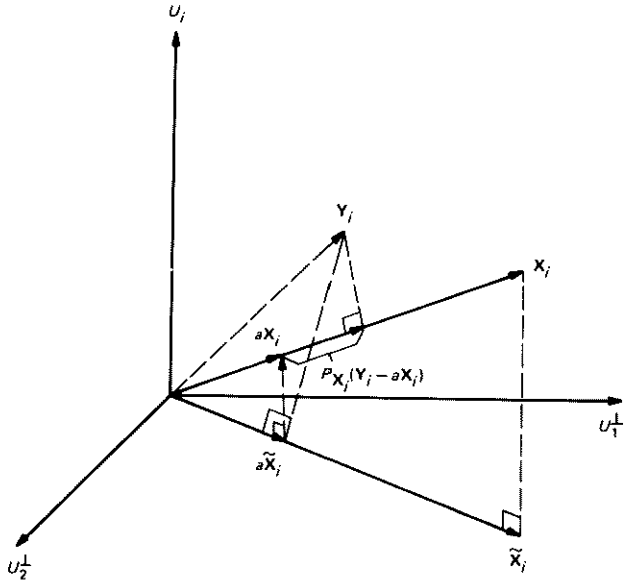



Fig. 2—Decomposition of $P_{\mathbf{X}_i}\mathbf{Y}_i$.

triangles,

$$\frac{\text{AB}}{\text{AD}} = \frac{\text{AC}}{\text{AE}} = a, \tag{26}$$

so that $\overline{\text{AC}} = a\mathbf{X}_i$. Figure 4 attempts to include vectors not shown in Fig. 2 and again illustrates the decomposition of $P_{\mathbf{X}_i}\mathbf{Y}_i$. (Only the endpoint of $\mathbf{Y}_i$ is in Fig. 4.)

Assume now that the vector $\mathbf{X}_i$ is replaced by the *subspace* $M_i$ spanned by the vectors $\mathbf{X}_{1,i}, \mathbf{X}_{2,i}, \cdots, \mathbf{X}_{n,i}$. Let

$$\mathbf{S}_i = [\mathbf{X}_{1,i}\ \mathbf{X}_{2,i}\ \cdots\ \mathbf{X}_{n,i}] \tag{27}$$

and

$$\tilde{\mathbf{S}}_i = [\tilde{\mathbf{X}}_{1,i}\ \tilde{\mathbf{X}}_{2,i}\ \cdots\ \tilde{\mathbf{X}}_{n,i}]. \tag{28}$$

We define the projection

$$P_{M_{i|i-1}}\mathbf{Y}_i \equiv \mathbf{S}_i[\tilde{\mathbf{S}}_i^T\tilde{\mathbf{S}}_i]^{-1}\tilde{\mathbf{S}}_i^T\tilde{\mathbf{Y}}_i = \mathbf{S}_i\mathbf{f}, \tag{29}$$

i.e., $P_{M_{i|i-1}}\mathbf{Y}_i$ lies in $M_i$, but uses regression coefficients computed at time $i-1$. Referring to Fig. 3, $P_{\mathbf{X}_{i|i-1}}\mathbf{Y}_i = a\mathbf{X}_i$. Appendix A shows that

$$P_{M_i}\mathbf{Y}_i = P_{M_{i|i-1}}\mathbf{Y}_i + P_{M_i}\mathbf{u}_i\langle \mathbf{u}_i, P_{M_i}^\perp\mathbf{Y}_i\rangle \sec^2\theta_i, \tag{30}$$

where

$$\sin^2\theta_i = \langle \mathbf{u}_i, P_{M_i}\mathbf{u}_i\rangle = \| P_{M_i}\mathbf{u}_i \|^2$$

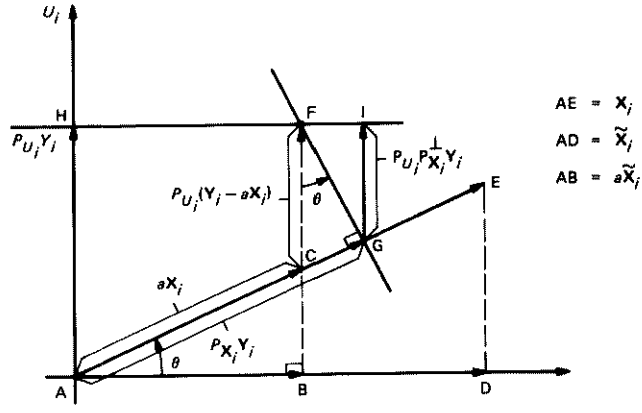



Fig. 3—Plane spanned by $\mathbf{X}_i$ and $\tilde{\mathbf{X}}_i$.

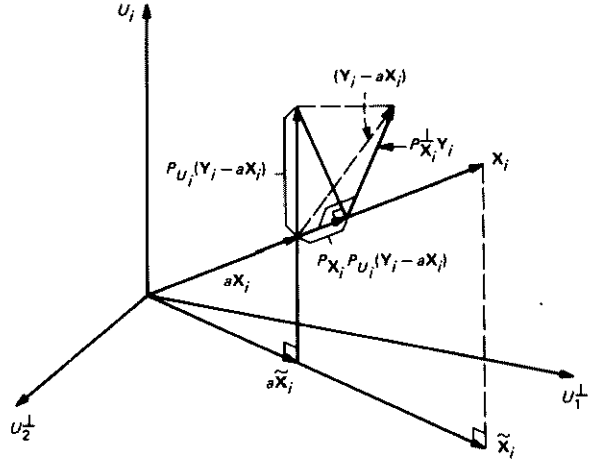



Fig. 4—Rotated view of Fig. 2.

$$= (\mathbf{u}_i^T\mathbf{S}_i)(\mathbf{S}_i^T\mathbf{S}_i)^{-1}(\mathbf{S}_i^T\mathbf{u}_i), \tag{31}$$

and

$$\sec^2\theta_i = \frac{1}{1 - \sin^2\theta_i}. \tag{32}$$

The variable $\theta_i$ can be interpreted as the angle between the spaces spanned by the matrices of basis vectors $\mathbf{S}_i$ and $\tilde{\mathbf{S}}_i$. Referring to Fig. 3, the angle $\theta$ is given by

$$\sin^2\theta = 1 - \frac{\|\tilde{\mathbf{X}}_i\|^2}{\|\mathbf{X}_i\|^2} = \frac{x_i^2}{\|\mathbf{X}_i\|^2}, \tag{33}$$

and measures the unexpectedness of the data received at time $i$. Notice that (33) can be rewritten as (31), where $M_i$ and $\mathbf{S}_i$ are replaced by $\mathbf{X}_i$.

We obtain the following time update for the orthogonal projection operator by subtracting both sides of (30) from $\mathbf{Y}_i$,

$$P_{M_i}^{\perp}\mathbf{Y}_i = P_{M_i|i-1}^{\perp}\mathbf{Y}_i - P_{M_i}\mathbf{u}_i\langle \mathbf{u}_i, P_{M_i}^{\perp}\mathbf{Y}_i\rangle \sec^2\theta_i. \tag{34}$$

One more relation that will be useful in the following section is a recursive equation for the inner product $\langle \mathbf{v}_i, P_{M_i}^{\perp}\mathbf{Y}_i\rangle$, where $\mathbf{v}_i$ is an arbitrary vector in $\mathbf{R}^{i-i_0+1}$. This recursion, which is derived in Appendix A, is

$$\langle \mathbf{v}_i, P_{M_i}^{\perp}\mathbf{Y}_i\rangle = \langle \tilde{\mathbf{v}}_i, P_{\tilde{M}_i}^{\perp}\tilde{\mathbf{Y}}_i\rangle + \langle \mathbf{u}_i, P_{M_i}^{\perp}\mathbf{v}_i\rangle\langle \mathbf{u}_i, P_{M_i}^{\perp}\mathbf{Y}_i\rangle \sec^2\theta_i, \tag{35}$$

where $\tilde{M}_i$ is the space spanned by $\tilde{\mathbf{S}}_i$.

### *4.3 Backward time updates*

Consider again the data vectors $\mathbf{X}_i$ and $\mathbf{Y}_i$ defined by (23). Suppose we wish to compute the linear least-squares estimate of $\mathbf{Y}_i$ given $\mathbf{X}_i$ in terms of a least-squares estimate that does not use the most *distant* or *past* values $y_{i_0}$ and $x_{i_0}$. Clearly, this problem can be solved in exactly the same fashion as the time-update problem stated at the beginning of the last section. By turning the vectors $\mathbf{Y}_i$ and $\mathbf{X}_i$ upside down, and assuming that $y_{i_0}$ and $x_{i_0}$ are the most recent samples, one can solve this problem by using time updates already derived. The same argument holds when $\mathbf{X}_i$ is replaced by the subspace $M_i$ spanned by vectors $\mathbf{X}_{1,i}, \mathbf{X}_{2,i}, \cdots, \mathbf{X}_{n,i}$. In this case we wish to calculate the projection $P_{M_i}\mathbf{Y}_i$ in terms of a projection onto the space spanned by the matrix of basis vectors $\mathbf{S}_i$ in which the *bottom row* has been replaced by zeros. This is in contrast to the previous time updates, which expressed $P_{M_i}\mathbf{Y}_i$ in terms of a projection onto the space spanned by $\mathbf{S}_i$ in which the *top* row has been replaced by zeroes (i.e., $\tilde{M}_i$).

In analogy with the notation defined in the last section, we define the unit vector

$$\mathbf{u}_{i_0}^T = [0\ 0\ \cdots\ 0\ 1] \in \mathbf{R}^{i-i_0+1}, \tag{36}$$

and the space spanned by $\mathbf{u}_{i_0}$ as $U_{i_0}$. We also define the following asterisk operator in analogy with the previous tilde operator,

$$\mathbf{Y}_i^* = P_{U_{i_0}}^{\perp}\mathbf{Y}_i = [y_i\ y_{i-1}\ \cdots\ y_{i_0+1}\ 0]^T. \tag{37}$$

Similarly,

$$\mathbf{S}_i^* = [\mathbf{X}_{1,i}^*\ \mathbf{X}_{2,i}^*\ \cdots\ \mathbf{X}_{n,i}^*]. \tag{38}$$

The projection of $\mathbf{Y}_i$ onto $M_i$ using regression coefficients computed from $\mathbf{S}_i^*$ is defined as

$$P_{M_{i_0|i_0+1}}\mathbf{Y}_i \equiv \mathbf{S}_i[\mathbf{S}_i^{*T}\mathbf{S}_i^*]^{-1}[\mathbf{S}_i^{*T}\mathbf{Y}_i]. \tag{39}$$

The regression coefficients that multiply the basis vectors of $M_i$ are in this case elements of the vector $[\mathbf{S}_i^{*T}\mathbf{S}_i^*]^{-1}[\mathbf{S}_i^{*T}\mathbf{Y}_i]$.

The derivation of (30) can be repeated with $\mathbf{u}_i$ replaced by $\mathbf{u}_{i_0}$, tildes replaced by asterisks, and $P_{M_{i|i-1}}$ replaced by $P_{M_{i_0|i_0+1}}$ to give the following projection decomposition,

$$P_{M_i}\mathbf{Y}_i = P_{M_{i_0|i_0+1}}\mathbf{Y}_i + P_{M_i}\mathbf{u}_{i_0}\langle \mathbf{u}_{i_0}, P_{M_i}^{\perp}\mathbf{Y}_i\rangle \sec^2\theta_i^*, \tag{40}$$

where

$$\begin{aligned}\sin^2\theta_i^* &= \| P_{M_i}\mathbf{u}_{i_0}\|^2 \\ &= (\mathbf{u}_{i_0}^T\mathbf{S}_i)(\mathbf{S}_i^T\mathbf{S}_i)^{-1}(\mathbf{S}_i^T\mathbf{u}_{i_0}) \\ &= \langle \mathbf{u}_{i_0}, P_{M_i}\mathbf{u}_{i_0}\rangle,\end{aligned} \tag{41}$$

and

$$\sec^2\theta_i^* = \frac{1}{1 - \sin^2\theta_i^*}. \tag{42}$$

Subtracting both sides of (40) from $\mathbf{Y}_i$ gives

$$P_{M_i}^{\perp}\mathbf{Y}_i = P_{M_{i_0|i_0+1}}^{\perp}\mathbf{Y}_i - P_{M_i}\mathbf{u}_{i_0}\langle \mathbf{u}_{i_0}, P_{M_i}^{\perp}\mathbf{Y}_i\rangle \sec^2\theta_i^*. \tag{43}$$

Finally, the following update for inner products is analogous to (35),

$$\langle \mathbf{v}_i, P_{M_i}^{\perp}\mathbf{Y}_i\rangle = \langle \mathbf{v}_i^*, P_{M_i^*}^{\perp}\mathbf{Y}_i^*\rangle + \langle \mathbf{u}_{i_0}, P_{M_i}^{\perp}\mathbf{v}_i\rangle\langle \mathbf{u}_{i_0}, P_{M_i}^{\perp}\mathbf{Y}_i\rangle \sec^2\theta_i^*. \tag{44}$$

This completes the presentation of projection-operator recursions needed to derive the least-squares recursions in Sections V and VII. All order updates for variables entering the least-squares algorithms to be presented can be derived from (22). Similarly, all forward and backward time updates for vectors entering these algorithms can be derived from (34) and (43), respectively, and all forward and backward time updates for inner products can be derived from (35) and (44), respectively.

## V. LEAST-SQUARES RECURSIONS

### *5.1 Notation*

Referring to the definition (23), a shift operator $z^{-j}$ is defined by

$$z^{-j}\mathbf{Y}_{i_0,i}^T = [y_{i-j}\ y_{i-j-1}\ \cdots\ y_{i_0-j}]. \tag{45}$$

Equations (1) and (2) can now be rewritten as

$$\epsilon_f(i\,|\,n) = \| \mathbf{Y}_{i_0+n,i} - \sum_{j=1}^{n} f_{j\,|\,n}(z^{-j}\mathbf{Y}_{i_0+n,i}) \|^2 \tag{46a}$$

and

$$\epsilon_b(i \mid n) = \| z^{-n}\mathbf{Y}_{i_0+n,i} - \sum_{j=1}^{n} b_{j\mid n}(z^{-j+1}\mathbf{Y}_{i_0+n,i}) \|^2, \tag{46b}$$

where $i'$ has been replaced by $i_0 + n$. A matrix of shifted data vectors is denoted as

$$\mathbf{S}_{i_0+n,i}(l, n) = [z^{-l}\mathbf{Y}_{i_0+n,i}\ z^{-l-1}\mathbf{Y}_{i_0+n,i} \cdots z^{-n}\mathbf{Y}_{i_0+n,i}], \tag{47}$$

where $l < n$. The space spanned by the columns of $\mathbf{S}_{i_0+n,i}(l, n)$, which is a subspace generated by past data values, is denoted as $M_{i_0+n,i}(l, n)$. For notational convenience we will omit the lower time index of $\mathbf{S}$ and $M$ and assume that it is always $i_0 + n$. Notice that we can write the covariance matrix defined by (7) as

$$\Phi_{i_0+n,i\mid n} = \mathbf{S}_i^T(0, n-1)\mathbf{S}_i(0, n-1). \tag{48}$$

Two types of updates exist for least-squares parameters: order updates and time updates. The time updates in this section generally fall into two categories. Given some LS parameter $\xi$ (i.e., the forward prediction vector $\mathbf{f}$ or the prediction residual), we wish to find (1) a recursion for $\xi$ computed from the data samples $\{y_{i_0}, y_{i_0+1}, \cdots, y_i\}$ in terms of $\xi$ computed from the data samples $\{y_{i_0}, y_{i_0+1}, \cdots, y_{i-1}\}$ (forward time update), and (2) a recursion for $\xi$ computed from the data samples $\{y_{i_0}, y_{i_0+1}, \cdots, y_i\}$ in terms of $\xi$ computed from the data samples $\{y_{i_0+1}, y_{i_0+2}, \cdots, y_i\}$ (backward time update). Associated with the variable $\xi$ is therefore an order index $n$ and the time indices of the data used in the least-squares computation. If the data values $\{y_{i_0}, y_{i_0+1}, \cdots, y_i\}$ are used to compute $\xi$, then the indices $i_0$ and $i$ must be specified. This is in contrast to the prewindowed case where only $i$ need be specified since $i_0$ is always zero.

Throughout the rest of this paper, the starting-time index of the generic parameter $\xi$ will appear as a subscript, and the current-time index will appear as a function argument. As an example $\xi_{i_0}(i \mid n)$ implies that the data values $\{y_{i_0}, y_{i_0+1}, \cdots, y_i\}$ are used to compute the $n$th order variable $\xi$. The following variables are needed to derive the LS algorithms in the next section:

1. Forward and backward prediction vectors [from (4)],

$$\mathbf{f}_{i_0}(i \mid n) = \Phi^{-1}_{i_0+n-1,i-1\mid n}[\mathbf{S}_i^T(1, n)\mathbf{Y}_{i_0+n,i}] \tag{49a}$$

and

$$\mathbf{b}_{i_0}(i \mid n) = \Phi^{-1}_{i_0+n,i\mid n}[\mathbf{S}_i^T(0, n-1)(z^{-n}\mathbf{Y}_{i_0+n,i})]. \tag{49b}$$

2. Forward and backward prediction residual vectors,

$$\mathbf{E}_{f,i_0}(i \mid n) \equiv \mathbf{Y}_{i_0+n,i} - \mathbf{S}_i(1, n)\mathbf{f}_{i_0}(i \mid n) \tag{50a}$$

and

$$\mathbf{E}_{b,i_0}(i \mid n) \equiv z^{-n}\mathbf{Y}_{i_0+n,i} - \mathbf{S}_i(0, n-1)\mathbf{b}_{i_0}(i \mid n). \tag{50b}$$

3. Forward and backward prediction residuals (scalars),

$$e_{f,i_0}(i \mid n) \equiv \langle \mathbf{u}_i, \mathbf{E}_{f,i_0}(i \mid n) \rangle = y_i - \mathbf{f}_{i_0}^T(i \mid n)\mathbf{y}_{i-1 \mid n} \tag{51a}$$

and

$$e_{b,i_0}(i \mid n) \equiv \langle \mathbf{u}_i, \mathbf{E}_{b,i_0}(i \mid n) \rangle = y_{i-n} - \mathbf{b}_{i_0}^T(i \mid n)\mathbf{y}_{i \mid n}. \tag{51b}$$

4. Forward and backward cost functions,

$$\epsilon_{f,i_0}(i \mid n) \equiv \| \mathbf{E}_{f,i_0}(i \mid n) \|^2, \qquad \epsilon_{b,i_0}(i \mid n) \equiv \| \mathbf{E}_{b,i_0}(i \mid n) \|^2. \tag{52}$$

5. PARtial CORrelation (PARCOR) coefficient,

$$k_{n,i_0}(i) \equiv \langle \mathbf{E}_{f,i_0+1}(i \mid n-1), \mathbf{E}_{b,i_0}(i-1 \mid n-1) \rangle. \tag{53}$$

6. Auxiliary variables, or gains,

$$\mathbf{g}_{i_0+1}(i \mid n) \equiv \Phi_{i_0+n,i \mid n}^{-1}\mathbf{y}_{i \mid n}, \tag{54a}$$

$$\mathbf{h}_{i_0+1}(i \mid n) \equiv \Phi_{i_0+n,i \mid n}^{-1}\mathbf{y}_{i_0+n \mid n}, \tag{54b}$$

$$\gamma_{i_0+1}(i \mid n) \equiv \langle \mathbf{u}_i, P_{M_i(0,n-1)}\mathbf{u}_i \rangle = \mathbf{y}_{i \mid n}^T\Phi_{i_0+n,i \mid n}^{-1}\mathbf{y}_{i \mid n}, \tag{55a}$$

$$\gamma_{i_0+1}^*(i \mid n) \equiv \langle \mathbf{u}_{i_0}, P_{M_i(0,n-1)}\mathbf{u}_{i_0} \rangle = \mathbf{y}_{i_0+n \mid n}^T\Phi_{i_0+n,i \mid n}^{-1}\mathbf{y}_{i_0+n \mid n}, \tag{55b}$$

and

$$\alpha_{i_0+1}(i \mid n) \equiv \langle \mathbf{u}_{i_0}, P_{M_i(0,n-1)}\mathbf{u}_i \rangle = \mathbf{y}_{i \mid n}^T\Phi_{i_0+n,i \mid n}^{-1}\mathbf{y}_{i_0+n \mid n}. \tag{55c}$$

Notice that

$$\mathbf{S}_i(0, n-1)\mathbf{g}_{i_0+1}(i \mid n) = P_{M_i(0,n-1)}\mathbf{u}_i; \tag{56a}$$

and

$$\mathbf{S}_i(0, n-1)\mathbf{h}_{i_0+1}(i \mid n) = P_{M_i(0,n-1)}\mathbf{u}_{i_0}; \tag{56b}$$

and that

$$\gamma_{i_0+1}(i \mid n) = \mathbf{g}_{i_0+1}^T(i \mid n)\mathbf{y}_{i \mid n}, \tag{57a}$$

$$\gamma_{i_0+1}^*(i \mid n) = \mathbf{h}_{i_0+1}^T(i \mid n)\mathbf{y}_{i_0+n \mid n}, \tag{57b}$$

and

$$\alpha_{i_0+1}(i \mid n) = \mathbf{g}_{i_0+1}^T(i \mid n)\mathbf{y}_{i_0+n \mid n} = \mathbf{h}_{i_0+1}^T(i \mid n)\mathbf{y}_{i \mid n}. \tag{57c}$$

Using the notation in the last section, the gains $\gamma$ and $\gamma^*$ are, respectively, $\sin^2\theta_i$ and $\sin^2\theta_i^*$ where $\theta_i$ and $\theta_i^*$ are, respectively, the angles between $M_i(0, n-1)$ and $\tilde{M}_i(0, n-1)$, and between $M_i(0, n-1)$ and $M_i^*(0, n-1)$.

At each time instant our objective is to minimize the cost functions

$\epsilon_{f,i_0}(i \mid n)$ and $\epsilon_{b,i_0}(i \mid n)$. From the discussion in Section III it follows that

$$\mathbf{E}_{f,i_0}(i \mid n) = P^{\perp}_{M_i(1,n)} \mathbf{Y}_{i_0+n,i} \tag{58a}$$

and

$$\mathbf{E}_{b,i_0}(i \mid n) = P^{\perp}_{M_i(0,n-1)}(z^{-n}\mathbf{Y}_{i_0+n,i}). \tag{58b}$$

The following variables, which are closely related to the prediction residuals, are also needed:

$$\begin{aligned}\mathbf{E}'_{f,i_0}(i \mid n) &\equiv P^{\perp}_{M_{i\mid i-1}(1,n)} \mathbf{Y}_{i_0+n,i} \\ &= \mathbf{Y}_{i_0+n,i} - \mathbf{S}_i(1,\ n)\mathbf{f}_{i_0}(i-1 \mid n),\end{aligned} \tag{59a}$$

and

$$\begin{aligned}\mathbf{E}'_{b,i_0}(i \mid n) &\equiv P^{\perp}_{M_{i\mid i-1}(0,n-1)}(z^{-n}\mathbf{Y}_{i_0+n,i}) \\ &= z^{-n}\mathbf{Y}_{i_0+n,i} - \mathbf{S}_i(0,\ n-1)\mathbf{b}_{i_0}(i-1 \mid n),\end{aligned} \tag{59b}$$

i.e., $\mathbf{E}'_f$ and $\mathbf{E}'_b$ are the forward and backward residual vectors obtained by using prediction vectors computed at the *previous* time interval. The top components of $\mathbf{E}'_{f,i_0}(i \mid n)$ and $\mathbf{E}'_{b,i_0}(i \mid n)$ are, respectively,

$$\begin{aligned}e'_{f,i_0}(i \mid n) &\equiv \langle \mathbf{u}_i,\ \mathbf{E}'_{f,i_0}(i \mid n)\rangle \\ &= y_i - \mathbf{f}^T_{i_0}(i-1 \mid n)\mathbf{y}_{i-1\mid n}\end{aligned} \tag{60a}$$

and

$$\begin{aligned}e'_{b,i_0}(i \mid n) &\equiv \langle \mathbf{u}_i,\ \mathbf{E}'_{b,i_0}(i \mid n)\rangle \\ &= y_{i-n} - \mathbf{b}^T_{i_0}(i-1 \mid n)\mathbf{y}_{i\mid n}.\end{aligned} \tag{60b}$$

The $n$th order forward prediction residual computed at time $i_0 + n$ using the tap vector $\mathbf{f}_{i_0}(i \mid n)$ is

$$\begin{aligned}e^*_{f,i_0}(i \mid n) &\equiv \langle \mathbf{u}_{i_0}, \mathbf{E}_{f,i_0}(i \mid n)\rangle \\ &= y_{i_0+n} - \mathbf{f}^T_{i_0}(i \mid n)\mathbf{y}_{i_0+n-1\mid n}.\end{aligned} \tag{61}$$

The forward residual vector at time $i$ using the forward prediction vector calculated from the data samples $\{y_{i_0+1},\ \cdots,\ y_i\}$ is

$$\begin{aligned}\mathbf{E}_{f,i_0\mid i_0+1}(i \mid n) &\equiv P^{\perp}_{M_{i_0\mid i_0+1}(1,n)} \mathbf{Y}_{i_0+n,i} \\ &= \mathbf{Y}_{i_0+n,i} - \mathbf{S}_i(1,\ n)\mathbf{f}_{i_0+1}(i \mid n).\end{aligned} \tag{62}$$

The variables $e^*_{b,i_0}(i \mid n)$ and $\mathbf{E}_{b,i_0\mid i_0+1}(i \mid n)$ are similarly defined.

Notice that the time indices associated with a residual vector change in accordance with the projection space, i.e.,

$$P^{\perp}_{M_i(1,n-1)} \mathbf{Y}_{i_0+n,i} = \mathbf{E}_{f,i_0+1}(i \mid n-1), \tag{63a}$$

and

$$P^{\perp}_{M_i(1,n-1)}(z^{-n}\mathbf{Y}_{i_0+n,i}) = \mathbf{E}_{b,i_0}(i-1\,|\,n-1). \tag{63b}$$

The recursions needed to derive the algorithms in the next section are now generated systematically. By appropriately defining the vectors and subspaces entering the projection order update (22), order updates are derived for all of the basic variables defined by (49) through (55). We then use the forward and backward time updates (34) and (43) to obtain forward and backward time updates for the basic vectors defined by (49), (50), and (54). Finally, the forward and backward time updates for inner products (35) and (44) are applied to $k_{n,i_0}(i)$, $\epsilon_{f,i_0}(i\,|\,n)$, and $\epsilon_{b,i_0}(i\,|\,n)$. It would take up too much space to explicitly define the vectors and subspaces that must be substituted in the projection update used to derive each recursion. Consequently, only the results are stated, with a few representative examples worked out in more detail.

### *5.2 Order updates*

The following order updates are obtained by using the projection order update (22) [or equivalently (21)]. The $l$th through the $m$th component of $\mathbf{f}_{i_0}(i\,|\,n)$ is denoted by $[\mathbf{f}_{i_0}(i\,|\,n)]_{l,m}$, and $[\mathbf{f}_{i_0}(i\,|\,n)]_j$ is the $j$th component of $\mathbf{f}_{i_0}(i\,|\,n)$. The same notation is used for the backward prediction vector $\mathbf{b}_{i_0}(i\,|\,n)$ and the gain vectors $\mathbf{g}_{i_0}(i\,|\,n)$ and $\mathbf{h}_{i_0}(i\,|\,n)$.

$$\mathbf{E}_{f,i_0}(i\,|\,n) = \mathbf{E}_{f,i_0+1}(i\,|\,n-1) - \frac{k_{n,i_0}(i)}{\epsilon_{b,i_0}(i-1\,|\,n-1)} \cdot \mathbf{E}_{b,i_0}(i-1\,|\,n-1), \tag{64a}$$

$$\mathbf{E}_{b,i_0}(i\,|\,n) = \mathbf{E}_{b,i_0}(i-1\,|\,n-1) - \frac{k_{n,i_0}(i)}{\epsilon_{f,i_0+1}(i\,|\,n-1)} \cdot \mathbf{E}_{f,i_0+1}(i\,|\,n-1), \tag{64b}$$

$$\epsilon_{f,i_0}(i\,|\,n) = \epsilon_{f,i_0+1}(i\,|\,n-1) - \frac{k^2_{n,i_0}(i)}{\epsilon_{b,i_0}(i-1\,|\,n-1)}, \tag{65a}$$

$$\epsilon_{b,i_0}(i\,|\,n) = \epsilon_{b,i_0}(i-1\,|\,n-1) - \frac{k^2_{n,i_0}(i)}{R_{f,i_0+1}(i\,|\,n-1)}, \tag{65b}$$

$$[\mathbf{f}_{i_0}(i\,|\,n)]_n = \frac{k_{n,i_0}(i)}{\epsilon_{b,i_0}(i-1\,|\,n-1)}, \tag{66a}$$

$$[\mathbf{f}_{i_0}(i\,|\,n)]_{1,n-1} = \mathbf{f}_{i_0+1}(i\,|\,n-1) - [\mathbf{f}_{i_0}(i\,|\,n)]_n \mathbf{b}_{i_0}(i-1\,|\,n-1), \tag{66b}$$

$$[\mathbf{b}_{i_0}(i\,|\,n)]_1 = \frac{k_{n,i_0}(i)}{\epsilon_{f,i_0+1}(i\,|\,n-1)}, \tag{67a}$$

$$[\mathbf{b}_{i_0}(i\,|\,n)]_{2,n} = \mathbf{b}_{i_0}(i-1\,|\,n-1) - [\mathbf{b}_{i_0}(i\,|\,n)]_1 \mathbf{f}_{i_0+1}(i\,|\,n-1), \tag{67b}$$

$$[\mathbf{g}_{i_0}(i\,|\,n+1)]_{n+1} = \frac{e_{b,i_0}(i\,|\,n)}{\epsilon_{b,i_0}(i\,|\,n)}, \tag{68a}$$

$$[\mathbf{g}_{i_0}(i\,|\,n+1)]_{1,n} = \mathbf{g}_{i_0+1}(i\,|\,n) - [\mathbf{g}_{i_0}(i\,|\,n+1)]_{n+1} \mathbf{b}_{i_0}(i\,|\,n), \tag{68b}$$

$$[\mathbf{g}_{i_0}(i\,|\,n+1)]_1 = \frac{e_{f,i_0}(i\,|\,n)}{\epsilon_{f,i_0}(i\,|\,n)}, \tag{69a}$$

$$[\mathbf{g}_{i_0}(i\,|\,n+1)]_{2,n+1} = \mathbf{g}_{i_0}(i-1\,|\,n) - [\mathbf{g}_{i_0}(i\,|\,n+1)]_1 \mathbf{f}_{i_0}(i\,|\,n), \tag{69b}$$

$$[\mathbf{h}_{i_0}(i\,|\,n+1)]_{n+1} = \frac{e^*_{b,i_0}(i\,|\,n)}{\epsilon_{b,i_0}(i\,|\,n)}, \tag{70a}$$

$$[\mathbf{h}_{i_0}(i\,|\,n+1)]_{1,n} = \mathbf{h}_{i_0+1}(i\,|\,n) - [\mathbf{h}_{i_0}(i\,|\,n+1)]_{n+1} \mathbf{b}_{i_0}(i\,|\,n), \tag{70b}$$

$$[\mathbf{h}_{i_0}(i\,|\,n+1)]_1 = \frac{e^*_{f,i_0}(i\,|\,n)}{\epsilon_{f,i_0}(i\,|\,n)}, \tag{71a}$$

$$[\mathbf{h}_{i_0}(i\,|\,n+1)]_{2,n+1} = \mathbf{h}_{i_0}(i-1\,|\,n) - [\mathbf{h}_{i_0}(i\,|\,n+1)]_1 \mathbf{f}_{i_0}(i\,|\,n), \tag{71b}$$

$$\gamma_{i_0}(i\,|\,n+1) = \gamma_{i_0+1}(i\,|\,n) + \frac{e^2_{b,i_0}(i\,|\,n)}{\epsilon_{b,i_0}(i\,|\,n)}, \tag{72a}$$

$$\gamma_{i_0}(i\,|\,n+1) = \gamma_{i_0}(i-1\,|\,n) + \frac{e^2_{f,i_0}(i\,|\,n)}{\epsilon_{f,i_0}(i\,|\,n)}, \tag{72b}$$

$$\gamma^*_{i_0}(i\,|\,n+1) = \gamma^*_{i_0+1}(i\,|\,n) + \frac{e^{*2}_{b,i_0}(i\,|\,n)}{\epsilon_{b,i_0}(i\,|\,n)}, \tag{73a}$$

$$\gamma^*_{i_0}(i\,|\,n+1) = \gamma^*_{i_0}(i-1\,|\,n) + \frac{e^{*2}_{f,i_0}(i\,|\,n)}{\epsilon_{f,i_0}(i\,|\,n)}, \tag{73b}$$

$$\alpha_{i_0}(i\,|\,n+1) = \alpha_{i_0+1}(i\,|\,n) + \frac{e_{b,i_0}(i\,|\,n) e^*_{b,i_0}(i\,|\,n)}{\epsilon_{b,i_0}(i\,|\,n)}, \tag{74a}$$

and

$$\alpha_{i_0}(i\,|\,n+1) = \alpha_{i_0}(i-1\,|\,n) + \frac{e_{f,i_0}(i\,|\,n) e^*_{f,i_0}(i\,|\,n)}{\epsilon_{f,i_0}(i\,|\,n)}. \tag{74b}$$

As an example, (64a) is derived from (22), where $M$ is replaced by $M_i(1, n-1)$, $\mathbf{X}$ is replaced by $z^{-n}\mathbf{Y}_{i_0+n,i}$, and $\mathbf{Y}$ is replaced by $\mathbf{Y}_{i_0+n,i}$.

By observing that $\mathbf{E}_{b,i_0}(i-1\,|\,n-1)$ is orthogonal to $M_i(1, n-1)$, it is clear that

$$\langle \mathbf{Y}_{i_0+n,i}, \mathbf{E}_{b,i_0}(i-1\,|\,n-1)\rangle = \langle \mathbf{E}_{f,i_0+1}(i\,|\,n-1), \cdot \mathbf{E}_{b,i_0}(i-1\,|\,n-1)\rangle = k_{n,i_0}(i). \tag{75}$$

Recursions (65) are obtained by taking norms of (64) respectively. The recursions (68) through (71) are obtained from (21), where $\mathbf{Y}$ is replaced by $\mathbf{u}_i$ and $\mathbf{u}_{i_0}$, respectively. Making the same substitutions in (21) and then taking inner products with $\mathbf{u}_i$ or $\mathbf{u}_{i_0}$ gives recursions (72) through (74).

### *5.3 Forward time updates*

The following forward time updates are obtained from the (orthogonal) projection operator forward time update (34):

$$\mathbf{E}_{f,i_0}(i\,|\,n) = \mathbf{E}'_{f,i_0}(i\,|\,n) - [P_{M_i(1,n)}\mathbf{u}_i]\,\frac{e_{f,i_0}(i\,|\,n)}{1-\gamma_{i_0}(i-1\,|\,n)}, \tag{76a}$$

$$\mathbf{E}_{b,i_0}(i\,|\,n) = \mathbf{E}'_{b,i_0}(i\,|\,n) - [P_{M_i(0,n-1)}\mathbf{u}_i]\,\frac{e_b(i\,|\,n)}{1-\gamma_{i_0+1}(i\,|\,n)}, \tag{76b}$$

$$\mathbf{f}_{i_0}(i\,|\,n) = \mathbf{f}_{i_0}(i-1\,|\,n) + \mathbf{g}_{i_0}(i-1\,|\,n)\,\frac{e_{f,i_0}(i\,|\,n)}{1-\gamma_{i_0}(i-1\,|\,n)}, \tag{77a}$$

$$\mathbf{b}_{i_0}(i\,|\,n) = \mathbf{b}_{i_0}(i-1\,|\,n) + \mathbf{g}_{i_0+1}(i\,|\,n)\,\frac{e_{b,i_0}(i\,|\,n)}{1-\gamma_{i_0+1}(i\,|\,n)}, \tag{77b}$$

$$\mathbf{h}_{i_0}(i\,|\,n) = \mathbf{h}_{i_0}(i-1\,|\,n) - \mathbf{g}_{i_0}(i\,|\,n)\,\frac{\alpha_{i_0}(i\,|\,n)}{1-\gamma_{i_0}(i\,|\,n)}, \tag{78}$$

$$\gamma^*_{i_0}(i\,|\,n) = \gamma^*_{i_0}(i-1\,|\,n) - \frac{\alpha^2_{i_0}(i\,|\,n)}{1-\gamma_{i_0}(i\,|\,n)}, \tag{79}$$

and

$$\alpha_{i_0}(i\,|\,n) = \mathbf{y}^T_{i|n}\mathbf{h}_{i_0}(i-1\,|\,n)[1-\gamma_{i_0}(i\,|\,n)]. \tag{80}$$

Equation (78) is obtained from (34), where $M_i$ is replaced $M_i(0, n)$ and $\mathbf{Y}_i$ is replaced by $\mathbf{u}_{i_0}$. Equations (79) and (80) are obtained by making the same substitutions in (34) and then taking inner products with $\mathbf{u}_{i_0}$ and $\mathbf{u}_i$, or by premultiplying (78) by $\mathbf{y}^T_{i_0+n-1|n}$ and $\mathbf{y}^T_{i|n}$, respectively. Taking the inner product of (76) with $\mathbf{u}_i$ and $\mathbf{u}_{i_0}$, respectively gives the following recursions:

$$e'_{f,i_0}(i\,|\,n) = \frac{e_{f,i_0}(i\,|\,n)}{1 - \gamma_{i_0}(i-1\,|\,n)}, \tag{81a}$$

$$e'_{b,i_0}(i\,|\,n) = \frac{e_{b,i_0}(i\,|\,n)}{1 - \gamma_{i_0+1}(i\,|\,n)}, \tag{81b}$$

$$e^*_{f,i_0}(i\,|\,n) = e^*_{f,i_0}(i-1\,|\,n) - e_{f,i_0}(i\,|\,n)\,\frac{\alpha_{i_0}(i-1\,|\,n)}{1 - \gamma_{i_0}(i-1\,|\,n)}, \tag{82a}$$

and

$$e^*_{b,i_0}(i\,|\,n) = e^*_{b,i_0}(i-1\,|\,n) - e_{b,i_0}(i\,|\,n)\,\frac{\alpha_{i_0+1}(i\,|\,n)}{1 - \gamma_{i_0+}(i\,|\,n)}. \tag{82b}$$

### *5.4 Backward time updates*

The following backward time updates are obtained from the projection operator backward time update (43):

$$\mathbf{E}_{f,i_0}(i\,|\,n) = \mathbf{E}_{f,i_0|\,i_0+1}(i\,|\,n) - [P_{M_i(1,n)}\mathbf{u}_{i_0}]\,\frac{e^*_{f,i_0}(i\,|\,n)}{1 - \gamma^*_{i_0}(i-1\,|\,n)}, \tag{83a}$$

$$\mathbf{E}_{b,i_0}(i\,|\,n) = \mathbf{E}_{b,i_0|\,i_0+1}(i\,|\,n) - [P_{M_i(0,n-1)}\mathbf{u}_{i_0}]\,\frac{e^*_{b,i_0}(i\,|\,n)}{1 - \gamma^*_{i_0+1}(i\,|\,n)}, \tag{83b}$$

$$\mathbf{f}_{i_0+1}(i\,|\,n) = \mathbf{f}_{i_0}(i\,|\,n) - \mathbf{h}_{i_0}(i-1\,|\,n)\,\frac{e^*_{f,i_0}(i\,|\,n)}{1 - \gamma^*_{i_0}(i-1\,|\,n)}, \tag{84a}$$

$$\mathbf{b}_{i_0+1}(i\,|\,n) = \mathbf{b}_{i_0}(i\,|\,n) - \mathbf{h}_{i_0+1}(i\,|\,n)\,\frac{e^*_{b,i_0}(i\,|\,n)}{1 - \gamma^*_{i_0+1}(i\,|\,n)}, \tag{84b}$$

$$\mathbf{g}_{i_0}(i\,|\,n) = \mathbf{g}_{i_0+1}(i\,|\,n) - \mathbf{h}_{i_0}(i\,|\,n)\,\frac{\alpha_{i_0}(i\,|\,n)}{1 - \gamma^*_{i_0}(i\,|\,n)}, \tag{85}$$

$$\gamma_{i_0}(i\,|\,n) = \gamma_{i_0+1}(i\,|\,n) - \frac{\alpha^2_{i_0}(i\,|\,n)}{1 - \gamma^*_{i_0}(i\,|\,n)}, \tag{86}$$

and

$$\alpha_{i_0}(i\,|\,n) = \mathbf{y}^T_{i_0+n-1|\,n}\mathbf{g}_{i_0+1}(i\,|\,n)[1 - \gamma^*_{i_0}(i\,|\,n)]. \tag{87}$$

Equation (85) is obtained by replacing $\mathbf{Y}_i$ by $\mathbf{u}_i$ in (43). Equations (86) and (87) are obtained by premultiplying (85) by $\mathbf{y}^T_{i|n}$ and $\mathbf{y}^T_{i_0+n-1|n}$, respectively. The following recursions are obtained by taking the inner product of (83) with $\mathbf{u}_i$, respectively:

$$e_{f,i_0+1}(i\,|\,n) = e_{f,i_0}(i\,|\,n) + e^*_{f,i_0}(i\,|\,n)\,\frac{\alpha_{i_0}(i-1\,|\,n)}{1 - \gamma^*_{i_0}(i-1\,|\,n)}, \tag{88a}$$

and

$$e_{b,i_0+1}(i \mid n) = e_{b,i_0}(i \mid n) + e^*_{b,i_0}(i \mid n) \frac{\alpha_{i_0+1}(i \mid n)}{1 - \gamma^*_{i_0+1}(i \mid n)}. \tag{88b}$$

The recursions that result from taking inner products with $\mathbf{u}_{i_0}$ will not be used and are therefore omitted.

### 5.5 Inner product updates

The following recursions are obtained from the forward time update for inner products (35):

$$k_{n,i_0}(i) = k_{n,i_0}(i-1) + e_{f,i_0+1}(i \mid n-1) e_{b,i_0}(i-1 \mid n-1) \cdot \frac{1}{1 - \gamma_{i_0+1}(i-1 \mid n-1)}, \tag{89}$$

$$\epsilon_{f,i_0}(i \mid n) = \epsilon_{f,i_0}(i-1 \mid n) + e^2_{f,i_0}(i \mid n) \frac{1}{1 - \gamma_{i_0}(i-1 \mid n)}, \tag{90a}$$

and

$$\epsilon_{b,i_0}(i \mid n) = \epsilon_{b,i_0}(i-1 \mid n) + e^2_{b,i_0}(i \mid n) \frac{1}{1 - \gamma_{i_0+1}(i \mid n)}. \tag{90b}$$

The following recursions are obtained from the backward time update for inner products (44):

$$k_{n,i_0}(i) = k_{n,i_0+1}(i) + e^*_{f,i_0+1}(i \mid n-1) e^*_{b,i_0}(i-1 \mid n-1) \cdot \frac{1}{1 - \gamma^*_{i_0+1}(i-1 \mid n-1)}, \tag{91}$$

$$\epsilon_{f,i_0}(i \mid n) = \epsilon_{f,i_0+1}(i \mid n) + e^{*2}_{f,i_0}(i \mid n) \frac{1}{1 - \gamma^*_{i_0}(i-1 \mid n)}, \tag{92a}$$

and

$$\epsilon_{b,i_0}(i \mid n) = \epsilon_{b,i_0+1}(i \mid n) + e^{*2}_{b,i_0}(i \mid n) \frac{1}{1 - \gamma^*_{i_0+1}(i \mid n)}. \tag{92b}$$

Equations (89) and (91) are obtained by using (35) and (44), where $\mathbf{v}_i$ is replaced by $\mathbf{Y}_{i_0+n,i}$, $\mathbf{Y}_i$ is replaced by $z^{-n}\mathbf{Y}_{i_0+n,i}$, and $M_i$ is replaced by $M_i(1, n-1)$, respectively. The previous set of recursions (64) through (92) are complete in the sense that any existing least-squares alogrithm can be derived by manipulating suitable subsets of these recursions.

## VI. RECURSIVE FIXED-ORDER COVARIANCE ALGORITHMS

### 6.1 *Sliding-window algorithm*

Combining (60), (61), (68) through (73), (77), (81a), (84), (90a), and (92a), gives the following sliding-window LS algorithm for the prediction coefficients. Where unspecified, the order of the variable is assumed to be $N$, the order of the least-squares filter. Also, the starting time index is denoted as $i_0$. If the sliding window contains $M$ data values, then $i_0 = i - M + 1$,

$$e'_{f,i_0}(i) = y_i - \mathbf{f}^T_{i_0}(i-1)\mathbf{y}_{i-1}, \tag{93a}$$

$$\mathbf{f}_{i_0}(i) = \mathbf{f}_{i_0}(i-1) + \mathbf{g}_{i_0}(i-1)e'_{f,i_0}(i), \tag{93b}$$

$$e_{f,i_0}(i) = e'_{f,i_0}(i)[1 - \gamma_{i_0}(i-1)], \tag{93c}$$

$$e^*_{f,i_0}(i) = y_{i_0+N} - \mathbf{f}^T_{i_0}(i)\mathbf{y}_{i_0+N-1}, \tag{93d}$$

$$\epsilon_{f,i_0}(i) = \epsilon_{f,i_0}(i-1) + e'_{f,i_0}(i)e_{f,i_0}(i), \tag{93e}$$

$$\mathbf{g}_{i_0}(i \mid N+1)]_1 = \frac{e_{f,i_0}(i)}{\epsilon_{f,i_0}(i)}, \tag{93f}$$

$$[\mathbf{g}_{i_0}(i \mid N-1)]_{2,N+1} = \mathbf{g}_{i_0}(i-1) - [\mathbf{g}_{i_0}(i \mid N+1)]_1\mathbf{f}_{i_0}(i), \tag{93g}$$

$$[\mathbf{h}_{i_0}(i \mid N+1)]_1 = \frac{e^*_{f,i_0}(i)}{\epsilon_{f,i_0}(i)}, \tag{93h}$$

$$[\mathbf{h}_{i_0}(i \mid N+1)]_{2,N+1} = \mathbf{h}_{i_0}(i-1) - [\mathbf{h}_{i_0}(i \mid N+1)]_1\mathbf{f}_{i_0}(i), \tag{93i}$$

$$\mathbf{f}_{i_0+1}(i) = \mathbf{f}_{i_0}(i) - \mathbf{h}_{i_0}(i-1)\frac{e^*_{f,i_0}(i)}{1 - \gamma^*_{i_0}(i-1)}, \tag{93j}$$

$$\epsilon_{f,i_0+1}(i) = \epsilon_{f,i_0}(i) - \frac{e^{*2}_{f,i_0}(i)}{1 - \gamma^*_{i_0}(i-1)}, \tag{93k}$$

$$e'_{b,i_0}(i) = y_{i-N} - \mathbf{b}^T_{i_0}(i-1)\mathbf{y}_i, \tag{93l}$$

$$\mathbf{b}_{i_0}(i) = \frac{\mathbf{b}_{i_0}(i-1) + e'_{b,i_0}(i)[\mathbf{g}_{i_0}(i \mid N+1)]_{1,N}}{1 - e'_{b,i_0}(i)[\mathbf{g}_{i_0}(i \mid N+1)]_{N+1}}, \tag{93m}$$

$$e^*_{b,i_0}(i) = y_{i_0} - \mathbf{b}^T_{i_0}(i)\mathbf{y}_{i_0+N}, \tag{93n}$$

$$\mathbf{g}_{i_0+1}(i) = [\mathbf{g}_{i_0}(i \mid N+1)]_{1,N} + [\mathbf{g}_{i_0}(i \mid N+1)]_{N+1}\mathbf{b}_{i_0}(i), \tag{93o}$$

$$\mathbf{h}_{i_0+1}(i) = [\mathbf{h}_{i_0}(i \mid N+1)]_{1,N} + [\mathbf{b}_{i_0}(i \mid N+1)_{N+1}\mathbf{b}_{i_0}(i), \tag{93p}$$

$$\gamma_{i_0+1}(i) = \frac{\gamma_{i_0}(i-1) + e_{f,i_0}(i)[\mathbf{g}_{i_0}(i \mid N+1)]_1 - e'_{b,i_0}(i)[\mathbf{g}_{i_0}(i \mid N+1)]_{N+1}}{1 - e'_{b,i_0}(i)[\mathbf{g}_{i_0}(i \mid N+1)]_{N+1}}, \tag{93q}$$

$$\gamma^*_{i_0+1}(i) = \gamma^*_{i_0}(i-1) + e^*_{f,i_0}(i)[\mathbf{h}_{i_0}(i \mid N+1)]_1 - e^*_{b,i_0}[\mathbf{h}_{i_0}(i \mid N+1)]_N, \tag{93r}$$

and

$$\mathbf{b}_{i_0+1}(i) = \mathbf{b}_{i_0}(i) - \mathbf{h}_{i_0+1}(i) \frac{e^*_{b,i_0}(i)}{1 - \gamma^*_{i_0+1}(i)}. \tag{93s}$$

The recursions (93m) and (93q) were not listed in the previous section, but are easily obtained by solving (77b) and (68b) simultaneously for $\mathbf{b}_{i_0}(i)$, and by substituting (68a), (69a), and (81b) into (72), and solving for $\gamma_{i_0+1}(i \mid n)$. Notice that all data samples in the sliding window $(y_{i-M+1}, \cdots, y_i)$ must be stored. This is also true of the order-recursive sliding-window algorithm presented in Ref. 14. If division is counted as multiplication, then the algorithm (93) requires $12N + 16$ multiplies and $12N + 12$ additions at each iteration. In contrast, the unnormalized sliding-window lattice predictor (see Appendix B) requires $16N$ multiplies and $10N$ additions per iteration, and the normalized lattice predictor[16] requires $30N$ multiplies, $18N$ additions, and $6N$ square roots per iteration.

Because sliding-window algorithms have finite memory, initialization for these algorithms is basically the same as for the prewindowed case, i.e., the data $y_i$ can be assumed to be zero for $i < 0$. After $M$ iterations, where $M$ is the window length, these data points are discarded. The algorithm (93) is therefore initialized by setting the gains $\gamma$ and $\gamma^*$, and the elements of the vectors **f**, **b**, **g**, and **h** equal to zero, and letting

$$\epsilon_{f,i_0}(0) = \delta, \tag{94}$$

where $\delta$ is chosen to ensure that the algorithm remains stable. It is easily verified that for time $i < M - N - 1$, where $M$ is the length of the sliding window, the algorithm (93) becomes a modified version of the prewindowed LS transversal (fast Kalman) algorithm.[1,22]

### 6.2 *Growing-memory covariance algorithm*

The following fixed-order growing-memory covariance algorithm is obtained by combining (60), (68b), (69), (77), (78), (80), (85), (87), and (90a). The lower index of the window $i_0$ is assumed to be zero. For notational convenience we define the following variables,

$$\beta_{i_0}(i \mid n) \equiv \frac{\alpha_{i_0}(i \mid n)}{1 - \gamma_{i_0}(i \mid n)} \tag{95a}$$

and

$$\beta^*_{i_0}(i \mid n) \equiv \frac{\alpha_{i_0}(i \mid n)}{1 - \gamma^*_{i_0}(i \mid n)}. \tag{95b}$$

Where unspecified, the lower time index and the order of the variables are equal to zero and $N$, respectively,*

$$e'_f(i) = y_i = \mathbf{f}^T(i-1)\mathbf{y}_{i-1}, \tag{96a}$$

$$\mathbf{f}(i) = \mathbf{f}(i-1) + \mathbf{g}(i-1)e'_f(i), \tag{96b}$$

$$e_f(i) = y_i - \mathbf{f}^T(i)\mathbf{y}_{i-1}, \tag{96c}$$

$$\epsilon_f(i) = \epsilon_f(i-1) + e_f(i)e'_f(i), \tag{96d}$$

$$[\mathbf{g}(i \mid N+1)]_1 = \frac{e_f(i)}{\epsilon_f(i)}, \tag{96e}$$

$$[\mathbf{g}(i \mid N+1)]_{2,N+1} = \mathbf{g}(i-1) - [\mathbf{g}(i \mid N+1)]_1\mathbf{f}(i), \tag{96f}$$

$$e'_b(i) = y_{i-N} - \mathbf{b}^T(i-1)\mathbf{y}_i, \tag{96g}$$

$$\mathbf{b}(i) = \frac{\mathbf{b}(i-1) + e'_b(i)[\mathbf{g}(i \mid N+1)]_{1,N}}{1 - e'_b(i)[\mathbf{g}(i \mid N+1)]_{N+1}}, \tag{96h}$$

$$\mathbf{g}_1(i) = [\mathbf{g}(i \mid N+1)]_{1,N} + [\mathbf{g}(i \mid N+1)]_{N+1}\mathbf{b}(i), \tag{96i}$$

$$\beta(i) = \mathbf{y}_i^T\mathbf{h}(i-1), \tag{96j}$$

$$\beta^*(i) = \mathbf{y}_{N-1}^T\mathbf{g}_1(i), \tag{96k}$$

$$\mathbf{g}(i) = \frac{\mathbf{g}_1(i) - \beta^*(i)\mathbf{h}(i-1)}{1 - \beta(i)\beta^*(i)}, \tag{96l}$$

and

$$\mathbf{h}(i) = \mathbf{h}(i-1) - \beta(i)\mathbf{g}(i). \tag{96m}$$

Notice that this algorithm can be applied only if $i > N$. Otherwise, the least-squares variables of order $N$ are undefined and cannot be used to compute the same least-squares variables at the successive time interval. Initialization of this algorithm can be performed, however, by using an order-recursive algorithm for $i < N$ to increase the order of the filter by one at each successive time iteration. An order-recursive algorithm for the prediction coefficients is obtained by combining (89), (66a), (67a), top components of (64a) and (64b), (66b), (67b), (65a) and (65b), (82a), (88a), (84a), (92a), (71), (73b), (72), and (74b). This algorithm is basically the same as the covariance lattice

* The author recently discovered that this algorithm has been independently derived in Ref. 23 using an algebraic approach.

algorithm presented in Refs. 7 and 19, except that additional order recursions for the prediction vectors **f** and **b** have been added. It is not explicitly stated in an effort to conserve space. Order-recursive computation of **f** and **b** requires order $N^2$ arithmetic operations per iteration, rather than order $N$ operations per iteration, as required by the fixed-order algorithm. Not all $N$ components of the vectors **f** and **b** need to be updated at each iteration for $i < N$, however. If data is first received at time $i = 0$, the recursions listed above can be used for $n = 0$ up to $n = i$. At time $i = N$ all of the variables that enter the fixed-order algorithm (96) have been computed by the order-recursive algorithm except for $\mathbf{g}(i)$, $\beta(i)$ and $\beta^*(i)$. The gain $\mathbf{g}(i)$ is the only variable needed at the next iteration of the fixed-order algorithm and can be computed by first using (96j) to calculate $\beta(i)$ and then using (96m) to solve for $\mathbf{g}(i)$.

Derivation of initial conditions for the order-recursive initialization routine is significantly more complicated than for the sliding-window algorithm. This is because for $i = n$, the matrix $\Phi_{n,i|n}$ is guaranteed to be singular, and hence all variables are technically undefined. Reference 14 gives a convenient solution to this startup problem. By using a generalized inverse of a singular or nonsingular matrix, the least-squares projection operator $P$, given by (15), can be defined even when the matrix $\mathbf{S}^T\mathbf{S}$ is singular. If this generalized inverse is defined appropriately, it can be shown that the projection updates in Section IV hold even when the covariance matrix is singular. This implies that all of the recursions listed in the last paragraph that constitute the order-recursive initialization routine can be used starting from $i = 0$ with the following initial conditions:

$$\mathbf{f}(0\,|\,0) = \mathbf{b}(0\,|\,0) = \mathbf{f}_1(-1\,|\,0) = \mathbf{h}(-1\,|\,0) = \mathbf{0}, \tag{97a}$$

$$k_n(-1) = 0, \qquad 1 \leq n \leq N, \tag{97b}$$

$$\gamma(-1\,|\,0) = \gamma^*(-1\,|\,0) = \alpha(-1\,|\,0) = 0, \tag{97c}$$

and

$$e_f^*(0\,|\,n) = \begin{cases} y_0 & \text{for} \quad n = 0 \\ 0 & \text{for} \quad n > 0. \end{cases} \tag{97d}$$

At each iteration $i < N$,

$$e_f(i\,|\,0) = e_b(i\,|\,0) = y_i \tag{97e}$$

and

$$\epsilon_f(i\,|\,0) = \epsilon_b(i\,|\,0) = \epsilon_f(i-1\,|\,0) + y_i^2. \tag{97f}$$

Counting division as multiplication, the algorithm (96) requires

$11N + 7$ multiplies and $11N + 1$ additions per iteration. To compare, the unnormalized growing-memory lattice predictor (see Appendix B) requires $22N$ multiplies and $12N$ additions per iteration. The normalized lattice algorithm requires $30N$ multiplies, $18N$ additions, and $6N$ square roots per iteration. We point out that the fixed-order covariance algorithm specified by (96) is not unique. In particular, equation (96c) can be replaced by (81a). The extra recursion (93q) must be added to compute $\gamma$, however. This type of modification has been applied to the fast Kalman algorithm, and has resulted in improved numerical properties.[22]

## VII. EXTENSIONS TO JOINT-PROCESS ESTIMATION

The algorithms presented so far solve the LS prediction problem wherein the sums (1) and (2) are minimized. In applications such as channel equalization, echo and noise cancellation, and adaptive line enhancement, two processes, $\{x_j\}$ and $\{y_j\}$, are given, and our objective is to estimate the $\{x_j\}$ process in terms of the $\{y_j\}$ process. The vector of estimation errors is denoted as

$$\begin{aligned}\mathbf{E}_{x,i_0+1}(i \mid n) &\equiv \mathbf{X}_{i_0+n,i} - \sum_{j=0}^{n-1} c_{j+1\mid n}(z^{-j}\mathbf{Y}_{i_0+n,i}) \\ &= \mathbf{X}_{i_0+n,i} - \mathbf{S}_i(0,\, n-1)\mathbf{c}_{i_0+1}(i \mid n), \end{aligned} \tag{98}$$

where $\mathbf{X}_{i_0,i}$ is defined by (23a), $\mathbf{c}_{i_0+1}(i \mid n)$ is the $n$-dimensional vector of regression coefficients at time $i$ used to estimate $\mathbf{X}_{i_0+n,i}$ [given by (4c)] where $i' = i_0 + n$), and the lower time subscript of $\mathbf{E}_x$ and $\mathbf{c}$ denotes the time index of the starting value from the $y$ sequence (i.e., $y_{i_0+1}$), which is used in the least-squares computation. Our objective is to choose $\mathbf{c}_{i_0+1}(i \mid n)$ such that

$$\epsilon_{x,i_0+1}(i \mid n) \equiv \| \mathbf{E}_{x,i_0+1}(i \mid n) \|^2 \tag{99}$$

is minimized. The discussion in Section III implies that

$$\mathbf{E}_{x,i_0+1}(i \mid n) = P^{\perp}_{M_i(0,n-1)}\mathbf{X}_{i_0+n,i}. \tag{100}$$

We now use the projection recursions in Section IV to derive order and time updates for $\mathbf{E}_{x,i_0}(i \mid n)$ and $\mathbf{c}_{i_0}(i \mid n)$. Details are again omitted since they are basically the same as before. Combining recursions in this section with the prediction algorithms of the last section results in recursive algorithms that solve the LS joint-process-estimation problem.

The following notation, which is analogous to the notation in Section 5.1, is first defined:

1. Cross-correlation coefficient,

$$k^{(x)}_{n+1,i_0}(i) \equiv \langle \mathbf{X}_{i_0+n,i},\, \mathbf{E}_{b,i_0}(i \mid n) \rangle$$
$$= \langle \mathbf{E}_{x,i_0+1}(i \mid n),\, \mathbf{E}_{b,i_0}(i \mid n) \rangle. \tag{101}$$

2. Current residual (scalar),

$$e_{x,i_0}(i \mid n) = \langle \mathbf{u}_i,\, \mathbf{E}_{x,i_0}(i \mid n) \rangle$$
$$= x_i - \mathbf{c}^T_{i_0}(i \mid n)\mathbf{y}_{i \mid n}. \tag{102}$$

3. Past residual (scalar),

$$e^*_{x,i_0+1}(i \mid n) = \langle \mathbf{u}_{i_0},\, \mathbf{E}_{x,i_0+1}(i \mid n) \rangle$$
$$= x_{i_0+n} - \mathbf{c}^T_{i_0+1}(i \mid n)\mathbf{y}_{i_0+n \mid n}. \tag{103}$$

4. Oblique residual

$$e'_{x,i_0}(i \mid n) \equiv x_i - \mathbf{c}^T_{i_0}(i-1 \mid n)\mathbf{y}_{i \mid n}. \tag{104}$$

The following order recursions are obtained from (22):

$$\mathbf{E}_{x,i_0}(i \mid n+1) = \mathbf{E}_{x,i_0+1}(i \mid n) - \frac{k^{(x)}_{n+1,i_0}(i)}{\epsilon_{b,i_0}(i \mid n)}\, \mathbf{E}_{b,i_0}(i \mid n), \tag{105}$$

$$\epsilon_{x,i_0}(i \mid n+1) = \epsilon_{x,i_0+1}(i \mid n) - \frac{k^{(x)2}_{n+1,i_0}(i)}{\epsilon_{b,i_0}(i \mid n)}, \tag{106}$$

$$[\mathbf{c}_{i_0}(i \mid n+1)]_{n+1} = \frac{k^{(x)}_{n+1,i_0}(i)}{\epsilon_{b,i_0}(i \mid n)}, \tag{107a}$$

and

$$[\mathbf{c}_{i_0}(i \mid n+1)]_{1,n} = \mathbf{c}_{i_0+1}(i \mid n) - [\mathbf{c}_{i_0}(i \mid n+1)]_{n+1}\mathbf{b}_{i_0}(i \mid n). \tag{107b}$$

Derivation of the following forward time updates involves a straightforward application of (34) and (35), where $\mathbf{Y}_i$ is replaced by $\mathbf{X}_{i_0+n,i}$ and $M_i$ is replaced by $M_i(0, n-1)$:

$$\mathbf{c}_{i_0}(i \mid n) = \mathbf{c}_{i_0}(i-1 \mid n) + e'_{x,i_0}(i \mid n)\mathbf{g}_{i_0}(i \mid n), \tag{108}$$

$$k^{(x)}_{n+1,i_0}(i) = k^{(x)}_{n+1,i_0}(i-1) + e_{x,i_0+1}(i \mid n)e_{b,i_0}(i \mid n)\, \frac{1}{1 - \gamma_{i_0+1}(i \mid n)}, \tag{109}$$

$$\epsilon_{x,i_0}(i \mid n) = \epsilon_{x,i_0}(i-1 \mid n) + e^2_{x,i_0}(i \mid n)\, \frac{1}{1 - \gamma_{i_0}(i \mid n)}, \tag{110}$$

$$e^*_{x,i_0}(i \mid n) = e^*_{x,i_0}(i-1 \mid n) - e_{x,i_0}(i \mid n)\, \frac{\alpha_{i_0}(i \mid n)}{1 - \gamma_{i_0}(i \mid n)}, \tag{111}$$

and

$$e'_{x,i_0}(i \mid n) = \frac{e_{x,i_0}(i \mid n)}{1 - \gamma_{i_0}(i \mid n)}. \tag{112}$$

Similarly, the following backward time updates are obtained from (43) and (44):

$$\mathbf{c}_{i_0}(i \mid n) = \mathbf{c}_{i_0+1}(i \mid n) + \mathbf{h}_{i_0}(i \mid n) \frac{e^*_{x,i_0}(i \mid n)}{1 - \gamma^*_{i_0}(i \mid n)}, \tag{113}$$

$$e_{x,i_0+1}(i \mid n) = e_{x,i_0}(i \mid n) + e^*_{x,i_0}(i \mid n) \frac{\alpha_{i_0}(i \mid n)}{1 - \gamma^*_{i_0}(i \mid n)}, \tag{114}$$

$$k^{(x)}_{n+1,i_0+1}(i) = k^{(x)}_{n+1,i_0}(i) - e^*_{x,i_0+1}(i \mid n) e^*_{b,i_0}(i \mid n) \frac{1}{1 - \gamma^*_{i_0+1}(i \mid n)}, \tag{115}$$

and

$$\epsilon_{x,i_0+1}(i \mid n) = \epsilon_{x,i_0}(i \mid n) - e^{*2}_{x,i_0}(i \mid n) \frac{1}{1 - \gamma^*_{i_0}(i \mid n)}. \tag{116}$$

Combining (104), (108), (103), and (113) (in that order) with the fixed-order sliding-window algorithm (93) gives the corresponding sliding-window joint-process-estimation algorithm. Adding these additional recursions results in a total computational complexity of $16N + 17$ multiplies and $16N + 13$ additions per iteration. This should be compared with $23N$ multiplies and $14N$ additions per iteration required by the unnormalized sliding-window lattice joint-process estimator. Initialization of these additional recursions is accomplished in a fashion analogous to the prediction recursions. In particular, the data $y_i$ and $x_i$ is assumed to be zero for $i < 0$, and $\mathbf{c}_{i_0}(-1 \mid n) = \mathbf{0}$.

The fixed-order growing-memory algorithm (96) is extended to the joint-process-estimation case by adding the recursions (104) and (108). The order-recursive prediction algorithm listed in Section 6.2 is extended to the joint-process-estimation case by adding the recursions (105) (top component only), (109), (107), (111), and (113). In each case the variable $i_0 = 0$. Adding (104) and (108) to (96) results in a total computational complexity of $13N + 7$ multiplies and $13N + 1$ additions per iteration. This should be compared with $28N$ multiplies and $16N$ additions per iterations required by the growing-memory covariance lattice joint-process estimator. The following accomplishes the initialization of the additional recursions for the order-recursive algorithm:

$$k^{(x)}_n(-1) = 0, \qquad 1 \leq n \leq N, \tag{117a}$$

$$\mathbf{c}(-1 \mid n) = \mathbf{0}, \qquad 0 \leq n \leq N, \tag{117b}$$

and

$$e_x^*(0 \mid n) = \begin{cases} x_0 & \text{for } n = 0 \\ 0 & \text{for } n > 0. \end{cases} \tag{117c}$$

The fixed-order algorithm is initialized by using the order-recursive algorithm for $i < N$.

## VIII. CONCLUSIONS

We have presented new fixed-order algorithms that recursively solve the sliding-window and growing-memory covariance least-squares estimation problems. The fixed-order growing-memory algorithm requires approximately one half the number of multiplies and divides required by the analogous unnormalized order-recursive or lattice algorithm. The fixed-order sliding-window algorithm requires approximately 70 percent of the number of multiplies and divides required by the analogous lattice algorithm. These fixed-order algorithms also help complete the list of computationally efficient LS algorithms currently available. In particular, each type of windowing technique that has been proposed for the LS computation (i.e., prewindowed, growing-memory covariance, and the sliding window) has resulted in both computationally efficient fixed-order and order-recursive algorithms. The order-recursive algorithms offer the advantage of being able to dynamically choose the order of the autoregressive model, while the fixed-order algorithms require less computation.

Associated with the algorithms mentioned in this paper are performance issues such as the relative convergence speed of each algorithm given different types of stationary and nonstationary random inputs, and the evaluation of finite word-length effects. As an example, the relative performance improvement offered by LS covariance algorithms over LS prewindowed algorithms has yet to be ascertained in applications where the prediction coefficients must be estimated from relatively few data samples. These issues will play a crucial role in determining the practical value of the LS algorithms presented in this paper.

## REFERENCES


1. D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," IEEE Trans. Commun., *COM-26* (October 1978), pp. 1439–46.
2. E. Satorius and J. Pack, "Application of Least Squares Lattice Algorithms to Adaptive Equalization," IEEE Trans. Commun., *COM-29* (February 1981), pp. 136–42.
3. T. L. Lim and M. S. Mueller, "Rapid Equalizer Start-Up Using Least Squares Algorithms" 1980 Proc. IEEE ICC, Seattle, WA.
4. M. Mueller, "On the Rapid Initial Convergence of Least Squares Equalizer Adjustment Algorithms," B.S.T.J., *60*, No. 10 (December 1981), pp. 2345–58.

5. F. K. Soong and A. M. Pederson, "Fast Least-Squares (LS) in the Voice Echo Cancellation Application," Proc. 1982 IEEE ICASSP, Paris, France, May 1982.
6. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, NJ: Prentice-Hall, 1978.
7. M. Morf and D. T. L. Lee, "Fast Algorithms for Speech Modeling," Technical Report No. M308-1, Information Systems Laboratory, Stanford University, Stanford, CA, December 1978.
8. B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory*, R. Kalman and N. De Claris, Eds. New York, NY: Holt, Rinehart, and Winston, 1971, pp. 563–87.
9. R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the Design of Gradient Algorithms for Digitally Implemented Adjustment Filters," IEEE Trans. Circuit Theory, *CT-20* (March 1973), pp. 125–36.
10. L. J. Griffiths, "A Continuously-Adaptive Filter Implemented as a Lattice Structure," Proc. 1977 IEEE ICASSP, Hartford, CT (May 1977), pp. 683–6.
11. M. Morf, B. Dickinson, T. Kailath, and A. Vieira, "Efficient Solution of Covariance Equations for Linear Prediction," IEEE Trans. ASSP, *ASSP-25* (October 1977), pp. 429–33.
12. L. Ljung, M. Morf, and D. Falconer, "Fast Calculation of Gain Matrices for Recursive Estimation Schemes," Int. J. Contr. *27*, No. 1 (1978), pp. 1–19.
13. D. T. Lee, B. Friedlander, and M. Morf, "Recursive Least Squares Ladder Estimation Algorithms," IEEE Trans. ASSP, *ASSP-29* (June 1981), pp. 627–41.
14. B. Porat, B. Friedlander, and M. Morf, "Square-Root Covariance Ladder Algorithms," IEEE Trans. Aut. Control, *AC-27*, No. 4 (August 1982), pp. 813–29.
15. C. Samson, "A Unified Treatment of Fast Kalman Algorithms for Identification," Int. J. Control, *35*, No. 5 (May 1982), pp. 909–34.
16. B. Friedlander, "Lattice Filters for Adaptive Processing," Proc. IEEE, *70*, No. 8 (August 1982), pp. 829–67.
17. S. L. Marple, Jr., "Efficient Least Squares FIR System Identification," IEEE Trans. ASSP, *ASSP-29* (Feb. 1981), pp. 62–73.
18. M. Morf, D. T. Lee, J. R. Nickolls, and A. Vieira, "A Classification of Algorithms for ARMA Models and Ladder Realizations," Proc. 1977 IEEE Conf. ASSP, Hartford, CT (April 1977), pp. 13–9.
19. M. Morf, D. T. Lee, "Recursive Least Squares Ladder Forms for Fast Parameter Tracking," Proc. 1978 IEEE Conf. D&C, San Diego, CA (Jan. 12, 1979), pp. 1326–67.
20. D. T. L. Lee and M. Morf, "Recursive Square-Root Ladder Estimation Algorithms," Proc. 1980 IEEE ICASSP, Denver, CO, April 1980.
21. A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science*, New York, NY: Holt, Rinehart and Winston, Inc., 1971.
22. J. Cioffi and T. Kailath, "Fast, Fixed-Order, Least-Squares Algorithms for Adaptive Filtering," Proc. 1983 IEEE ICASSP, Boston MA, April 1983.
23. C. C. Halkias et al., "A New Generalized Recursion for the Fast Computation of the Kalman Gain to Solve the Covariance Equations," Proc. 1982 IEEE ICASSP, Paris, France (April 1982), pp. 1760–3.
24. M. L. Honig, "Performance of FIR Adaptive Filters Using Recursive Algorithms," Ph.D. Dissertation, Univ. of California, Berkeley, April 1981.

## APPENDIX A

### *Derivation of (30)*

We wish to prove (30). By definition,

$$P_{M_i|i-1}\mathbf{Y}_i = P_{\tilde{M}_i}\mathbf{Y}_i + P_{U_i}P_{M_i|i-1}\mathbf{Y}_i, \tag{118}$$

where $\tilde{M}$ is the subspace spanned by the column vectors of $\tilde{\mathbf{S}}_i$. Projecting both sides of (118) onto $M_i$ gives

$$P_{M_i}P_{M_i|i-1}\mathbf{Y}_i = P_{M_i}P_{\tilde{M}_i}\mathbf{Y}_i + P_{M_i}P_{U_i}P_{M_i|i-1}\mathbf{Y}_i. \tag{119}$$

Now $P_{M_i|i-1}\mathbf{Y}_i$ lies in $M_i$, and hence

$$P_{M_i}P_{M_i|i-1}\mathbf{Y}_i = P_{M_i|i-1}\mathbf{Y}_i. \tag{120}$$

Also,

$$P_{M_i}P_{\tilde{M}_i}\mathbf{Y}_i = \mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}(\mathbf{S}_i^T\tilde{\mathbf{S}}_i)(\tilde{\mathbf{S}}_i^T\tilde{\mathbf{S}}_i)^{-1}\tilde{\mathbf{S}}_i^T\mathbf{Y}_i$$

$$= \mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\tilde{\mathbf{S}}_i^T\mathbf{Y}_i$$

$$= P_{M_i}(\mathbf{Y}_i - P_{U_i}\mathbf{Y}_i). \tag{121}$$

Combining (118) through (121) gives

$$P_{M_i}\mathbf{Y}_i = P_{M_{i|i-1}}\mathbf{Y}_i + P_{M_i}P_{U_i}P^{\perp}_{M_{i|i-1}}\mathbf{Y}_i$$

$$= P_{M_{i|i-1}}\mathbf{Y}_i + (P_{M_i}\mathbf{u}_i)\langle \mathbf{u}_i,\ P^{\perp}_{M_{i|i-1}}\mathbf{Y}_i\rangle. \tag{122}$$

Subtracting both sides of (122) from $\mathbf{Y}_i$, and then taking inner products of both sides with $\mathbf{u}_i$ gives

$$\langle \mathbf{u}_i,\ P^{\perp}_{M_i}\mathbf{Y}_i\rangle = \langle \mathbf{u}_i,\ P_{M_{i|i-1}}\mathbf{Y}_i\rangle[1 - \langle \mathbf{u}_i,\ P_{M_i}\mathbf{u}_i\rangle]. \tag{123}$$

Combining (122) and (123), and using the definition (31) gives (30). [Ref. 24 gives a purely geometric proof of (30) for the case where $M_i$ is spanned by one vector (as illustrated in Fig. 2).]

To derive the inner product update (35), we first rewrite (34) as

$$P^{\perp}_{M_i}\mathbf{Y}_i = P^{\perp}_{U_i}P^{\perp}_{M_{i|i-1}}\mathbf{Y}_i + P_{U_i}P^{\perp}_{M_{i|i-1}}\mathbf{Y}_i - P_{M_i}\mathbf{u}_i\langle \mathbf{u}_i, P^{\perp}_{M_i}\mathbf{Y}_i\rangle\sec^2\theta_i$$

$$= P^{\perp}_{U_i}P^{\perp}_{\tilde{M}_i}\mathbf{Y}_i + \mathbf{u}_i\langle \mathbf{u}_i,\ P^{\perp}_{M_{i|i-1}}\mathbf{Y}_i\rangle - P_{M_i}\mathbf{u}_i\langle \mathbf{u}_i,\ P^{\perp}_{M_i}\mathbf{Y}_i\rangle\sec^2\theta_i$$

$$= P^{\perp}_{U_i}P^{\perp}_{\tilde{M}_i}\mathbf{Y}_i + P^{\perp}_{M_i}\mathbf{u}_i\langle \mathbf{u}_i,\ P^{\perp}_{M_i}\mathbf{Y}_i\rangle\sec^2\theta_i. \tag{124}$$

Taking the inner product of both sides with $\mathbf{v}_i$ and using the fact that

$$\langle \mathbf{v}_i,\ P^{\perp}_{M_i}\mathbf{u}_i\rangle = \langle \mathbf{u}_i,\ P^{\perp}_{M_i}\mathbf{v}_i\rangle \tag{125}$$

gives (35).

## APPENDIX B

### *Other Recursive Least-Squares Algorithms*

The recursions in Section V and VII are complete in the sense that any of the existing computationally efficient LS prediction or joint-process-estimation algorithms can be derived from suitable subsets of these recursions. The purpose of this appendix is to illustrate this point by listing the recursions that enter the prewindowed LS transversal (fast Kalman) and lattice algorithms, the unnormalized sliding-window and growing-memory covariance lattice algorithms,[14] and the nonrecursive LS algorithm presented in Refs. 11 and 17. The list of recursions presented below does not completely describe each algorithm. For example, initialization is not discussed. Consistent steady-state algorithms can be formulated, however, by choosing the time indices and order of the variables in each recursion appropriately. The

following algorithms apply to the more general joint-process-estimation case (eliminating the recursions from Section VII gives the analogous prediction algorithm):

1. Prewindowed transversal (fast Kalman) algorithm:[1] (60a), (77a), (51a), (90a), (69), (60b), (93m), (68b), (104), and (108).
2. Prewindowed lattice algorithm* (See Refs. 18 and 16): (89), (64a)† and (64b), (65a) and (65b), (72a), (109) and (105).
3. Sliding-window lattice algorithm:[14,16] (89), (64a) and (64b), (65a) and (65b), (72a), (73a), (91), (109), (105), and (115).
4. Growing-memory covariance lattice algorithm:[14,16] (89), (64a) and (64b), (65a) and (65b), (82a), (88a), (92a), (72b) and (72a), (73b), (74b), (109), (105), (111), and (114).
5. Nonrecursive LS algorithm:[11,17]

The following set of recursions, which represents a modified version of the algorithm presented in Ref. 17, can be used to compute $\mathbf{f}(i \mid N)$, $\mathbf{b}(i \mid N)$, and $\mathbf{c}(i \mid N)$, given by (4), in an order-recursive fashion starting with first-order least squares variables at time $i$. Initialization consists of computing these first-order variables via the definitions given in Section V.

(78) (for computing $\mathbf{h}(i-1 \mid n)$), (85) (for computing $\mathbf{g}_1(i \mid n)$), (79), (86), (84a), (77b), (92a), (90b), (53), (66), (67), (65a) and (65b), (51b), (61), (68), (71), (57c), (72a), (73b), (103), (113), (101), (107).

Assuming that the covariance matrix $\Phi_{N,i|N+1}$ has been computed, a more convenient form for (53) is

$$\begin{aligned} k_n(i) &= \langle \mathbf{Y}_{n,i},\ \mathbf{E}_b(i-1 \mid n-1) \rangle \\ &= \mathbf{Y}_{n,i}^T [z^{-n}\mathbf{Y}_{n,i} - \mathbf{S}_i(1,\ n-1)\mathbf{b}(i-1 \mid n-1)] \\ &= R_n - \sum_{j=1}^{n-1} R_{n-j}[\mathbf{b}(i-1 \mid n-1)]_j, \end{aligned} \tag{126}$$

where $R_j = \mathbf{Y}_{n,i}^T(z^{-j}\mathbf{Y}_{n,i})$, and is the $(1,\ j+1)$st element of $\Phi_{n,i|n+1}$. Equation (101) can be similarly modified.

## AUTHOR

**Michael L. Honig,** B.S. (Electrical Engineering), 1977, Stanford University; M.S. and Ph.D. (Electrical Engineering), University of California, Berkeley, 1978 and 1981, respectively; AT&T Bell Laboratories, 1981–1982, AT&T

---

* The normalized lattice algorithms presented in Refs. 13 and 14 can be obtained by making substitutions for the variables entering the unnormalized recursions presented here.[16,20,24]

† In the prewindowed and growing-memory covariance cases, the inner products of (64) and (105) with $\mathbf{u}_i$ are required. In the sliding-window case, the inner products of (64) and (105) with both $\mathbf{u}_i$ and $\mathbf{u}_{i_0}$ are required.

Information Systems, 1983—. At AT&T Bell Laboratories and AT&T Information Systems Mr. Honig has worked on modulation, coding, and echo cancellation of voiceband data signals; and performance analysis of local area networks. He is currently working on office information networks. Member, IEEE, Tau Beta Pi, Phi Beta Kappa.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# On the Average Product of Gauss-Markov Variables

By B. F. LOGAN, Jr.,* J. E. MAZO,* A. M. ODLYZKO* and L. A. SHEPP*

(Manuscript received April 8, 1983)

Let $x_i$ be members of a stationary sequence of zero mean gaussian random variables having correlations $Ex_ix_j = \sigma^2\rho^{|i-j|}$, $0 < \rho < 1$, $\sigma > 0$. We address the behavior of the averaged product $q_m(\rho, \sigma) \equiv Ex_1x_2 \cdots x_{2m-1}x_{2m}$ as $m$ becomes large. Our principal result when $\sigma^2 = 1$ is that this average approaches zero (infinity) as $\rho$ is less (greater) than the critical value $\rho_c = 0.563007169\ldots$. To obtain this we introduce a linear recurrence for the $q_m\cdot(\rho, \sigma)$, and then continue generating an entire sequence of recurrences, where the $(n + 1)$-st relation is a recurrence for the coefficients that appear in the $n$th relation. This leads to a new, simple continued fraction representation for the generating function of the $q_m(\rho, \sigma)$. The related problem with $\tilde{q}_m(\rho, \sigma) = E|x_1 \cdots x_m|$ is studied via integral equations and is shown to possess a smaller critical correlation value.

## I. INTRODUCTION

The problem that we consider in this paper is as follows: Let $\{x_i\}_1^\infty$ be a stationary sequence of zero mean, gaussian random variables with covariances

$$\rho_{ij} \equiv Ex_ix_j = \sigma^2\rho^{|i-j|},\ 0 < \rho < 1,\ \sigma > 0; \quad i, j = 1, 2, \cdots, \quad (1)$$

where $E(\cdot)$ denotes mathematical expectation. What is the behavior of

*Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

$$q_m(\rho, \sigma) \equiv Ex_1x_2 \cdots x_{2m-1}x_{2m} \tag{2}$$

as $m$ becomes large?

In other words, the product in (2) is formed from samples of a gauss-markov process that are taken at regular intervals. Only an even number of samples is considered in (2) since an odd number would result in a zero average.

Originally, the problem was conceived as a simple model for averages of multiplicative structures having infinite memory between the factors of the product. Such products arise in the analysis of learning curves for many adaptive systems, and for these problems one encounters products whose factors are noncommuting matrices. We felt that the analysis of a simple problem, such as that described above, would serve as a valuable guide to what results might be achievable for more realistic situations. However as one may readily imagine, as soon as the problem described in (1) and (2) was written down it became of interest in its own right, consisting as it does of a simple question about long familiar quantities.

Our principal result is that for large $m$ the behavior of the average product $q_m(\rho, \sigma)$ in (2) depends on the relationship of $\rho$ to a critical value, $\rho_c = \rho_c(\sigma)$. If $\rho < \rho_c$, then $q_m(\rho, \sigma)$ will approach zero exponentially fast; if $\rho > \rho_c$, $q_m(\rho, \sigma)$ approaches infinity exponentially fast; finally, if $\rho = \rho_c$, $q_m(\rho, \sigma) \to q_\infty(\sigma)$. We find for $\sigma = 1$, $\rho_c(1) = 0.563007169391816 \cdots$, and $q_\infty(1) = 0.50900853 \ldots$. A plot of $\rho_c(\sigma)$ is given in Fig. 1. All of these results were obtained from a continued fraction representation for the generating function

$$Q(z, \rho, \sigma) = \sum_{m=0}^{\infty} q_m(\rho, \sigma) z^m. \tag{3}$$

Since $q_m(\rho, \sigma) = \sigma^{2m} q_m(\rho, 1)$, we have

$$Q(z, \rho, \sigma) = Q(z\sigma^2, \rho, 1), \tag{4}$$

so it is without loss of generality that we will set $\sigma = 1$, $Q(z, \rho) = Q(z, \rho, 1)$, and $q_m(\rho) = q_m(\rho, 1)$. By introducing a sequence of generating functions, we show in Section II that

$$Q(z, \rho) = \cfrac{1}{1 - \cfrac{\rho z}{1 - \cfrac{2\rho^3 z}{1 - \cfrac{3\rho^5 z}{1 - \ddots}}}} \tag{5}$$

The value $\rho_c^{(\sigma)}$ is then the smallest $\rho$ for which $Q(\sigma^2, \rho) = \infty$, while the value $q_\infty(\sigma)$ is the limit as $z \to 1$ of $(1 - z)Q(z\sigma^2, \rho_c)$.

Since methods are as interesting as results, Section III presents another approach involving integral equations for discussing the $q_m(\rho)$ behavior. Although this method is not rigorously justified for the present problem due to a non-hermitian kernel, it is applicable to a

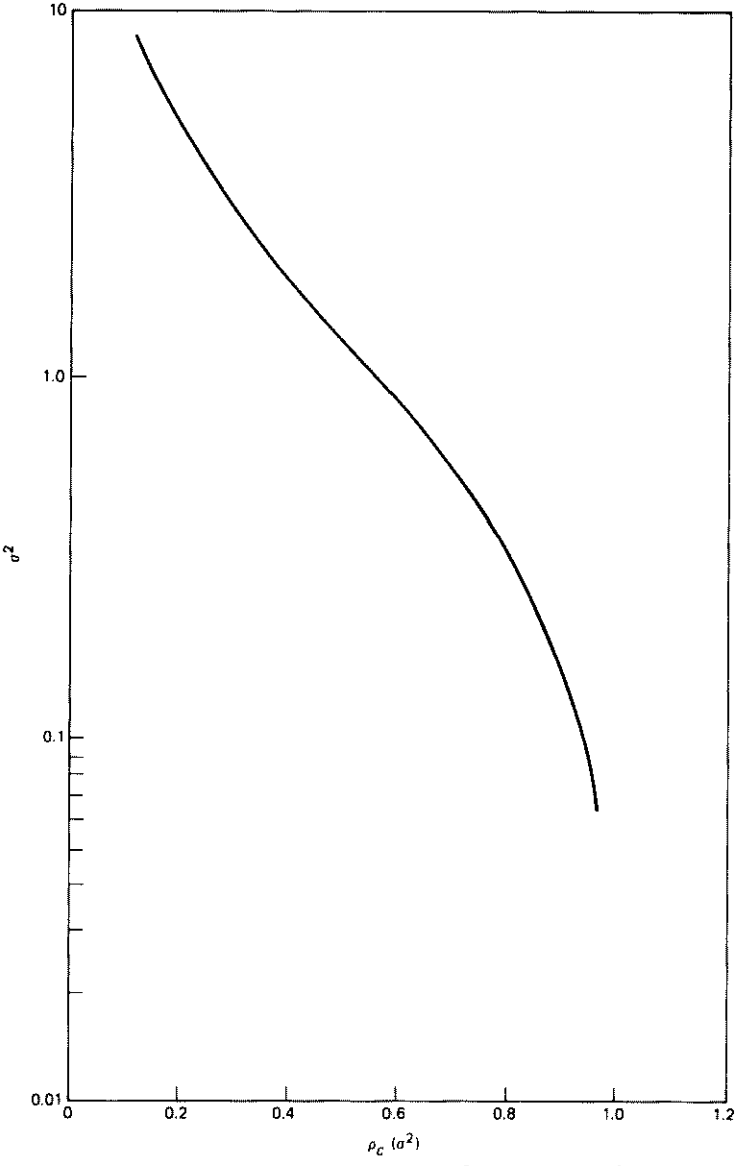



Fig. 1—Critical correlation value $\rho_c(\sigma^2)$ vs. variance $\sigma^2$.

related problem, the behavior of $E|x_1 \cdots x_m|$ as $m \to \infty$ [still assuming (1)]. Using the integral equation we show that $\bar{\rho}_c$, the critical value of $\rho$ for this new problem, is strictly smaller than the $\rho_c$ defined above. This is of interest since it shows that the behavior of $q_m(\rho)$ is determined both by how large $|x_1 \cdots x_{2m}|$ is on the average, and by the extent of cancellation between positive and negative values of $q_m$.

Although we do not give the details here, it is not difficult to show that for all $\rho < 1$, $q_m(\rho)$ approaches zero with probability one as $m$ becomes large.

## II. LINEAR RECURRENCES AND GENERATING FUNCTIONS

Given $2m$ zero-mean jointly gaussian random variables $x_i$ of unit variance and correlations $Ex_ix_j = \rho_{ij}$, then a known formula[1] states that

$$Ex_1 \cdots x_{2m} = \sum_{\substack{\text{all} \\ \text{pairs}}} \rho_{i_1 i_2} \rho_{i_3 i_4} \cdots \rho_{i_{2m-1} i_{2m}}, \tag{6}$$

where the unordered set $\{i_1, \cdots, i_{2m}\}$ is equal to the unordered set $\{1, 2, \cdots, 2m\}$. The sum in (6) is over all distinct, unordered pairs of subscripts. That is, we do not count twice terms which differ only by interchanging the values within one or more subscript pairs, nor do we count twice terms which differ only by permuting subscript pairs. Thus there are $(2m)!/(2^m m!)$ terms in the sum (6).

If we denote permutations of $2m$ objects by $\sigma(i)$: $i \to \sigma(i)$, $i = 1, 2, \cdots, 2m$, then a succinct way of writing (6) when (1) holds is

$$q_m(\rho) = \frac{1}{2^m m!} \sum_{\sigma \in S_{2m}} \rho^{\sum_{j=1}^{m} |\sigma(2j) - \sigma(2j-1)|}, \tag{7}$$

the sum in (7) being over all $(2m)!$ permutations of $S_{2m}$, the group of permutations of $2m$ symbols. Formula (7) shows immediately that $q_m(\rho) > 0$ if $\rho > 0$.

Now define $q_0(\rho) = 1$ and write

$$q_m(\rho) = \sum_{s=1}^{m} b_s(\rho) q_{m-s}(\rho), \qquad m = 1, 2, \cdots. \tag{8}$$

We evaluate a few of the $b_s(\rho)$, writing for convenience $b_i(\rho) = b_i$, $q_i(\rho) = q_i$. The evaluation is done from (8) by explicitly evaluating the $q_m(\rho)$ as needed. A partial list of $b_i(\rho)$ follows:

$$b_1 = \rho$$

$$b_2 = 2\rho^4$$

$$b_3 = 4\rho^7 + 6\rho^9$$

$$b_4 = 8\rho^{10} + 24\rho^{12} + 18\rho^{14} + 24\rho^{16}$$

$$b_5 = 16\rho^{13} + 72\rho^{15} + 108\rho^{17} + 150\rho^{19} + 144\rho^{21} + 96\rho^{23} + 120\rho^{25}. \tag{9}$$

Equation (9) suggests the possibility that, for small $\rho$, only a few terms in (8) would need to be kept for an accurate description of $q_m(\rho)$. For example, keeping only one term yields

$$q_m = \rho q_{m-1}, \tag{10}$$

or $q_m = \rho^m$. Since $Ex_1x_2 = \rho$, this approximation corresponds to treating the successive pairs of gaussian variables which determine $q_m(\rho)$, via (2), as independent. The next step after (10) would be to write

$$q_m = b_1 q_{m-1} + b_2 q_{m-2}. \tag{11}$$

This equation, involving $b_2$ as well, would be a correction to the "independence assumption," but one involving only up to fourth-order correlations, since, from (8) the highest average appearing in $b_2$ is $E(x_1x_2x_3x_4)$. Further corrections are obtained by including more terms of (8), with higher order correlations entering.†

Assuming the $b_i(\rho)$ to be known, the natural procedure would be to "solve" (8) using generating functions. We define these as follows: if $y_0, y_1, y_2, \cdots$ is a bounded sequence of numbers, then the generating function, $Y(z)$, of the sequence is defined for complex $z$, $|z| < 1$, by

$$Y(z) = \sum_{i=0}^{\infty} y_i z^i. \tag{12}$$

Given $Y(z)$, the $y_i$ are, in principle, uniquely determined. We assume that the reader is familiar with the use of generating functions. If not, consult Chapters XI and XIII of Feller.[2]

We define $b_0(\rho) = 0$, $q_0(\rho) = 1$, and call the generating functions of the $b_i(\rho)$, and $q_i(\rho)$ sequences $B(z; \rho)$ and $Q(z; \rho)$, respectively. The $\rho$ dependence is explicitly indicated.

If we multiply (8) by $z^m$ and sum from $m = 1$ to $\infty$ (treating $q_m = 0$, $m < 0$ and $b_m = 0$, $m < 0$), we obtain the basic relation

$$Q(z; \rho) = \frac{1}{1 - B(z; \rho)}. \tag{13}$$

Equation (13) thus allows us to determine, in principle, the $q_m$ from the $b_m$. In particular, we have

†The above interpretation prompts us to advocate consideration of the ideas represented by (8) for analyzing more complex multiplicative structures, particularly when connections to some sort of independence approximation are a natural thing to seek.

$$\sum_{m=1}^{\infty} q_m(\rho) = Q(1; \rho) = \frac{1}{1 - B(1; \rho)}, \tag{14}$$

and the critical value $\rho_c$ will be given by the equation

$$B(1; \rho_c) = \sum_{1}^{\infty} b_k(\rho_c) = 1. \tag{15}$$

Although we could work with the $b_m(\rho)$ themselves, a more convenient approach for finding $\rho_c$ numerically is to set up a continued fraction representation for the generating functions $Q(z; \rho)$, or equivalently, $B(z; \rho)$. It is this approach that we follow now.

Recall (8) defining $b_s(\rho)$. Since these $b_s$ coefficients are a numerical sequence themselves, we can use the same reasoning that took us from the $q_k$ to the $b_s$ and use it to suggest going from the $b_s$ to a new set of coefficients, $b_k^{(2)}$, via the following recurrence

$$b_k(\rho) = \sum_{s=1}^{k} b_s^{(2)}(\rho) b_{k-s}(\rho), \qquad k = 2, 3, \cdots, \tag{16}$$

where we define $b_0(\rho) = 0$. The recurrence (16) yields

$$B(z; \rho) = \frac{b_1(\rho) z}{1 - B^{(2)}(z; \rho)}, \tag{17}$$

$B^{(2)}(z; \rho)$ being the generating function for the $b_k^{(2)}(\rho)$. To continue this procedure with a uniform notation, we define

$$\begin{aligned} b_s^{(1)}(\rho) &= b_s(\rho) \\ b_0^{(m)}(\rho) &= 0, \qquad m = 1, 2, \cdots \end{aligned} \tag{18}$$

and write

$$b_k^{(m)}(\rho) = \sum_{s=1}^{k} b_s^{(m+1)}(\rho) b_{k-s}^{(m)}(\rho), \qquad \begin{matrix} m = 1, 2, \cdots \\ k = 2, 3, \cdots \end{matrix}. \tag{19}$$

The corresponding sequence of generating functions are related by

$$B^{(m)}(z; \rho) = \frac{b_1^{(m)}(\rho) z}{1 - B^{(m+1)}(z; \rho)}. \tag{20}$$

We use this repeatedly in (13) and obtain the continued fraction representation†

† The fact that this continued fraction does not terminate implies that $Q(z; \rho)$ is not a rational function of $z$, and thus one cannot find a (finite-order) difference equation for the $q_m(\rho)$. See Ref. 3, Theorem 99.1, p. 400.

$$Q(z,\rho)=\cfrac{1}{1-\cfrac{b_1^{(1)}z}{1-\cfrac{b_1^{(2)}z}{1-\cfrac{b_1^{(3)}z}{1-\ddots}}}} \tag{21}$$

In (21) we have, for simplicity, written $b_1^{(m)}(\rho)=b_1^{(m)}$.

A relation which will be used later to aid in finding the $b_1^{(m)}$ follows by setting $k=2$ in (19), to obtain

$$b_1^{(m+1)}(\rho)=\frac{b_2^{(m)}(\rho)}{b_1^{(m)}(\rho)}. \tag{22}$$

We can calculate some of the $b_1^{(m)}(\rho)$ using the partial list of the $b_k(\rho)$ given in (9) to derive several $b_s^{(m)}(\rho)$ from (19). Using (22) we then obtain

$$\begin{aligned} b_1^{(1)} &= \rho \\ b_1^{(2)} &= 2\rho^3 \\ b_1^{(3)} &= 3\rho^5 \\ b_1^{(4)} &= 4\rho^7 \\ b_1^{(5)} &= 5\rho^9. \end{aligned} \tag{23}$$

The obvious guess that

$$b_1^{(m)}=m\rho^{2m-1}, \qquad m=1,2,\cdots \tag{24}$$

follows from a direct proof of the continued fraction given in the appendix. Assuming (24) to hold yields the simple representation

$$Q(z;\rho)=\cfrac{1}{1-\cfrac{\rho z}{1-\cfrac{2\rho^3 z}{1-\cfrac{3\rho^5 z}{1-\ddots}}}} \tag{25}$$

The accurate numerical value

$$\rho_c=0.563007169391816\cdots \tag{26}$$

was obtained by using this representation along with (14) and (15).

When $\rho = \rho_c$,

$$q_\infty = \lim_{k\to\infty} q_k(\rho_c) = \frac{1}{\sum_1^\infty k b_k(\rho_c)}. \tag{27}$$

Using the computed value of $\rho_c$ and the definition of the $b_k(\rho)$, we found numerically that

$$q_\infty = 0.50900853 \cdots. \tag{28}$$

It was quite surprising to us that $Q(z, \rho)$ turned out to be a new, but simple, continued fraction.

## III. INTEGRAL EQUATION METHOD

The purpose of this section is to introduce the integral equation method and to show that $\tilde{\rho}_c < \rho_c$, where $\tilde{\rho}_c$ is the critical correlation value for the related problem involving $E|x_1 \cdots x_n|$.

We begin by developing an expression for $Ex_1 \cdots x_n$. We have, from the Markov property of the $x_i$ sequence,

$$Ex_1 \cdots x_n = \int \cdots \int x_n p(x_n|x_{n-1}) \cdots x_1 p(x_1|x_0)\phi(x_0)dx_0 \cdots dx_n, \tag{29}$$

where

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \tag{30}$$

is the standard normal density and

$$p(y|x) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp[-(y-\rho x)^2/2(1-\rho^2)] \tag{31}$$

is the generic form of the conditional densities occurring in (29). Define a kernel $K(x, y)$ by

$$K(x, y) = yp(y|x),$$

$$Kf(x) = \int_{-\infty}^{\infty} K(x, y)f(y)dy.$$

Then (29) may be written in the inner product notation of Hilbert Space

$$Ex_1 \cdots x_n = (K^n \underline{1}, \phi), \tag{32}$$

where $\phi$ is the normal density (30), $\underline{1}$ is the unit constant function,

and $K^n$ is the $n$th iterated kernel. Now assume, heuristically, that $K^n$ has the usual expansion

$$K^n(x, y) = \sum_{j=1}^{\infty} \lambda_j^n \psi_j(x)\psi_j(y) \tag{33}$$

in terms of eigenfunctions $\psi_j(x)$ and eigenvalues $\lambda_j$ of $K$. Then $Ex_1 \cdots x_n$ would remain bounded, if, and only if, the largest eigenvalue $\lambda_1 = \lambda_1(\rho)$ is less or equal to one; thus $\lambda_1(\rho_c) = 1$ would determine $\rho_c$. Unfortunately there is no general eigenexpansion theory available for $K$ since it is not symmetric and is not symmetrizable.

Fortunately symmetry holds for the integral equation method when one expresses $E|x_1 \cdots x_n|$ via kernels. Define, in analogy to (32),

$$\bar{K}(x, y) = |y| p(y|x). \tag{34}$$

If we further define

$$J(x, y) = \frac{h(x)}{h(y)} \bar{K}(x, y), \tag{35}$$

where

$$h(x) = \sqrt{|x|} \exp(-x^2/4), \tag{36}$$

then

$$J(x, y) = \frac{1}{\sqrt{2\pi(1 - \rho^2)}} \sqrt{|xy|} \exp\left[-\frac{1}{4}\frac{1 + \rho^2}{1 - \rho^2}(x^2 + y^2)\right] \cdot \exp\left[\frac{\rho xy}{1 - \rho^2}\right] \tag{37}$$

is a symmetric kernel.

As in (29),

$$\begin{aligned} E|x_1 \cdots x_n| &= \int \cdots \int \bar{K}(x_{n-1}, x_n)\bar{K}(x_{n-2}, x_{n-1}) \\ &\qquad \cdots \bar{K}(x_0, x_1)\, \phi(x_0) dx_0 \cdots dx_n \\ &= \int \cdots \int dx_0 \cdots dx_n\, J(x_0, x_1) \\ &\qquad \cdots J(x_{n-1}, x_n) \frac{h(x_n)}{h(x_0)} \phi(x_0) \\ &= \left(J^n h, \frac{\phi}{h}\right). \end{aligned} \tag{38}$$

Since $J$ is symmetric and square-integrable, it is a Hilbert-Schmidt kernel and so has a discrete spectrum. Further its maximum eigenvalue, $\lambda$, is given by

$$\lambda = \sup_f \frac{(Jf, f)}{(f, f)}. \tag{39}$$

Since $J(x, y) \geq 0$, we see that the maximum eigenfunction $g = g(x)$ is nonnegative and $\lambda > 0$ as well. Further since $h$ and $\phi/h$ are nonnegative, $(h, g) > 0$ and $(\phi/h, g) > 0$ so that $E|x_1 \cdots x_n| = (J^n h, \phi/h) \to \infty$ if and only if $\lambda > 1$.

Define $f_\alpha$ by

$$f_\alpha(x) = \sqrt{|x|}\exp(-\alpha x^2/4), \tag{40}$$

and note that from (39),

$$\lambda > (Jf_\alpha, f_\alpha)/(f_\alpha, f_\alpha). \tag{41}$$

Now

$$(f_\alpha, f_\alpha) = \int |x|\exp(-\alpha x^2/2)dx = 2/\alpha \tag{42}$$

and

$$(Jf_\alpha, f_\alpha) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |xy| \cdot \exp\left[-\frac{c}{2}(x^2+y^2) + \frac{\rho xy}{1-\rho^2}\right] dxdy, \tag{43}$$

where

$$c = \frac{1}{2}\left(\frac{1+\rho^2}{1-\rho^2} + \alpha\right). \tag{44}$$

Set $y = xu$ and integrate over $x$ to obtain

$$(Jf_\alpha, f_\alpha) = \frac{4}{\sqrt{2\pi(1-\rho^2)}} \int_{-\infty}^{\infty} \frac{|u|du}{(c(1+u^2) - \beta u)^2},$$

$$\beta = \frac{2\rho}{1-\rho^2}. \tag{45}$$

Using Ref. 4 (p. 68, 2.175) we evaluate the last integral as

$$(Jf_\alpha, f_\alpha) = \frac{4}{\sqrt{2\pi(1-\rho^2)}}\left[\frac{4}{\Delta} + \frac{4\beta}{\Delta^{3/2}}\tan^{-1}\frac{\beta}{\sqrt{\Delta}}\right],$$

$$(\Delta = 4c^2 - \beta^2. \quad (46)$$

Setting $\alpha = 0.5$, $\rho = 0.55$, we find that $c = 1.18369$, $\beta = 1.57706$, $\Delta = 3.11738$, $(Jf_\alpha, f_\alpha) = 4.0824$, $(f_\alpha, f_\alpha) = 4$, so that $\lambda > 1.012$. Thus for $\rho = 0.55$, $E|X_1 \cdots X_n| \to \infty$, and $\bar{\rho}_c < 0.55$. We have seen that $\rho_c > 0.563$ so the claim is proven.

## REFERENCES


1. D. Middleton, *An Introduction to Statistical Communication Theory*, New York: McGraw-Hill, 1960, p. 343.
2. W. Feller, *An Introduction to Probability Theory and Its Applications*, Volume 1, 2nd ed, New York: John Wiley and Sons, 1957.
3. H. S. Wall, *Analytic Theory of Continued Fractions*, New York: Van Nostrand Co., 1948.
4. I. S. Gradshteyn and I. R. Ryzhik, *Table of Integrals, Series, and Products*, New York: Academic Press, 1965.


## APPENDIX

### *Combinatorial Derivation of Continued Fraction*

In this appendix we give a direct combinatorial proof of the continued fraction representation (25) of $Q(z, \rho)$. This derivation is complete in itself, but we preferred the method of the text for showing where the continued fraction comes from. Our starting point is the formula (7). Let us define, for $\sigma \in S_{2m}$,

$$V(\sigma) = \sum_{i=1}^{m} |\sigma(2i) - \sigma(2i-1)|. \quad (47)$$

For $1 \leq k \leq m$, let

$$\begin{aligned} S(m, k) = \{\sigma \in S_{2m} : &\sigma(2m) = 2m, \\ &\sigma(2m-2) = 2m-1, \\ \sigma(2m-4) = 2m-2, &\cdots, \\ &\sigma(2m-2k+2) = 2m-k+1\}. \end{aligned} \quad (48)$$

For $k = 0$, we adopt the convention that $S(m, 0) = S_{2m}$. We also define

$$u(m, k) = \frac{1}{2^{m-k}(m-k)!}\sum_{\sigma \in S(m,k)} \rho^{V(s)}, \quad (49)$$

so that $u(m, 0) = q_m$. (We take $u(0, 0) = q_0 = 1$, and $u(m, k) = 0$ for $k < 0$ and $k > m$.) Our key result is:

*Lemma. If* $m \geq 1$, $k \geq 0$, *then*

$$u(m, k) = k\rho^{2k-1}u(m-1, k-1) + u(m, k+1). \tag{50}$$

*Proof.* We will prove this for $1 \leq k \leq m-1$, as the other cases are easy. Let

$$S' = \{\sigma \in S(m, k): 2m-k \in \{\sigma(2m-1), \sigma(2m-3), \cdots, \sigma(2m-2k+1)\}\},$$
$$S'' = S(m, k) - S'. \tag{51}$$

If $\sigma \in S''$, we construct a permutation $\sigma^* \in S(m, k+1)$ by changing the action of $\sigma$ on four letters in such a way that $V(\sigma) = V(\sigma^*)$ and $\sigma^*(2m-2k) = 2m-k$. To define $\sigma^*$ precisely, let $p$ and $r$ be such that $\{r, 2m-k\} = \{\sigma(2p-1), \sigma(2p)\}$. Then, if we associate to $\sigma$ the vector $A(\sigma) = (\sigma(1), \sigma(2), \cdots, \sigma(2m))$, the vector $A(\sigma^*)$ is obtained from $A(\sigma)$ by interchanging the pairs $\{\sigma(2m-2k-1), \sigma(2m-k)\}$ and $\{r, 2m-k\}$ so as to keep the same ordering in the first pair, but possibly reversing it in the second, so as to have $\sigma^*(2m-2k) = 2m-k$. As an example, if $m = 5$, $k = 3$, and $A(\sigma) = (7, 2, 4, 1, 6, 8, 3, 9, 5, 10)$, then $A(\sigma^*) = (4, 1, 2, 7, 6, 8, 3, 9, 5, 10)$. It is clear that $\sigma^* \in S(m, k+1)$ and $V(\sigma^*) = V(\sigma)$. Moreover, every $\tau \in S(m, k+1)$ can be represented in exactly $2(m-k)$ ways as $\tau = \sigma^*$, $\sigma^* \in S''$. Therefore,

$$\frac{1}{2^{m-k}(m-k)!} \sum_{\sigma \in S''} \rho^{V(\sigma)} = u(m, k+1). \tag{52}$$

Suppose now that $\sigma \in S'$. Then $2m-k = \sigma(2m-2r+1)$ for some $r$, $1 \leq r \leq k$. We now define a permutation $\sigma' \in S(m-1, k-1)$ as follows: In $A(\sigma)$, delete $a = \sigma(2m-2r+1)(= 2m-k)$ and $b = \sigma \cdot (2m-2r+2)$ and reduce the remaining entries that are between $a$ and $b$ by 1, and those that are larger than $\max(a, b) = a$ by 2. As an example, if $m = 5$, $k = 3$, and $A(\sigma) = (2, 1, 6, 3, 5, 8, 7, 9, 4, 10)$, then $A(\sigma') = (2, 1, 6, 3, 5, 7, 4, 8)$. The resulting vector clearly equals $A(\sigma')$ for some $\sigma' \in S(m-1, k-1)$, and each $\tau \in S(m-1, k-1)$ has exactly $k$ such representations. Further, $V(\sigma)$ equals the sum of (i) $V(\sigma')$, (ii) $a-b$ for the pair that was dropped, (iii) 2 for each of the $r-1$ pairs $(\sigma(2m-2j+1), \sigma(2m-2j+2))$ for $1 \leq j \leq r-1$, since in each such pair $\sigma(2m-2j+2) > a$, $\sigma(2m-2j+1) < b$, and finally (iv) 1 for each of the $k-r$ pairs $(\sigma(2m-2j+1), \sigma(2m-2j+2))$, $r+ \leq j \leq k$, since in each of them $\sigma(2m-j+1) < b$, $b < \sigma(2m-2j+2) < a$. Hence,

$$V(\sigma) = V(\sigma') + a - b + 2(r-1) + k - r. \tag{53}$$

But $a = 2m-k$ and $b = \sigma(2m-2r+2) = 2m-r+1$ from the definitions of $S(m, k)$, so

$$V(\sigma) = V(\sigma') + 2k - 1. \tag{54}$$

Hence, we have

$$\frac{1}{2^{m-k}(m-k)!}\sum_{\sigma\in S'}\rho^{V(\sigma)} = k\rho^{2k-1}u(m-1,\,k-1), \tag{55}$$

which proves the lemma.

We now can use the recurrence of the Lemma to derive the continued fraction expansion of the generating function. Let

$$f_k = f_k(z) = \sum_{m=0}^{\infty} u(m,\,k)z^m, \qquad k = 0,\,1,\,\cdots,$$

which for the moment we regard as formal power series in $z$. Then the Lemma gives us

$$f_1 = f_0 - 1, \tag{56}$$

and for $k \geqslant 2$,

$$\begin{aligned} f_k &= \sum_m u(m,\,k-1)z^m - (k-1)\rho^{2k-3}\sum_m u(m-1,\,k-2)z^m \\ &= f_{k-1} - (k-1)\rho^{2k-3}zf_{k-2}. \end{aligned} \tag{57}$$

Relations (56) and (57) show that for $k \geqslant 0$,

$$f_k = s_k f_0 - r_k, \tag{58}$$

where $s_0 = s_1 = 1$, $r_0 = 0$, $r_1 = 1$, and for $k \geqslant 2$ both $s_k$ and $r_k$ satisfy the recurrence

$$x_k = x_{k-1} - (k-1)\rho^{2k-3}zx_{k-2}.$$

Hence the quotients $r_k/s_k$ are the partial quotients of the continued fraction $R(z,\,\rho)$ on the right side of (25), and $s_k$ and $r_k$ converge as $k \to \infty$ to power series (in $z$) $s(z,\,\rho)$ and $r(z,\,\rho)$, respectively, for which

$$R(z,\,\rho) = \frac{r(z,\,\rho)}{s(z,\,\rho)}. \tag{59}$$

On the other hand, since $f_k$ starts with a term involving $z_k$, we conclude that $f_k$ converges to 0 in the ring of formal power series as $k \to \infty$. Therefore, from (58),

$$f_0 = \frac{r(z,\,\rho)}{s(z,\,\rho)} = R(z,\,\rho). \tag{60}$$

Since $f_0 = Q(z,\,\rho)$, we obtain the relation (25), at least in the ring of formal power series in $z$. However, the continued fraction (25) is clearly a meromorphic function of $z$ for $\rho$ fixed, $0 < \rho < 1$, and it is analytic at 0. Hence (25) holds as an equality among meromorphic functions, and we can obtain from this the exponential decrease of the $q_m(\rho)$ for $\rho < \rho_c$ and the exponential increase for $\rho > \rho_c$.

## AUTHORS

**Benjamin F. Logan, Jr.,** B.S. (Electrical Engineering), 1946, Texas Technological College; M.S., 1951, Massachusetts Institute of Technology; Eng.D.Sc. (Electrical Engineering), 1965, Columbia University; Bell Laboratories, 1956—. While at MIT, Mr. Logan was a research assistant in the Research Laboratory of Electronics, investigating characteristics of high-power electrical discharge lamps. Also at MIT he engaged in analog computer development at the Dynamic Analysis and Control Laboratory. From 1955 to 1956 he worked for Hycon-Eastern, Inc., where he was concerned with the design of airborne power supplies. He joined Bell Laboratories as a member of the Visual and Acoustics Research Department, where he was concerned with the processing of speech signals. Currently, he is a member of the Mathematical Research Department. Member, Sigma Xi, Tau Beta Pi.

**J. E. Mazo,** B.S. (Physics), 1958, Massachusetts Institute of Technology; M.S. (Physics), 1960, and Ph.D. (Physics), 1963, Syracuse University; Research Associate, Department of Physics, University of Indiana, 1963–1964; Bell Laboratories, 1964—. At the University of Indiana, Mr. Mazo worked on studies of scattering theory. At Bell Laboratories, he has been concerned with problems in data transmission and is now working in the Mathematics and Statistics Research Center. Member, IEEE.

**Andrew M. Odlyzko,** Ph.D., 1975, Massachusetts Institute of Technology; Bell Laboratories, 1975—. Mr. Odlyzko works in various areas of mathematics, including error-correcting codes, combinatorics, analysis, probability theory, and analysis of algorithms.

**Lawrence A. Shepp,** B.S. (Applied Mathematics), 1958, Brooklyn Polytechnic Institute; M.A., Ph.D. (Mathematics), 1961, Princeton; Bell Laboratories, 1962—. Mr. Shepp has worked mainly on problems involving probability theory. Since 1972 he has also worked in reconstruction of images from X-ray (and other) projections. He won the Paul Levy prize in probability, an IEEE (Nuclear Science) Distinguished Scientist award, and the Bell Laboratories Distinguished Scientist award. He is Professor of Radiology at Columbia University, biomathematician at The Neurological Institute, NYC, and a member of the Mathematics and Statistics Research Center at Bell Laboratories.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Series Solutions of Companding Problems

By B. F. LOGAN, Jr.*

(Manuscript received March 10, 1982)

A formal power series solution (i) $x(t) = \sum_1^\infty m^k x_k(t)$ is given for the companding problem (ii) $Bf\{x(t)\} = my(t)$, $B\{x(t)\} = x(t)$, where $B$ is the bandlimiting operator defined by $Bg = (Bg)(t) = \int_{-\infty}^{\infty} g(s)[\sin \lambda(t-s)]/[\pi(t-s)]ds$ and $f(t)$ has a Taylor series with $f(0) = 0, f'(0) \neq 0$. Expressions for the $x_k$ are given in terms of the coefficients of $f$, and operations on $y$, and in a different form in terms of the coefficients of the inverse function $\phi$, $\phi\{f(x)\} = x$. A series development is given for a bandlimited $z(t)$, $Bz = z$, such that the solution of (ii) is given by $x = B\phi(z)$. Also a series development is given for the "approximate identity", $x \doteq B\phi\{Bf(x)\}$, where $x = x(t)$, $Bx = x$, which is shown to be a good approximation to $x$ for fairly linear $f(x)$, not necessarily having a Taylor series expansion. As an example of one application of the results, a few terms are given for correction of the "inband" distortion arising in envelope detection of "full-carrier" single-sideband signals. The results should prove useful in correcting small distortions in other transmission systems. Finally, it is shown that the formal series solution (*i*) actually converges for sufficiently small $|m|$. This involves proving that the companding problem (*ii*) has a unique solution for arbitrary complex-valued $y(t)$ and complex $m$ of sufficiently small magnitude, the solution $x(t; m)$ being, for each $t$, an analytic function of the complex variable $m$ in a neighborhood of the origin. It is a curious fact, as shown by an interesting example, that the series (*i*) may converge for values of $m$ for which it is not a solution of (*ii*).

## I. INTRODUCTION

Suppose $x(t)$ is a bandlimited signal whose Fourier transform vanishes outside the interval $[-\lambda, \lambda]$. If such a signal is instantaneously

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

distorted by a nonlinear (companding) function $f(x)$, the distorted signal $f\{x(t)\}$ will, in general, have frequency components outside the interval $[-\lambda, \lambda]$. If the out-of-band components of the distorted signal are removed by ideal low-pass filtering, the result is a bandlimited signal $y(t)$ whose Fourier transform agrees with that of $f\{x(t)\}$ over $(-\lambda, \lambda)$. How, and under what conditions, may $x(t)$ be recovered from $y(t)$? When the signals are real-valued, this is known as the *companding problem* of Landau and Miranker (Refs. 1 and 2), hereafter referred to as the *r.v. companding problem.* Before stating their result, and our purpose, we introduce some notation.

The symbol $\mathscr{B}_2(\lambda)$ will denote the subspace of $L_2 = L_2(-\infty, \infty)$ whose elements are those (square-integrable) functions whose Fourier transforms vanish outside $[-\lambda, \lambda]$. Associated with this subspace is the *bandlimiting operator* $B_\lambda$, defined for $g$ in $L_2$ by

$$B_\lambda g = (B_\lambda g)(t) = \int_{-\infty}^{\infty} g(s) \frac{\sin \lambda(t-s)}{\pi(t-s)}\, ds.$$

The Fourier transforms of $g$ and $B_\lambda g$ agree over $(-\lambda, \lambda)$, the transform of the latter vanishing outside $[-\lambda, \lambda]$. In the language of Hilbert space, $B_\lambda g$ is the projection of $g$ on $\mathscr{B}_2(\lambda)$, being the best approximation to $g$ in the subspace $\mathscr{B}_2(\lambda)$. In case $g$ belongs to $\mathscr{B}_2(\lambda)$, we have

$$B_\lambda g = g.$$

It follows that

$$B_\lambda^n g = B_\lambda g, \qquad g \text{ in } L_2, \qquad n = 1, 2, \cdots.$$

The operator $B_\lambda$ may be applied also to functions belonging to $L_p$, $1 \leq p < \infty$; i.e., to functions $g$ satisfying

$$\|g\|_p = \left\{\int_{-\infty}^{\infty} |g(t)|^p dt\right\}^{1/p} < \infty \quad (1 \leq p < \infty).$$

Here the notation $\|g\|_p$ designates the norm of $g$ in $L_p$, or simply the $L_p$-norm of $g$. The space $L_\infty$ consists of those functions $g$ whose magnitude is bounded on the real line, their norm $\|g\|_\infty$ being the "essential supremum" of $|g(t)|$, which for functions we will be dealing with here, is simply the maximum value of $|g(t)|$. The operator $B_\lambda$ may not be applied to an arbitrary bounded function, since the associated integral may not converge. However, the integral may converge conditionally for a large class of functions; in particular, $B_\lambda g = g$, for any constant function $g$.

The operator $B_\lambda$ is a "contraction" operator on $L_2$; i.e.,

$$\|B_\lambda g\|_2 \leq \|g\|_2,$$

with equality attaining only for $g$ in $\mathscr{B}_2(\lambda)$. This follows from Parseval's theorem and the definition of $B_\lambda$.

Applying Schwarz's inequality to the integral equation denoted by $B_\lambda g = g$, we obtain the useful inequality

$$\|g\|_\infty \leq \sqrt{\lambda/\pi}\, \|g\|_2, \qquad g \quad \text{in} \quad \mathscr{B}_2(\lambda).$$

Also, it is easy to show from the integral equation that

$$\lim_{t\to\pm\infty} g(t) = 0, \qquad g \quad \text{in} \quad \mathscr{B}_2(\lambda).$$

We shall also make use of the *high-pass operator* $H_\lambda$, defined by

$$H_\lambda = I - B_\lambda,$$

where $I$ is the identity operator. $H_\lambda$ is an identity operator for functions $h$ of $L_2$ whose Fourier transforms vanish over $(-\lambda, \lambda)$, and it is also a contraction operator on $L_2$,

$$\|H_\lambda g\|_2 \leq \|g\|_2,$$

with equality attaining only for $H_\lambda g = g$, i.e., for $B_\lambda g = 0$. (In these operational equations, 0 is interpreted as the null function.)

It is clear from the operator definitions and the associated Fourier transform relations that any function $f$ in $L_2$ has the decomposition

$$f = g + h,$$

where

$$g = B_\lambda f, \qquad h = H_\lambda f.$$

Since $\lambda$ will be fixed throughout the paper, we will, except where emphasis is desired, simply write $B$, $H$, and $\mathscr{B}_2$ for $B_\lambda$, $H_\lambda$, $\mathscr{B}_2(\lambda)$, respectively.

Now, using our notation, we may state the important result of Landau and Miranker as follows:

*Theorem* (Landau and Miranker): *Let $f(x)$ be a real-valued function of the real variable $x$, satisfying*

(*i*) $f(0) = 0$

(*ii*) $0 < m_1 \leq f'(x) \leq m_2 < \infty, \qquad (-\infty < x < \infty).$

*Then to each real-valued $y$ in $\mathscr{B}_2$ there corresponds a unique $x$ in $\mathscr{B}_2$, also real-valued, satisfying*

(*iii*) $Bf(x) = y.$

*The solution $x$ of (iii) may be obtained as the limit of the sequence of approximants $\{x_n\}$ defined iteratively by*

(*iv*) $x_{n+1} = x_n - cB\{f(x_n) - y\},$

*provided only that $x_1$ is a real-valued function in $\mathscr{B}_2$ and the real constant $c$ is so chosen that*

(v) $\max_x |1 - cf'(x)| \leq r < 1.$

The beauty of this result is that, under the hypotheses on $f$, every (r.v.) $y$ in $\mathscr{B}_2$ has the representation (*iii*) where $x$ is a unique (r.v.) function in $\mathscr{B}_2$. In some r.v. companding problems of interest, $f(x)$ may not be defined outside some interval and/or the condition on $f'(x)$ may not be satisfied over the whole real axis, but rather over some interval including the origin. Then the conclusion will apply only to $y$ of sufficiently small norm. In such cases, the companding problem has two essentially different interpretations. The first is the *recovery problem*: $y$ is known to be of the form (*iii*); recover $x$. The second is the *design problem*: $y$ is a prescribed (desired) signal; find $x$, if possible, so that $y$ is given by (*iii*). In this case, one is faced with the problem of determining for what $y$ the problem has a solution.

The speed of covergence of the iterative solution of Landau and Miranker is a matter of practical concern. They show that

$$\| x_{n+1} - x_n \|_2 \leq r \| x_n - x_{n-1} \|_2.$$

Then the constant $c$ in (v) should be chosen to make $r$ as small as possible. Assuming that equality may attain on both sides in (*ii*), one should choose

$$c = \frac{2}{m_1 + m_2}, \qquad \text{giving} \quad r = \frac{m_2 - m_1}{m_2 + m_1}.$$

Thus rapid convergence is assured if $(m_2/m_1)$ is not much larger than 1. If this is not the case, a large number of iterations are, in general, required to obtain a close approximation to the solution of the problem. In a practical implementation of the iterative scheme of solution (Ref. 1), the ideal bandlimiting operator is replaced by an approximate operator, incurring a certain delay, in addition to (eventually) significant spectral distortions, with the result that the sequence $\{x_n\}$ will not converge to the solution $x$. Thus, in practice, the number of iterations to be performed is limited both by practical and theoretical considerations. The conclusion is that good approximate solutions to companding problems may be conveniently obtained in practice only in those cases where the companding function $f(x)$ is fairly linear over the range of $x(t)$.

We should remark at this point that there is only one known (nonlinear) r.v. companding problem (see Ref. 3) admitting of an explicit noniterative solution; viz.,

$$B\{\log(1 + x)\} = y, \qquad x > -1, \qquad x \text{ in } \mathscr{B}_2,$$

which has a solution if, and only if, the function†

† Here we are applying the $B$ operator to a function not in $L_2$, the proper interpretation being $w = 1 + B\{-1 + \exp(\cdot)\}$.

$$w(t) = B\{\exp\ 1/2[\,y(t) + \hat{y}(t)]\},$$

where $\hat{y}$ is the Hilbert transform of $y$, extends as a function zero-free in the upper half-plane, which will be the case if $\| y \|_2$ is sufficiently small. Then the solution is given by

$$x(t) = |\,w(t)\,|^2 - 1.$$

Motivated by the above considerations, pure curiosity, and the fact that in many cases of practical interest the companding function and/or its inverse can be well approximated by a polynomial of low degree over the range of interest, we are led to consider the case where the companding function has a Taylor series expansion, allowing the possibility of developing a corresponding series solution to the problem. To obtain the terms (1st order, 2nd order, etc.) in the series solution it is convenient to multiply $y$ by a scalar parameter $m$, and consider the problem

$$Bf(x) = my \tag{1}$$

to be solved for $x$ in $\mathscr{B}_2$, given $y$ in $\mathscr{B}_2$, for companding functions

$$f(x) = \sum_1^\infty b_k x^k, \qquad |\,x\,| < R_0 \tag{2}$$

$$b_1 \neq 0.$$

For sufficiently small $|\,x\,|$, $f$ will have an inverse $\phi$,

$$x = \phi\{f(x)\}$$

$$\phi(y) = \sum_1^\infty a_k y^k, \qquad |\,y\,| \leq R_0^*. \tag{3}$$

We assume that the solution $x = x(t; m)$ of (1) has a series expansion in the parameter $m$,

$$x(t; m) = \sum_1^\infty m^k x_k(t), \tag{4}$$

where the $x_k(t)$, aptly described as $k$th order corrections, (not to be confused with the Landau-Miranker approximants) depend only on $y(t)$ and $f$. Presumably, in cases of small distortion, a few terms of the series would give a satisfactory approximation to the solution.

Explicit expressions for the first five of the $x_k(t)$ are given in the sequel, first in formulas involving the coefficients of $\phi$, and next, the coefficients of $f$, together with certain operations on $y$. These formulas reveal how the Fourier transforms of the $x_k(t)$ may be calculated from the Fourier transform of $y(t)$, if this be given.

Next, we find a series development of $z$ in $\mathscr{B}_2$,

$$z(t) = z(t; m) = \sum_1^\infty m^k z_k(t) \tag{5}$$

such that the solution to (1) is given (presumably for sufficiently small $|m|$) by

$$x = B\phi(z). \tag{6}$$

We find that $z_1 = y$, $z_2 = 0$, and in case $a_2$ in (3), or $b_2$ in (2), vanishes, we have, in addition, $z_3 = z_4 = 0$. That is, under certain conditions, $z \doteq my$, implying that $B\phi\{Bf(x)\}$ is an "approximate identity" ($\doteq x$) for $x$ in $\mathscr{B}_2$, especially if $f$ is odd and fairly linear, or if $x(t)$ is a predominantly low-frequency function.

To further investigate the approximate identity, we introduce the parameter $m$ again, and obtain expressions for $u_k$ in

$$B\phi\{Bf(mx)\} = \sum_1^\infty m^k u_k, \qquad x \text{ in } \mathscr{B}_2. \tag{7}$$

To see how interchanging $f$ and $\phi$ affects the approximate identity, we compare $u_k$ with $v_k$ in

$$Bf\{B\phi(mx)\} = \sum_1^\infty m^k v_k, \qquad x \text{ in } \mathscr{B}_2. \tag{8}$$

As expected from the series development of $z$, we find $u_1 = v_1 = x$, and $u_2 = v_2 = 0$. Further comparisons [with the same $m$ in (7) and (8)] should be made for the case $f'(0) = \phi'(0) = 1$. For this case, we find $u_3 = v_3 = 2b_2^2 B(x \cdot Hx^2)$, which may be small if $b_2$ is small or if $Hx^2$ is small. In case $b_2 = 0$, we find $u_k = v_k = 0$ for $k = 2, 3, 4$, and $u_5 = v_5 = 3b_3^2 B(x^2 \cdot Hx^3)$.

These series developments of the approximate identity suggest that it would be useful in obtaining an approximate solution to the r.v. companding problem for fairly linear companding functions, not necessarily having a Taylor series expansion, but merely satisfying $f(0) = 0$ and

$$0 < m_1 \leqslant f'(x)\ m_2 < \infty, \qquad (-\infty < x < \infty). \tag{9}$$

Compelled by this suggestion, we digress in the Appendix to show for such $f$ that

$$\| x - B\phi\{Bf(x)\} \|_2 \leqslant \gamma \| x \|_2, \qquad x \text{ in } \mathscr{B}_2, \tag{10}$$

where

$$\gamma = \frac{\epsilon^2}{4(1+\epsilon)}, \qquad \epsilon = \frac{m_2}{m_1} - 1.$$

(Note that $\gamma = 1/8$ for $m_2/m_1 = 2$.) Thus in many companding

problems, $B\phi(y)$, involving only one filtering operation, would be an adequate approximation to the solution $x$. We go on to define an iterative procedure, involving both $f$ and its inverse $\phi$, obtaining approximants converging to $x$ for $\gamma < 1$, offering an alternative to the solution of Landau and Miranker in cases where $(m_2/m_1) < 3 + 2\sqrt{2}$. In any case, $B\phi(y)$ is suggested as a good choice for $x_1$ in their iterative solution. We note, in leaving this topic, that the inequality (10) is invariant to the interchange of $f$ and its inverse $\phi$.

Returning to the series solution, we apply the results to the problem of compatible single-sideband transmission (Ref. 4), obtaining a few terms for correction of the "in-band" distortion arising in envelope detection of "full-carrier" single-sideband signals.

Although the original intent of the work here was to obtain expressions for the first few terms of the series solution (4), supposedly adequate for correcting small distortions, the mathematical question naturally arises in the end as to whether the series actually converges for sufficiently small $|m|$ (or equivalently, for $|m| = 1$ and $\|y\|_2$ sufficiently small), or whether it is merely an asymptotic series. It is indeed a pertinent mathematical question, since the expressions for the $x_k(t)$ were obtained by purely formal manipulations of power series and application of the operators $B$ and $H$. The resulting expressions become progressively cumbersome and complicated, with no obvious general form, offering no possibility of establishing (from them) bounds on $|x_k(t)|$ which would ensure convergence of the series. The remainder of the paper is addressed to the problem of establishing the convergence of the series.

If we suppose in the r.v. companding problem that the series converges for real-valued $m$ of sufficiently small magnitude, then it would also converge for similar complex $m$, suggesting that the companding problem [for fixed $y(t)$] would have a solution for all complex $m$ of sufficiently small magnitude. This, in turn, suggests that the problem would have a solution for arbitrary *complex-valued* $y(t)$ in $\mathscr{B}_2$ and all complex $m$ of sufficiently small magnitude, depending on $f$ and the norm of $y$. That this is a fact has been established previously (Ref. 3) only for complex-valued $y(t)$ whose Fourier transforms vanish outside $[0, \lambda]$, (or $[-\lambda, 0]$), the Fourier transform of the solution $x(t)$ having the same property. In this case (with $m = 1$), the solution is given by $x = B\phi(y)$ for $y$ of sufficiently small norm; i.e., in case the Fourier transform of $x(t)$ vanishes outside $[0, \lambda]$, (or $[-\lambda, 0]$) the "approximate identity" is an exact identity,

$$x = B\phi\{Bf(x)\} = \phi\{f(x)\}$$

for $x$ of sufficiently small norm. This result can be explained, roughly, by the fact that nonlinear (analytic) distortion of such $x(t)$ does not

produce both "sum and difference" frequency components, but only "sum" components.

In order to prove that the series solution actually converges for sufficiently small $|m|$, we show that the companding problem (1), where $y(t)$ is an arbitrary *complex-valued* function in $\mathscr{B}_2$, has a solution $x(t; m)$ for all complex $m$ of sufficiently small magnitude, this solution being, for each fixed $t$, an anaytic function of the complex variable $m$, from which it follows that the solution has a Taylor series expansion in $m$; i.e.,

$$x(t;m) = \sum_1^\infty m^k x_k(t), \qquad |m| < m_0. \tag{11}$$

To obtain this result, we first have to establish for the complex-valued (c.v.) companding problem the analogue of A. Beurling's uniqueness theorem (see Ref. 1) for the r.v. companding problem.

We then examine in detail a specific problem illustrative of the theory and some of its nuances; viz., the problem (taking $\lambda = 2$ for convenience)

$$B\left\{\frac{x}{1-x}\right\} = m\,\frac{\sin 2t}{2t} \tag{12}$$

for which the solution is (at least for sufficiently small $|m|$)

$$x = x(t;m) = 2\beta\,\frac{\sin 2t}{2t} - \beta^2\left(\frac{\sin t}{t}\right)^2, \tag{13}$$

where

$$\beta = m/(2 + m).$$

The rather surprising revelation of this example is that, although the series expansion in $m$ of $x(t;m)$ converges, uniformly in $t$, for $|m| < 2$, it is not a solution of (12) for all such $m$. Furthermore, one might reasonably assume that (13) is a solution of (12) for all $m$ other than $-2$, but this is not true either. As an illuminating exercise, we determine precisely the set of $m$ for which (13) is a solution of (12).

## II. THE INVERSE SERIES METHOD

To obtain a series solution to (1), we first think of recovering from $y(t)$ the out-of-band components of $f\{x(t)\}$, so that we might apply the inverse function $\phi$ to the whole in order to recover $x(t) = x(t;m)$. We have

$$f\{x(t)\} = my(t) + h(t), \tag{14}$$

where $h(t) = h(t;m)$ is some unknown "high-pass" function satisfying

$$Bh(t) = 0 \tag{15}$$

and hence

$$x(t) = \phi\{my(t) + h(t)\}. \tag{16}$$

It is convenient at this point to introduce the high-pass operator defined by

$$H = I - B, \tag{17}$$

where $I$ is the identity operator. Thus applying $H$ to (16) we have

$$H\phi\{my(t) + h(t)\} = 0. \tag{18}$$

We would like to solve (18) for $h(t)$, which we think of as small in cases of interest.

Now we assume that

$$\phi(y) = \sum_1^\infty a_k y^k \quad \text{for sufficiently small} \quad |y|, \tag{19}$$

$$a_1 \neq 0$$

and that in (16)

$$h(t) = h(t;m) = \sum_2^\infty m^k h_k(t), \qquad Hh_k = h_k, \qquad k \geqslant 2, \tag{20}$$

$$x(t) = x(t;m) = \sum_1^\infty m^k x_k(t), \qquad Bx_k = x_k, \qquad k \geqslant 1, \tag{21}$$

where $h_k(t)$ and $x_k(t)$ do not depend on $m$.

We want to expand $\phi\{my(t) + h(t)\}$ as a power series in $m$. To do this it is convenient to write

$$my(t) + h(t) = \sum_1^\infty m^k h_k(t), \tag{22}$$

where we identify

$$y(t) = h_1(t) \quad \text{in} \quad \mathscr{B}_2. \tag{23}$$

Then we write

$$\phi\left\{\sum_1^\infty m^k h_k(t)\right\} = F(m;t) = \sum_1^\infty m^k F_k(t). \tag{24}$$

For convenience we suppress the variable $t$ and write simply $x_k$, $h_k$, $F_k$. In terms of the coefficients $a_k$ in

$$\phi(y) = \sum_{1}^{\infty} a_k y^k,$$

we find, equating coefficients of $m^k$ in (24),

$$F_1 = a_1 h_1 \tag{25.1}$$

$$F_2 = a_1 h_2 + a_2 h_1^2 \tag{25.2}$$

$$F_3 = a_1 h_3 + a_2(2h_1 h_2) + a_3 h_1^3 \tag{25.3}$$

$$F_4 = a_1 h_4 + a_2(2h_1 h_3 + h_2^2) + a_3(3h_1^2 h_2) + a_4 h_1^4 \tag{25.4}$$

$$F_5 = a_1 h_5 + a_2(2h_1 h_4 + 2h_2 h_3) + a_3(3h_1^2 h_3 + 3h_1 h_2^2) + a_4(4h_1^3 h_2) + a_5 h_1^5 \tag{25.5}$$

$$F_6 = a_1 h_6 + a_2(2h_1 h_5 + 2h_2 h_4 + h_3^2) + a_3(3h_1^2 h_4 + 6h_1 h_2 h_3 + h_2^3) + a_4(4h_1^3 h_3 + 6h_1^2 h_2^2) + a_5(5h_1^4 h_2) + a_6 h_1^6 \tag{25.6}$$

$$F_7 = a_1 h_7 + a_2(2h_1 h_6 + 2h_2 h_5 + 2h_3 h_4) + a_3(3h_1^2 h_5 + 6h_1 h_2 h_4 + 3h_1 h_3^2 + 3h_2^2 h_3) + a_4(4h_1^3 h_4 + 12h_1^2 h_2 h_3 + 4h_1 h_2^3) + a_5(5h_1^4 h_3 + 10h_1^3 h_2^2) + a_6(6h_1^5 h_2) + a_7 h_1^7 \tag{25.7}$$

$$F_8 = a_1 h_8 + a_2(2h_1 h_7 + 2h_2 h_6 + 2h_3 h_5 + h_4^2) + a_3(3h_1^2 h_6 + 6h_1 h_2 h_5 + 6h_1 h_3 h_4 + 3h_2^2 h_4 + 3h_2 h_3^2) + a_4(4h_1^3 h_5 + 12h_1^2 h_2 h_4 + 6h_1^2 h_3^2 + 12h_1 h_2^2 h_3 + h_2^4) + a_5(5h_1^4 h_4 + 20h_1^3 h_2 h_3 + 10h_1^2 h_2^3) + a_6(6h_1^5 h_3 + 15h_1^4 h_2^2) + a_7(7h_1^6 h_2) + a_8 h_1^8 \tag{25.8}$$

$$F_9 = a_1 h_9 + a_2(2h_1 h_8 + 2h_2 h_7 + 2h_3 h_6 + 2h_4 h_5) + a_3(3h_1^2 h_7 + 6h_1 h_2 h_6 + 6h_1 h_3 h_5 + 3h_1 h_4^2 + 3h_2^2 h_5 + 6h_2 h_3 h_4 + h_3^3) + a_4(4h_1^3 h_6 + 12h_1^2 h_2 h_5 + 12h_1^2 h_3 h_4 + 12h_1 h_2^2 h_4 + 12h_1 h_2 h_3^2 + 4h_2^3 h_3) + a_5(5h_1^4 h_5 + 20h_1^3 h_2 h_4 + 10h_1^3 h_3^2 + 30h_1^2 h_2^2 h_3 + 5h_1 h_2^4) + a_6(6h_1^5 h_4 + 30h_1^4 h_2 h_3 + 20h_1^3 h_2^3) + a_7(7h_1^6 h_3 + 21h_1^5 h_2^2) + a_8(8h_1^7 h_2) + a_9 h_1^9 \tag{25.9}$$

$$F_{10} = a_1 h_{10} + a_2(2h_1 h_9 + 2h_2 h_8 + 2h_3 h_7 + 2h_4 h_6 + h_5^2) + a_3(3h_1^2 h_8 + 6h_1 h_2 h_7 + 6h_1 h_3 h_6 + 6h_1 h_4 h_5 + 3h_2^2 h_6 + 6h_2 h_3 h_5 + 3h_2 h_4^2 + 3h_3^2 h_4) + a_4(4h_1^3 h_7 + 12h_1^2 h_2 h_6$$

$$+ 12h_1^2h_3h_5 + 6h_1^2h_4^2 + 12h_1h_2^2h_5 + 24h_1h_2h_3h_4 + 4h_1h_3^3$$
$$+ 4h_2^3h_4 + 6h_2^2h_3^2) + a_5(5h_1^4h_6 + 20h_1^3h_2h_5 + 20h_1^3h_3h_4$$
$$+ 30h_1^2h_2^2h_4 + 30h_1^2h_2h_3^2 + 20h_1h_2^3h_3 + h_2^5) + a_6(6h_1^5h_5$$
$$+ 30h_1^4h_2h_4 + 15h_1^4h_3^2 + 60h_1^3h_2^2h_3 + 15h_1^2h_2^4)$$
$$+ a_7(7h_1^6h_4 + 42h_1^5h_2h_3 + 35h_1^4h_2^3) + a_8(8h_1^7h_3$$
$$+ 28h_1^6h_2^2) + a_9(9h_1^8h_2) + a_{10}h_1^{10}. \tag{25.10}$$

In general the coefficient of $a_m$ in the expansion of $F_n$ consists of sums of products of the $h_k$ corresponding to partitions of $n$ into $m$ parts. The coefficient of the product is $m!$ divided by the product of the factorials of the exponents of the $h_k$ (the multinomial theorem). For example, in $F_{10}$ the coefficient of $a_5$ is found by writing down the partitions of 10 into 5 parts and proceeding thus (see Table 24.2, Ref. 5):

| | | | |
|---|---|---|---|
| $1^4, 6$ | $\rightarrow$ $h_1^4h_6$ | coef. $= 5!/4!$ | $= 5$ |
| $1^3, 2, 5$ | $h_1^3h_2h_5$ | $5!/3!$ | $= 20$ |
| $1^3, 3, 4$ | $h_1^3h_3h_4$ | $5!/3!$ | $= 20$ |
| $1^2, 2^2, 4$ | $h_1^2h_2^2h_4$ | $5!/2!2!$ | $= 30$ |
| $1^2, 2, 3^2$ | $h_1^2h_2h_3^2$ | $5!/2!2!$ | $= 30$ |
| $1, 2^3, 3$ | $h_1h_2^3h_3$ | $5!/3!$ | $= 20$ |
| $2^5$ | $h_2^5$ | $5!/5!$ | $= 1.$ |

Now we may obtain a formal series solution (21) by successively solving for the $h_k$ by requiring

$$HF_k(t) = 0, \qquad k = 1, 2, \cdots \tag{26}$$

and setting

$$x_k(t) = BF_k(t) = F_k(t), \qquad k = 1, 2, \cdots. \tag{27}$$

Recall that $h_1 = y$, the given bandlimited (low-pass) function, and all the other $h_k$ are high-pass functions.† We have $Hh_k = h_k$, $k \geq 2$ and

$$HF_1 = a_1Hh_1 = 0 \tag{28.1}$$
$$HF_2 = a_1h_2 + a_2Hh_1^2 = 0$$
$$h_2 = -\frac{a_2}{a_1}Hh_1^2 \tag{28.2}$$

† Actually, for $k \geq 2$, $h_k$ is a bandpass function whose Fourier transform vanishes over $(-\lambda, \lambda)$ and outside $[-k\lambda, k\lambda]$. This can be seen from (28.1)–(28.k).

$$HF_3 = a_1h_3 + 2a_2H(h_1h_2) + a_3Hh_1^3 = 0$$

$$h_3 = \frac{2a_2^2}{a_1^2} H(h_1 \cdot Hh_1^2) - \frac{a_3}{a_1} Hh_1^3 \tag{28.3}$$

$$HF_4 = a_1h_4 + a_2(2H(h_1h_3) + Hh_2^2) + 3a_3H(h_1^2h_2) + a_4Hh_1^4 = 0$$

$$h_4 = -\frac{4a_2^3}{a_1^3} H[h_1 \cdot H(h_1 \cdot Hh_1^2)] + \frac{2a_2a_3}{a_1^2} H(h_1 \cdot Hh_1^3)$$
$$- \left(\frac{a_2}{a_1}\right)^3 H[(Hh_1^2)^2] + \frac{3a_2a_3}{a_1^2} H(h_1^2 \cdot Hh_1^2) - \frac{a_4}{a_1} Hh_1^4 \tag{28.4}$$

$$HF_5 = a_1h_5 + a_2[2H(h_1h_4) + 2H(h_2h_3)] + a_3[3H(h_1^2h_3)$$
$$+ 3H(h_1h_2^2)] + 4a_4H(h_1^3h_2) + a_5Hh_1^5 = 0$$

$$h_5 = \frac{8a_2^4}{a_1^4} H\{h_1 \cdot H[h_1 \cdot H(h_1 \cdot Hh_1^2)]\} - \frac{4a_2^2a_3}{a_1^3} H[h_1 \cdot H(h_1 \cdot Hh_1^3)]$$
$$+ \frac{2a_2^4}{a_1^4} H[h_1 \cdot (Hh_1^2)^2] - \frac{6a_2^2a_3}{a_1^3} H[h_1 \cdot H(h_1^2 \cdot Hh_1^2)]$$
$$+ \frac{2a_2a_4}{a_1^2} H(h_1 \cdot Hh_1^4) + \frac{4a_2^4}{a_1^4} H[(Hh_1^2) \cdot H(h_1 \cdot Hh_1^2)]$$
$$- \frac{2a_2^2a_3}{a_1^3} H[(Hh_1^2) \cdot Hh_1^3] - \frac{6a_2^2a_3}{a_1^3} H[h_1^2 \cdot H(h_1 \cdot Hh_1^2)]$$
$$+ \frac{3a_3^2}{a_1^2} H(h_1^2 \cdot Hh_1^3) - \frac{3a_2^2a_3}{a_1^3} H[h_1 \cdot (Hh_1^2)^2]$$
$$+ \frac{4a_2a_4}{a_1^2} H(h_1^3 \cdot Hh_1^2) - \frac{a_5}{a_1} Hh_1^5. \tag{28.5}$$

Now replacing $h_1$ by $y$ we have from (25), (27), and (28)

$$x_1 = BF_1 = a_1y \tag{29.1}$$

$$x_2 = BF_2 = a_2By^2 \tag{29.2}$$

$$x_3 = BF_3 = 2a_2B(yh_2) + a_3By^3$$
$$= -\frac{2a_2^2}{a_1} B(y \cdot Hy^2) + a_3By^3 \tag{29.3}$$

$$x_4 = BF_4 = a_2(2B(yh_3) + Bh_2^2) + 3a_3B(y^2h_2) + a_4By^4$$
$$= \frac{4a_2^3}{a_1^2} B[y \cdot H(y \cdot Hy^2)] - \frac{2a_2a_3}{a_1} B(y \cdot Hy^3)$$

$$+\frac{a_2^3}{a_1^2}B(Hy^2)^2-\frac{3a_2a_3}{a_1}B(y^2\cdot Hy^2)+a_4By^4 \tag{29.4}$$

$$\begin{aligned}x_5=BF_5&=a_2[2B(yh_4)+2B(h_2h_3)]+3a[B(y^2h_3)+B(yh_2^2)]\\&\quad+4a_4B(y^3h_2)+a_5By^5\\&=-\frac{8a_2^4}{a_1^3}B\{y\cdot H[y\cdot H(y\cdot Hy^2)]\}+\frac{4a_2^2a_3}{a_1^2}B[y\cdot H(y\cdot Hy^3)]\\&\quad-\frac{2a_2^4}{a_1^3}B[y\cdot H(Hy^2)^2]+\frac{6a_2^2a_3}{a_1^2}B[y\cdot H(y^2\cdot Hy^2)]\\&\quad-\frac{2a_2a_4}{a_1}B(y\cdot Hy^4)-\frac{4a_2^4}{a_1^3}B[(Hy^2)\cdot H(y\cdot Hy^2)]\\&\quad+\frac{2a_2^2a_3}{a_1^2}B[(Hy^2)\cdot(Hy^3)]+\frac{6a_2^2a_3}{a_1^2}B[y^2\cdot H(y\cdot Hy^2)]\\&\quad-\frac{3a_3^2}{a_1}B(y^2\cdot Hy^3)+\frac{3a_2^2a_3}{a_1^2}B[y\cdot(Hy^2)^2]\\&\quad-\frac{4a_2a_4}{a_1}B(y^3\cdot Hy^2)+a_5By^5.\end{aligned} \tag{29.5}$$

If in (29) we replace the $H$ operator by $I-B$ and collect terms we obtain

$$x_1=a_1y \tag{30.1}$$

$$x_2=a_2By^2 \tag{30.2}$$

$$x_3=\frac{2a_2^2}{a_1}B(y\cdot By^2)+\left(a_3-\frac{2a_2^2}{a_1}\right)By^3 \tag{30.3}$$

$$\begin{aligned}x_4&=\frac{4a_2^3}{a_1^2}B[y\cdot B(y\cdot By^2)]-\left(\frac{4a_2^3}{a_1^2}-\frac{2a_2a_3}{a_1}\right)B(y\cdot By^3)\\&\quad+\frac{a_2^3}{a_1^2}B(By^2)^2-\left(\frac{6a_2^3}{a_1^2}-\frac{3a_2a_3}{a_1}\right)B(y^2\cdot By^2)\\&\quad+\left(a_4-\frac{5a_2a_3}{a_1}+\frac{5a_2^3}{a_1^2}\right)By^4\end{aligned} \tag{30.4}$$

$$\begin{aligned}x_5&=\frac{8a_2^4}{a_1^3}B\{y\cdot B[y\cdot B(y\cdot By^2)]\}+\left(\frac{4a_2^2a_3}{a_1^2}-\frac{8a_2^4}{a_1^3}\right)B[y\cdot B(y\cdot By^3)]\\&\quad+\left(\frac{6a_2^2a_3}{a_1^2}-\frac{12a_2^4}{a_1^3}\right)B[y\cdot B(y^2\cdot By^2)]+\frac{2a_2^4}{a_1^3}\end{aligned}$$

$$\cdot B[y\cdot B(By^2)^2] + \left(\frac{3a_2^2a_3}{a_1^2} - \frac{6a_2^4}{a_1^3}\right) B[y\cdot(By^2)^2]$$

$$+ \left(\frac{2a_2a_4}{a_1} - \frac{10a_2^2a_3}{a_1^2} + \frac{10a_2^4}{a_1^3}\right) B(y\cdot By^4) + \frac{4a_2^4}{a_1^3} B[(By^2)$$

$$\cdot B(y\cdot By^2)] + \left(\frac{6a_2^2a_3}{a_1^2} - \frac{12a_2^4}{a_1^3}\right) B[y^2\cdot B(y\cdot By^2)]$$

$$+ \left(\frac{2a_2^2a_3}{a_1^2} - \frac{4a_2^4}{a_1^3}\right) B[(By^2)\cdot(By^3)]$$

$$+ \left(\frac{3a_3^2}{a_1} - \frac{12a_2^2a_3}{a_1^2} + \frac{12a_2^4}{a_1^3}\right) B(y^2\cdot By^3)$$

$$+ \left(\frac{4a_2a_4}{a_1} - \frac{20a_2^2a_3}{a_1^2} + \frac{20a_2^4}{a_1^3}\right) B(y^3\cdot By^2)$$

$$+ \left(a_5 - \frac{6a_2a_4}{a_1} + \frac{21a_2^2a_3}{a_1^2} - \frac{14a_2^4}{a_1^3} - \frac{3a_3^2}{a_1}\right) By^5. \tag{30.5}$$

Note that if $y$ belongs to $\mathscr{B}_2(\lambda/n)$, then $By^k = y^k$ for $k = 1, 2, \cdots, n$. In this case we will have $x_n = a_n y^n$. So the sum of all coefficients in the expressions for $x_n$ must be $a_n$. If $\phi$ is an odd function these formulas simplify considerably. It is rather curious that if $a_2 = 0$, $a_3 \neq 0$, the coefficient of $B(y^3\cdot By^2)$ vanishes, whereas the coefficient of $B(y^2\cdot By^3)$ does not. The coefficients in (30) are more simply expressed in terms of the coefficients in the power series for $f$ as we see below.

## III. FORWARD SERIES METHOD

We can also solve (1) in the "forward" direction by writing

$$Bf(mx_1 + m^2x_2 + m^3x_3 + \cdots) = my, \tag{31}$$

where

$$f(x) = \sum_1^\infty b_k x^k. \tag{32}$$

Then applying the expansion (24) and (25) to $f(\sum m^k x_k)$ we have, equating coefficients of $m^k$,

$$Bb_1x_1 = y \tag{33.1}$$

$$Bb_1x_2 + Bb_2x_1^2 = 0 \tag{33.2}$$

$$Bb_1x_3 + Bb_2(2x_1x_2) + Bb_3x_1^3 = 0 \tag{33.3}$$

$$Bb_1x_4 + Bb_2(2x_1x_3 + x_2^2) + Bb_3(3x_1^2x_2) + Bb_4x_1^4 = 0 \tag{33.4}$$

$$Bb_1x_5 = Bb_2(2x_1x_4 + 2x_2x_3) + Bb_3(3x_1^2x_3 + 3x_1x_2^2)$$
$$+ Bb_4(4x_1^3x_2) + Bb_5x_1^5 = 0. \tag{33.5}$$

Then solving (33) successively for $x_k$, $(Bx_k = x_k)$, we have

$$x_1 = y/b_1 \tag{34.1}$$

$$x_2 = -\frac{b_2}{b_1^3}By^2 \tag{34.2}$$

$$x_3 = \frac{2b_2^2}{b_1^5}B(y\cdot By^2) - \frac{b_3}{b_1^4}By^3 \tag{34.3}$$

$$x_4 = -\frac{4b_2^3}{b_1^7}B[y\cdot B(y\cdot By^2)] + \frac{2b_2b_3}{b_1^6}B(y\cdot By^3)$$
$$-\frac{b_2^3}{b_1^7}B(By^2)^2 + \frac{3b_2b_3}{b_1^6}B(y^2\cdot By^2) - \frac{b_4}{b_1^5}By^4 \tag{34.4}$$

$$x_5 = \frac{8b_2^4}{b_1^9}B\{y\cdot B[y\cdot B(y\cdot By^2)]\} + \frac{4b_2^4}{b_1^9}B[(By^2)\cdot B(y\cdot By^2)]$$
$$+\frac{2b_2^4}{b_1^9}B[y\cdot B(By^2)^2] - \frac{4b_2^2b_3}{b_1^8}B[y\cdot B(y\cdot By^3)]$$
$$-\frac{6b_2^2b_3}{b_1^8}B[y\cdot B(y^2\cdot By^2)] - \frac{2b_2^2b_3}{b_1^8}B[(By^2)\cdot(By^3)]$$
$$-\frac{6b_2^2b_3}{b_1^8}B[y^2\cdot B(y\cdot By^2)] - \frac{3b_2^2b_3}{b_1^8}B[y\cdot(By^2)^2]$$
$$+\frac{2b_2b_4}{b_1^7}B(y\cdot By^4) + \frac{4b_2b_4}{b_1^7}B(y^3\cdot By^2)$$
$$+\frac{3b_3^2}{b_1^7}B(y^2\cdot By^3) - \frac{b_5}{b_1^6}By^5. \tag{34.5}$$

These correspond to the solutions for the $h_k$ in (28) with $H$ and $B$, and $a$'s and $b$'s interchanged, except here $x_1 = y/b_1$ as compared by $h_1 = y$ in (28). They agree with the formulas in (30) according to the identities in reversion of series (see 3.6.25, Ref. 5)

$$a_1b_1 = 1 \tag{35.1}$$

$$a_1^3b_2 = -a_2 \tag{35.2}$$

$$a_1^5b_3 = 2a_2^2 - a_1a_3 \tag{35.3}$$

$$a_1^7b_4 = 5a_1a_2a_3 - a_1^2a_4 - 5a_2^3 \tag{35.4}$$

$$a_1^9 b_5 = 6a_1^2 a_2 a_4 + 3a_1^2 a_3^2 + 14a_4^2 - a_1^3 a_5 - 21a_1 a_2^2 a_3 \tag{35.5}$$

$$a_1^{11} b_6 = 7a_1^3 a_2 a_5 + 7a_1^3 a_3 a_4 + 84a_1 a_1^3 a_3 - a_1^4 a_6 - 28a_1^2 a_2 a_3^2 - 42a_2^5 - 28a_1^2 a_2^2 a_4 \tag{35.6}$$

$$a_1^{13} b_7 = 8a_1^4 a_2 a_6 + 8a_1^4 a_3 a_5 + 120a_1^2 a_2^3 a_4 + 180a_1^2 a_2^2 a_3^2 + 132a_2^6 - a_1^5 a_7 - 36a_1^3 a_2^2 a_5 - 72a_1^3 a_2 a_3 a_4 - 12a_1^3 a_2 a_3 a_4 - 12a_1^3 a_3^3 - 330a_1 a_1^4 a_3. \tag{35.7}$$

Of course the $a$'s and $b$'s may be interchanged in (35). Actually the expressions for $x_k$ in (34) are analogous in a way to the coefficients $a_k$ expressed in terms of the $b_k$ as given by (35) (with the $a$'s and $b$'s interchanged). That is, if it were not for the $B$ operator in eq. (33.k), we would have simply $x_k = a_k y^k$ according to the determining equations for the inverse coefficients. Because of the $B$ operator, successive solutions for the $x_k$ generate powers of $y$ interposed with $B$ operators and powers of $\{B(\cdot)\}$ in all combinations so that, for example, in $x_5$ we have a number of terms with coefficients $b_2^2 b_3 / b_1^8$ that combine only when $B$ is replaced by $I$ to give $-21 b_2^2 b_3 y^5 / b_1^8$, corresponding to the last term in (35.5). Note if all the $b_k = 1$, the sum of the integer coefficients in $x_n$ is $(-1)^{n+1}$ as this is the case $y = f(x) = x/(1 - x)$, $x = \phi(y) = y/(1 + y)$. Note also that in the expression (34.5) for $x_5$, for example, there are five groups of functions with common "$b$" coefficients. These combine with certain weights, depending only on the coefficients of $f(x)$, to give $x_5$. Similarly, in the expression (29.5) obtained from the inverse function, there are again five groups of functions with common "$a$" coefficients that combine with certain weights, still depending only on the coefficients of $f(x)$, to give $x_5$. The interesting and rather puzzling fact is that the groups are not identical but overlap.

## IV. A SOLUTION OF THE FORM $x(t) = B\phi\{x(t)\}$, $z$ in $\mathscr{B}_2$

For the solution to (1) we have $x = \sum_1^\infty m^k x_k$, where according to (25) and (29), we have

$$x_1 = a_1 y$$

$$x_2 = a_1 h_2 + a_2 y^2 = a_2 B y^2$$

$$x_3 = a_1 h_3 + 2a_2 y h_2 + a_3 y^3 = 2a_2 B(y h_2) + a_3 B y^3.$$

Since

$$\phi(my) = a_1 my + a_2 m^2 y^2 + a_3 m^3 y^3 + \cdots,$$

we see that $x = B\phi(my) + \mathscr{O}(m^3)$, $m \to 0$. Then setting $m = 1$ (with $y$ sufficiently small) we can conclude that

$$x = B\phi(y) + \mathscr{O}(y^3), \qquad y \to 0. \tag{36}$$

At least as $y \to 0$, $B\phi(y)$ is a better approximation than $\phi(y) = x + \mathscr{O}(y^2)$. Also $B\phi(y)$ could be a good approximation to $x$ without $y$ being small, as would be the case if $y$ were a predominately low-frequency function (compared with its top frequency). This suggests that given $y$ in (1) we determine a bandlimited function $z$, perhaps close to $y$, such that the solution to (1) is given by

$$x = B\phi(z). \tag{37}$$

To determine a series solution for $z$ we set

$$z = \sum_1^\infty z_k m^k, \qquad Bz_k = z_k \tag{38}$$

and expand $\phi(z)$ in a power series in $m$:

$$\phi(mz_1 + m^2 z_2 + m^3 z_3 + \cdots) = \sum_1^\infty m^k F_k. \tag{39}$$

The $F_k$ are given by (25) with $z_k$ replacing $h_k$. The difference now is that the $F_k$ are bandlimited to $[-k\lambda, k\lambda]$, i.e., $\phi(z)$ is not bandlimited (in general) with $z$ in $\mathscr{B}_2$. However, we must have

$$BF_k = x_k, \tag{40}$$

where in terms of operations on $y$ the $x_k$ are given conveniently by (29). We have

$$Ba_1 z_1 = x_1 = a_1 y \tag{41.1}$$

$$B(a_1 z_2 + a_2 z_1^2) = x_2 = a_2 B y^2 \tag{41.2}$$

$$B(a_1 z_3 + 2a_2 z_1 z_2 + a_3 z_1^3) = x_3 = -\frac{2a_2^2}{a_1} B(y \cdot Hy^2) + a_3 B y^3 \tag{41.3}$$

$$\begin{aligned} B[a_1 z_4 + a_2(2z_1 z_3 + z_2^2) + a_3(3z_1^2 z_2) + a_4 z_1^4] &= x_4 \\ &= \frac{4a_2^3}{a_1^2} B[y \cdot H(y \cdot Hy^2)] - \frac{2a_2 a_3}{a_1} B(y \cdot Hy^3) \\ &\quad + \frac{a_2^3}{a_1^2} B(Hy^2)^2 - \frac{3a_2 a_3}{a_1} B(y^2 \cdot Hy^2) + a_4 B y^4 \end{aligned} \tag{41.4}$$

$$\begin{aligned} &B[a_1 z_5 + a_2(2z_1 z_4 + 2z_2 z_3) + a_3(3z_1^2 z_3 + 3z_1 z_2^2) \\ &\quad + a_4(4z_1^3 z_2) + a_5 z_1^5] = x_5 = -\frac{8a_2^4}{a_1^3} B\{y \cdot H[y \cdot H(y \cdot Hy^2)]\} \\ &\quad + \frac{4a_2^2 a_3}{a_1^2} B[y \cdot H(y \cdot Hy^3)] - \frac{2a_2^4}{a_1^3} B[y \cdot H(Hy^2)^2] \end{aligned}$$

$$+\frac{6a_2^2a_3}{a_1^2}B[y\cdot H(y^2\cdot Hy^2)]-\frac{2a_2a_4}{a_1}B(y\cdot Hy^4)$$

$$-\frac{4a_2^4}{a_1^3}B[(Hy^2)\cdot H(y\cdot Hy^2)]+\frac{2a_2^2a_3}{a_1^2}B[(Hy^2)\cdot(Hy^3)]$$

$$+\frac{6a_2^2a_3}{a_1^2}B[y^2\cdot H(y\cdot Hy^2)]-\frac{3a_3^2}{a_1}B(y^2\cdot Hy^3)$$

$$+\frac{3a_2^2a_3}{a_1^2}B[y\cdot(Hy^2)^2]-\frac{4a_2a_4}{a_1}B(y^3\cdot Hy^2)+a_5By^5. \tag{41.5}$$

Solving these equations successively for $z_k$ we find

$$z_1=y \tag{42.1}$$

$$z_2=0 \tag{42.2}$$

$$z_3=-\frac{2a_2^2}{a_1^2}B(y\cdot Hy^2) \tag{42.3}$$

$$z_4=\frac{4a_2^3}{a_1^3}B[y\cdot H(y\cdot Hy^2)]-\frac{2a_2a_3}{a_1^2}B(y\cdot Hy^3)+\frac{a_2^3}{a_1^3}B(Hy^2)^2$$
$$-\frac{3a_2a_3}{a_1^2}B(y^2\cdot Hy^2)+\frac{4a_2^3}{a_1^3}B[y\cdot B(y\cdot Hy^2)]. \tag{42.4}$$

The first and last terms in (42.4) combine ($H+B=I$) to give $\frac{4a_2^3}{a_1^3}B(y^2\cdot Hy^2)$, which then combines with the fourth term to give

$$z_4=\frac{a_2}{a_1^3}(4a_2^2-3a_1a_3)B(y^2\cdot Hy^2)-\frac{2a_2a_3}{a_1^2}B(y\cdot Hy^3)$$
$$+\frac{a_2^3}{a_1^3}B(Hy^2)^2 \tag{42.4a}$$

$$z_5=-\frac{8a_2^4}{a_1^4}B\{y\cdot H[y\cdot H(y\cdot Hy^2)]\}$$
$$-\frac{8a_2^4}{a_1^4}B[y\cdot B(y^2\cdot Hy^2)]+\frac{a_3}{a_1^3}(4a_2^2-3a_1a_3)B(y^2\cdot Hy^3)$$
$$+\frac{a_2^2}{a_1^4}(3a_1a_3-2a_2^2)B[y\cdot(Hy^2)^2]$$
$$+\frac{4a_2}{a_1^3}(3a_2a_3-a_1a_4)B(y^3\cdot Hy^2)-\frac{2a_2a_4}{a_1^2}B(y\cdot Hy^4)$$

$$-\frac{4a_2^4}{a_1^3}B[(Hy^2)\cdot H(y\cdot Hy^2)]+\frac{2a_2^2a_3}{a_1^3}B[(Hy^2)\cdot(Hy^3)]. \tag{42.5}$$

The expressions for the third- and fourth-order terms, $z_3$ and $z_4$, are quite simple, owing to the fact that $z_1 = y$, $z_2 = 0$. Note in eq. (41.n) the "diagonal" term $a_nBz_1^n$ on the left is cancelled by the term $a_nBy^n$ appearing in $x_n$ on the right. Also in case $a_2 = 0$ we have

$$z_1 = y \tag{43.1}$$

$$z_2 = 0 \tag{43.2}$$

$$z_3 = 0 \tag{43.3}$$

$$z_4 = 0 \tag{43.4}$$

$$z_5 = -\frac{3a_3^2}{a_1^2}B(y^2\cdot Hy^3). \tag{43.5}$$

## V. THE APPROXIMATE IDENTITY

The series development in the previous section suggests that as a practical expedient one might take

$$x \doteq B\phi(y) = B\phi\{Bf(x)\} \doteq \phi\{f(x)\} = x.$$

That is, what appears to be the naive thing to do may in fact be quite good, especially for odd functions $\phi$ (or $f$) that are not severely nonlinear. The interposition of the bandlimiting operator between a nonlinear function and its inverse and then subsequent bandlimiting is an interesting "approximate identity" that we examine further in the Appendix. One might ask how the interchange in the order of a particular function and its inverse in the transformation affects the approximate identity. The series expansion of the approximate identity may shed some light on the general problem. To keep track of the various orders, it is convenient to introduce the parameter $m$ as before. We have

$$Bf(mx) = \sum_1^\infty m^k b_k Bx^k, \tag{44}$$

$$\phi\{Bf(mx)\} = \sum_1^\infty m^k F_k, \tag{45}$$

where the $F_k$ are given by (25) with $b_kBx^k$ replacing $h_k$, and $Bx = x$. Thus,

$$F_1 = a_1b_1x = x \tag{46.1}$$

$$F_2 = a_1b_2Bx^2 + a_2b_1^2x^2 \tag{46.2}$$

$$F_3 = a_1b_3Bx^3 + 2a_2b_1b_2x\cdot Bx^2 + a_3b_1^3x^3 \tag{46.3}$$

$$\begin{aligned} F_4 = {} & a_1b_4Bx^4 + 2a_2b_1b_3x\cdot Bx^3 + a_2b_2^2(Bx^2)^2 \\ & + 3a_3b_1^2b_2x^2\cdot Bx^2 + a_4b_1^4x^4 \end{aligned} \tag{46.4}$$

$$\begin{aligned} F_5 = {} & a_1b_5Bx^5 + 2a_2b_1b_4x\cdot Bx^4 + 2a_2b_2b_3(Bx^2)\cdot(Bx^3) \\ & + 3a_3b_1^2b_3x^2\cdot Bx^3 + 3a_3b_1b_2^2x\cdot(Bx^2)^2 \\ & + 4a_4b_1^3b_2x^3\cdot Bx^2 + a_5b_1^5x^5. \end{aligned} \tag{46.5}$$

Now we set

$$B\phi\{Bf(mx)\} = \sum_1^\infty m^k u_k, \tag{47}$$

where

$$u_k = BF_k. \tag{48}$$

Now note in (46) that if all the terms in $BF_k$ involving $x$ were of the form $Bx^k$ then $BF_k$ would vanish identically for $k \geq 2$ because $\phi\{f(x)\} = x$. So we will introduce the high-pass operator $H = I - B$ to collect the terms $Bx^k$ that cancel. For example, to collect terms $Bx^3$ in $BF_3$ we write

$$B(x\cdot Bx^2) = B(x^3 - x\cdot Hx^2) = Bx^3 - B(x\cdot Hx^2).$$

Thus,

$$u_1 = BF_1 = x \tag{49.1}$$

$$u_2 = BF_2 = a_1b_2Bx^2 + a_2b_1^2Bx^2 = 0 \tag{49.2}$$

$$\begin{aligned} u_3 = BF_3 &= a_1b_3Bx^3 + 2a_2b_1b_2B(x\cdot Bx^2) + a_3b_1^3Bx^3 \\ &= a_1b_3Bx^3 + 2a_2b_1b_2B(x^3 - x\cdot Hx^2) + a_3b_1^3Bx^3 \\ &= -2a_2b_1b_2B(x\cdot Hx^2) \end{aligned} \tag{49.3}$$

$$\begin{aligned} u_4 = BF_4 &= a_1b_4Bx^4 + 2a_2b_1b_3B(x\cdot Bx^3) + a_2b_2^2B(Bx^2)^2 \\ &\quad + 3a_3b_1^2b_2B(x^2\cdot Bx^2) + a_4b_1^4Bx^4 \\ &= a_1b_4Bx^4 + 2a_2b_1b_3B(x^4 - x\cdot Hx^3) + a_2b_2^2B(x^2 - Hx^2)^2 \\ &\quad + 3a_3b_1^2b_2B(x^4 - x^2\cdot Hx^2) + a_4b_1^4Bx^4 \\ &= -2a_2b_1b_3B(x\cdot Hx^3) - (2a_2b_2^2 + 3a_3b_1^2b_2)B(x^2\cdot Hx^2) \\ &\quad + a_2b_2^2B(Hx^2)^2 \end{aligned} \tag{49.4}$$

$$
\begin{aligned}
u_5 = BF_5 &= a_1b_5Bx^5 + 2a_2b_1b_4B(x\cdot Bx^4) \\
&\quad + 2a_2b_2b_3B[(Bx^2)\cdot(Bx^3)] \\
&\quad + 3a_3b_1^2b_3B(x^2\cdot Bx^3) + 3a_3b_1b_2^2B[x\cdot(Bx^2)^2] \\
&\quad + 4a_4b_1^3b_2B(x^3\cdot Bx^2) + a_5b_1^5Bx^5 \\
&= a_1b_5Bx^5 + 2a_2b_1b_4B(x^5 - x\cdot Hx^4) \\
&\quad + 2a_2b_2b_3B[(x^2 - Hx^2)\cdot(x^3 - Hx^3)] \\
&\quad + 3a_3b_1^2b_3B(x^5 - x^2\cdot Hx^3) \\
&\quad + 3a_3b_1b_2^2B[x\cdot(x^2 - Hx^2)^2] + 4a_4b_1^3b_2B(x^5 - x^3\cdot Hx^2) \\
&\quad + a_5b_1^5Bx^5 \\
&= -2a_2b_1b_4B(x\cdot Hx^4) - (2a_2b_2b_3 + 3a_3b_1^2b_3)B(x^2\cdot Hx^3) \\
&\quad - (2a_2b_2b_3 + 6a_3b_1b_2^2 + 4a_4b_1^3b_2)B(x^3\cdot Hx^2) \\
&\quad + 2a_2b_2b_3B[(Hx^2)\cdot(Hx^3)] + 3a_3b_1b_2^2B[x\cdot(Hx^2)^2]. \qquad (49.5)
\end{aligned}
$$

Now in order to assess the symmetry or lack of symmetry in interchanging $\phi$ and $f$ we can use the identities (35) to express the mixed coefficients of $u_n$ in terms of the $b_k$ or the $a_k$ alone. We have

$$u_1 = x \qquad (50.1)$$

$$u_2 = 0 \qquad (50.2)$$

$$u_3 = {}_3C_1B(x\cdot Hx^2) \qquad (50.3)$$

$$u_4 = {}_4C_1B(x\cdot Hx^3) + {}_4C_2B(x^2\cdot Hx^2) + {}_4C_3B(Hx^2)^2 \qquad (50.4)$$

$$
\begin{aligned}
u_5 &= {}_5C_1B(x\cdot Hx^4) + {}_5C_2B(x^2\cdot Hx^3) + {}_5C_3B(x^3\cdot Hx^2) \\
&\quad + {}_5C_4B[(Hx^2)\cdot(Hx^3)] + {}_5C_5B[x\cdot(Hx^2)^2], \qquad (50.5)
\end{aligned}
$$

where

$${}_3C_1 = -2a_2b_1b_2 = \frac{2b_2^2}{b_1^2} = \frac{2a_2^2}{a_1^4} \qquad (50.3a)$$

$${}_4C_1 = -2a_2b_1b_3 = \frac{2b_2b_3}{b_1^2} = \frac{2a_2a_3}{a_1^5} - \frac{4a_2^3}{a_1^6} \qquad (50.4a)$$

$${}_4C_2 = -(2a_2b_2^2 + 3a_3b_1^2b_2) = -\frac{4b_2^3}{b_1^3} + \frac{3b_2b_3}{b_1^2} = -\frac{2a_2^3}{a_1^6} + \frac{3a_2a_3}{a_1^6}$$

$${}_4C_3 = a_2b_2^2 = -\frac{b_2^3}{b_1^3} = \frac{a_2^3}{a_1^6}$$

$$
{}_5C_1 = -2a_2b_1b_4 = \frac{2b_2b_4}{b_1^2} = -\frac{10a_2^2a_3}{a_1^7} + \frac{2a_2a_4}{a_1^6} + \frac{10a_2^4}{a_1^8} \tag{50.5a}
$$

$$
{}_5C_2 = -(2a_2b_2b_3 + 3a_3b_1^2b_3) = -\frac{4b_2^2b_3}{b_1^3} + \frac{3b_3^2}{b_1^2}
$$

$$
= (2a_2^2 - a_1a_3)(2a_2^2 - 3a_1a_3)/a_1^8
$$

$$
{}_5C_3 = -(2a_2b_3 + 6a_3b_1b_2^2 + 4a_4b_1^3b_2)
$$

$$
= \frac{8b_2^4}{b_1^4} - \frac{12b_2^2b_3}{b_1^3} + \frac{4b_2b_4}{b_1^2} = \frac{4a_2^4}{a_1^7} - \frac{8a_2^2a_3}{a_1^7} + \frac{4a_2a_4}{a_1^3}
$$

$$
{}_5C_4 = 2a_2b_2b_3 = -\frac{2b_2^2b_3}{b_1^3} = \frac{2a_2^2a_3}{a_1^7} - \frac{4a_2^4}{a_1^8}
$$

$$
{}_5C_5 = 3a_3b_1b_2^2 = \frac{6b_2^4}{b_1^4} - \frac{3b_2^2b_3}{b_1^3} = \frac{3a_2^2a_3}{a_1^7}.
$$

Now if we interchange the order of $\phi$ and $f$ in (45) and write

$$
Bf\{B\phi(mx)\} = \sum_1^\infty m^k v_k, \tag{51}
$$

we obtain the $v_k$ by replacing $u_k$ by $v_k$ in (49.k) and then interchanging the $a$'s and $b$'s. We have $u_1 = v_1 = x$ and $u_2 = v_2 = 0$. We should compare $u_k$ and $v_k$, $k \geqslant 3$, for $f'(0) = 1 = a_1 = b_1$. Then we have

$$
u_k = v_k \qquad k = 1, 2, 3. \tag{52}
$$

But we have, for example (with $a_1 = b_1 = 1$),

$$
u_4 = 2b_2b_3B(x\cdot Hx^3) + (3b_2b_3 - 4b_2^3)B(x^2\cdot Hx^2) - b_2^3B(Hx^2)^2 \tag{53.1}
$$

$$
v_4 = (2b_2b_3 - 4b_2^3)B(x\cdot Hx^3)
$$

$$
+ (3b_2b_3 - 2b_2^3)B(x^2\cdot Hx^2) + b_2^3B(Hx^2)^2. \tag{53.2}
$$

If, however, $b_2 = 0$ ($a_2 = 0$) we have

$$
u_k = v_k = 0, \qquad k = 2, 3, 4, \tag{53.3}
$$

and if $a_1 = b_1 = 1$,

$$
u_5 = v_5 = 3b_3^2B(x^2\cdot Hx^3). \tag{53.4}
$$

In case $b_2 = b_4 = b_6 = 0$, ($a_2 = a_4 = a_6 = 0$), we have

$$
u_7 = \frac{3b_3b_5}{b_1^2}B(x^2\cdot Hx^5) + \left(\frac{5b_3b_5}{b_1^2} - \frac{9b_3^3}{b_1^3}\right)B(x^4\cdot Hx^3)
$$

$$
- \frac{3b_3^3}{b_1^3}B[x\cdot(Hx^3)^2], \tag{54}
$$

where the coefficients expressed in terms of the $a$'s are

$$\frac{3b_3b_5}{b_1^2} = \frac{3a_3a_5}{a_1^8} - \frac{9a_3^3}{a_1^9},$$

$$\frac{5b_3b_5}{b_1^2} - \frac{9b_3^3}{b_1^3} = \frac{5a_3a_5}{a_1^8} - \frac{6a_3^3}{a_1^9} - \frac{3b_3^3}{b_1^3} = \frac{3a_3^3}{a_1^9}.$$

So we do not have, in general, $u_7 = v_7$ for odd functions $f(x)$ with $f'(0) = 1$.

## VI. APPLICATION TO COMPATIBLE SINGLE-SIDEBAND TRANSMISSION

The mathematical problem of compatible single-sideband transmission was formulated in Ref. 4. Given a signal $y(t)$ in $\mathscr{B}_2$ the problem is to determine $m$ such that the equation

$$B\{\sqrt{(1 + s(t))^2 + \hat{s}(t)^2}\} = my(t) + 1 \tag{55}$$

has a solution $s(t)$ in $\mathscr{B}_2$. In (55) $\hat{s}(t)$, sometimes called the quadrature signal, is the Hilbert transform of $s(t)$, and $B$ is operating on the envelope of the single-sideband signal. The idea is to transmit a single-sideband signal that is compatible with receivers designed for double-sideband (AM) reception. Setting

$$2s(t) + s^2(t) + \hat{s}^2(t) = x(t), \tag{56}$$

we may write (55) as

$$Bf\{x(t)\} = my(t), \tag{57}$$

where

$$f(x) = \sqrt{1 + x} - 1, \qquad x \geqslant -1. \tag{58}$$

Then $s(t)$ may be found from the solution $x(t)$ of (57). (This requires factoring $1 + x(t)$ in the form $g(t)\bar{g}(t)$, where the bandwidth of $g$ is half the bandwidth of $x$.) Then with $y = f(x)$ we have for the inverse

$$x = \phi(y) = 2y + y^2. \tag{59}$$

Setting

$$x = \sum_1^\infty m^k x_k, \tag{60}$$

we have from (29) with $a_1 = 2$, $a_2 = 1$, $a_k = 0$ for $k \geqslant 3$,

$$x_1 = 2y \tag{61.1}$$

$$x_2 = By^2 \tag{61.2}$$

$$x_3 = -B(y \cdot Hy^2) \tag{61.3}$$

$$x_4 = B[y \cdot H(y \cdot Hy^2)] + 1/4B(Hy^2)^2 \tag{61.4}$$

$$x_5 = -B\{y \cdot H[y \cdot H(y \cdot Hy^2)]\} - 1/4B[y \cdot H(Hy^2)^2] - 1/2B[(Hy^2) \cdot H(y \cdot Hy^2)]. \tag{61.5}$$

Replacing $H$ by $I - B$ we have from (30) the alternate forms

$$x_3 = B(y \cdot By^2) - By^3 \tag{61.3a}$$

$$x_4 = B[y \cdot B(y \cdot By^2)] - B(y \cdot By^3) + 1/4B(By^2)^2 - 3/2B(y^2 \cdot By^2) + 5/4By^4 \tag{61.4a}$$

$$\begin{aligned} x_5 = {} & B\{y \cdot B[y \cdot B(y \cdot By^2)]\} - B[y \cdot B(y \cdot By^3)] \\ & - 3/2B[y \cdot B(y^2 \cdot By^2)] + 1/4B[y \cdot (By^2)^2] \\ & - 3/4B[y \cdot (By^2)^2] + 5/4B(y \cdot By^4) \\ & + 1/2B[By^2) \cdot B(y \cdot By^2)] \\ & - 3/2B[y^2 \cdot B(y \cdot By^2)] - 1/2B[(By)^2 \cdot (By^3)] \\ & + 3/2B(y^2 \cdot By^3) \\ & + 5/2B(y^3 \cdot By^2) - 7/4By^5. \end{aligned} \tag{61.5a}$$

The factoring of $1 + x$ can be avoided by developing a series solution for $s$, $Bs = s$. We have

$$x = 2s + s^2 + \hat{s}^2$$

$$x = mx_1 + m^2x_2 + m^3x_3 + \cdots.$$

Then setting

$$s = ms_1 + m^2s_2 + m^3s_3 + \cdots \tag{62}$$

$$\hat{s} = m\hat{s}_1 + m^2\hat{s}_2 + m^3\hat{s}_3 + \cdots, \tag{63}$$

we have

$$s^2 = m^2s_1^2 + m^3 2s_1s_2 + m^4(2s_1s_3 + s_2^2) + m^5(2s_1s_4 + 2s_2s_3) + \cdots \tag{64}$$

$$\hat{s}^2 = m^2\hat{s}_1^2 + m^3 2\hat{s}_1\hat{s}_2 + m^4(2\hat{s}_1\hat{s}_3 + \hat{s}_2^2) + m^5(2\hat{s}_1\hat{s}_4 + 2\hat{s}_2\hat{s}_3) + \cdots. \tag{65}$$

Note that if $s$ belongs to $\mathscr{B}_2(\lambda)$, then the Fourier transform* of $(s + i\hat{s})$ vanishes outside $[0, \lambda]$ and that of its complex conjugate $(s - i\hat{s})$

*Here the Fourier transform of $\hat{s}$ is $-i(sgn\omega)S(\omega)$ where $S(\omega)$ is the Fourier transform of $s$.

vanishes outside $[-\lambda, 0]$. Thus the Fourier transform of $e^{-i\lambda t/2}(s + i\hat{s})$ vanishes outside $[-(\lambda/2), \lambda/2]$. It follows that the Fourier transform of $s^2 + \hat{s}^2$ vanishes outside $[-\lambda, \lambda]$, and hence that the sums of the coefficients of $m^n$ in (64) and (65) are functions whose Fourier transforms vanish outside $[-\lambda, \lambda]$.

Now we can solve successively for $s_k$. It is convenient to introduce the Hilbert transform (Quadrature) operator $Q$

$$\hat{g} = Qg \tag{66}$$

to indicate the "hat" of complicated expressions.

Equating coefficients of $m_k$ in

$$x = 2s + s^2 + \hat{s}^2,$$

we have

$$s_1 = x_1/2 = y \tag{67.1}$$

$$\begin{aligned} s_2 &= 1/2x_2 - 1/2(s_1^2 + \hat{s}_1^2) \\ &= 1/2By^2 - 1/2(y^2 + \hat{y}^2), \end{aligned} \tag{67.2}$$

which may be written, using $Bs_2 = s_2$, as

$$s_2 = -1/2B\hat{y}^2 \tag{67.2a}$$

$$\begin{aligned} s_3 &= 1/2x_3 - (s_1 s_2 + \hat{s}_1 \hat{s}_2) \\ &= -1/2B(y \cdot Hy^2) + 1/2yB\hat{y}^2 + 1/2\hat{y} \cdot QB\hat{y}^2. \end{aligned} \tag{67.3}$$

Here we may write

$$yB\hat{y}^2 = y \cdot \hat{y}^2 - y \cdot H\hat{y}^2$$

and then use $Bs_3 = s_3$ to obtain

$$\begin{aligned} s_3 &= -1/2B(y \cdot Hy^2) - 1/2B(y \cdot H\hat{y}^2) + 1/2B(y \cdot \hat{y}^2) + 1/2B(\hat{y} \cdot QB\hat{y}^2) \\ &= -1/2B[y \cdot H(y^2 + \hat{y}^2)] + 1/2B(y \cdot \hat{y}^2) + 1/2B(\hat{y} \cdot QB\hat{y}^2). \end{aligned}$$

Then since $H(y^2 + \hat{y}^2) = 0$, we have

$$s_3 = 1/2\ B(y \cdot \hat{y}^2) + 1/2\ B(\hat{y} \cdot QB\hat{y}^2) \tag{67.3a}$$

$$\begin{aligned} s_4 &= 1/2\ x_4 - (s_1 s_3 + 1/2\ s_2^2) - (\hat{s}_1 \hat{s}_3 + 1/2\ \hat{s}_2^2) \\ &= 1/2\ B[y \cdot H(y \cdot Hy^2)] + 1/4\ B(Hy^2)^2 \\ &\quad - 1/2\ yB(y \cdot \hat{y}^2) - 1/2\ \hat{y}QB(y \cdot \hat{y}^2) \\ &\quad - 1/8\ (B\hat{y}^2)^2 - 1/8\ (QB\hat{y}^2)^2. \end{aligned} \tag{67.4}$$

There appears to be no simplification here. One may prefer the alternate expression (61.4a) for $x_4$ to eliminate the $H$ operator. Note

that $QB$ can be replaced by the bandlimiting quadrature operator $\hat{B}$ where

$$\hat{B}g = (\hat{B}g)(t) = \int_{-\infty}^{\infty} g(s) \frac{1 - \cos \lambda(t - s)}{\pi(t - s)} \, ds. \tag{68}$$

## VII. THE COMPLEX-VALUED COMPANDING PROBLEM

The c.v. companding problem is considerably more complicated than the r.v. companding problem, even for the same analytic companding function. For example, if

$$f(x) = \tan^{-1} x + \epsilon x, \qquad (\epsilon > 0),$$

we know from the Landau-Miranker theory that the r.v. companding problem

$$Bf(x) = y$$

has a unique solution $x$ in $\mathscr{B}_2$ corresponding to every real-valued $y$ in $\mathscr{B}_2$. However, in the case of complex-valued $y$, this may not be true because $x$ must then take complex values which, if the norm of $y$ is not restricted, may be singularities of $f$. In addition, we are confronted with the problem of establishing the uniqueness of the solution, which may require still more severe restrictions on the norm of $y$.

Beurling's uniqueness proof (see Ref. 1) for the r.v. companding problem is elegant and simple: Suppose $f(x) = \mathscr{O}(|x|)$, $|x| \to 0$, and is monotone increasing, and further that

$$Bf(x_1) = y \qquad \text{and} \qquad Bf(x_2) = y,$$

with $x_1$, $x_2$, (and $y$) in $\mathscr{B}_2$. Then $f(x_1)$ and $f(x_2)$ belong to $L_2$ and $B\{f \cdot (x_1) - f(x_2)\} = 0$, i.e., the Fourier transform of $\{f(x_1) - f(x_2)\}$ vanishes over $(-\lambda, \lambda)$, and therefore $\{f(x_1) - f(x_2)\}$ must be orthogonal to $(x_1 - x_2)$. But this is impossible unless $x_1 \equiv x_2$, for otherwise $(x_1 - x_2) \cdot \{f(x_1) - f(x_2)\}$, which is everywhere non-negative, will be positive everywhere on the real axis, except at the isolated zeros (if any) of $(x_1 - x_2)$.

For establishing uniqueness in the c.v. companding problem, it would seem that the weakest analogue of monotocity should be "schlichtness" of $f$, i.e., that $x$ should be confined to a region, where $f(x_1) = f(x_2)$ implies $x_1 = x_2$. This suffices to establish uniqueness of the solution in the special case where $x$ has a one-sided Fourier transform, but we are not able to see that it suffices in the general case. We can establish the following analogue of Buerling's theorem, where, without loss of generality, we assume $f'(0)$ is positive.

*Theorem* 1: *Suppose* $f(0) = 0$, $f'(0) > 0$, *and* $f(z)$ *is analytic in a convex region* $G$ *including the origin, wherein*

$$Re\{f'(z)\} > 0.$$

*If $x_1(t)$ and $x_2(t)$ belong to $\mathscr{B}_2$ and are confined to $G$ for all real $t$, then*

$$Bf(x_1) = y \qquad \text{and} \qquad Bf(x_2) = y$$

*imply*

$$x_1(t) \equiv x_2(t).$$

*Proof of Theorem* 1: Since $G$ is convex, any two points $x_1$ and $x_2$ in $G$ can be connected by a straight line segment in $G$. Suppose $(x_1 - x_2) = re^{i\theta}$, where $r > 0$. Then integrating $f'$ along the connecting line segment, we have

$$f(x_1) - f(x_2) = e^{i\theta} \int_0^r f'(x_2 + se^{i\theta})ds,$$

and hence

$$\operatorname{Re} \frac{f(x_1) - f(x_2)}{x_1 - x_2} = \operatorname{Re} \frac{1}{r} \int_0^r f'(x_2 + se^{i\theta})ds > 0.$$

In case $x_1 \to x_2$, the limit is $\operatorname{Re} f'(x_2) > 0$.

Now $\{f[x_1(t)] - f[x_2(t)]\}$ belongs to $L_2$ and must be orthogonal to all members of $\mathscr{B}_2$, in particular to $\{x_1(t) - x_2(t)\}$; i.e., setting

$$P(t) = (\bar{x}_1 - \bar{x}_2)\{f(x_1) - f(x_2)\} = |x_1 - x_2|^2 \frac{\{f(x_1) - f(x_2)\}}{x_1 - x_2},$$

the integral of $P(t)$ must vanish. However, we see that the real part of $P(t)$ is non-negative everywhere on the real axis, and vanishes only where $x_1 = x_2$. Since the integral of $P$ is zero, its real part vanishes a.e. Thus the function $\{x_1(t) - x_2(t)\}$ in $\mathscr{B}_2$ vanishes a.e., and hence everywhere. □

Now we can establish that the c.v. companding problem,

$$Bf(x) = y,$$

will have a solution $x$, which will take values in a disk centered on the origin, wherein $\operatorname{Re}\{f'(z)\} > 0$, provided $\|y\|_2$ is sufficiently small. Then the uniqueness of the solution follows from Theorem 1.

An objectionable, but inherent, feature of companding problems (as formulated here) is that a restriction on $\|y\|_\infty$ is not sufficient to give a corresponding restriction on $\|x\|_\infty$. We can, however, establish that $\|x\|_2$ will be small if $\|y\|_2$ is small, and hence that $\|x\|_\infty$ will be small, according to the inequality (given in the introduction) for a function $g$ in $\mathscr{B}_2(\lambda)$,

$$\|g\|_\infty \leq \sqrt{\lambda/\pi}\,\|g\|_2. \tag{69}$$

In the sequel, we assume, for convenience and without loss of generality, that $f'(0) = 1$,

$$f(z) = z + \sum_{2}^{\infty} b_k z^k, \qquad |z| < R_0, \tag{70}$$

where $R_0$ (perhaps $\infty$) is the radius of convergence of the series. We exclude the trivial case $f(z) = z$, and define

$$M(r) = \max_{|z|=r} |f'(z) - 1| \leqslant \sum_{2}^{\infty} k|b_k| r^{k-1}, \qquad r < R_0, \tag{71}$$

which increases steadily from 0 to $\infty$, allowing us to define $p$ uniquely by

$$M(p) = 1. \tag{72}$$

Then it is clear that

$$\mathrm{Re}\{f'(z)\} > 0 \qquad \text{for} \quad |z| < p. \tag{73}$$

We are now able to establish the following result.

*Theorem* 2: *Let $y(t)$ be any complex-valued function in $\mathscr{B}_2 = \mathscr{B}_2(\lambda)$, satisfying*

$$\sqrt{\lambda/\pi}\, \|y\|_2 \leqslant \max_{0<r<p} \{r[1 - M(r)]\} = r_0[1 - M(r_0)]. \tag{74}$$

*Then the companding problem*

$$Bf(x) = y$$

*has a unique solution $x = x(t)$ in $\mathscr{B}_2$.*

*Proof of Theorem* 2: We can use the method of Landau and Miranker to obtain a Cauchy sequence $\{x_n\}$ converging to the solution $x$, provided we restrict $\|y\|_2$, in the end, to be sufficiently small that all the approximants satisfy $|x_n| < p$.

Assuming the norm of $y$ to be sufficiently small, we take

$$x_1 = y = By, \tag{75}$$

which should be a good approximation to $x$ for small $y$. Then we set, following Landau and Miranker,

$$x_{n+1} = x_n + y - Bf(x_n), \qquad n \geqslant 1, \tag{76}$$

so that, by induction, $Bx_n = x_n$, i.e., $x_n$ belongs to $\mathscr{B}_2$. We have, writing the same equation for $n - 1$ and subtracting,

$$\begin{aligned} x_{n+1} - x_n &= x_n - Bf(x_n) - \{x_{n-1} - Bf(x_{n-1})\} \\ &= B[x_n - f(x_n) - \{x_{n-1} - f(x_{n-1})\}]. \end{aligned} \tag{77}$$

Now we write

$$f(x_n) - x_n - \{f(x_{n-1}) - x_{n-1}\} = \int_{x_{n-1}}^{x_n} \{f'(z) - 1\}dz. \tag{78}$$

Then assuming that

$$|x_n| \leq r < p \qquad \text{for all} \quad n \geq 1 \tag{79}$$

[and all $t$, suppressed in the notation $x_n = x_n(t)$] we have in (78)

$$\left| \int_{x_{n-1}}^{x_n} \{f'(z) - 1\}dz \right| \leq M(r)\,|x_n - x_{n-1}|, \tag{80}$$

where

$$M(r) < M(p) = 1.$$

Substituting in (77) the inequality (80) for the magnitude of the function in (78), we obtain

$$\|x_{n+1} - x_n\|_2 \leq M(r)\,\|x_n - x_{n-1}\|_2. \tag{81}$$

So, under the assumption (79), $\{x_n\}$ forms a Cauchy sequence converging to $x$ in $\mathscr{B}_2$ [cf. Landau, Ref. 1]. It follows from (76) that

$$Bf(x) = y. \tag{82}$$

Now we would like to see how large $\|y\|_2$ may be in order that (79) hold, giving the conclusion in (82). We write

$$x_n = x_1 + (x_2 - x_1) + (x_3 - x_2) + \cdots + (x_n - x_{n-1}) \tag{83}$$

from which follows

$$\|x_n\|_2 \leq \sum_1^n \|x_k - x_{k-1}\|_2, \tag{84}$$

where

$$x_1 = y \qquad \text{and} \qquad x_0 = 0.$$

Applying (81) to (84), we have

$$\|x_n\|_2 \leq \frac{1 - \alpha^n}{1 - \alpha}\,\|y\|_2, \tag{85}$$

where $\alpha = M(r) < 1$, provided (79) holds. This will be the case, according to (69), if

$$\|x_n\|_2 \leq \sqrt{\pi/\lambda}\,r \qquad \text{for all} \quad n \geq 1, \tag{86}$$

which, in turn, will hold if in (85) we have

$$\sqrt{\pi/\lambda}\ \|y\|_2 \leq r[1 - M(r)]. \tag{87}$$

Here we are free to take the maximum over $r$. Thus the problem will have a solution $x$ satisfying the hypotheses, provided the norm of $y$ satisfies

$$\sqrt{\pi/\lambda}\ \|y\|_2 \leq \max_{0<r<p}\{r(1 - M(r)\} = r_0[1 - M(r_0)], \tag{88}$$

and the solution is unique according to Theorem 1. □

We note that if (88) is satisfied, then the solution $x$ satisfies, according to (85) and (69),

$$\|x\|_\infty \leq r_0 < p. \tag{89}$$

So, in fact, the restriction (88) on the norm of $y$ is too severe. We obtain a slightly better result later, using a different method.

We now wish to show that if $y_1$ and $y_2$ are close to each other, then the corresponding solutions, $x_1$ and $x_2$, are also close to each other.

*Lemma: Let $y_1$ and $y_2$ satisfy the hypotheses on $y$ in Theorem 2. Then the solutions of*

$$Bf(x_1) = y_1 \qquad \textit{and} \qquad Bf(x_2) = y_2$$

*satisfy*

$$\|x_1 - x_2\|_2 \leq \frac{\|y_1 - y_2\|_2}{1 - M(r_0)} \leq 2\sqrt{\pi/\lambda}\ r_0 \tag{90}$$

$$\|x_1 - x_2\|_\infty \leq \sqrt{\pi/\lambda}\ \frac{\|y_1 - y_2\|_2}{1 - M(r_0)} \leq 2r_0. \tag{91}$$

*Proof of the Lemma*: We have

$$x_1 - x_2 = y_1 - y_2 - B[f(x_1) - x_1 - f(x_2) + x_2], \tag{92}$$

giving

$$\|x_1 - x_2\|_2 \leq \|y_1 - y_2\|_2 + \|B[\cdot]\|_2 \leq \|y_1 - y_2\|_2 + \|[\cdot]\|_2. \tag{93}$$

Also, since, according to (89),

$$|x_1| \leq r_0 < p \qquad \text{and} \qquad |x_2| \leq r_0 < p,$$

we have from (78) and (80),

$$\|f(x_1) - x_1 - f(x_2) + x_2\|_2 \leq M(r_0)\|x_1 - x_2\|_2, \tag{94}$$

which with (93) gives

$$\|x_1 - x_2\|_2 \leq \frac{\|y_1 - y_2\|_2}{1 - M(r_0)}. \tag{95}$$

This, with (69) and the assumptions on $y_1$ and $y_2$, establishes the lemma. □

With this Lemma and Theorem 2 we can show that the problem

$$Bf\{x(t;m)\} = my(t),$$

for fixed $y$ in $\mathscr{B}_2$, has a unique solution in $\mathscr{B}_2$ for all complex $m$ of sufficiently small magnitude, the solution $x(t; m)$ being a continuous function of the complex variable $m$ in a certain disk centered on the origin. To establish for each $t$ that $F(m;t) = x(t;m)$ is an analytic function of $m$ in that disk, we show that $F(m)$ has a derivative (nondirectional) there. Working with the derivative we are able to improve on Theorem 2. It is convenient now to set $\sqrt{\lambda/\pi}\,\|y\|_2 = 1$ so that $|y(t)| \leqslant 1$.

*Theorem* 3: *Let $y(t)$ be any complex-valued function in $\mathscr{B}_2 \equiv \mathscr{B}_2(\lambda)$ satisfying*

$$\sqrt{\lambda/\pi}\,\|y\|_2 = 1.$$

*Then the problem*

$$Bf\{x(t;m)\} = my(t)$$

*has a unique solution $x(t;m)$ in $\mathscr{B}_2$ for all complex $m$ satisfying*

$$|m| \leqslant \alpha(p) = \int_0^p [1 - M(r)]dr, \tag{96}$$

*where $M(r)$ and $p$ are defined in* (71) *and* (72). *Furthermore, for each fixed real $t$, $x(t;m)$ is an analytic function of $m$, $|m| \leqslant \alpha(p)$, and hence, since $x(t; 0) \equiv 0$,*

$$x(t;m) = \sum_1^\infty m^k x_k(t), \qquad |m| \leqslant \alpha(p), \qquad (-\infty < t < \infty) \tag{97}$$

*where the $x_k(t)$ depend only on $y(t)$ and $f$.*

We note, before proving Theorem 3, that in Theorem 2, $0 < r_0 < p$, and in Theorem 3

$$\alpha(p) = \int_0^{r_0} [1 - M(r)]dr + \int_{r_0}^p [1 - M(r)]dr,$$

where $M(r)$ increases from 0 to 1 over $(0, p)$. Thus

$$\int_0^{r_0} [1 - M(r)]dr > r_0[1 - M(r_0)],$$

and hence

$$r_0[1 - M(r_0)] < \alpha(p) < p. \tag{98}$$

Then, according to Theorem 3, the c.v. companding problem

$$Bf(x) = y$$

has solutions for $y$ of larger norm than specified in Theorem 2.

*Proof of Theorem* 3: We first consider the solutions $x_1 = x(t;m_1)$, $x_2 = x(t;m_2)$, corresponding to $y_1 = m_1 y(t)$, $y_2 = m_2 y(t)$, where

$$m_2 = m_1 + \epsilon, \qquad \epsilon = |\epsilon| e^{i\theta} \tag{99}$$

and

$$|m_1| + |\epsilon| \leq r_0[1 - M(r_0)], \tag{100}$$

so that Theorem 2 and the Lemma apply.

We have

$$y_2 - y_1 = \epsilon y(t), \tag{101}$$

and hence, from the Lemma,

$$\| x_2 - x_1 \|_2 \leq \frac{|\epsilon|}{1 - M(r_0)} \cdot \| y \|_2 = \frac{|\epsilon| \sqrt{\pi/\lambda}}{1 - M(r_0)}. \tag{102}$$

Now

$$B\{f(x_2) - f(x_1)\} = \epsilon y, \tag{103}$$

which we rewrite as

$$\frac{x_2 - x_1}{\epsilon} = y - B\left\{\frac{f(x_2) - f(x_1) - (x_2 - x_1)}{\epsilon}\right\}. \tag{104}$$

We intend to let $\epsilon \to 0$ (with any argument) and show that the quantity on the left tends to a limit, independent of $\arg(\epsilon)$; viz., $F'(m_1;t)$, where

$$F'(m;t) = \frac{\partial}{\partial m} x(t;m), \qquad |m| \leq r_0[1 - M(r_0)]. \tag{105}$$

From (78), (80), and (102) we have

$$\| f(x_2) - f(x_1) - (x_2 - x_1) \|_2 \leq M(r_0) \| x_2 - x_1 \|_2 \leq \frac{|\epsilon| \sqrt{\pi/\lambda} M(r_0)}{1 - M(r_0)}. \tag{106}$$

So

$$\frac{f(x_2) - f(x_1) - (x_2 - x_1)}{\epsilon} \quad \text{belongs to} \quad L_2. \tag{107}$$

We also have from (102), or the Lemma,

$$\| x_2 - x_1 \|_\infty \leq \frac{|\epsilon|}{1 - M(r_0)}. \tag{108}$$

Thus we may write

$$x_2 = x_1 + \epsilon u, \tag{109}$$

where

$u = u(t;m_1, \epsilon)$ belongs to $\mathscr{B}_2$ and $|u|$

$$= \mathscr{O}(1) \quad \text{as} \quad \epsilon \to 0. \tag{110}$$

Then

$$f(x_2) - f(x_1) = f(x_1 + \epsilon u) - f(x_1)$$
$$= \epsilon u f'(x_1) + \mathscr{O}(\epsilon^2 u^2).$$

So (104) may be rewritten as

$$u = y - B\{uf'(x_1) - u + \mathscr{O}(\epsilon u^2)\}. \tag{111}$$

Now letting $\epsilon \to 0$ and replacing $m_1$ by $m$ and $x_1$ by $x(t;m)$, we obtain, setting $u(t;m, o) = u(t;m)$, the equation

$$u(t;m) = y(t) - B\{u(t;m)[f'\{x(t;m)\} - 1]\}, \tag{112}$$
$$|m| \leq A < r_0[1 - M(r_0)].$$

Here we make the identification

$$u(t;m) = F'(m;t) = \frac{\partial}{\partial m} x(t;m), \qquad |m| \leq A \tag{113}$$

by verifying that (112) has a solution $u(t;m)$ in $\mathscr{B}_2$, in fact, for $|m|$ larger than $r_0[1 - M(r_0)]$. We observe, since $x(t;0) \equiv 0$, and $f'(0) = 1$, that

$$u(t;0) = y(t). \tag{114}$$

Actually, we can obtain better estimates for $|x(t;m)|$ by integrating its partial derivative from 0 to $m$.

We consider the equation for $u$,

$$u = y - B\{u \cdot [f'(x) - 1]\}, \tag{115}$$

assuming $x = x(t; m)$ is known and satisfies

$$\| x \|_\infty \leq r < p, \tag{116}$$

so that

$$|f'(x) - 1| \leq M(r) < M(p) = 1. \tag{117}$$

Using this inequality in (115) we obtain

$$\| u \|_2 \leq \| y \|_2 + M(r) \| u \|_2, \tag{118}$$

or

$$\| u \|_2 \leq \frac{\| y \|_2}{1 - M(r)} = \frac{\sqrt{\pi/\lambda}}{1 - M(r)}. \tag{119}$$

The last inequality implies that (115) has a solution $u$ in $\mathscr{B}_2$ (obtained iteratively), in fact, for $\| x \|_\infty < p$, since $M(r) < 1$ for $r < p$. We note further that the inequality (119) is crude, with equality possible only for $r = 0$, for we cannot have

$$| f'\{x(t;m)\} - 1 | \equiv M(r), \qquad (-\infty < t < \infty)$$

unless $x(t;m) \equiv 0$. Therefore, in (115) we have

$$\| B\{u[f'(x) - 1]\|_2 < M(r) \| u \|_2 \quad \text{for} \quad 0 < \| x \|_\infty \leq r < p.$$

So we have strict inequality in (119) for $0 < r < p$. Hence,

$$\| u \|_\infty < \frac{1}{1 - M(r)} \quad \text{for} \quad 0 < \| x \|_\infty \leq r < p. \tag{120}$$

Now let us set

$$m = \alpha e^{i\theta}, \qquad \alpha > 0 \tag{121}$$

and

$$r(\alpha) = \max_\theta \| x(t;\alpha e^{i\theta}) \|_\infty. \tag{122}$$

We want to see how large we can make $\alpha$, say $\alpha(p)$, and have $r(\alpha) < p$. Using (120) in

$$x(t;m) = \int_0^m u(t;\xi) d\xi, \tag{123}$$

we obtain the inequality

$$r(\alpha) < \int_0^\alpha \frac{d\xi}{1 - M[r(\xi)]}, \qquad 0 < \alpha \leq \alpha(p). \tag{124}$$

Then, after defining

$$s(\alpha) = \int_0^\alpha \frac{d\xi}{1 - M[s(\xi)]}, \qquad 0 < \alpha \leq \alpha(p), \tag{125}$$

it is clear, since $M(r)$ is an increasing function of $r$, that we will have

$$r(\alpha) < s(\alpha), \qquad 0 < \alpha \leq \alpha(p). \tag{126}$$

Differentiating (125) with respect to $\alpha$, we obtain the simple equation

$$s'(\alpha)\{1 - M[s(\alpha)]\} = 1,$$

or, considering $\alpha$ as a function of $s$,

$$\frac{d\alpha}{ds} = 1 - M(s).$$

Thus

$$\alpha(s) = \int_0^s [1 - M(r)]dr, \qquad 0 < s \leqslant p. \tag{127}$$

We have $s(\alpha(p)) = p$ and $r(\alpha) < s(\alpha)$ for $\alpha > 0$. So we will have

$$\| x(t;m) \|_\infty < p \quad \text{for} \tag{128}$$

$$| m | \leqslant \alpha(p) = \int_0^p [1 - M(r)]dr. \tag{129}$$

According to (120) and (128), the partial derivative $u(t;m)$ will exist for $| m |$ somewhat larger than $\alpha(p)$. This completes the proof of Theorem 3. □

## VIII. AN ILLUSTRATIVE EXAMPLE

It will be shown in a future paper that the r.v. companding problem

$$B\left\{\frac{x}{1 - x}\right\} = y, \qquad x < 1, \qquad x, y \text{ in } \mathscr{B}_2(\lambda), \tag{130}$$

is equivalent to finding the reproducing kernel for a certain Hilbert space of bandlimited functions. The specific problem with $\lambda = 2$ (for convenience) and

$$y = m \frac{\sin 2t}{2t} \tag{131}$$

is quite easily solved. For real $m > -2$, the solution is

$$x(t;m) = 2\beta \frac{\sin 2t}{2t} - \beta^2 \left(\frac{\sin t}{t}\right)^2, \tag{132}$$

where

$$\beta = m/(2 + m).$$

We need not be concerned here with the derivation of this solution, as we will later show directly that it satisfies

$$B\left\{\frac{x(t;m)}{1 - x(t;m)}\right\} = m \frac{\sin 2t}{2t} \tag{133}$$

for all $m$ in a certain region of the complex plane, but for no other $m$.

We know from Theorem 3 that (133) has a unique solution for sufficiently small $|m|$, and that the solution is an analytic function of $m$. It follows that (132) is the solution of (133) for all complex $m$, $|m| < |m_0|$, for some $|m_0| > 0$. Since $m = -2$ is the only point where $x(t;m)$ is not analytic, we might suppose $|m_0| = 2$. The series expansion of $x(t;m)$ certainly converges (uniformly in $t$) for all $|m| < 2$, but it is not a solution of (133) for all such $m$. For example, we have

$$x\left(\pm \frac{\pi}{2};m\right) = -\left(\frac{2\beta}{\pi}\right)^2$$

and

$$m = \frac{2\beta}{1-\beta}.$$

Then for $\beta = \pm i\pi/2$, we have

$$m = \frac{-2}{1 \pm i\dfrac{2}{\pi}}, \quad \text{and} \quad x\left(\pm \frac{\pi}{2};m\right) = 1.$$

Therefore, the meromorphic function of $t$,

$$f\{x(t;m)\} = \frac{x(t;m)}{1 - x(t;m)}$$

will have poles at $t = \pm \pi/2$ for $m = -2/(1 \pm i2/\pi)$. Thus we have here an example, $|m| < 2$, for which (132) is not a solution of (133). However, it is, according to Theorem 3, for all $m$ satisfying $|m| \leqslant 3 - 2\sqrt{2}$. This, as it turns out, is an overly conservative upper bound for $|m|$.

We now turn to the problem of determining precisely those $m$ for which (132) is a solution of (133).

First we can easily show that the r.v. problem has no solution for $m \leqslant -2$ by convolving both sides of (133) with $1/\pi K(t)$, where $K(t) = (\sin t)^2/t^2$. The result is

$$\int_{-\infty}^{\infty} \frac{x(s;m)}{1 - x(s;m)} \cdot \frac{1}{\pi} K(t-s)ds = \frac{m}{2} K(t). \tag{134}$$

Since $x/(1-x) > -1$, and $K(t) \geqslant 0$, we have

$$\int_{-\infty}^{\infty} \frac{x(s;m)}{1 - x(s;m)} \cdot \frac{1}{\pi} K(t-s)ds > -\frac{1}{\pi} \int_{-\infty}^{\infty} K(t)dt = -1$$

This gives, setting $t = 0$ in (134),

$$m > -2. \tag{135}$$

Now to proceed towards our stated goal, we first write

$$1 - x(t;m) = \left(1 - \beta e^{it} \frac{\sin t}{t}\right)\left(1 - \beta e^{-it} \frac{\sin t}{t}\right). \tag{136}$$

Then in order for $x(t;m)$ to be a solution of the problem, the Fourier transform of the function

$$h(t;m) = \frac{x(t;m)}{1 - x(t;m)} - m \frac{\sin 2t}{2t} \tag{137}$$

must vanish over $(-2,2)$. With a bit of manipulation we arrive at the expression

$$h(t;m) = g(t;m) + g(-t;m), \tag{138}$$

where

$$g(t;m) = \frac{-\beta^2}{1 - \beta} \frac{e^{2it}\left(1 - e^{it} \dfrac{\sin t}{t}\right)}{2it\left(1 - \beta e^{it} \dfrac{\sin t}{t}\right)} \tag{139}$$

and

$$\beta = \frac{(m/2)}{1 + (m/2)} \neq 1.$$

We now introduce the complex variable $\tau = t + iu$, and observe that

$$|g(t + iu;\, m)| = \mathscr{O}\left\{\frac{e^{-2u}}{|t + iu|}\right\}, \qquad u \to +\infty. \tag{140}$$

Then if the denominator satisfies the condition,

$$\left(1 - \beta e^{i\tau} \frac{\sin \tau}{\tau}\right) \textit{ is zero-free} \quad \text{for} \quad u \geqslant 0, \tag{141}$$

it is easy to see (by contour integration in the upper half-plane) that

$$G(\omega;m) = \int_{-\infty}^{\infty} g(t;m)e^{-i\omega t}dt = 0 \quad \text{for} \quad \omega < 2. \tag{142}$$

On the other hand, if the function in (141) has zeros $\tau_k$ in the upper half plane $u > 0$, (it must have no real zeros in order for $g$ to have a Fourier transform) we will have, by the calculus of residues,

$$G(\omega;m) = \sum_1^n c_k e^{-i\omega\tau_k}, \qquad c_k \neq 0, \tag{143}$$

where $n$, depending on $\beta$, is finite, since it is clear that the function in (141) can have only a finite number of zeros in the upper half-plane. Since the Fourier transform of $h(t;m)$ is given by

$$H(\omega, m) = G(\omega;m) + G(-\omega;m) \tag{144}$$

it will vanish over $(-2,2)$ if, and only if, the condition (141) is satisfied.

Now the values taken by $e^{i\tau} \sin \tau/\tau$ in the upper half-plane, $u \geqslant 0$ are precisely those values on the boundary and interior of the cardioid-like region whose boundary is described parametrically by

$$e^{it} \frac{\sin t}{t}, \qquad -\pi \leqslant t \leqslant \pi.$$

(Some values are taken more than once.) Then $x(t;m)$ will be a solution to the problem except for those values of $m$ such that $1/\beta$ is a point on the boundary or in the interior of the cardioid-like region. By the mapping

$$m = \frac{2}{\frac{1}{\beta} - 1},$$

$x(t;m)$ is a solution to the problem for precisely those (finite) $m$ lying in the region to the right of the boundary line described parametrically by

$$m = \frac{2}{e^{it} \frac{\sin t}{t} - 1}, \qquad -\pi \leqslant t \leqslant \pi. \tag{145}$$

This region (see Fig. 1) includes the half-plane $\mathrm{Re}\{m\} \geqslant -4/3$, its boundary indenting more to the left near the real axis, having a cusp at its leftmost point, $m = -2$, where it is tangent to the real axis. It is found that the distance from the origin to the boundary is minimal (see circle in Fig. 1) at the point $m_0$ and its conjugate, where

$$\begin{aligned} m_0 &= (-2 + \xi) + i\xi, \\ \xi &= \frac{1}{t_0} \doteq .4895273114 \doteq (2.42786943)^{-1}, \end{aligned} \tag{146}$$

$t_0$ is the smallest positive root of $\sin t/t = \cos t + \sin t$,

$$|m_0| = \frac{\sqrt{2}}{\sin t_0} \doteq 1.58781760, \ \arg\{m_0\} = \frac{\pi}{4} + t_0.$$

So $|m_0|$ is the largest number such that $x(t;m)$ is a solution for *all* $m$

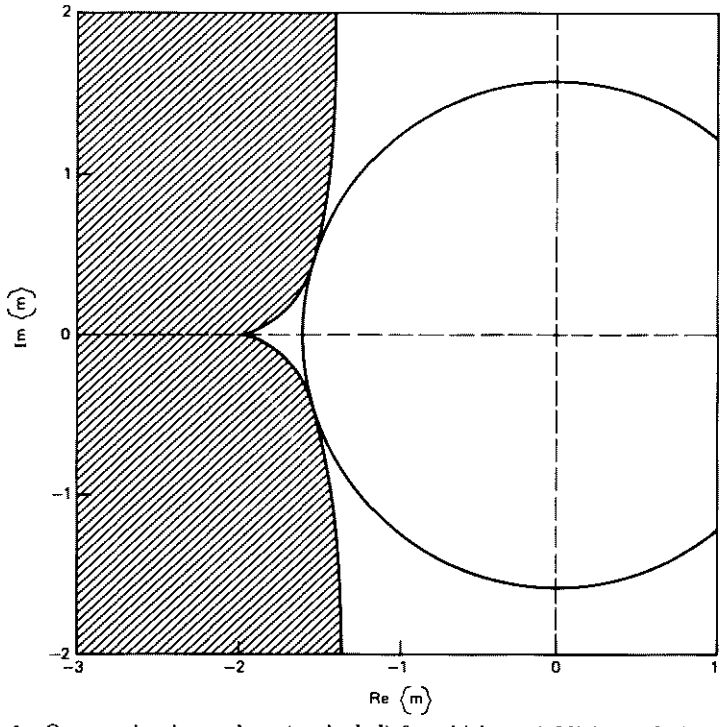



Fig. 1—Open region in $m$-plane (unshaded) for which eq. (132) is a solution of eq. (133). Shown is the largest disk (centered on the origin) contained in the region.

satisfying $|m| < |m_0|$. Also,

$$|m_0| = \min_t R_0(t),$$

where $R_0(t)$ is the radius of convergence of the series

$$\frac{x(t;m)}{1 - x(t;m)} = m \frac{\sin 2t}{2t} + m^2 h_2(t) + m^3 h_3(t) + \cdots,$$

the minimum occurring for $t = \pm t_0$.

## IX. CONCLUSION

The expressions for the $n$th order components $x_n(t)$ of the series solution to the companding problem become so complicated that, for practical purposes, only the first few are of interest. These should be useful in correcting small distortions in nonlinear transmission systems which fit the companding model. It would appear that the corrections applied internally to the inverse function (the $z_k$ in Section V) would be more effective for correcting larger distortions, especially if the lower frequency components are predominant in the signals. In

this connection, the simpler "approximate identity" should be quite effective for correcting small to moderate distortions of a more general nature, as evidenced by the inequality given in the appendix. Experimental evidence of the effectiveness of these correction schemes would be desirable.

The question of the convergence of the series solution is a matter of little practical concern, but the fact that it does attaches more mathematical significance to the results. To settle this question we had to show that the complex-valued companding problem has solutions for functions of sufficiently small norm. This generalizes the result for functions of one-sided spectra; and whether or not the general result will ever find practical application, it is an interesting addition to a theory, though still incomplete in many respects. For example, it is doubtful that the condition that $x(t)$ be confined to a convex region $G$ where $\mathrm{Re}\{f'\} > 0$ is a necessary condition for uniqueness of the solution. In this connection, one could probably use analytic continuation arguments to show that the specific problem examined in Section VIII has solutions only for those values of $m$ for which the (particular) solution given is the only solution, this being unique for sufficiently small $|m|$, and being an analytic function of $m$ having no branch points. Also there is the difficult question of determining for what $y(t)$ the companding problem has a solution, where particular interest is attached to the real-valued problem with analytic companding functions. It can be shown for the case $f(x) = x/(1 - x)$, $x < 1$, that the problem has a solution for every (r.v.) $y$ in $\mathscr{B}_2$ satisfying $y > -1$. This suggests (conjecture) that the r.v. companding problem with $f(x) = x/(1 - x^2)$, $-1 < x < 1$, has a solution for every (r.v.) $y$ in $\mathscr{B}_2$, or more generally for monotone $f(x)$ defined on $(-1, 1)$ having singularities as strong as poles at $\pm 1$. In general, it is not enough for $f(x)$ to increase from $-\infty$ to $+\infty$ over its range of definition in order to draw the same conclusion; e.g., $f(x) = \log(1 + x)$, $x > -1$. The questions raised here are certainly worthy of future consideration.

In connection with the series solution, one naturally inquires whether an explicit formula (albeit complicated and involving partitions of various kinds) can be given for the general term $x_n$. Perhaps the combinatorics experts will consider this question.

We note that the solution $x = B\phi(y)$, valid for $y$ (of sufficiently small norm) whose Fourier transforms vanish outside $[0, \lambda]$, is verified by the fact that in (29.n) the expression reduces to $x_n = a_n B y^n$, the other terms vanishing because $B$ is operating on functions whose Fourier transforms vanish over $(-\infty, \lambda)$. The same reduction occurs in the expression (34.n), because, in this case, $B$ is operating on functions whose Fourier transforms vanish over $(-\infty, 0)$ and agree over $[0, \lambda]$; i.e.,

$$B[\cdots] = By^n$$

holds for each term in (34.n), the sum of all the coefficients being $a_n$.

Some interesting identities are obtained by equating the expressions for $x_n$ in the series solution of the general problem to those obtained from the explicit solution (given in the introduction) to the special problem

$$B \log(1 + x) = my,$$

which involve $\hat{y}$, the Hilbert transform of $y$, which does not appear in the more general expressions. For example, we find from (34.3) that $x_3$ in the series solution of this problem is given by

$$x_3 = 1/2B(y \cdot By^2) - 1/3By^3,$$

and, from the series expansion of the explicit solution, by

$$x_3 = 1/8yB(y^2 - \hat{y}^2) - 1/8B(y\hat{y}^2) + 1/24\,By^3 + 1/4\hat{y}B(y\hat{y}).$$

It is an interesting exercise to show directly that these two expressions are identical.

Finally, since truly bandlimited signals exist only as mathematical abstractions, some attention should be given to developing a mathematical theory of practical companding problems,

$$\int_{-\infty}^{\infty} f\{x(s)\}k(t - s)ds = y(t),$$

where $k(t)$ is the (absolutely-integrable) impulse response of a practical low-pass filter, so that the theory may be extended to signals that are merely bounded. Here one may not be interested, for various reasons, in the exact solution of this problem but, instead, a compromise problem, where the equation is nearly satisfied with both $x(t)$ and $y(t)$ being close to bandlimited functions. For example, in many cases $f\{x(t)\}$ is given (say) by an $n$th order differential operator acting on $y(t)$. Then the (exact) solution $x(t) = \phi\{f(x)\}$ may be far from a bandlimited function. However, if $y(t)$ is close to a bandlimited function there should be an approximate solution which is also close to a bandlimited function. A case in point is found in Landau's simulation of the iterative solution of the companding problem (Ref. 2), where, in fact, the equation he was obtaining approximate solutions to was the case $y(t) = k(t)$, (approximately bandlimited) for which the unique solution is (a multiple of) the Dirac delta function.

## X. ACKNOWLEDGMENT

I am grateful to H. J. Landau for helpful discussions, to J. C.

Lagarias for aid in preparation of the manuscript, and to A. M. Odlyzko for the production of Fig. 1.

## REFERENCES

1. H. J. Landau, "On the Recovery of a Band-Limited Signal, After Instantaneous Companding and Subsequent Band-Limiting," B.S.T.J., *39*, No. 2 (March 1960), pp. 351–64.
2. H. J. Landau and W. L. Miranker, "The Recovery of Band-Limited Signals," J. Math. Anal. and App., *2*, No. 1 (February 1961), pp. 97–104.
3. B. F. Logan, "Theory of Analytic Modulation Systems," B.S.T.J., *57*, No. 3 (March 1978), pp. 491–576.
4. B. F. Logan and M. R. Schroeder, "A Solution to Problem of Compatible Single-Sideband Transmission," IRE Trans. Info. Th., *IT-8*, No. 5 (September 1962), pp. 252–9.
5. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, New York: Dover, 1965.

## APPENDIX

Suppose $f(x)$ is a monotone increasing function of the real variable $x$, satisfying

$$(i) \qquad f(0) = 0$$

$$(ii) \qquad 0 < m_1 \leqslant f'(x) \leqslant m_2 < \infty, \qquad (-\infty < x < \infty).$$

Then $f$ has an inverse $\phi$

$$(iii) \qquad x = \phi\{f(x)\}, \qquad (-\infty < x < \infty),$$

satisfying, since $1 = f'(x)\phi'\{f(x)\}$,

$$(iv) \qquad 0 < \frac{1}{m_2} \leqslant \phi'(y) \leqslant \frac{1}{m_1} < \infty, \qquad (-\infty < y < \infty).$$

Now let $x = x(t)$ be any function in $\mathscr{B}_2$. We wish to establish

$$\| x - B\phi\{Bf(x)\}\|_2 \leqslant \gamma \| x \|_2, \tag{147}$$

where $$\gamma = \frac{\epsilon^2}{4(1 + \epsilon)}, \qquad \epsilon = \frac{m_2}{m_1} - 1.$$

Set

$$y(t) = y = Bf(x). \tag{148}$$

Then

$$f(x) = y + h, \qquad Bh = 0. \tag{149}$$

Now set

$$x_1 = B\phi(y) \tag{150}$$

so that, since $x = \phi(y + h) = Bx = B\{\phi(y + h)\}$,

$$x - x_1 = B\{\phi(y + h) - \phi(y)\}. \tag{151}$$

Since $|\phi(y+h)-\phi(y)| \leq \frac{1}{m_1} h$, we see that

$$\|\phi(y+h)-\phi(y)\|_2 \leq \frac{1}{m_1}\|h\|_2$$

and hence that

$$\|x-x_1\|_2 \leq \frac{1}{m_1}\|h\|_2,$$

but we can improve this inequality by writing

$$\phi(y+h)-\phi(y) = \int_y^{y+h} \{\phi'(\xi)-\alpha\}d\xi + \alpha h, \tag{152}$$

where

$$\alpha = \frac{1}{2}\left(\frac{1}{m_1}+\frac{1}{m_2}\right).$$

Then, since

$$|\phi'(\xi)-\alpha| \leq \frac{1}{2}\left(\frac{1}{m_1}-\frac{1}{m_2}\right), \tag{153}$$

we have

$$\phi(y+h)-\phi(y) = u + \alpha h, \tag{154}$$

where $u = u(t)$ and

$$|u| \leq \frac{1}{2}\left(\frac{1}{m_1}-\frac{1}{m_2}\right)|h|. \tag{155}$$

Thus

$$x - x_1 = B(u+\alpha h) = Bu, \tag{156}$$

and hence

$$\|x-x_1\|_2 \leq \|u\|_2 \leq \frac{1}{2}\left(\frac{1}{m_1}-\frac{1}{m_2}\right)\|h\|_2. \tag{157}$$

Now we need an inequality of the form $\|h\|_2 \leq c\|x\|_2$. We have

$$h = Hf(x), \tag{158}$$

where $H = I - B$ is the high-pass operator. So, clearly

$$\|h\|_2 \leq \|f(x)\|_2 \leq m_2\|x\|_2.$$

We can improve this inequality by setting

$$v(t) = v = f(x) - \beta x, \tag{159}$$

where

$$\beta = 1/2(m_1+m_2).$$

Then

$$|v| \leq 1/2(m_2 - m_1)|x|, \tag{160}$$

and, since $Hx = 0$,

$$h = Hf(x) = H\{f(x) - \beta x\} = Hv. \tag{161}$$

Hence

$$\|h\|_2 \leq \|v\|_2 \leq 1/2(m_2 - m_1)\|x\|_2. \tag{162}$$

This, with (157) gives

$$\|x - x_1\|_2 \leq \frac{1}{4}\left(\frac{1}{m_1} - \frac{1}{m_2}\right)(m_2 - m_1)\|x\|_2 = \gamma\|x\|_2, \tag{163}$$

which is the result (147). The number $\gamma$ in the inequality is invariant under the interchange of $\phi$ and $f$ in the approximate identity (147), as we would expect from using only (*ii*) and (*iv*).

So $x_1 = B\phi\{Bf(x)\}$ is a good approximation to $x$ if $\gamma$ is small. The manipulations leading to the inequality (163) suggest an iterative scheme for solving, given $y$ in $\mathscr{B}_2$,

$$BF(x) = y, \qquad x \text{ in } \mathscr{B}_2, \tag{164}$$

provided $\gamma < 1$, which will be the case if $(m_2/m_1) < 3 + 2\sqrt{2}$.

We set

$$x_n = B\phi(y + h_{n-1}), \qquad n \geq 1, \tag{165}$$

where

$$\mathrm{h}_n = \mathrm{Hf}(\mathrm{x}_n), \qquad \mathrm{n} \geq 0, \tag{166}$$

and $\mathrm{x}_0 = \mathrm{h}_0 = 0,$

giving $\mathrm{x}_1 = \mathrm{B}\phi(\mathrm{y})$ as in (150).

Now we wish to show that

$$\|x - x_n\|_2 \leq \gamma^n\|x\|_2, \qquad n \geq 1. \tag{167}$$

We have

$$x - x_n = B\{\phi(y + h) - \phi(y + h_{n-1})\}. \tag{168}$$

Following the previous pattern we write

$$\phi(y + h) - \phi(y + h_{n-1}) = u_n + \alpha(h - h_{n-1}), \tag{169}$$

where

$$u_n = \int_{y+h_{n-1}}^{y+h} \{\phi'(\xi) - \alpha\}d\xi,$$

and hence

$$|u_n| \leqslant \frac{1}{2}\left(\frac{1}{m_1} - \frac{1}{m_2}\right)|h - h_{n-1}|. \tag{170}$$

Then

$$x - x_n = B\{u_n + \alpha(h - h_{n-1})\} = Bu_n, \tag{171}$$

giving

$$\|x - x_n\|_2 \leqslant \frac{1}{2}\left(\frac{1}{m_1} - \frac{1}{m_2}\right)\|h - h_{n-1}\|_2. \tag{172}$$

Now

$$h - h_{n-1} = H\{f(x) - f(x_{n-1})\}, \quad n \geqslant 1, \quad \text{with } x_0 = h_0 = 0. \tag{173}$$

Here we write

$$\begin{aligned} f(x) - f(x_{n-1}) &= \int_{x_{n-1}}^{x} \{f'(\xi) - \beta\}d\xi + \beta(x - x_{n-1}) \\ &= v_n + \beta(x - x_{n-1}), \end{aligned} \tag{174}$$

where

$$|v_n| \leqslant \frac{m_2 - m_1}{2}|x - x_{n-1}|. \tag{175}$$

Then

$$h - h_{n-1} = H\{v_n + \beta(x - x_{n-1})\} = Hv_n, \tag{176}$$

giving, with (175),

$$\|h - h_{n-1}\|_2 \leqslant \frac{m_2 - m_1}{2}\|x - x_{n-1}\|_2. \tag{177}$$

This, with (172), gives

$$\|x - x_n\|_2 \leqslant \gamma\|x - x_{n-1}\|_2, \tag{178}$$

whence follows, with $x_0 = 0$,

$$\|x - x_n\|_2 \leqslant \gamma^n\|x\|_2, \qquad n \geqslant 1. \tag{179}$$

Note that there is a bonus attached to $x_1 = B\phi(y)$, in that only one filtering operation is required to obtain it. Thereafter, two filtering operations are required to obtain $x_n$ from $y$ and $x_{n-1}$.

## AUTHOR

**Benjamin F. Logan, Jr.**, B.S. (Electrical Engineering), 1946, Texas Technological College; M.S., 1951, Massachusetts Institute of Technology;

Eng.D.Sc. (Electrical Engineering), 1965, Columbia University; Bell Laboratories, 1956—. While at MIT, Mr. Logan was a research assistant in the Research Laboratory of Electronics, investigating characteristics of high-power electrical discharge lamps. Also at MIT he engaged in analog computer development at the Dynamic Analysis and Control Laboratory. From 1955 to 1956 he worked for Hycon-Eastern, Inc., where he was concerned with the design of airborne power supplies. He joined Bell Laboratories as a member of the Visual and Acoustics Research Department, where he was concerned with the processing of speech signals. Currently, he is a member of the Mathematical Research Department. Member, Sigma Xi, Tau Beta Pi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Bandwidth-Conserving Independent Amplitude and Phase Modulation

By B. F. LOGAN, Jr.*

(Manuscript received March 26, 1982)

Given two baseband signals $f(t)$ and $g(t)$, suitably restricted in amplitude and bandlimited to $[\lambda, \mu]$ and $[-\mu, -\lambda]$, $0 < \lambda < \mu < \infty$, it is shown how to generate a carrier signal, $s(t) = A(t)\cos\{ct + \phi(t)\}$, bandlimited to $[c - \beta, c + \beta]$ and $[-(c + \beta), -(c - \beta)]$, where $\beta$ need be only sightly larger than $\mu$, and such that $f(t)$ and $g(t)$ may be recovered by bandlimiting $\log A(t)$ and $\phi(t)$, respectively. The restriction $\lambda > 0$, i.e., that the baseband signals be bandpass, is not essential but it is a practical constraint in approximating the required operations. Also a modification is given for conserving bandwidth in case the signals $f(t)$ and $g(t)$ are of disparate bandwidths.

## I. INTRODUCTION

Double-sideband amplitude modulation is wasteful of bandwidth, but it offers the advantage of envelope detection (with full carrier). A simple way to utilize the same bandwidth in transmitting two independent signals, $f(t)$ and $g(t)$, is the so-called in-phase and quadrature modulation

$$S_1(t) = f(t)\cos ct - g(t)\sin ct,$$

where synchronous demodulation is required to recover $f$ and $g$. A modification that allows $f$ to be recovered (approximately) by envelope detection is

$$S_2(t) = \{1 + f(t)\}\cos ct - g(t)\sin ct.$$

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

The envelope of $S_2(t)$ is

$$A_2(t) = \sqrt{\{1 + f(t)\}^2 + g^2(t)}.$$

Then if $g$ is made small (compared to $\min\{1 + f(t)\}$), we have

$$A_2(t) \cong 1 + f(t).$$

The phase of $S_2$ (i.e., the part due to signals) is

$$\phi_2(t) = \tan^{-1} \frac{g(t)}{1 + f(t)} \cong \frac{g(t)}{1 + f(t)}.$$

So making $g$ small allows envelope and phase detection to be used so as to approximately recover $f$ and $g$ (multiplying the phase output by the envelope output).

A still further modification is

$$S_3(t) = \{1 + x_1(t)\}\cos ct - \{1 + x_2(t)\}\sin ct,$$

where $x_1$ and $x_2$ are both small. The envelope of $S_3$ is

$$A_3(t) = \sqrt{(1 + x_1)^2 + (1 + x_2)^2} = \sqrt{2 + 2x_1 + 2x_2 + x_1^2 + x_2^2}$$
$$\cong \sqrt{2}\left(1 + \frac{x_1 + x_2}{2}\right).$$

The phase of $S_3$ is

$$\phi_3(t) = \tan^{-1} \frac{1 + x_2}{1 + x_1} \cong \frac{\pi}{4} + \frac{x_2 - x_1}{2}.$$

So if

$$\frac{x_1 + x_2}{2} = f \quad \text{and} \quad \frac{x_1 - x_2}{2} = g,$$

i.e.,

$$x_1 = f + g$$
$$x_2 = f - g,$$

then envelope and phase detection of $S_3$ will give (approx.) the desired independent signals $f$ and $g$.

An exact result of this type may be obtained using log of the envelope, rather than the envelope, and then *bandlimiting* the phase and log of the envelope to obtain the desired independent signals $f$ and $g$. A slight increase in bandwidth is required to allow a guard band in the filtering operations. Also $|f|$ and $|g|$ cannot be too large if the increase in bandwidth is to be small. The basic theory is that of

Exponential Single-Sideband Modulation (ESSB) developed in Ref. 1.

## II. THE EXACT METHOD

We assume that the desired signals, $f$ and $g$, are bounded band-pass signals whose Fourier transforms vanish (in the sense detailed in Ref. 1) outside $[\lambda, \mu]$ and $[-\mu, -\lambda]$, $0 < \lambda < \mu < \infty$, which then (automatically) have bounded Hilbert transforms, $\hat{f}$ and $\hat{g}$. The band-pass assumption is not essential to the theory, but affords important practical simplifications in approximating the Hilbert transform operations as well as in effecting the subsequent analytic exponential modulation.

Now suppose $z_1(t)$ and $z_2(t)$ are bandlimited "analytic signals" whose Fourier transforms vanish outside $[0, \beta]$ and which signals are zero-free in the upper half-plane with

$$|z_{1,2}(t + iu)| \geq \epsilon \quad \text{for} \quad u \geq 0, \qquad -\infty < t < \infty. \tag{1}$$

Then $\log z_1$ and $\log z_2$ are analytic and bounded in the upper half-plane, and hence their Fourier transforms vanish over $(-\infty, 0)$.

Writing

$$z_1(t) = A_1(t)e^{i\phi_1(t)}, \qquad A_1 = |z_1| \tag{2}$$

$$z_2(t) = A_2(t)e^{i\phi_2(t)}, \qquad A_2 = |z_2|, \tag{3}$$

we have

$$\log z_1(t) = \log A_1(t) + i\phi_1(t) \tag{4}$$

$$\log z_2(t) = \log A_2(t) + i\phi_2(t). \tag{5}$$

Under further assumptions on $z$, e.g.,

$$z_{1,2}(t + iu) = 1 + 0(e^{-\lambda u}), \qquad u \to \infty, \tag{6}$$

$\log A$ and $\phi$ will be Hilbert transform pairs:

$$\phi_1(t) = \log^{\wedge} A_1(t), \qquad \log A_1(t) = -\hat{\phi}_1(t) \tag{7}$$

$$\phi_2(t) = \log^{\wedge} A_2(t) \qquad \log A_2(t) = -\hat{\phi}_2(t). \tag{8}$$

Now we consider the product

$$z_1(t)\overline{z_2(t)} = A_1(t)A_2(t)e^{i[\phi_1(t)-\phi_2(t)]},$$

where the bar denotes the complex conjugate. The F.T. (Fourier transform) of $z_1(t)$ vanishes outside $[0, \beta]$ and the F.T. of $\overline{z_2(t)}$ vanishes outside $[-\beta, 0]$. Therefore, the F.T. of $z_1(t)\overline{z_2(t)}$ vanishes outside $[-\beta, \beta]$. Then we form the signal

$$s(t) = \text{Re } e^{ict} z_1(t)\overline{z_2(t)}$$
$$= A(t)\cos\{ct + \phi(t)\}, \tag{9}$$

where $c > \beta$,

$$A(t) = A_1(t)A_2(t) \tag{9a}$$

$$\phi(t) = \phi_1(t) - \phi_2(t), \tag{9b}$$

and the spectrum of $s(t)$ is confined to $[c - \beta, c + \beta]$ and $[-c - \beta, -c + \beta]$.

We require

$$\mathbf{B}_{\mu,\alpha}\{\log A(t)\} = f(t) \tag{10}$$

$$\mathbf{B}_{\mu,\alpha}\{\phi(t)\} = g(t), \tag{11}$$

where $\mu < \alpha < \beta$, and, in general, $\mathbf{B}_{p,q}$ is any bandlimiting operator [with passband $(-p, p)$ and cut-off frequency $\pm q$] defined by

$$\mathbf{B}_{p,q}\{x(t)\} = \int_{-\infty}^{\infty} x(s)K_{p,q}(t - s)ds \tag{12}$$

and

$$\overset{>}{K}_{p,q}(\omega) = \int_{-\infty}^{\infty} K_{p,q}(t)e^{-i\omega t}dt = 1, \qquad -p < \omega < p$$
$$= 0, \qquad |\omega| > q. \tag{12a}$$
$$0 < p < q < \infty. \tag{12b}$$

The definition of $\overset{>}{K}_{p,q}(\omega)$ in the cut-off region $(p, q)$ and $(-q, -p)$ is not important, but $\overset{>}{K}_{p,q}(\omega)$ must be sufficiently smooth to give

$$\int_{-\infty}^{\infty} |K_{p,q}(t)|\,dt < \infty \tag{12c}$$

so that $\mathbf{B}_{p,q}\{x(t)\}$ is defined for any bounded $x(t)$.

Writing (10) as

$$\mathbf{B}_{\mu,\alpha}\{\log A_1(t) + \log A_2(t)\} = f(t)$$

and taking Hilbert transforms of both sides of (10) and (11), using (7) and (8), we have

$$\mathbf{B}_{\mu,\alpha}\{\phi_1(t) + \phi_2(t)\} = \hat{f}(t) \tag{13}$$

$$\mathbf{B}_{\mu,\alpha}\{\phi_1(t) - \phi_2(t)\} = g(t) \tag{14}$$

or

$$\mathbf{B}_{\mu,\alpha}\{\phi_1(t)\} = \tfrac{1}{2}\{\hat{f}(t) + g(t)\} \tag{15}$$

$$\mathbf{B}_{\mu,\alpha}\{\phi_2(t)\} = \tfrac{1}{2}\{\hat{f}(t) - g(t)\}, \tag{16}$$

which according to (7) and (8) is equivalent to

$$\mathbf{B}_{\mu,\alpha}\{\log A_1(t)\} = \tfrac{1}{2}\{f(t) - \hat{g}(t)\} \tag{17}$$

$$\mathbf{B}_{\mu,\alpha}\{\log A_2(t)\} = \tfrac{1}{2}\{f(t) + \hat{g}(t)\}. \tag{18}$$

Setting

$$h_1(t) = \tfrac{1}{2}\{f(t) - \hat{g}(t)\}; \qquad \hat{h}_1(t) = \tfrac{1}{2}\{\hat{f}(t) + g(t)\}, \tag{19}$$

$$H_1(t) = h_1(t) + i\hat{h}_1(t), \tag{19a}$$

$$h_2(t) = \tfrac{1}{2}\{f(t) + \hat{g}(t)\}; \qquad \hat{h}_2(t) = \tfrac{1}{2}\{\hat{f}(t) - g(t)\}, \tag{20}$$

$$H_2(t) = h_2(t) + i\hat{h}_2(t), \tag{20a}$$

the four equations (15), (16), (17), and (18) are equivalent to the two equations, implying (6),

$$\mathbf{B}_{\mu,\alpha}\{\log z_1(t)\} = H_1(t) \tag{21}$$

$$\mathbf{B}_{\mu,\alpha}\{\log z_2(t)\} = H_2(t), \tag{22}$$

where $H_1$ and $H_2$ are given "analytic" band-pass signals whose Fourier transforms vanish outside the single interval $[\lambda, \mu]$ and $z_1$ and $z_2$ are bandlimited "analytic" signals whose Fourier transforms vanish outside the single interval $[0, \beta]$. The problem of finding $z_1$ and $z_2$ has been solved (see Ref. 1):

$$z_1(t) = \mathbf{B}_{\alpha,\beta}\{\exp H_1(t)\} \tag{23}$$

$$z_2(t) = \mathbf{B}_{\alpha,\beta}\{\exp H_2(t)\}, \tag{24}$$

where $\mathbf{B}_{\alpha,\beta}$ is any bandlimiting operator with passband $(-\alpha, \alpha)$ and cut-off frequency $\pm\beta$.

Now $z_1$ and $z_2$ given by (23) and (24) satisfy (21) and (22), *provided* $z_1(\tau)$ and $z_2(\tau)$, $\tau = t + iu$, are zero free in the upper half-plane $u > 0$. The filter characteristic $\overset{>}{K}_{\alpha,\beta}(\omega)$ in the cut-off region $(\alpha, \beta)$ becomes important, but not critical, in this respect. From theoretical considerations the linear cut-off characteristic is desirable (see Ref. 1):

$$\overset{>}{K}_{\alpha,\beta}(\omega) = \frac{\beta - \omega}{\beta - \alpha}, \qquad \alpha < \omega < \beta. \tag{25}$$

For a given $\alpha$ and $\beta$, and a smooth cut-off characteristic, $z_1$ and $z_2$ will be zero free in the upper half-plane provided $|H_1|$ and $|H_2|$ are not too large. In practice this means that the levels of $f$ and $g$ must not be

too large if $\alpha$ and $\beta$ are not much larger than $\mu$, the top signal frequency, i.e., in the bandwidth conserving case. The results in Ref. 1 may be used as a rough guide. For example, if $\alpha$ is only slightly larger than $\mu$ and $\beta/\alpha = 1.1$ (relatively sharp cut-off), then $z_1$ and $z_2$ will be zero free in the upper half-plane if $|H_1|$ and $|H_2|$ are less than 0.6. (See the appendix for a modification of signals $f$ and $g$ of disparate bandwidths.)

## III. IMPLEMENTATION

The block diagram of an implementation is shown in Fig. 1. The transmitter is shown in Fig. 1a. The inputs are labeled $f(t + T)$ and $g(t + T)$ to account for a delay $T$ incurred in the Hilbert transform filters. The delay $T$ need not be more than one or two periods of the lower signal frequency $\lambda$ to obtain a good approximation to the Hilbert transforms, $\hat{f}(t)$ and $\hat{g}(t)$. (The inputs $f(t + T)$ and $g(t + T)$ must be delayed accordingly to obtain $f(t)$ and $g(t)$.) The signals $f(t)$, $\hat{f}(t)$, $g(t)$, and $\hat{g}(t)$ are summed to obtain

$$h_1 = \tfrac{1}{2}(f - \hat{g})$$

$$\hat{h}_1 = \tfrac{1}{2}(\hat{f} + g)$$

$$h_2 = \tfrac{1}{2}(f + \hat{g})$$

$$\hat{h}_2 = \tfrac{1}{2}(\hat{f} - g)$$

in accord with (19) and (20). (The gain factor of the summing networks, shown as 1/2, may be any constant, which may be simply reflected as a gain factor on the inputs.) Then these outputs are fed to two analytic exponential modulators that furnish outputs

$$X_1 = e^{h_1}\cos \hat{h}_1 = \mathrm{Re}\{\exp H_1\}$$

$$Y_1 = e^{h_1}\sin \hat{h}_1 = \mathrm{Im}\{\exp H_1\}$$

$$X_2 = e^{h_2}\cos \hat{h}_2 = \mathrm{Re}\{\exp H_2\}$$

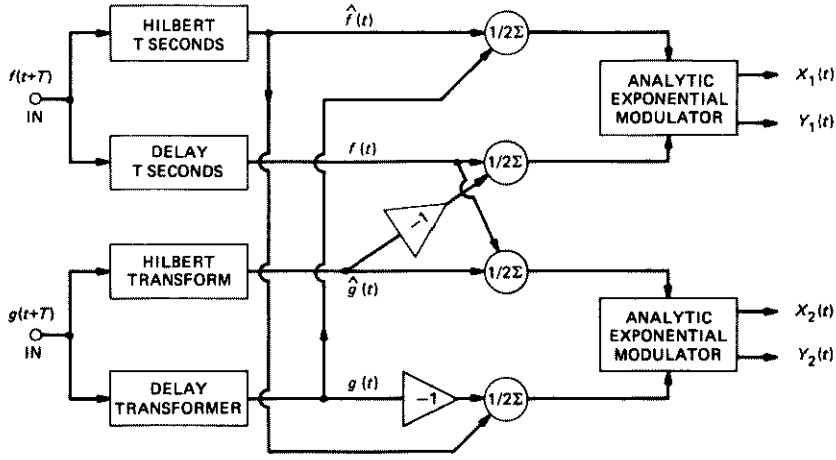
$$Y_2 = e^{h_2}\sin \hat{h}_2 = \mathrm{Im}\{\exp H_2\}.$$

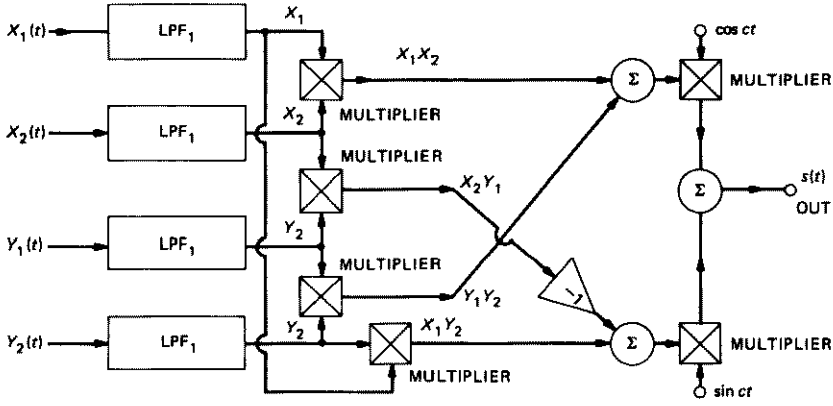
A feedback circuit for accomplishing the analytic exponential modulation is described in Ref. 2. The outputs of the modulators are then bandlimited with identical low-pass filters $\mathrm{LPF}_1$ having the characteristic shown in Fig. 2a to obtain

$$x_1 = \mathrm{Re}\{z_1\}, \qquad y_1 = \mathrm{Im}\{z_1\}$$

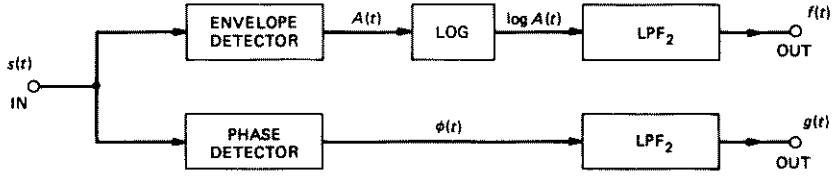
$$x_2 = \mathrm{Re}\{z_2\}, \qquad y_2 = \mathrm{Im}\{z_2\}.$$

These outputs are then combined to form




(a)




(b)

Fig. 1a—Transmitter.
Fig. 1b—Receiver.

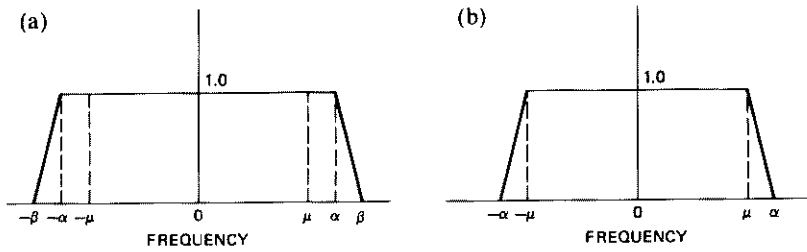



Fig. 2a—Characteristic of $LPF_1$.

Fig. 2b—Characteristic of $LPF_2$.

$$\begin{aligned} s(t) &= \mathrm{Re}\{[x_1(t) + iy_1(t)][x_2(t) - iy_2(t)]e^{ict}\} \\ &= \mathrm{Re}\{[x_1x_2 + y_1y_2 + i(y_1x_2 - y_2x_1)]e^{ict}\} \\ &= (x_1x_2 + y_1y_2)\cos ct - (y_1x_2 - y_2x_1)\sin ct \\ &= A(t)\cos\{ct + \phi(t)\}. \end{aligned}$$

The signal $s(t)$ is then transmitted to the receiver, Fig. 1b, where an envelope detector is used to obtain the envelope $A(t)$, which is then fed to a device having a logarithmic characteristic to furnish the output $\log A(t)$. This output is then filtered with $LPF_2$ to obtain $f(t)$. A phase detector, e.g., a phase-locked loop, is used to detect the phase $\phi(t)$, which is subsequently filtered with another $LPF_2$ to obtain $g(t)$. The characteristic of the filters $LPF_2$ is shown in Fig. 2b.

Note that $\phi(t)$ is high pass with lower frequency $\lambda$; so $\phi(t)$ may be recovered from $\{\phi'(t) + c\}$, if desired.

## REFERENCES

1. B. F. Logan, "Theory of Analytic Modulation Systems," B.S.T.J., 57, No. 3 (March 1978), pp. 491–576.
2. B. F. Logan, "Click Modulation," unpublished work.

## APPENDIX

### *Modification for Signals of Disparate Bandwidths*

Note that the bandwidth of the transmitted signal is the sum (or twice the sum, counting positive and negative frequencies) of the bandwidths of the analytic signals $z_1(t)$ and $z_2(t)$, which need be only slightly larger than the sum of the bandwidths of the analytic signals $H_1(t)$ and $H_2(t)$. Owing to the linear combinations in (19) and (20), the bandwidths of $H_1(t)$ and $H_2(t)$ will be the same, equal to the larger of the bandwidths of $f(t)$ and $g(t)$. In case the bandwidth of, say, $g(t)$ is (considerably) larger than that of $f(t)$, the bandwidth of the transmitted signal may be reduced by setting

$$H_1(t) = f(t) + i\hat{f}(t) \tag{26}$$

$$H_2(t) = g(t) + i\hat{g}(t). \tag{27}$$

Here we assume that the Fourier transforms of $H_1(t)$ and $H_2(t)$ vanish outside the intervals $[\lambda, \mu_1]$ and $[\lambda_2, \mu_2]$, respectively. Now we set

$$z_1(t) = \mathbf{B}_{\alpha_1,\beta_1}\{\exp H_1(t)\}, \qquad \mu_1 < \alpha_1 < \beta_1 \tag{28}$$

$$z_2(t) = \mathbf{B}_{\alpha_2,\beta_2}\{\exp H_2(t)\}, \qquad \mu_2 < \alpha_2 < \beta_2, \tag{29}$$

where $\beta_1$ and $\beta_2$ need be only slightly larger than $\mu_1$ and $\mu_2$ (respectively), the top frequencies of $f(t)$ and $g(t)$ (respectively). The Fourier transform of $z_1(t)\overline{z_2(t)}$ now vanishes outside the interval $[-\beta_2, \beta_1]$, which is smaller than would obtain in the previous scheme. Thus the Fourier transform of the transmitted signal,

$$s(t) = \operatorname{Re} e^{ict} z_1(t)\overline{z_2(t)}, \qquad c > \beta_2 \tag{30}$$

vanishes outside the interval $[c - \beta_2, c + \beta_1]$ (and its reflection about the origin). The price paid for the saving in bandwidth is another Hilbert transform operation required in separating the signals at the receiver.

We have

$$s(t) = A(t)\cos[ct + \phi(t)], \tag{31}$$

where

$$A(t) = |z_1(t)z_2(t)|$$

$$\phi(t) = \phi_1(t) - \phi_2(t).$$

Then, assuming as before that $z_1$ and $z_2$ are zero free in the upper half-plane, we have

$$L(t) = \log A(t) = L_1(t) + L_2(t), \tag{32}$$

where $L_1(t) = \log |z_1(t)|$, $L_2(t) = \log |z_2(t)|$ and $L(t)$ is related to $\phi(t)$ by

$$\phi(t) = \phi_1(t) - \phi_2(t) = \hat{L}_1(t) - \hat{L}_2(t) \tag{33}$$

$$\hat{\phi}(t) = \hat{\phi}_1(t) - \hat{\phi}_2(t) = -L_1(t) + L_2(t). \tag{34}$$

In accord with (28) and (29) and the zero-free hypothesis, we have (as shown in Ref. 1)

$$\mathbf{B}_{\mu_1,\alpha_1}\{L_1(t)\} = f(t) \tag{35}$$

$$\mathbf{B}_{\mu_2,\alpha_2}\{L_2(t)\} = g(t) \tag{36}$$

To obtain $L_1(t)$ and $L_2(t)$ from $\log A(t)$ and $\phi(t)$, we need the Hilbert transform $\hat{\phi}(t)$, where according to (32) and (34),

$$L_1(t) = \tfrac{1}{2} \log A(t) - \tfrac{1}{2}\hat{\phi}(t) \tag{37}$$

$$L_2(t) = \tfrac{1}{2} \log A(t) + \tfrac{1}{2}\hat{\phi}(t). \tag{38}$$

However, to recover $f(t)$ and $g(t)$, we may use a modified version of $\hat{\phi}(t)$. We define

$$\hat{\phi}_{\lambda,\mu}(t) = \mathbf{H}_{\lambda,\mu}\{\phi(t)\}, \tag{39}$$

where $\mathbf{H}_{\lambda,\mu}$ is a modified (e.g., band-pass) Hilbert transform operator defined by

$$\mathbf{H}_{\lambda,\mu}\{\phi(t)\} = \int_{-\infty}^{\infty} h_{\lambda,\mu}(t - x)\phi(x)dx \tag{40a}$$

$$h_{\lambda,\mu}(\omega) = \int_{-\infty}^{\infty} h_{\lambda,\mu}(t)e^{i\omega t}dt = -i \operatorname{sgn} \omega, \tag{40b}$$

$$\text{for} \quad 0 < \lambda \leqslant |\omega| \leqslant \mu.$$

Now $\phi_1(t)$ and $\phi_2(t)$ are high-pass functions with lower frequencies $\lambda_1$ and $\lambda_2$ (respectively). Thus, if we require

$$0 < \lambda \leqslant \min(\lambda_1, \lambda_2), \qquad \mu \geqslant \max(\mu_1, \mu_2), \tag{41}$$

then we have

$$f(t) = \tfrac{1}{2}\mathbf{B}_{\mu_1,\alpha_1}\{\log A(t) - \hat{\phi}_{\lambda,\mu}(t)\} \tag{42}$$

$$g(t) = \tfrac{1}{2}\mathbf{B}_{\mu_2,\alpha_2}\{\log A(t) + \hat{\phi}_{\lambda,\mu}(t)\}. \tag{43}$$

## AUTHOR

**Benjamin F. Logan, Jr.,** B.S. (Electrical Engineering), 1946, Texas Technological College; M.S., 1951, Massachusetts Institute of Technology; Eng.D.Sc. (Electrical Engineering), 1965, Columbia University; Bell Laboratories, 1956—. While at MIT, Mr. Logan was a research assistant in the Research Laboratory of Electronics, investigating characteristics of high-power electrical discharge lamps. Also at MIT he engaged in analog computer development at the Dynamic Analysis and Control Laboratory. From 1955 to 1956 he worked for Hycon-Eastern, Inc., where he was concerned with the design of airborne power supplies. He joined Bell Laboratories as a member of the Visual and Acoustics Research Department, where he was concerned with the processing of speech signals. Currently, he is a member of the Mathematical Research Department. Member, Sigma Xi, Tau Beta Pi.

# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

Beeri C., Fagin R., Maier D., Yannakakis M., **On the Desirability of Acyclic Database Schemes.** J ACM 30(3):479–513, 1983.

Cargill T. A., **A Robust Distributed Solution to the Dining Philosophers Problem—Response (Letter)** Software 13(6):549, 1983.

Coffman, E. G., **An Introduction to Combinatorial Models of Dynamic Storage-Allocation.** SIAM Rev 25(3):311–325, 1983.

Coffman, E. G., Sethi R., **Instruction Sets for Evaluating Arthimetic Expressions.** J ACM 30(3):457–478, 1983.

Du D. Z., Hwang F. K., **A New Bound for the Steiner Ratio.** T Am Math S 278(1):137–148, 1983.

Du D. Z., Hwang F. K., Weng J. F., **Steiner Minimal-Trees on Zig-Zag Lines.** T Am Math S 278(1):149–156, 1983.

Gehani N. H., **An Electronic Form System—An Experience in Prototyping.** Software 13(6):479–486, 1983.

Heck W. S., **Large Families of Incomparable *A*-Isols.** J Symb Log 48(2):250–252, 1983.

Hwang F. K., Weng J. F., Du D. Z., **A Class of Full Steiner Minimal-Trees.** Discr Math 45(1):107–112, 1983.

Lagarias J. C., **Sets of Primes Determined by Systems of Polynomial Congruences.** Ill J Math 27(2):224–239, 1983.

Liebesma B. S., Saperstein B., **A Proposed Attribute Skip-Lot Sampling Program (Review).** J Qual Tech 15(3):130–140, 1983.

Matula R. A., **The Need for an Online Compilation of Compilations.** Online 7(4):61–63, 1983.

Murphy B. T., **Microcomputers—Trends, Technologies, and Design Strategies.** IEEE J Soli 18(3):236–244, 1983.

Ramsey H. R., Atwood M. E., Vandoren J. R., **Flowcharts Versus Program Design Languages—An Experimental Comparison.** Comm ACM 26(6):445–449, 1983.

Raoult J. C., Sethi R., **Properties of a Notation for Combining Functions.** J ACM 30(3):595–611, 1983.

Sleator D. D., Tarjan R. E., **A Data Structure For Dynamic Trees.** J Comput Syst 26(3):362–391, 1983.

Slepian D., **Some Comments on Fourier-Analysis, Uncertainty and Modeling.** Siam Rev 25(3):379–393, 1983.

Tarjan R. E., **Citation Classic—Depth-1st Search and Linear Graph Algorithms.** CC/Eng Tech 1983(30):20, 1983.

Venkataraman K. N., Yasuhara A., Hawrusik F. M., **A View of Computability on Term Algebras.** J Comput Sy 26(3):410–471, 1983.

## ENGINEERING

Agrawal G. P., Joyce W. B., Dixon R. W., Lax M., **Beam-Propagation Analysis of Stripe-Geometry Semiconductor-Lasers—Threshold Behavior.** Appl Phys L 43(1):11–13, 1983.

Anthony P. J., Pawlik J. R., Swaminathan V., Tsang W. T., **Reduced Threshold Current Temperature-Dependence in Double Heterostructure Lasers Due to Separate P-N and Heterojunctions.** IEEE J Q El 19(6):1030–1035, 1983.

Antler M., **Effect of Lubricants on Frictional Polymerization of Palladium Electrical Contacts.** ASLE Trans 26(3):376–380, 1983.

Auston D. H., Freeman R. R., Smith P. R., Mills D. M., Siemann R. H., **High-Speed X-Ray Sensitive Photoconducting Detector.** Appl Phys L 42(12):1050–1052, 1983.

Beni G., Hackwood S., **Robotic Electroplating of Gold on Quaternary Semiconductors.** J Appl Elec 13(4):531–533, 1983.

Bokor J., Eichner L., Storz R. H., Bucksbaum P. H., Freeman R. R., **Wavelength Conversion With Excimer Lasers.** AIP Conf PR1983(100):143–150, 1983.

Bowers J. E., Coldren L. A., Hemenway B. R., Miller B. I., Martin R. J., **1.55 μm Multisection Ridge Lasers.** Electr Lett 19(14):523–525, 1983.

Campbell J. C., Dentai A. G., Qua G. J., Ferguson J. F., **Avalanche InP/InGaAs Heterojunction Photo-Transistor.** IEEE J Q El 19(6):1134–1138, 1983.

Capasso F., **Band-Gap Engineering Via Graded Gap, Superlattice, and Periodic Doping Structures: Applications to Novel Photodetectors and Other Devices.** J Vac Sci B 1(2):457–461, 1983.

Chen C. Y., Cho A. Y., Bethea C. G., Garbinski P. A., Pang Y. M., Levine B. F., Ogawa K., **Ultrahigh Speed Modulation-Doped Heterostructure Field-Effect Photodetectors.** Appl Phys L 42(12):1040–1042, 1983.

Cheng J., Thurston R. N., Boyd G. D., Meyer R. B., **A New Low-Voltage Electrically-Addressed Bistable Nematic Liquid-Crystal Boundary-Layer Display.** IEEE Device 30(5):520–525, 1983.

Coldren L. A., Ebeling K. J., Miller B. I., Rentschler J. A., **Single Longitudinal Mode-Operation of Two-Section GaLnAsP/InP Lasers Under Pulsed Excitation.** IEEE J Q El 19(6):1057–1062, 1983.

Cox D. C., **Antenna Diversity Performance in Mitigating the Effects of Portable Radiotelephone Orientation and Multipath Propagation.** IEEE Commun 31(5):620–628, 1983.

Dvorak C. A., **Text Sharpness, Its Components and Text Quality.** J Appl Phot 9(3):109–111, 1983.

Dvorak C. A., Hamerly J. R. **Just-Noticeable Differences for Text Quality Components.** J Appl Phot 9(3):97–100, 1983.

Finegan S. N., Swartz R. G., McFee J. H., **A UHV-Compatible Round Wafer Heater for Silicon Molecular-Beam Epitaxy.** J Vac Sci B 1(2):497–500, 1983.

Geyling F. T., Walker K. L., Csencsits R., **The Viscous Collapse of Thick-Walled Tubes.** J Appl Mech 50(2):303–310, 1983.

Gilbert J. A. et al., **The Monomode Fiber—A New Tool for Holographic-Interferometry.** Exp Mech 23(2):190–195, 1983.

Henry C. H., Levine B. F., Logan R. A., Bethea C. G., **Minority-Carrier Lifetime and Luminescence Efficiency of 1.3-μm InGaAsP-InP Double Heterostructure Layers.** IEEE J Q El 19(6):905–912, 1983.

Henry C. H., Logan R. A., Merritt F. R., Luongo J. P., **The Effect of Intervalence Band Absorption on the Thermal-Behavior of InGaAsP Lasers.** IEEE J Q El 19(6):947–952, 1983.

Henry C. H., Logan R. A., Temkin H., Merritt F. R., **Absorption, Emission, and Gain Spectra of 1.3-μm InGaAsP Quaternary Lasers.** IEEE J Q El 19(6):941–946, 1983.

Howard R. E., Liao P. F., Skocpol W. J., Jackel L. D., Craighead H. G., **Microfabrication as a Scientific Tool.** Science 221(4606):117–121, 1983.

Johnson L. F., Ingersoll K. A., Dalton J. V., **Planarizing of Phosphosilicate Glass-Films on Patterned Silicon Wafers.** J Vac Sci B 1(2):487–489, 1983.

Kmetz A. R., Penz P. A., **Joint Special Issue on Displays—Foreword (Editorial).** IEEE Device 30(5):429–430, 1983.

Meloni A., Lanzerotti L. J., Gregori G. P., **Induction of Currents in Long Submarine Cables by Natural Phenomena (Review).** Rev Geophys 21(4):795–803, 1983.

Miller B., Rosamilia J. M., **Hydrodynamically Modulated Rotating-Disk Electrode Analysis in Derivative Mode (Review).** Analyt Chem 55(8):1281–1285, 1983.

Miller D. A. B., **Dynamic Non-Linear Optics in Semiconductors—Physics and Applications.** Laser Focus 19(7):61+, 1983.

Nygren S. F., **A Non-Destructive Method for Predicting Laser-Emission Wavelength From Photocurrent Spectra of GaAlAs Double Heterostructure Wafers.** IEEE J Q El 19(6):898–905, 1983.

Phillips J. C., **Realization of a Zachariasen Glass.** Sol St Comm 47(3):203–206, 1983.

Phillips J. C., **Why Localized and Extended Impurity Band States Can Coexist and Be Separated.** Sol St Comm 47(3):191–193, 1983.

Pini R. et al., **Ultraviolet Stimulated Raman-Scattering in Multimode Silica Fibers Pumped with Excimer Lasers.** Appl Phys L 43(1):6–8, 1983.

Smith J. S., Chiu L. C., Margalit S., Yariv A., Cho A. Y., **A New Infrared Detector Using Electron-Emission.** J Vac Sci B 1(2):376–378, 1983.

Sorace R., **Trellis Coding for a Multiple-Access Channel (Letter).** IEEE Info T 29(4):606–611, 1983.

Spencer E. G., **Citation Classic—Dielectric Materials for Electrooptic, Elastooptic, and Ultrasonic Device Applications.** CC/Eng Tech1983(31):22, 1983.

Tomita A., **Cross Talk Caused by Stimulated Raman-Scattering In Single-Mode Wavelength-Division Multiplexed Systems.** Optics Lett 8(7):412–414, 1983.

Tsang W. T., Olsson N. A., Logan R. A., **Stable Single-Longitudinal-Mode Operation Under High-Speed Direct Modulation in Cleaved-Coupled-Cavity GaInAsP Semiconductor-Lasers.** Electr Lett 19(13):488–490, 1983.

Tsang W. T., Olsson N. A., Logan R. A., **Threshold-Wavelength and Threshold-Temperature Dependences of GaInAsP-InP Lasers with Frequency Selective Feedback Operating in the 1.3-$\mu$m and 1.5-$\mu$m Regions.** Appl Phys L 43(2):154–156, 1983.

Tucker R. S., Eisenstein G., Kaminow I. P., **10 GHz Active Mode-Locking of a 1.3-$\mu$m Ridge-Waveguide Laser in an Optical-Fibre Cavity.** Electr Lett 19(14):552–553, 1983.

Weschler C. J., Kelty S. P., Lingousky J. F., **The Effect of Building Fan Operation on Indoor-Outdoor Dust Relationships.** J Air Pollu 33(6):624–628, 1983.

White J. C., Henderson D., **Tuning and Saturation Behavior of the Anti-Stokes Raman Laser.** AIP Conf PR1983(100):121–127, 1983.

White J. C., Miller T. A., Heaven M., **Anti-Stokes Raman-Scattering of Excimer and Tunable UV Lasers in Chemically Pumped CO.** AIP Conf PR1983(100):195–199, 1983.

Wilson T., Boyd G. D., Thurston R. N., Cheng J., Storz F. G., Westerwick E. H., **A Matrix Addressable Bistable Nematic Liquid-Crystal Display with Electric-Field Writing and Thermal Erasure.** IEEE Device 30(5):513–520, 1983.

Wilson T. G., Whelan E. W., Rodriguez R., Dishman J. M., **DC-to-DC-Converter Power-Train Optimization for Maximum Efficiency.** IEEE Aer El 19(3):413–427, 1983.

## MANAGEMENT/ECONOMICS

Gordon R. H., Slemrod J., **A General Equilibrium Simulation Study of Subsidies to Municipal Expenditures.** J Finance 38(2):585–594, 1983.

Heiser W. J., Meulman J., **Analyzing Rectangular Tables by Joint and Constrained Multidimensional-Scaling.** J Economet 22(1–2):139–167, 1983.

Hill S. et al., **Putting a Lid on Pension Costs (Editorial).** Inst Invest 17(6):87+, 1983.

## PHYSICAL SCIENCES

Agrawal G. P., **Nonperturbative Analysis of Zeeman-Coherence Effects on Resonant Phase Conjugation.** Optics Lett 8(7):359–361, 1983.

Alavi K., Petroff P. M., Wagner W. R., Cho A. Y., **Substrate Rotation-Induced Compositional Oscillation in Molecular-Beam Epitaxy (MBE).** J Vac Sci B 1(2):146–148, 1983.

Anderson P. W. et al., **Theory of the Universal Degradation of Tc in High-Temperature Superconductors.** Phys Rev B 28(1):117–120, 1983.

Ballman A. A., Glass A. M., Nahory R. E., Brown H., **Double Doped Low Etch Pit Density InP With Reduced Optical-Absorption.** J Cryst Gr 62(1):198–202, 1983.

Bates F. S., Baker G. L., **Polyacetylene Single-Crystals (Letter).** Macromolec 16(6):1013–1015, 1983.

Becker G. E., Cardillo M. J., Serri J. A., Hamann D. R., **He-Ag {001} – *C* (2×2)Cl Attractive Potential From Resonance Scattering.** Phys Rev B 28(2):504–514, 1983.

Blasie J. K. et al., **The Location of Redox Centers in the Profile Structure of a Reconstituted Membrane Containing a Photosynthetic Reaction Center-Cytochrome-C Complex by Resonance X-Ray-Diffraction.** Bioc Biop A 723(3):350–357, 1983.

Brasen D., Karlicek R. F., Donnelly V. M., **TEM Observations of Laser-Induced Pt and Au Deposition on InP.** J Elchem So 130(7):1473–1475, 1983.

Brillson L. J., Shapira Y., Heller A., **InP Surface-States and Reduced Surface Recombination Velocity.** Appl Phys L 43(2): 174–176, 1983.

Brummell M. A., Nicholas R. J., Portal J. C., Cheng K. Y., Cho A. Y., **Two-Dimensional Magnetophonon Resonance 2. GaInAs-AlInAs Heterojunctions (Letter).** J Phys C 16(17):L579–L584, 1983.

Bucksbaum P. H., Bokor J., **Time Resolved Amorphous-Silicon Formation From Laser Melted Liquid Silicon Films.** AIP Conf PR1983(100):279–287, 1983.

Cardillo M. J., Becker G. E., Hamann D. R., Serri J. A., Whitman L., Mattheiss L. F., **Geometry of the Ag{001} – *C* (2×2)Cl Structure as Determined by He Diffraction.** Phys Rev B 28(2):494–503, 1983.

Cassanho A., Guggenheim H., Walstedt R. E., **Ionic-Conductivity and NMR Study of Fluorite-Structured $K_{0.4}Bi_{0.6}F_{2.2}$.** Phys Rev B 27(11):6587–6592, 1983.

Cheng A. F., Maclennan C. G., Lanzerotti L. J., Paonessa M. T., Armstrong T. P., **Energetic Ion Losses Near Io's Orbit.** J Geo R-S P 88(NA5):3936–3944, 1983.

Chin A. K., Zipfel C. L., Chin B. H., Digiuseppe M. A., **Degradation of 1.3-μm InP InGaAsP Light-Emitting Diodes With Misfit Dislocations.** Appl Phys L 42(12):1031–1033, 1983.

Chiu L. C., Smith J. S., Margalit S., Yariv A., Cho A. Y., **Application of Internal Photoemission From Quantum-Well and Heterojunction Super-Lattices to Infrared Photodetectors.** Infrar Phys 23(2): 93–97, 1983.

Cohen J. D., Lang D. V., Harbison J. P., Sergent A. M., **Photoinduced Changes in the Bulk-Density of Gap States in Hydrogenated Amorphous-Silicon Associated With the Staebler-Wronski Effect.** Solar Cells 9(1–2):119–131, 1983.

Dubois L. H., Somorjai G. A., **Why $CO_2$ Does Not Dissociate on RH At Low-Temperature—Comment (Letter).** Surf Sci 128(2–3):L231–L235, 1983.

Dutt B. V., Ludwig R. A., Ermanis F., **The Origin and Elimination of Si Pyramids in (Ga, Al) As-Si LEDS.** J Cryst Gr 62(1):21–26, 1983.

Englert T., Maan J. C., Uihlein C., Tsui D. C., Gossard A. C., **Cyclotron-Resonance of Two-D-Electrons in $GaAs/Al_xGa_{1-x}As$ Heterostructures at Low-Densities.** J Vac Sci B 1(2):427–430, 1983.

Fiory A. T., **Brownian-Dynamics Simulations of Melting of Finite Two-Dimensional Systems With Logarithmic Interaction.** Phys Rev B 28(1):236–243, 1983.

Gibbs H. M. et al., **Optical Bistability, Regenerative Pulsations, and Transverse Effects in Room-Temperature GaAs-AlGaAs Super-Lattice Etalons.** J Physique 44(NC-2):195–204, 1983.

Gill P. S., Graedel T. E., Weschler C. J., **Organic Films on Atmospheric Aerosol-Particles, Fog Droplets, Cloud Droplets, Raindrops, and Snowflakes (Review).** Rev Geophys 21(4):903–920, 1983.

Gilmer G. H. et al., **Simulation-Models of the Crystal Vapor Interface.** J Vac Sci B 1(2):298–304, 1983.

Greenside H. S., Schulter M., **Pseudopotentials for the Three-D Transition-Metal Elements.** Phys Rev B 28(2):535–543, 1983.

Greis N. P., **Flood Frequency-Analysis—A Review of 1979–1982.** Rev Geophys 21(3):699–706, 1983.

Griscom D. L., Friebele E. J., Long K. J., Fleming J. W., **Fundamental Defect Centers in Glass-Electron-Spin Resonance and Optical-Absorption Studies of Irradiated Phosphorus-Doped Silica Glass and Optical Fibers.** J Appl Phys 54(7):3743–3762, 1983.

Gurvitch M., **Experimental-Evidence Versus Exchange Theory of Resistivity Saturation.** Phys Rev B 28(2):544–549, 1983.

Haddon R. C., Hirani A. M., Kroloff N. J., Marshall J. H., **1,3,4,9-Tetramethoxyphenalenyl system.** J Org Chem 48(12):2115–2117, 1983.

Hilinski E. F., Rentzepis P. M., **Chemical Applications of Picosecond Spectroscopy (Review).** Acc Chem Re 16(6):224–232, 1983.

Jelinski L. W., Dumais J. J., Stark R. E., Ellis T. S., Karasz F. E., **Interaction of Epoxy-Resins With Water—A Quadrupole Echo Deuterium NMR-Study (Letter).** Macromolec 16(6):1019–1021, 1983.

Kinsbron E., Murarka S. P., Sheng T. T., Lynch W. T., **Oxidation of Arsenic Implanted Polycrystalline Silicon.** J Elchem So 130(7):1555–1560, 1983.

Klauder J. R., **A Langevin Approach to Fermion and Quantum Spin Correlation-Functions (Letter).** J Phys A 16(10):L317–L319, 1983.

Kleinman D. A., **Binding-Energy of Biexcitons and Bound Excitons in Quantum Wells.** Phys Rev B 28(2):871–879, 1983.

Klem J., Fischer R., Drummon T. J., Morkoc H., Cho A. Y., **Incorporation of Sb in $GaAs_{1-x}Sb_x$ (X-Less-Than-0.15) By Molecular-Beam Epitaxy.** Electr Lett 19(12):453–455, 1983.

Lake G., Norman C., **Stellar and Gaseous Dynamics of Triaxial Galaxies.** Astrophys J 270(1):51–70, 1983.

Lifshitz N., **Study of Breakdown Fields of Oxides Grown on Reactive Ion Etched Silicon Surface—Improvement of Breakdown Limits by Oxidation of the Surface.** J Elchem So 130(7):1549–1550, 1983.

Luongo J. P., **Infrared Characterization of Alpha-Crystalline and Beta-Crystalline Silicon-Nitride.** J Electrochem Soc 130(7):1560–1562, 1983.

Lyons A. M., Wilkins C. W., Robbins M., **Thin Pinhole-Free Carbon-Films.** Thin Sol Fi 103(4):333–341, 1983.

Malyj M., Griffiths J. E., **Stokes Anti-Stokes Raman Vibrational Temperatures—Reference Materials, Standard Lamps, and Spectrophotometric Calibrations.** Appl Spectr 37(4):315–333, 1983.

McWhan D. B., Gurvitch M., Rowell J. M., Walker L. R., **Structure and Coherence of NbAl Multilayer Films.** J Appl Phys 54(7):3886–3891, 1983.

Olego D., Pinczuk A., Gossard A. C., Wiegmann W., **Plasma-Oscillations of Layered Electron Gases in Semiconductor Heterostructures.** J Vac Sci B 1(2):412–414, 1983.

People R., Wecht K. W., Alavi K., Cho A. Y., **Measurement of The Conduction-Band Discontinuity of Molecular-Beam Epitaxial Grown $In_{0.52}Al_{0.48}As/In_{0.53}Ga_{0.47}As$, N-N Heterojunction by C-V Profiling.** Appl Phys L 43(1):118–120, 1983.

Petroff P. M., Wilson R. J., **Surface Dislocation Process for Surface Reconstruction and Its Application to the Silicon (111) 7×7 Reconstruction.** Phys Rev L 51(3):199–202, 1983.

Phillips J. M., Feldman L. C., Gibson J. M., Mcdonald M. L., **Rutherford Backscattering Channeling and Transmission Electron-Microscopy Analysis of Epitaxial $BaF_2$ Films on Ge and InP.** J Vac Sci B 1(2):246–249, 1983.

Pian T. R. et al., **Electron and Photon Stimulated Desorption of Positive-Ions From Alkali-Halide Surfaces.** Surf Sci 128(1):13–21, 1983.

Schiavone J. A., **Microwave Radio Meteorology—Seasonal Fading Distributions.** Radio Sci 18(3):369–380, 1983.

Schneemeyer L. F., Cohen U., **Electrochemical Synthesis of Photoactive $MoS_2$.** J Elchem So 130(7):1536–1539, 1983.

Schubert R., Augis J. A., **Sub-PPMA Gas-Analysis in 40-$\mu$l Volumes by rf Mass Spectrometry.** J Vac Sci A 1(2):248–251, 1983.

Schwartz G. P., **Analysis of Native Oxide-Films and Oxide Substrate Reactions on III-V-Semiconductors Using Thermochemical Phase-Diagrams.** Thin Sol Fi 103(1–2):3–16, 1983.

Schwartz G. P., Bondybey V. E., English J. H., Gualtieri G. J., **Thermal and Pulsed Laser Evaporation of Single Phase $As_xP_{1-x}$ Alloys.** Appl Phys L 42(11):952–954, 1983.

Schwartz G. P., Dutt B. V., Malyj M., Griffiths J. E., Gualtieri G. J., **Thermal-Oxidation and Anodic Film Substrate Reactions on $In_xGa_{1-x} - As_yP_{1-y}$.** J Vac Sci B 1(2):254–259, 1983.

Shaw E. D. et al., **Helical Undulatory Study.** J Physique 44(NC-1):153–161, 1983.

Silberg E., Chang T. Y., Caridi E. A., Evans C. A., Hitzman C. J., **Manganese and**

**Germanium Redistribution in $In_{0.53}Ga_{0.47}As$ Grown by Molecular-Beam Epitaxy.** J Vac Sci B 1(2):178–181, 1983.

Simmons J. G., Taylor G. W., **Generalized Theory of Conduction in Schottky Barriers.** Sol St Elec 26(7):705–709, 1983.

Stall R. A., **Growth of Refractory Oxide-Films Using Solid Oxygen Sources in a Molecular-Beam Epitaxy Apparatus.** J Vac Sci B 1(2):135–137, 1983.

Stavola M. et al., **Symmetry Determination For Deep States in Semiconductors From Stress-Induced Dichroism of Photocapacitance.** J Appl Phys 54(7):3897–3901, 1983.

Stormer H. L., Haavasoja T., Narayanamurti V., Gossard A. C., Wiegmann W., **Observation of the Dehaas-Vanalphen Effect in a Two-Dimensional Electron-System.** J Vac Sci B 1(2):423–426, 1983.

Stormer H. L., Schlesinger Z., Chang A., Tsui D. C., Gossard A. C., Wiegmann W., **Energy Structure and Quantized Hall-Effect of Two-Dimensional Holes.** Phys Rev L 51(2):126–129, 1983.

Subramaniam P., Zemanian A. H., **Analysis of $Delta_2Phi$-$c_2Phi = g$ in a Semi-Infinite Medium With an Irregular Boundary by Means of Network Manipulations.** IEEE Circ S 30(5):300–307, 1983.

Swaminatthan V., Wagner W. R., Anthony P. J., Henein G., Koszi L. A., **Bonding Pad Induced Stresses in (Al, Ga)As Double Heterostructure Lasers.** J Appl Phys 54(7):3763–3768, 1983.

Tennant D. M., **Metal on Polymer Ion-Implantation Mask.** J Vac Sci B 1(2):494–496, 1983.

Teo B. K., Antonio M. R., Averill B. A., **Molybdenum K-Edge Extended X-Ray Absorption Fine Structure (EXAFS) Studies of Synthetic Mo-Fe-S Clusters Containing the $MoS_4$ Unit: Development of a Fine Adjustment Technique Based on Models.** J Am Chem S 105(12):3751–3762, 1983.

Tompkins H. G., Allara D. L., Pasteur G. A., **Molecular-Orientation and Structure of Copper Benzimidazole Films.** Surf Int An 5(3):101–104, 1983.

Tsang W. T., Olsson N. A., **New Current Injection 1.5-$\mu$m Wavelength $Ga_xAl_yIn_{1-x-y}As$/InP Double-Heterostructure Laser Grown by Molecular Beam Epitaxy.** Appl Phys L 42(11):922–924, 1983.

Tsang W. T., Olsson N. A., **Preparation of 1.78-$\mu$m Wavelength $Al_{0.2}Ga_{0.8}Sb$/GaSb Double-Heterostructure Lasers by Molecular-Beam Epitaxy.** Appl Phys L 43(1):8–10, 1983.

Tsang W. T., Olsson N. A., Logan R. A., **Transient Single-Longitudinal Mode Stabilization in Double Active Layer GaInAsP InP Laser Under High-Bit Rate Modulation.** Appl Phys L 42(12):1003–1005, 1983.

Varma C. M., Simons A. L., **Strong-Coupling Theory of Charge-Density-Wave Transitions.** Phys Rev L 51(2):138–141, 1983.

Verter F., Knapp G. R., Stark A. A., Wilson R. W., **Regions of Low-Molecular Column Density Near the Galactic Plane.** Astroph J S 52(3):289–292, 1983.

Weber T. A. et al., **Simulation of Polyethylene (Review).** Adv Chem SE1983(204):487–500, 1983.

Wilson L. O., Nelson T. J., **Magnetization Distributions in Ion-Implanted Bubble Garnet-Films.** J Appl Phys 54(7):4163–4167, 1983.

Wilson R. J., Mills A. P., **Electron and Positron Work-Functions of Cr(100).** Surf Sci 128(1):70–80, 1983.

Wilson T., Boyd G. D., Westerwick E. H., Storz F. G., **Alignment of Liquid-Crystals on Surfaces With Films Deposited Obliquely at Low and High-Rates.** Molec Cryst 94(3-4):359–366, 1983.

Wudl F. et al., **Tetramethyldithiadiselenafulvalene (TMDTDSF)—Properties of its Hexafluorophosphate Salt and Alloys With Tetramethyltetraselenafulvalene.** J Chem Phys 79(2):1004–1012, 1983.

## SOCIAL AND LIFE SCIENCES

Hopfield J. J., Feinstein D. I., Palmer R. G., **Unlearning Has a Stabilizing Effect in Collective Memories.** Nature 304(5922):158–159, 1983.

Pierrehumbert J., Liberman M., **On Finding the Iguana (Letter).** Cont Psycho 28(7):569–570, 1983.

## SPEECH AND ACOUSTICS

Atal B. S., **Speech Coding—Recognizing What We Do Not Hear in Speech.** Ann NY Acad 405(Jun):18–32, 1983

Pols L. C. W., Olive J. P., **Intelligibility of Consonants in CVC Utterances Produced by Dyadic Rule Synthesis.** Speech Comm 2(1):3–13, 1983.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF COMPUTING

Volume 62 December 1983 Number 10, Part 2

# Theory of Program Testing—An Overview

By R. E. PRATHER*

(Manuscript received January 18, 1983)

In this paper, we provide a detailed survey of the various approaches to program testing that have been proposed in recent years. Particular attention is given to a discussion of the developing theory of program testing and to the decomposition of the testing problem into the program graph construction, test path selection, and test case generation phases. Examples are included to illustrate the different testing strategies. Comparisons are made from one method to another, all in a uniform terminology and notation, to facilitate an understanding of various combinations of strategies that might lead to a more workable testing methodology.

## I. INTRODUCTION

The general goal of software testing is to affirm the quality of a program through systematic exercising of the code in a carefully controlled environment. The execution of a program test scheme should validate an expected prespecified behavior, ideally serving to demonstrate the absence of program errors. Considering the difficulty of obtaining actual proofs of program correctness, program testing

* University of Denver, Colorado.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

may be the only effective means for assuring the quality of software systems of nontrivial complexity.

The state of the art in software testing as of a decade ago is broadly surveyed in the book by Hetzel,[1] representing an ad hoc approach at best. During the intervening years, computer programming methodology has made great strides toward improving the quality of our product. And yet, software testing has remained a kind of "black art", only vaguely understood by its practitioners. Happily, this situation is changing. The development of the beginnings of a theory of testing are well under way, and the more recent literature shows great promise for brighter days ahead. Some of these ideas are discussed in a new book by Myers,[2] and further elaboration can be found in the survey papers by Miller.[3-6]

In this overview, we summarize in detail the more recent literature on software testing and present the more important results in a uniform framework, style, and notation. We hope that this perspective will help to focus attention on the more viable alternatives and to point the way toward the most promising directions for future research and development.

## II. GENERAL THEORY—THE FUNCTIONAL APPROACH

The first attempt to describe a generalized theory of testing is found in the work of Goodenough and Gerhart,[7,8] A related study is that of Hamlet.[9] In the former, a *program* is viewed as a function $F{:}D \rightarrow R$ over an *input domain* $D$ with values in an *output range* $R$. The program *specification* can also be viewed as a function $G{:}D \rightarrow R$, whether completely specified or not. For testing purposes, we must compare $F(d)$ with $G(d)$ for selected inputs $d$ in $D$. Though such an exhaustive test is not feasible in general, we say that the program $F$ is *correct* if we have

$$F(d) = G(d) \text{ (for all } d \text{ in } D),$$

recognizing that this is simply a theoretical notion, one not necessarily capable of direction verification.

In any practical setting, we will only be able to examine the behavior of the program for a few selected input values. Realizing this, we say that a *test* for the program $F$ is a (finite) subset $T$ of $D$. Recalling the 'goal of software testing,' $T$ is said to be an *ideal test* (for $F$) if

$$\text{success}(T) \Rightarrow \text{correct}(F),$$

i.e., if $F(t) = G(t)$ for $t$ in $T$ implies the same for all $t$ in $D$. Note that the successful execution of an ideal test would constitute a proof of correctness. Given the difficulty in finding proofs of correctness, however, we should not be surprised to learn that ideal tests, in this

sense, are difficult to discover. (We note that the 'trivial' ideal test, the exhaustive one with $T = D$, though easily stated is ordinarily unmanageable in size.)

As a matter of fact, we would prefer not to 'discover' our tests at all, but to have them 'selected' on the basis of some sensible criterion. Formally, a *test selection criterion* (for a program $F$) is a (true-false) predicate $C$ over the subsets of $D$. Following Goodenough and Gerhart once again, such a criterion $C$ is *reliable* (for $F$) if

$$C(T1) \text{ and } C(T2) \Rightarrow \text{success}(T1) = \text{success}(T2),$$

and, on the other hand, $C$ is said to be *valid* (for $F$) if

$$\sim\text{correct}(F) \Rightarrow \sim\text{success}(T)$$

for some $T$ satisfying $C(T)$. In general, reliability refers to the consistency with which results will be produced within the selection criterion, whereas validity refers to the ability to produce meaningful results, regardless of their consistency.

It is clear that these notions of reliability and consistency are quite strong. Perhaps the most convincing statement to this effect is given by the following:

*Theorem (Goodenough and Gerhart): If C is reliable and valid, then C(T) implies that T is an ideal test.*

On the other hand, Weyuker and Ostrand[10] have argued that these notions are not strong enough, referring as they do to a particular program. If the same ideas are extended, however, so as to apply "uniformly" over all programs $F$, then one obtains the following:

*Theorem (Weyuker and Ostrand): If C is uniformly reliable and uniformly valid, then C(T) implies that $T = D$, i.e., T is an exhaustive test.*

Surely this carries the original ideas too far. And in fact, the theorem can be understood to say, "If nothing is known about the errors in the program, a test criterion is guaranteed ideal (in the sense of Goodenough and Gerhart) if and only if it selects the entire input domain." What is probably needed to arrive at a more practical alternative is a weakening of the Goodenough and Gerhart theory. This is the general thrust of Hamlet's work, but results along these lines thus far are less than satisfactory, showing perhaps more promise toward applications to program maintenance than to testing. The interested reader should consult Ref. 9 for details.

A test selection criterion $C$ should outline the properties of a program that must be exercised to constitute a "thorough" test, ideally one whose successful execution implies an error-free program. Following Goodenough and Gerhart once again, we may suppose that $C$ is described as a finite set $\{c\}$ of *test predicates* (i.e., logical conditions on

the input data), and we then choose $T$ subject to the condition(s):

$$C(T) \Leftrightarrow \begin{array}{l}\text{for all } c \text{ in } C, \text{ there is } t \text{ in } T \text{ with } c(t) \\ \text{for all } t \text{ in } T, \text{ there is } c \text{ in } C \text{ with } c(t).\end{array} \qquad (*)$$

In words, every test predicate belonging to $C$ should be satisfied by at least one test datum $t$ in $T$, and conversely, every $t$ in $T$ must satisfy at least one test predicate.

It is suggested that the test predicates be derived from the program specifications—this is the essence of the *functional approach* (or "black box" approach) to testing. But the claim is made[7] that to have a reasonable chance of constituting a reliable criterion, $C$ must be composed of test predicates satisfying (at least) the following set of conditions:

*Condition 1:* Every individual branching condition in the program must be represented by an equivalent test predicate.

*Condition 2:* Every potential termination condition (e.g., error, overflow, etc.) must be represented by a corresponding test predicate.

*Condition 3:* The range of every variable appearing in a test predicate must be partitioned into classes that are "treated in the same way" by the program.

*Condition 4:* Every condition relevant to the proper functioning of the program that is implicit in the program specification or of one's knowledge of the program must be represented by a corresponding test predicate.

*Condition 5:* The test predicates must be "independent," in that all data satisfying a particular test predicate must exercise the same path in the program and must test the same branch conditions.

We note that only the second and fourth of these conditions are of a "functional" nature. The others are "structural," that is, relating more to the topology of the underlying flowchart. It would seem, therefore, that any reasonable testing strategy should address both points of view.

Consider the following example, the often cited problem of classifying triangles:

Specification:

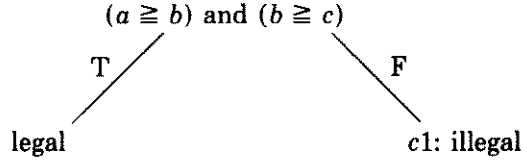
*Input:* Three positive integers $a \geqq b \geqq c$.

*Output:* An indication as to whether:

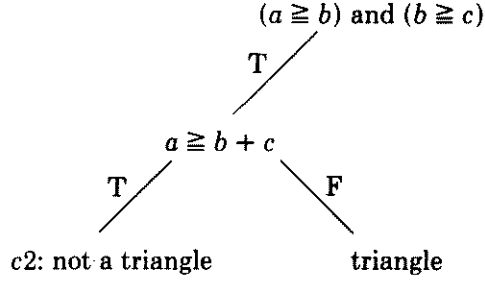
1. They do not represent the sides of a triangle
2. They are the sides of an equilateral triangle
3. They are the sides of an isosceles triangle
4. They are the sides of a scalene right triangle
5. They are the sides of a scalene obtuse triangle
6. They are the sides of a scalene acute triangle.

This problem is especially well suited to the "functional" approach.

Since the whole purpose of the problem is to classify its input domain, there is an obvious specification-based derivation of test predicates. We may first divide the universe of triples $(a, b, c)$ into legal and illegal forms:




For the legal entries, we may further distinguish two cases:




and the triangles may then be subdivided into six subclasses:

$c3 : (a = b)$ and $(b = c)$ equilateral
$c4 : (a = b)$ and $(b > c)$ isosceles
$c5 : (a > b)$ and $(b = c)$ and $(a < b + c)$ isosceles
$c6 : (a > b)$ and $(b > c)$ and $(a*a = b*b + c*c)$ right scalene
$c7 : (a > b)$ and $(b > c)$ and $(a*a < b*b + c*c)$ acute scalene
$c8 : (a > b)$ and $(b > c)$ and $(a*a > b*b + c*c)$ and $(a < b + c)$ obtuse scalene.

If we set $C = \{c(i) : i = 1 \text{ to } 8\}$ and choose one triple from each input subdomain, we may obtain the test set:

$t1 = (1, 2, 3)$
$t2 = (14, 6, 4)$
$t3 = (1, 1, 1)$
$t4 = (2, 2, 1)$
$t5 = (3, 2, 2)$
$t6 = (5, 4, 3)$
$t7 = (6, 5, 4)$
$t8 = (4, 3, 2).$

Such a test set will automatically satisfy $(*)$ and the test selection criteria will more than likely meet Conditions 2 and 4 above. But we have no guarantee that the "structural" conditions 1, 3, 5 will be met, since we haven't looked at the program!

Weyuker and Ostrand[10,11] have made the cogent suggestion that the input domain be partitioned *both* on the basis of the specification-driven, program-independent properties mentioned above, and on the structural properties of the program as well. It seems that this is the only way to meet all five of the Goodenough and Gerhart conditions, and to thus have a chance of approaching a reliable test selection criterion $C = \{c\}$ defined by a set of test predicates.

Suppose we add a sixth (implicit) condition to the five that are outlined above, namely:

*Condition 6:* The test predicates must be "complete" in that every input of the domain $D$ must satisfy (exactly—see condition 5) one of the test predicates.

Then Conditions 5 and 6 ensure that $C = \{c\}$ defines a partition

$$\kappa = \{C\}$$

on the input domain. When we concentrate only on the problem specifications, as above, we obtain the *problem partition* consisting of *problem domains* $C$. Having a completed version of the program in hand, we may speak as well of a *path partition*

$$\pi = \{P\}$$

of the same domain $D$, where each *path domain* $P$ comprises a class of inputs that traverse the same path through the program. Thus, the path partition separates $D$ into classes of inputs that are treated the same way by the program, whereas the problem partition separates $D$ into classes that should be treated the same.

There is no assurance that these two partitions will coincide, nor is it necessary that they do. But ultimately (or at least, hopefully), the program and its algorithm all derive from the original problem specification, so we should not expect the two paritions to differ markedly. On the other hand, those differences that do exist are fruitful places to look for errors! Recognizing this, Weyuker and Ostrand have suggested that the problem and path partitions be intersected, yielding a finer partition

$$\sigma = \kappa \wedge \pi = \{C \cap P\} = \{S\}$$

of nonoverlapping subdomains $S$ of the domain $D$, and they further suggest that this be used as the ultimate test selection criterion, choosing one test case from each subdomain as before.

In the terminology of Weyuker and Ostrand,[10,11] a subdomain $S$ of $D$ is said to be *revealing* (of errors) if

$$\text{success}(s \text{ in } S) \Rightarrow \text{correct}(F, S),$$

i.e., if the successful execution of any input from $S$ implies correctness

of the program over the whole subdomain. Since the inputs of a subdomain $S$ (in the intersection above) should be and in fact are treated the same by the program, the hope is extended that these are, in essence, the revealing subdomains. A successful execution of one test datum from each of the subdomains $S$ then implies (or at least suggests) the correctness of the program over the whole domain.

Consider the "triangle classification problem" once again, and suppose we are presented with the program (flowchart) of Fig. 1 as

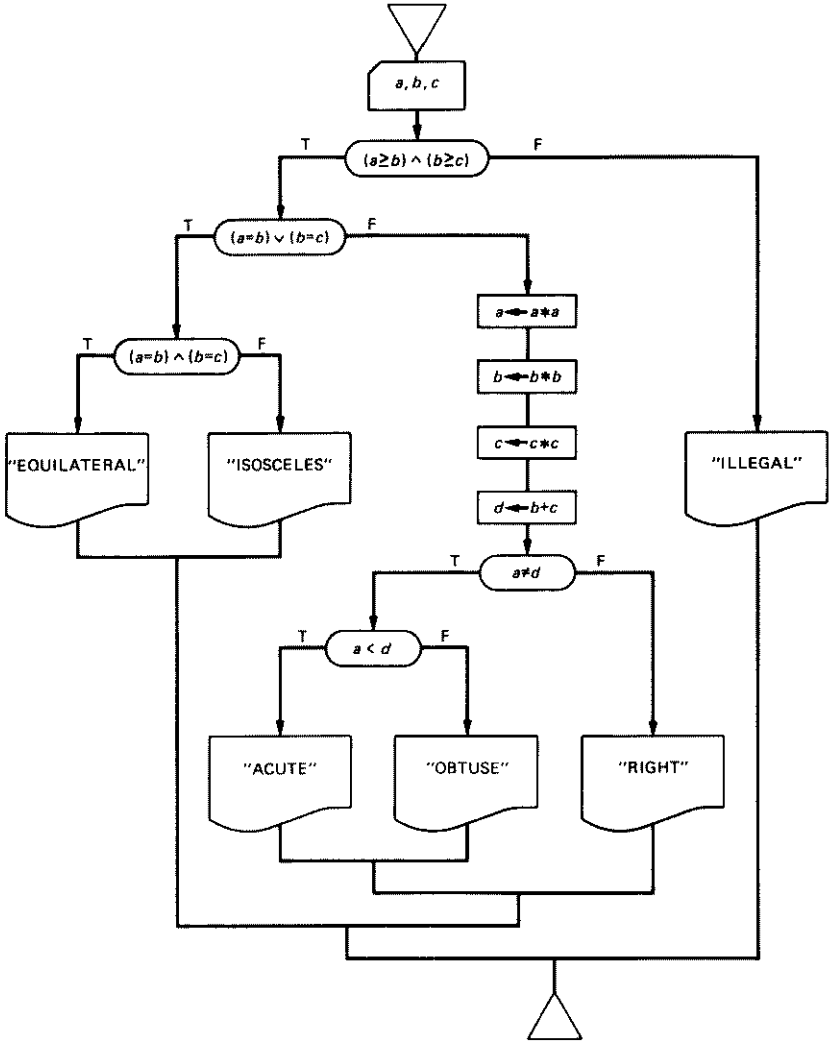



Fig. 1—Flowchart for classifying triangles.

representing a solution to the problem. There are six paths through the program, as described by the conjunctions of branch conditions defined by each path, as follows:

$p1 : \sim[(a \geqq b) \text{ and } (b \geqq c)] = (a < b) \text{ or } (b < c)$
$p2 : (a > b > c) \text{ and } (a*a = b*b + c*c)$
$p3 : (a > b > c) \text{ and } (a*a < b*b + c*c)$
$p4 : (a > b > c) \text{ and } (a*a > b*b + c*c)$
$p5 : (a \geqq b \geqq c) \text{ and } [(a = b) \text{ or } (b = c)] \text{ and } \sim[(a = b) \text{ and } (b = c)]$
$= (a = b > c) \text{ or } (a > b = c)$
$p6 : a = b = c.$

Intersecting the six corresponding path domains $P[i]$ ($i = 1$ to 6) with the eight earlier problem domains $C[i]$ results in a partition $\{S\}$ of nine subdomains characterized as follows:

$S1 = C1 = C1 \cap P1 : (a < b) \text{ or } (b < c)$
$S2 = C2 \cap P4 : (a > b > c) \text{ and } (a \geqq b + c)$
$S3 = C2 \cap P5 : (b = c) \text{ and } (a \geqq b + c)$
$S4 = C3 = C3 \cap P6 : a = b = c$
$S5 = C4 = C4 \cap P5 : a = b > c$
$S6 = C5 = C5 \cap P5 : (a > b = c) \text{ and } (a < b + c)$
$S7 = C6 = C6 \cap P2 : (a > b > c) \text{ and } (a*a = b*b + c*c)$
$S8 = C7 = C7 \cap P3 : (a > b > c) \text{ and } (a*a < b*b + c*c)$
$S9 = C8 = C8 \cap P4 : (a > b > c) \text{ and } (a*a > b*b + c*c)$
$\text{and } (a < b + c)$

in very close agreement with the problem partition $\{C\}$ obtained earlier.

The problem we are discussing has a rather precise functional specification so that we would expect that the problem and path partitions might nearly coincide. Nevertheless, there is a slight discrepancy, and in place of the test datum $t2 = (14, 6, 4)$ we would now have to choose two, say (14, 6, 4) and (3, 1, 1). A test of the resulting nine data points would then reveal two errors, as shown in Table I below.

Table I—Test of nine data points

| Domain | Test Data | Correct Output | Actual Output |
|---|---|---|---|
| $S1$ | (1, 2, 3) | Illegal | illegal |
| $S2$ | (14, 6, 4) | Not a triangle | obtuse |
| $S3$ | (3, 1, 1) | Not a triangle | isosceles |
| $S4$ | (1, 1, 1) | Equilateral | equilateral |
| $S5$ | (2, 2, 1) | Isosceles | isosceles |
| $S6$ | (3, 2, 2) | Isosceles | isosceles |
| $S7$ | (5, 4, 3) | Right | right |
| $S8$ | (6, 5, 4) | Acute | acute |
| $S9$ | (4, 3, 2) | Obtuse | obtuse |

The programmer has failed to take account of those situations where $a \geqq b + c$ (not a triangle). And our test criteria are able to detect such errors. In fact, so detailed is the specification for this example that a test set based on the problem partition alone would have served equally well.

In spite of the obvious relevance of the ideas presented here, particularly those of Weyuker and Ostrand, a good deal of work remains to be done to apply the theory to a wide class of programs. One of the more important tasks is to find more systematic methods for constructing the problem partition. This will not be easy, since finding a good problem partition is quite similar to the task of creating the program itself. It is suggested, however, that the development of formal specification languages would be helpful here, particularly if such developments are made with a specification-driven testing methodology in mind, along the lines presented here.

An equally important consideration when thinking of applying the above theory to larger programs is that of obtaining the domains of the path partition. How are the paths to be described, generated, and selected with programs of increasing size and complexity? It is certainly clear that our one example is misleading in this respect. We had only a small number of paths to consider, whereas a typical program of any size will have a very large number of paths, most likely an infinity of paths, owing to the presence of loops. If our test is still to be finite, how do we then choose paths judiciously? How are they described? And most importantly, how do we generate test cases that will traverse these paths, if indeed this is possible? These are some of the questions that we begin to address in the following sections.

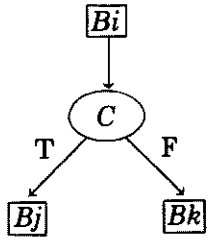
## III. GENERAL THEORY—THE STRUCTURAL APPROACH

In a *structural approach* to the theory of testing, a program $F$ is represented by a "skeleton" of its underlying flowchart, a directed graph symbolizing the flow of control. This point of view is advanced most effectively in the extremely lucid survey paper by Huang.[12] We should keep in mind, however, that the flowchart graph must be an accurate representation of the program flow in the code itself. Ordinarily, this is ensured through the use of a "tool" that automatically generates the flowgraph from the source program listing.

Using Huang's terminology,[12-14] a *program block* is a maximal sequence of program statements having the property that if the first member of the sequence is executed, then all other statements in the sequence will also be executed. A *program graph* $F = (V, E)$ is then a directed graph with vertex set $V$ and edge set $E$, where each vertex is associated with a program block and in which there are pairs of edges:

$(i, j)$ labeled by the condition $C$
$(i, k)$ labeled by the condition $\sim C$

according as the flowchart segment:




encountered for blocks *Bi*, *Bj*, and *Bk*. (For convenience, we permit an empty block as a vertex in good standing, e.g., for treating an "if ··· then ···" statement with vacuous "else" clause.) It is further assumed that the graph has a single entry point, the *start vertex*, and a single exit point, the *stop vertex*, and that every vertex lies on some path from 'start' to 'stop.'

A *path* in a (program) graph is defined in the usual way, as a sequence of edges

$$p = e1, e2, \cdots, en,$$

though we ordinarily assume as well that we begin the sequence at 'start' and end at 'stop.' Each such path has an associated *path predicate*

$$P = P1 \wedge \cdots \wedge Pn$$

written as a conjunction of the individual interpreted branch condition labels on the edges $e$, as discussed below (see Section V). The path predicates $P$ are to be identified in one-to-one correspondence with the path domains of the previous section. Thus we may write (somewhat ambiguously):

$$D = \cup P \quad \text{for} \quad P = \{d \text{ in } D : P(d)\}$$

so that the program function $F : D \rightarrow R$ is a union of functions $F(P)$: $P \rightarrow R$ restricting $F$ to the individual path domains $P$.

In structured testing, we examine the program (as a digraph) and we seek to choose a finite set of paths that will cover the program with a certain degree of thoroughness. It is then hoped that test data causing the program to be successfully executed when traversing these paths are sufficient to warrant our confidence in the program's correctness. The theoretical underpinnings of such a testing plan have been studied by Howden,[15–18] who relates his work to the earlier study by Goodenough and Gerhart.[7] Rather than speaking of a "test criteria," however, Howden refers to a *testing strategy*, as a uniform computable function

$$(F : D \rightarrow R) \xrightarrow{H} (T, \text{ subset of } D)$$

that associates with each program $F$ a finite test set $T$ of $D$. $H$ is said to be an *ideal strategy* if each $T = H(F)$ is an ideal test (for every program $F$). As is so often the case with testing theory, the first result is of a negative character:

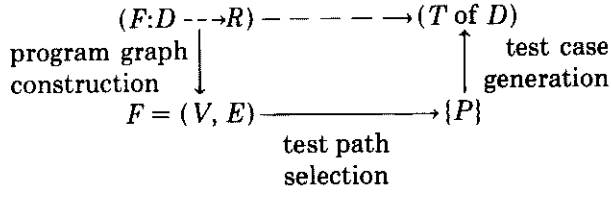
*Theorem (Howden): An ideal testing strategy does not exist.*

Nevertheless, Howden was able to show that "path testing" can be a reliable approach, at least for detecting certain types of errors. He takes the view that the program being treated is a member of a class of programs differing only as to whether they are correct, and for which the incorrect programs have errors of various (known) types. His objective was then to find, if possible, a restricted set of programs for which certain forms of structured testing (i.e., path testing) would be reliable. Typical of Howden's results is that which assumes that the error in a program does not change its control flow, i.e., that the set of path domains is not affected.

*Theorem (Howden): Path testing is a reliable method for distinguishing correct from incorrect programs, as long as the errors of incorrect programs do not affect the path partition.*

Of course, there are theoretical limitations in applying results such as this, since Howden has in mind our choosing one test datum from each path domain $P$, and these may be infinite in number. On the other hand, he has also devised a classification of error types that can be expected to lead to new insights into the testing problem generally. The reader should consult Howden's work (particularly Refs. 16 and 17) for further detail.

In a practical test setting, we require that the subset $T$ of $D$ be finite. Moreover, if we are speaking of a "path testing strategy," the above schema will be decomposed into the three-stage process,




summarized as follows:

1. Program graph construction
2. Test path selection
3. Test case generation.

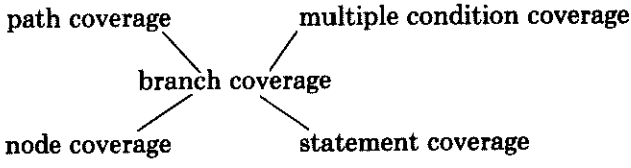
The first phase of the process, to construct the program graph from a source code listing, is fairly straightforward, and for most of the conventional programming languages, e.g., FORTRAN, Pascal, etc., such implementations are already in existence.

As a matter of fact, implementations of testing tools are in various stages of development for treating the entire process outlined above (e.g., see Clarke[19]). But, as we shall see, there are serious problems associated with the latter stages of any proposed implementation along these lines. There are many alternative strategies to choose from, and seemingly, none of these is best for all situations. All we can do at this point is to outline the several alternatives and comment on their general suitability. We begin by introducing the various path selection criteria, continuing this discussion in the next section. The last, and perhaps the most difficult, of our three subprocesses, the generation of test data, is treated in Section V.

There are, as we have indicated, a number of path selection criteria that can be used in attempting to devise a testing strategy that will provide a reasonable coverage of a program graph. Among these criteria are:

1. *Statement coverage*: Execute all statements (blocks) in the graph.
2. *Node coverage*: Encounter all decision node entry points in the graph.
3. *Branch coverage*: Encounter all exit branches of each decision node in the graph.
4. *Multiple condition coverage*: Achieve all possible combinations of condition outcomes at each decision node of the graph.
5. *Path coverage*: Traverse all paths in the graph.

These five strategies are related in their strength of coverage as shown below:




with the weaker criteria at the bottom and the stronger criteria at the top.

As an example illustrating the differing requirements of these criteria, consider the flowchart segment (program graph) shown in Fig. 2. In order to achieve node coverage, the single test:

$$abe\text{: } A = 2, B = 1, X = 1$$

will suffice (but it will not achieve statement coverage because the assignment $X \leftarrow X/A$ will not have been executed). On the other hand, the single test:

$$ace\text{: } A = 2, B = 0, X = 3$$

will be sufficient for complete statement coverage (and node coverage

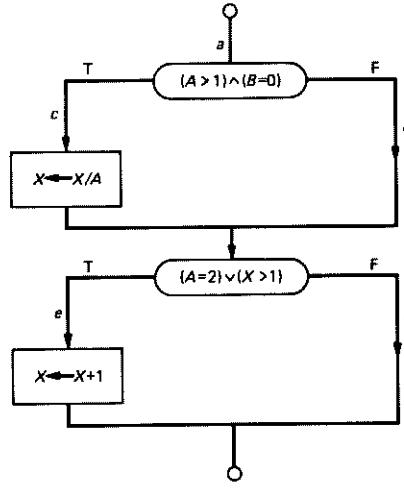



Fig. 2—Flowchart segment illustrating coverage criteria.

as well). For branch coverage, however, at least two tests would be required, e.g.,

$$acd\text{: } A = 3, B = 0, X = 3$$
$$abe\text{: } A - 2, B = 1, X = 1.$$

In the multiple condition coverage criterion, there are 2*2 + 2*2 or 8 outcomes to achieve in combination for the two simple conditions at each decision node. These may be satisfied, for example, by the selection of four separate tests, e.g.,

$$ace\text{: } A = 2, B = 0, X = 4$$
$$abe\text{: } A = 2, B = 1, X = 1$$
$$abe\text{: } A = 1, B = 0, X = 2$$
$$abd\text{: } A = 1, B = 1, X = 1.$$

The first test satisfies the conditions $A > 1, B = 0$ in the first decision and $A = 2, X > 1$ in the second decision. The second test ensures that $A > 1, B \neq 0$ for the first decision and $A = 2, X \leqq 1$ for the second decision. Further analysis shows that all eight combinations are achieved. For the path coverage criterion to be met, we again require four tests, e.g.,

$$ace\text{: } A = 2, B = 0, X = 4$$
$$acd\text{: } A = 3, B = 0, X = 3$$
$$abe\text{: } A = 1, B = 0, X = 2$$
$$abd\text{: } A = 1, B = 1, X = 1.$$

Note, however, that this test set would not satisfy the multiple condition coverage criterion.

It is clear that "statement coverage" and "node coverage" are in themselves rather weak strategies for testing, representing necessary but by no means sufficient criteria for a reasonable structural test. The "branch coverage" criterion, however, implies the other two (as seen in the diagram above) and has come to be regarded as a minimal standard of achievement in structure-based testing. The stronger criteria of "multiple condition coverage" and "path coverage" are difficult to achieve in a program of any complexity. In fact, the path testing criterion is usually relaxed to the extent that only "equivalence classes" of paths are represented. In a program of any size, particularly in the presence of program loops, there is a virtual infinity of paths through the program graph. Two paths are then considered "equivalent" if they differ only in their number of loop traversals. One then chooses only one representative from each such equivalence class in devising a test set. But still, this *modified path coverage* criterion is difficult to achieve in practice.

A survey of the literature shows that there is little common agreement as to what would be considered as an 'adequate' structural test criterion. As we have noted, the "branch coverage" criterion has been widely recognized as a basic measure of testing thoroughness. This is evidenced by the fact that most of the major software testing tools in existence or in development do indeed include some provision for achieving this particular test goal. The disagreement seems to be in deciding how much more (or less) is needed beyond this basic requirement to entitle a structural testing strategy to be considered adequate.

If total branch coverage is indeed used as a measure of testing thoroughness, a simple calibration scheme can be invoked, using a set of *software counters.* One "prepares" the program for testing by inserting counters at appropriate points in a modified copy of the program, and after running through the test set, one can determine the degree of thoroughness from a listing of the resulting counter values. This is the method of *test instrumentation.* We first define a *decision to decision* (DD) *path* of a program to be a sequence of a statements leading from a decision box (or the "start" node) to a decision box (or the "stop" node), having no intervening decisions. To determine whether every branch of our program has been encountered at least once (branch coverage) in our testing, it is sufficient to insert a counter at the 'head' of each DD path.

Consider the classical flowchart solution (Fig. 3) to the problem of computing $z =$ "$x$ to the power $y$". Here, there are five DD paths:

*abc, d, efhi, gfhi, jk*

and we therefore insert our software counters at the points *a, d, e, g,*

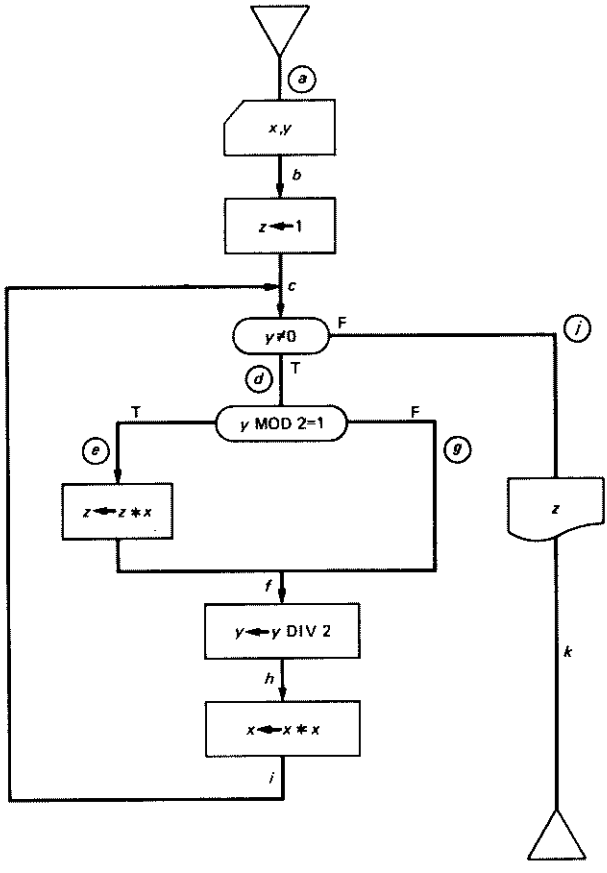



Fig. 3—Flowchart for computing $x^y$.

$j$, as shown. If we have run two test cases as shown in the table below,

| $x$ | $y$ | $a$ | $d$ | $e$ | $g$ | $j$ |
|---|---|---|---|---|---|---|
| 10 | 0 | 1 | 0 | 0 | 0 | 1 |
| 20 | 1 | 1 | 1 | 1 | 0 | 1 |

we would find that 'counter $g$' has not yet been activated. Inspection of the flowchart then shows that we need a test to traverse the path $a$, $b$, $c$, $d$, $g$, etc., requiring $y \neq 0$, $y \bmod 2 \neq 1$. So we may use $x = 24$ and $y = 2$, say, as an additional test case, thus ensuring complete branch coverage. Ideally, this latter phase, directing the tester to the area of untested code, would also be automated.

Of course, we would like to automate as much of the testing methodology as possible, recalling the three-stage process mentioned earlier. On the other hand, in lieu of a complete mechanization, the testing

instrumentation scheme presented here can be of great help in isolating areas in need of further testing. Furthermore, it can be argued that for the little extra cost entailed, it is a worthwhile investment in any testing process, fully automated or not. [Parenthetically, we might note as an indication of the expense associated with the development of testing tools generally, that a package that does little more than "test instrumentation," as described here, has been announced recently (Computer, May 1982) by Management and Computer Services, selling for $12,000.00!.] It may be that some other criterion than "branch coverage" is being used as a measure of test thoroughness. It is still good practice to be concerned as to what extent this standard measure is being met. Moreover, it is reasonable to suppose that the "instrumentation concept," as exemplified here, might generalize to settings where other thoroughness criteria are being used.

## IV. TEST PATH SELECTION

As we have indicated, there are a number of criteria that can be used in selecting program paths to achieve an adequate testing coverage. But the question then becomes: How do we automatically generate a collection of paths meeting a given criterion? The literature is somewhat "hazy" on this point. Perhaps the most explicit treatment of the problem is that of Paige,[20–22] in reference to programs built up from a strict adherence to the structured programming methodology. In fact, we know of no more general approach to the problem, one that would handle structured or unstructured code in relation to the whole spectrum of path selection criteria.

Recall that a *structured program* F is one that has been built up inductively from certain "simple statements" as a base (typically, the assignment, input and output statements, and procedure calls), using only the three familiar constructs:

1. *Sequence:* begin $P1$; $P2$; $\cdots$ ; $Pn$ end
2. *Selection:* if $C$ then $P$ else $Q$
3. *Repetition:* while $C$ do $P$

for structured (but possibly themselves compound) statements $Pi$, $P$, $Q$, respectively. The resulting program graph $F = (V, E)$ is then of a correspondingly restricted form, greatly facilitating the path analysis problem. Perhaps collapsing sequences of simple statements to a single block (graph node), one may then use a "regular expression" $r(F)$ to characterize the program flow, associating

1. The '$\cdot$' operator to sequences
2. The '+' operator to selections
3. The (Kleene) '*' operator to repetitions,

respectively.

For example, if $F$ is the (structured) program graph shown in Fig.

4, we have

$$r(F) = a(b(d + e)(k + 1) + c(f + g(h(i + j))^*m))$$

as the corresponding regular expression. Note the loop $(h(i + j))^*$ resulting from a "while" statement.

We have mentioned earlier, in reference to the modified path coverage criterion, how an equivalence relation is often used to obtain a finite representation of the path alternatives in the presence of loops. Accordingly, if we make the substitution $x^* = x + 1$ (1 = null) in the regular expression $r(F)$, we acknowledge that a loop is either executed or not. Multiplying out so as to obtain a "sum of products" expression, one then obtains the desired collection of paths satisfying the modified path coverage criterion, e.g.,

| | |
|---|---|
| *abdk* | *acf* |
| *abdl* | *acghim* |
| *abek* | *acghjm* |
| *abel* | *acgm* |

in reference to the program graph of Fig. 4. On the other hand, it does not appear that this technique can be extended to handle unstructured programs.

But if we continue to deal with a structured program graph, we can describe a method for deriving a minimum number of paths sufficient to meet the "branch coverage" criterion. We assign a set of paths $S(r)$ to each regular expression $r = r(F)$ inductively, as follows. We let $S(a) = \{a\}$ for each simple statement $a$, and then, assuming that $S(r)$

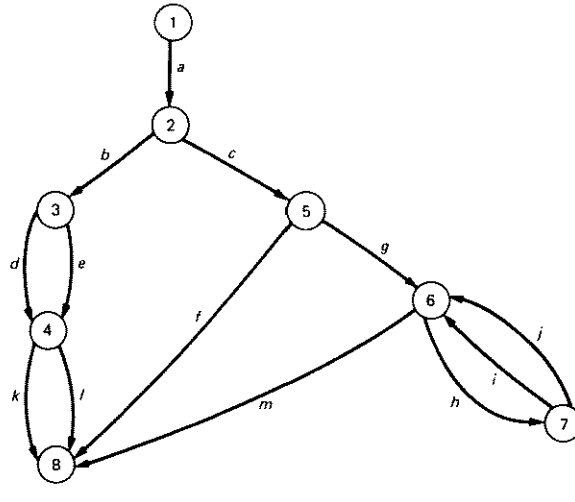



Fig. 4—Structured program graph illustrating modified path coverage criterion.

and $S(t)$ have been defined, for regular expressions $r$ and $t$, we set:

1. $S(r \cdot t) = S(r) \cdot S(t)$
2. $S(r + t) = S(r) \cup S(t)$
3. $S(r^*) = S(r)^*$.

Here, $S(r)^*$ is the singleton set obtained by concatenating (in any order) all of the paths in $S(r)$, and similarly, the set product $S(r) \cdot S(t)$ is obtained by concatenating paths in $S(r)$ with those in $S(t)$—but retaining only enough products so that each of the factors from $S(r)$ and $S(t)$ are represented. It follows that

1. $|S(r \cdot t)| = \max\{|S(r)|, |S(t)|\}$
2. $|S(r + t)| = |S(r)| + |S(t)|$
3. $|S(r^*)| = 1$.

By way of illustration, in considering once again the example from Fig. 4, we may compute:

$$S(i + j) = \{i, j\}$$
$$S(h(i + j)) = \{hi, hj\}$$
$$S((h(i + j))^*) = \{hihj\}$$
$$S(g(h(i + j))^*m) = \{ghihjm\},$$

etc., and finally,

$$S(r) = \{abdk, abel, acf, acghihjm\},$$

yielding four paths that together cover all of the branches of the program.

Once again, as in the case of the previous algorithm, there seems to be no easy extension of this technique that would handle unstructured programs as well. However, a general *upper bound* is readily available regarding the number of paths necessary for total branch coverage. Whether our program is structured or not, we make the observation that if a test path reaches a particular node of the program graph, then it must exit this node through one of the (two) branches leaving the node. If the graph $F$ has $e$ edges and $n$ nodes, it follows that

$$v(F) = e - (n - 2) = e - n + 2$$

is an upper bound on the number of paths necessary to achieve total branch coverage. Coincidentally, this is the formula for McCabe's *cyclomatic complexity measure*,[23] a figure that has proved to be useful in estimating overall "program complexity". At the same time, the graph theoretic derivation of a program's "independent" circuits (paths) yields a branch covering of paths, $v(F)$ in number—though generally somewhat in excess of the minimum number of paths that would be required. In the context of our running example, we have

$v(F) = 13 - 8 + 2 = 7$ and a corresponding set of *basis paths*:

$$\begin{array}{ll} acf & acgm \\ abdk & acghim \\ abek & acghjm. \\ abel & \end{array}$$

Note that the single path *abdl* from our "path coverage" list that is not present here is itself a linear combination of paths already listed. Note as well that we obtain seven paths here, whereas we know from the preceding analysis that four paths will suffice (for branch coverage).

The whole notion that McCabe's *basis of program paths* should constitute a goal of program testing has attracted considerable attention, and we feel obliged to comment on this point. Perhaps this is best done by listing what we think are the pros and cons to the approach. On the positive side, we cite the following:

1. The method is sufficiently general as to be applicable to both structured and unstructured programs.
2. The resulting "basis" does indeed ensure total branch coverage.
3. The paths of a basis are feasibly computable, using standard graph theoretic techniques.

On the other hand, these aspects must be counterbalanced with the following:

1. A single basis is not uniquely determined—there are many, and we must make a choice.
2. The number $v(F)$ of paths in a basis can greatly exceed the minimum number of paths required to achieve branch coverage.
3. The notion that, in some sense, *every* path in the program graph is accounted for by our having selected a basis is somewhat specious.

Note that we do not comment here on the inadequacies of McCabe's $v(F)$ as a measure of overall program complexity—we leave this discussion to a separate paper. On the other hand, the arguments for and against the use of the associated "basis of program paths" as a testing strategy are inconclusive at best, particularly in comparison with the "level paths" of Paige[21,22] that we now describe.

In a program graph $F = (V, E)$, a *level-0 path* is a simple (acyclic) path from "start" to "stop". In effect, these paths trace the "fall through" conditions in the program. Then, inductively, a *level-i path* ($i > 0$) is a simple path (perhaps a circuit) that begins and ends on nodes of a path of lower level, but has none of its other nodes previously appearing on paths of a lower level. Intuitively, the level-$i$ paths for $i > 0$ account for program loops of increasing nesting level and for feedback paths, etc., in the case of an unstructured program.

Considering our earlier structured program graph (Fig. 4) and the

unstructured program graph of Fig. 5, we tabulate the respective level-$i$ paths as shown in Tables II and III.

In any case, we are able to say that a given level-$(i + 1)$ path is "associated with" a certain level-$i$ path according as the given path begins and ends on nodes of the parent path. This relationship orders the level paths in a tree-like structure, in such a way that one can readily construct test paths that again effect a total branch coverage. In so doing, only level paths that associate can be combined to form a program test path. Thus, for example, in the case of the program graph of Fig. 4 above, we may construct the path *acghihjm* as the linear combination:

$$acghihjm = (6) + (7) + (8)$$

using the notation in Table II.

It is clear that the level paths of a program graph span the space of

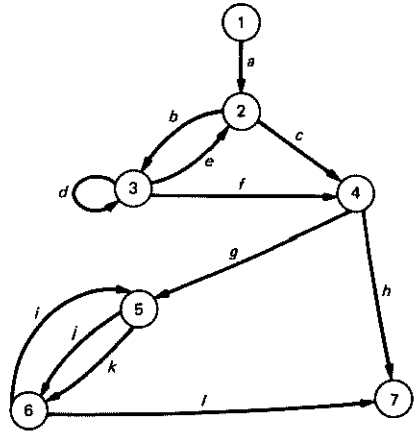



Fig. 5—Unstructured program graph.

Table II—Level-$i$ paths for structured program graph (see Fig. 4)

| Level | Level Paths | $V[i]$ | $E[i]$ |
|---|---|---|---|
| −1 | | {1, 8} | 0 |
| 0 | (1) *abdk*<br>(2) *abdl*<br>(3) *abek*<br>(4) *abel*<br>(5) *acf*<br>(6) *acgm* | {2, 3, 4, 5, 6} | {*a*, *b*, *c*, *d*, *e*, *f*, *g*, *k*, *l*, *m*} |
| 1 | (7) *hi*<br>(8) *hj* | {7} | {*h*, *i*, *j*} |

Note: The sets $V[i]$ and $E[i]$ of vertices and edges at level-$i$ are useful in the computation of the level-$(i + 1)$ paths.

Table III—Level-$i$ paths for unstructured program graph (see Fig. 5)

| Level | Level Paths | $V[i]$ | $E[i]$ |
|---|---|---|---|
| −1 | | {1, 7} | 0 |
| 0 | (1) *acgjl*<br>(2) *acgkl*<br>(3) *abfgjl*<br>(4) *abfgkl*<br>(5) *ach*<br>(6) *abfh* | {2, 3, 4, 5, 6} | {*a*, *b*, *c*, *f*, *g*, *h*, *j*, *k*, *l*} |
| 1 | (7) *d*<br>(8) *e*<br>(9) *i* | 0 | {*d*, *e*, *i*} |

Note: The sets $V[i]$ and $E[i]$ of vertices and edges at level-$i$ are useful in the computation of the level-$(i + 1)$ paths.

program paths. But taken together, they do not usually constitute a basis. Thus, again in Fig. 4 above, we have eight-level paths, whereas we know from our previous analysis that this graph has rank $v = 7$. On the other hand, Paige's level paths have a definite uniqueness, an advantage over the notion of a basis as developed by McCabe, and leading to a graduated *level path testing strategy* as follows:

1. First test all level-0 paths—in effect, keeping all loops in the "nonexecuting" mode.
2. Next test all level-1 paths, reaching them through their associated level-0 paths, etc.

The result is a highly structured testing strategy where segments of the program are treated in successive layers of nesting depth.

The level path testing strategy provides for a rather exhaustive treatment of a program's path structure at successive depths of nesting. In this sense, the approach has a potential thoroughness rivaling that of the "modified path coverage" criterion. On the other hand, Paige's strategy is readily applicable to both structured and unstructured programs. At the same time, his method lends itself to a convenient algorithmic solution,[22] though one must be prepared to compute all simple paths (or circuits) between various identified pairs of nodes, along edges not previously used—most likely requiring the use of a "depth first search" strategy. Except for this computational difficulty, the approach is quite orderly; it provides for a more thorough testing than simple branch coverage, and it compares favorably against McCabe's "basis of program paths" in that:

1. The level paths are uniquely determined.
2. The number of level paths will exceed $v(F)$.
3. The notion that somehow every path in the program graph is accounted for by our successive treatment of its levels has a good deal more credibility.

In conclusion, it must be noted that all of the methods we have discussed for selecting program test paths are subject to one overriding criticism. There is absolutely no assurance that the paths selected will be *feasible*, i.e., executable with an appropriate choice of input data. We suggest that this problem becomes more serious (and is surely more difficult to analyze) in the case of paths selected in an attempt to minimize the number required for branch coverage. Paige's "level path" strategy would seem to be easier to handle in this respect, since we build up paths from the simple to the more complex, starting with those that are more likely to be feasible.

## V. TEST CASE GENERATION

The whole question of path feasibility is related to the "test case generation" problem. This is the one remaining phase to be discussed of the three that were outlined in the rectangular problem-decomposition paradigm of Section III. Considering the question of feasibility, however, we can see that it is difficult to so trichotomize the automation of the overall testing program. Though useful as a paradigm, we must admit that this partition of the problem is overly idealistic in relation to the real world of program testing that we are likely to encounter.

Suppose we have selected a set of program paths because they meet one or another of the test coverage criteria, or for whatever reason. There still remains the problem of generating corresponding test cases that will drive the program through the indicated paths. This again turns out to be a nontrivial (and in some cases, unsolvable) problem. All we can do at this point is to summarize the approaches that have been taken by researchers in the field and to give a few suggestions that might aid in developing a workable methodology.

Perhaps the most comprehensive treatment of the problem is that of Clarke.[19,24,25] Consider a single path $p$ from "start" to "stop" through a program $F = (V, E)$, again viewed as a directed graph. We intend to show how $p$ may be characterized as a *path predicate*, i.e., a logical condition

$$P = P1 \wedge P2 \wedge \cdots \wedge Pn$$

expressed as a conjunction of "interpreted" branch conditions derived from the labels on the edges of $F$. Inductively, we may think of $p$ as having developed as a sequence of "partial paths:"

$$p(k) = (v[0] = \text{start}, v[1], \cdots, v[k])$$

leading from "start" to some intermediate vertex $v(k)$ on the way to

"stop". Correspondingly, we may give an inductive derivation of the path predicate, writing $P(0) = \text{true}$ and

$$P(k) = P(k-1) \wedge ibp(v[k-1], v[k]),$$

where the latter conjunct is the *interpreted branch predicate* associated with the transition from vertex $v[k-1]$ to $v[k]$. More precisely, $ibp(e)$ for an edge $e$ labeled with the Boolean condition $C$ will be computed by substituting (in $C$) the current "symbolic values" of all variables according to their updating along the partial path $p(k)$.

For example, consider the flowchart solution (Fig. 6) for estimating the point where a function $f$ takes on its maximum value. For the path

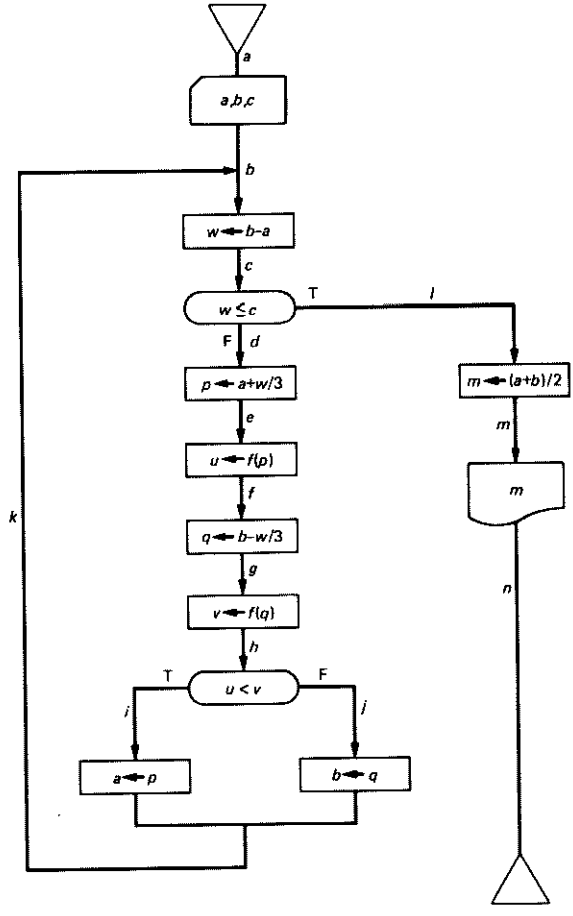



Fig. 6—Flowchart for estimating the point where a function is maximized.

$p = abclmn$, we compute

$$\begin{aligned} P(0) &= \text{true} \\ P(1) &= P(0) \wedge \text{true} = \text{true} \\ P(2) &= P(1) \wedge \text{true} = \text{true} \\ P(3) &= P(2) \wedge (b - a \leqq c) = (b - a \leqq c) \\ P(4) &= P(3) \wedge \text{true} = (b - a \leqq c) \end{aligned}$$

and finally,

$$P = P(5) = P(4) \wedge \text{true} = (b - a \leqq c),$$

noting that it was necessary to substitute $a - b$ for $w$ in the condition for traversing edge 1 because of the earlier assignment statement.

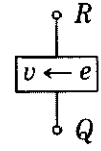
In general, this process of continually updating the symbolic values of program variables as we proceed along a path is called *symbolic execution* (or *symbolic evaluation*). The data descriptions generated in symbolic execution provide a precise representation of the changing *program state*. Initially, the program state is the three-place vector:

$$\begin{aligned} \text{state} &= [\text{start, values (start), pathpred (start)}] \\ &= (\text{start, } (\perp, \perp, \cdots, \perp), \text{true}), \end{aligned}$$

where "values" tabulates the symbolic values of all program variables ($\perp$ = undefined), and "pathpred" stores the inductively generated path predicate $P$ as described earlier. Symbolic names are assigned (in "values") to input variables whenever a read statement is encountered on the program path. Throughout the symbolic evaluation, all symbolic representations of variable and branch predicate values are then expressed in terms of these symbolic names, as representatives of input values. In particular, as one encounters an assignment statement ($v \leftarrow e$), the symbolic value of the program variable $v$ is updated (as in the example above) through substitution of the symbolic value of the expression $e$. In this way, "state" and especially the "values" vector will provide a continually updated snapshot of the program's development along the path. Moreover, the final value of the "pathpred" component of "state" provides the logical conjunction described above.

This path predicate $P$ defines a corresponding (path) subdomain of the input space $D$, and by the nature of the symbolic evaluation technique, $P$ is expressed as a set of conditions on the input variables alone. To generate a test case (of input data) that will cause the program to traverse the path $p$, it is then only necessary to find input values that satisfy all of these conditions. As we might expect, however, this is often easier said than done.

Before discussing this problem in any detail, it is better that we first describe an alternative to the above approach, one that proceeds in reverse—from the end of the path to its beginning. This technique, known appropriately as *backward substitution*, is best described in the survey paper by Huang.[12] To traverse a path $p$, certain conditions must be met, i.e., the set of branch conditions ($C$ or $\sim C$) along the path must be satisfied as they are encountered. On the other hand, suppose that an assignment statement ($v \leftarrow e$) intervenes, between "start" and the predicate $Q$, the latter representing a given branch condition (though modified by "partial backward substitution" as we are now describing). In the following flowchart segment:




if we want $Q$ to be true after the assignment ($v \leftarrow e$) has been executed, then the predicate $Q(v \leftarrow e)$ must be satisfied prior to its execution. Here, $Q(v \leftarrow e)$ is the predicate obtained by substituting the expression $e$ for each occurrence of $v$ in $Q$ [and we speak of $Q(v \leftarrow e)$ as the predicate obtained by *dragging* $Q$ *backward* through the indicated assignment statement]. It follows that the conjunction

$$R \wedge Q(v \leftarrow e)$$

is necessary for our passage along the edge with condition $R$ (through the assignment) and then to satisfy $Q$.

Altogether, if we want the specific path $p$ to be traversed in a program's execution, then we must drag each of its edge conditions backward to "start", and the conjunction of all resulting predicates must be satisfied by the corresponding test case of input data. Note once again that we obtain in this way a corresponding path predicate:

$$P = P1 \wedge P2 \wedge \cdots \wedge Pn,$$

i.e., a conjunction of modified branch conditions, each expressed in terms of the input variables to the program.

Consider once again the example of Fig. 6, and suppose we wish to traverse the path *abcdefghikclmn*. The listing shown below traces the dragging of the three necessary branch conditions backward along this

path:

| | | | |
|---|---|---|---|
| $l$ | $w \leqq c$ | | |
| $c$ | $w \leqq c$ | | |
| $k$ | $b - a \leqq c$ | | |
| $i$ | $b - p \leqq c$ | $u < v$ | |
| $h$ | $b - p \leqq c$ | $u < v$ | |
| $g$ | $b - p \leqq c$ | $u < f(q)$ | |
| $f$ | $b - p \leqq c$ | $u < f(b - w/3)$ | |
| $e$ | $b - p \leqq c$ | $f(p) < f(b - w/3)$ | |
| $d$ | $b - (a + w/3) \leqq c$ | $f(a + w/3) < f(b - w/3)$ | $\sim(w \leqq c)$ |
| $c$ | $b - (a + w/3) \leqq c$ | $f(a + w/3) < f(b - w/3)$ | $\sim(w \leqq c)$ |
| $b$ | $b - a - (b - a)/3 \leqq c$ | $f(a + (b - a)/3) < f(b - (b - a)/3)$ | $\sim(w \leqq c)$ |
| $a$ | $b - a - (b - a)/3 \leqq c$ | $f(a + (b - a)/3) < f(b - (b - a)/3)$ | $\sim(w \leqq c)$ |

One finally obtains the conjunction of three predicates:

$$P1 : b - a - (b - a)/3 \leqq c$$
$$P2 : f(a + (b - a)/3) < f(b - (b - a)/3)$$
$$P3 : \sim(b - a \leqq c)$$

all expressed in terms of the inputs $a$, $b$, $c$ (and the "called" function $f$). For purposes of comparison, the reader may try to compute an equivalent predicate using the symbolic execution method.

In an overall comparison of these two methods, one can identify an obvious "trade-off." With backward substitution, we avoid the costly storage facility needed for the continuous updating of all the symbolic program variable values. On the other hand, an important advantage accrues to the symbolic execution method, one that is not available for the backward substitution technique. Namely, we are more easily able to determine whether a given path is (or will be) feasible. And we can make the determination early in the symbolic evaluation. We need only check that the inductively generated predicates $P(k)$ are noncontradictory, as far as they go. We begin with $P(0)$ = true—certainly there is no contradiction here. Then, in the expression for $P(k)$ in terms of $P(k - 1)$, we have only to see whether $ibp(v[k - 1], v[k])$ contradicts $P(k - 1)$. If so, $P(k)$ and hence $P$ itself is contradictory, and the path $p$ is infeasible. Otherwise, we keep going. Note, in comparison, that with the backward substitution method, we wouldn't know whether a path was feasible until all of the calculation (of $P$) was completed—a definite disadvantage.

One must note, however, that all such "logical satisfiability" problems as we are now beginning to consider are exceedingly difficult to handle in practice. We include here the satisfiability question that results from the use of the "backward substitution" technique or the forward "symbolic evaluation" method, whichever is used. At the

conclusion of the backward substitution, we have a system of constraints on the inputs to the problem, and unless these constraints can be "solved" for the input data, we don't have a test case at all. The same may be said for the forward symbolic execution, except for the slight advantage that we can be determining the satisfiability (or lack thereof) as we go.

Huang, in his survey paper,[12] presents a systematic approach for handling the satisfiability problem, and we now outline the major features of his plan. The simplifying assumption is made that the path predicate takes the form:

$$P = P1 \wedge P2 \wedge \cdots \wedge Pn,$$

where the $Pi$ are nonnegated atomic expressions:

$$d\ R\ e$$

with $d$, $e$ arithmetic expressions in the input variables and $R$ one of the six relational operators: $<, \leqq, =, \langle\rangle, \geqq, >$. Such a system of atomic logical expressions can readily be rewritten in the *prenex normal form*:

$$(Ex1)(Ex2) \cdots (Exn)(x1 = e1) \wedge (x2 = e2) \wedge \cdots \wedge (xn = en),$$

where the $E$'s are "existential quantifiers" on auxiliary variable $x$'s, and the new expressions (the $e$'s) are differences of $d$, $e$ above, sufficient to transform any inequalities to equalities. The inequalities are, in effect, shifted to the auxiliary variables, thereby serving to normalize the solution space. Thus, in place of the inequality $2(b - a)/3 \leqq c$ at the end of the table above, we would have $(Ex1 \geqq 0)[x1 = c - 2(b - a)/3]$. Altogether, the three inequalities of that problem are similarly transformed, and we have instead the prenex normal form:

$$(Ex1 \geqq 0)(Ex2 \geqq 0)(Ex3 > 0)$$

$$\begin{aligned} x1 &= c - 2(b - a)/3 \\ x2 &= b + 2a - 6 \\ x3 &= b - a - c, \end{aligned}$$

one that is somewhat easier to handle.

From this point, standard techniques of linear algebra can be used to further transform the system into one where a minimum number of variables are involved. Thus, in the case of our running example, we can simplify the system so as to finally obtain:

$$\begin{aligned} &(Ex1 \geqq 0)(Ex2 \geqq 0)(Ex3 > 0) \\ &\quad 3x1 - x2 + 3x3 = 6 - 3a. \end{aligned}$$

From here, one may "guess" a solution, e.g., $x1 = x2 = 0$ and $x3 = 0.1$

say. One thereby obtains:

$$a = 1.9$$
$$b = 2.2$$
$$c = 0.2,$$

an input set that will cause the program to traverse the path *abcdefghikclmn* in Fig. 6, as originally required.

If we are going to have to "guess" a solution to the feasibility question in the end, however, the outright "trial and error" approach of Ramamoorthy et al.[26] offers an attractive alternative. One makes the assumption, as before, that the path predicate $P$ is expressed as a logical conjunction:

$$P = P1 \wedge P2 \wedge \cdots \wedge Pn,$$

where each of the $Pi$ is a constraint on the program's input variables. Moreover, it is assumed that the input variables have been ordered as $v[1], v[2], \cdots, v[m]$. With each variable $v[i]$, we associate a set $S[i]$ of conjuncts from $P$, namely:

$$S[i] = \{Pj : \text{only } v[1], \cdots, v[i] \text{ occur in } Pj\}$$

and these sets are then used as the basis for the "trial and error" algorithm shown in Fig. 7.

Assuming that values have been found for $v[1], \cdots, v[i-1]$ satisfying all the conjuncts in $S[1], \cdots, S[i-1]$, we either solve for $v[i]$ or randomly choose $v[i]$, depending on whether the set $S[i]$ contains an equality relation in $v[i]$. We then substitute this value in the conjuncts of $S[i]$. Should we thereby arrive at a contradiction, we "backtrack" to the iteration $i-1$, generating a different value for $v[i-1]$. Otherwise, we go ahead to the iteration $i+1$. If the complete iteration on $i$ concludes successfully, we arrive thereby at a "satisfiable" test case for the input variables of the program; otherwise we do not. Note that the "key" to the method is the fact that at each stage $i$, only the variable $v[i]$ has not yet been resolved. Note, however, that the loop at the right of Fig. 7 must include some criterion for deciding that "enough" random numbers have been tried in the current iteration. But however this is decided, it must be conceded that such an approach as presented here has much to offer in its favor, particularly considering the difficulty of the "satisfiability" question in general.

The authors[26] provide an example of the use of their algorithm on the "triangle classification problem" considered earlier. More generally, they suggest that the method has proven to be successful in treating a much wider class of problems. We would note further that the method could conceivably be applied to the "running satisfiability" questions that arise in the use of the "symbolic execution" technique.

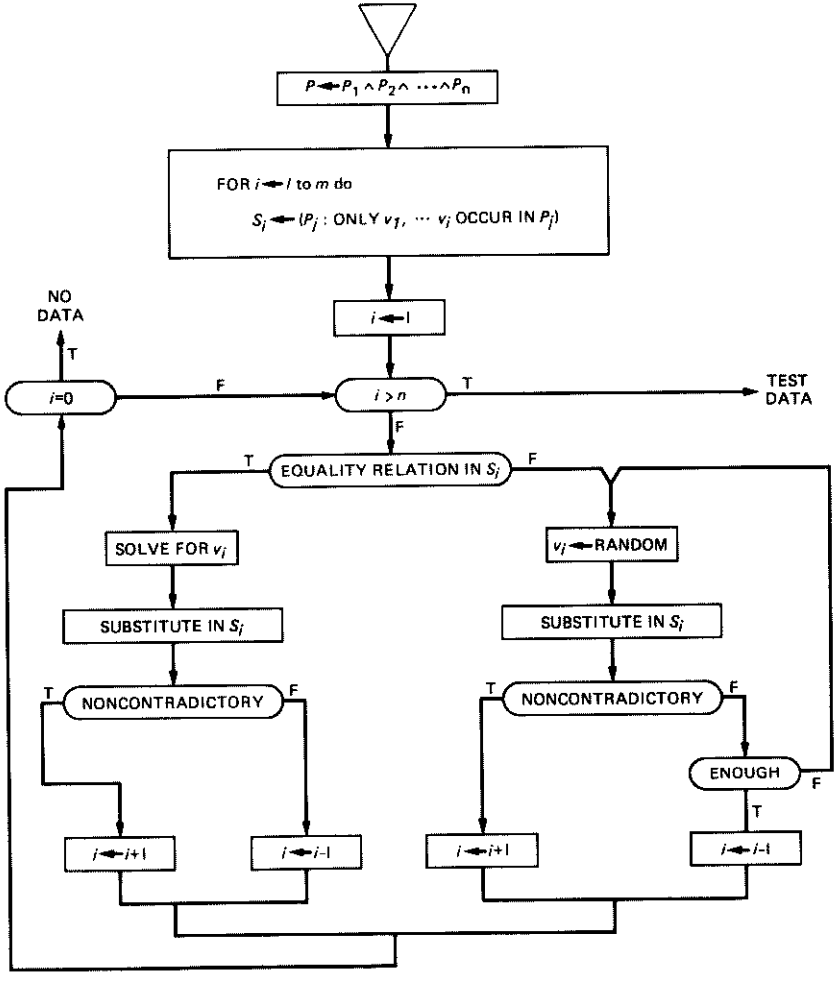



Fig. 7—Trial and error algorithm for solving the satisfiability problem.

In fact, it seems that this "trial and error" approach has a definite place—at least as a method of last resort, to be used as a component of any overall testing methodology.

Without some technique such as this, we are forced to rely on the extremely costly and not wholly reliable methods of "mathematical programming," particularly those routines that are designed to generate solutions to systems of inequalities. We cannot always assume that our systems are linear, in spite of the assumptions made by some authors. And in the absence of such an assumption, the problem is quite a difficult one, generally beyond the capability of the packages that are currently available.

Recognizing this, a most unusual and quite promising approach has been suggested by Kundu.[27] The idea is to combine the "test path selection" and "test case generation" phases of the solution, using the previous test case(s) $t[k]$ to help in determining the next test case $t[k+1]$. The result is a sequence of determinations:

$$(t[0]\rightarrow)p[0]\rightarrow t[1]\rightarrow p[1]\rightarrow \cdots$$

starting from an initial test case $t[0]$, chosen at random. The method is as follows:

1. Analyze $t[k]$: Execute the program with input $t[k]$, and determine its execution path $p[k]$. Then perform a (partial) symbolic execution of $p[k]$, so as to determine (an approximation to) its path predicate $P[k]$.
2. Select next test case: Determine the next test case $t[k+1]$ so that it violates at least one constraint in each of the path predicates $P[j]$, for $j < k$.

We are thus assured that each new test care $t[k+1]$ causes the program to traverse a genuinely new path, different from all those previously chosen.

In comparison with the previous methods we have discussed, Kundu reverses the roles of the test paths and the test data. The path $p[k]$ is determined from $t[k]$ in order to guide the next test case $t[k+1]$ away from previous paths. That is, $p[k]$ is not used for finding an input that corresponds to that path itself. Therein lies the novelty of the approach.

Moreover, Kundu's method is definitely not designed with any specific measure of test thoroughness in mind. (He asserts that no good measures of testedness are available, anyway.) It is clear, however, that one could easily augment his procedure with test instrumentation devices, as discussed earlier, for the purpose of assuring that some standard test coverage criterion (e.g., branch coverage) has been met.

The primary advantage of Kundu's method is easily understood. Consider the constraint on $t[k+1]$ as described in (2) above, i.e.,

$$t[k+1] \text{ not in } P[1] \cup P[2] \cup \cdots \cup P[k].$$

It is clear that the "forbidden region" for $t[k+1]$ thus represents only a small portion of the total input domain $D$ (see Fig. 8). This is so because the number of test cases generated in the testing activity is very small compared with the total number of executable paths in the program. Intuitively, the determination of the required $t[k+1]$ should thus be relatively easy. And for the same reason, the determination of a test datum in a given path domain (as is required in the usual strategy) should be more difficult. Kundu (see Ref. 27, pp. 176–177)

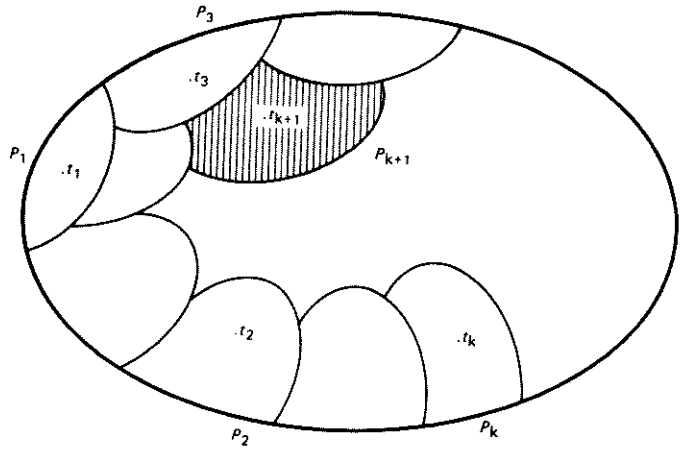



Fig. 8—Illustration of the forbidden region for selecting test cases.

gives a more detailed account of this reasoning, and the thrust of his argument is quite compelling. The reader may wish to consult Kundu's article for these additional details.

## VI. CONCLUDING REMARKS

We have attempted to describe the many interesting and varied approaches to the program testing problem. Whereas no single approach to the problem may hold all the answers, it seems that there are enough good ideas around as to suggest the feasibility of a workable methodology, based on one or another combination of the strategies that have been advanced to date.

It must be remembered, however, that the thrust of our presentation, and, indeed, the main thrust in the literature has been toward the "unit test" level, where smaller programs are encountered. Thus, the ideas we have presented are, at the present time, feasible only in the case of programs of limited size. To think that we are nearing the point where we are ready to apply all of these techniques to the testing of an entire operating system or a compiler would be to miss the point completely. Nevertheless, our study has shown that indeed a start has been made.

We have tried to present a reasonably balanced survey of the recent contributions to the research literature on software testing methodology. It is perhaps likely that one or more worthwhile studies have escaped the author's attention, and therefore, their omission from this survey should not reflect on their importance to the development of the field. Moreover, the author can only hope that the studies that have been cited here have been presented in their best light. Limita-

tions of time and space have prevented a more complete treatment of these works, and for this, apologies to the authors are in order. At the same time, this author would like to acknowledge the use of the many cogent examples from the literature cited, hoping as well that these and other contributions have been faithfully reported.

In conclusion, the author would like to thank W. H. Leung, K. A. Gluck, and N. H. Petschenik for their most helpful comments in reviewing an earlier draft of the manuscript.

## REFERENCES

1. W. C. Hetzel (ed.), *Program Test Methods*, Englewood Cliffs, NJ: Prentice Hall, 1973.
2. G. J. Myers, *The Art of Software Testing*, New York: John Wiley and Sons, 1979.
3. E. F. Miller, "Program Testing: Art Meets Theory," *Tutorial: Software Testing and Validation Techniques*, Miller and Howden (eds.), New York: IEEE, 1978, pp. 390–8.
4. E. F. Miller, "Program Testing Technology in the 1980's," *Tutorial: Software Testing and Validation Techniques*, Miller and Howden (eds.), New York: IEEE, 1978, pp. 399–406.
5. E. F. Miller, "Introduction to Software Testing Technology," *Tutorial: Software Testing and Validation Techniques*, Miller and Howden (eds.), New York: IEEE, 1978, pp. 3–14.
6. E. F. Miller, M. R. Paige, J. P. Benson, and W. R. Wisehart, "Structural Techniques of Program Validation," *Tutorial: Software Testing and Validation Techniques*, Miller and Howden (eds.), New York: IEEE, 1978, pp. 262–5.
7. J. B. Goodenough and S. L. Gerhart, "Toward a Theory of Test Data Selection," IEEE Trans. on Software Eng., *SE-1*, No. 2 (June 1975), pp. 156–73.
8. J. B. Goodenough, "A Survey of Program Testing Issues," in *Research Directions in Software Technology*, P. Wegner (ed.), Cambridge, MA: MIT Press, 1979, pp. 316–40.
9. R. G. Hamlet, "Test Reliability and Software Maintenance," Proc. Computer Software and Applications Conf. *COMPSAC 78*, November 13–16, 1978, Chicago, IL, New York: IEEE, 1978, pp. 315–20.
10. E. J. Weyuker and T. J. Ostrand, "Theories of Program Testing and the Application of Revealing Subdomains," IEEE Trans. Software Eng., *SE-6*, No. 3 (May 1980), pp. 236–46.
11. E. J. Weyuker and T. J. Ostrand, "Current Directions in the Theory of Testing," Proc. Computer Software and Applications Conf., *COMPSAC 80*, October 27–31, 1980, Chicago, IL, New York: IEEE, 1980, pp. 386–9.
12. J. C. Huang, "An Approach to Program Testing," Computing Surveys, 7, No. 3 (September 1975), ppl. 114–28.
13. J. C. Huang, "Program Instrumentation and Software Testing," Computer, *11*, No. 4 (April 1978), pp. 25–32.
14. J. C. Huang, "Program Instrumentation: A Tool for Software Testing," *INFOTECH State of the Art Report, Software Testing*, Infotech Intl. Ltd. (1979), pp. 149–59.
15. W. E. Howden, "Methodology for the Generation of Program Test Data," IEEE Trans. Computers, *C-24*, No. 5 (May 1975), pp. 554–9.
16. W. E. Howden, "Reliability of the Path Analysis Testing Strategy," IEEE Trans. Software Eng., *SE-2*, No. 3 (September 1976), pp. 208–15.
17. W. E. Howden, "Introduction to the Theory of Testing," in *Tutorial: Software Testing and Validation*, Miller and Howden (eds.), New York: IEEE, 1978, pp. 16–19.
18. W. E. Howden, "A Survey of Dynamic Analysis Methods," in *Tutorial: Software Testing and Validation*, Miller and Howden (eds.), New York: IEEE, 1978, pp. 184–206.
19. L. A. Clarke, "Automatic Test Data Selection Techniques," INFOTECH State of the Art Report, Software Testing, Infotech Intl. Ltd., 1979, pp. 43–63.
20. M. R. Paige, "On Sizing Software Testing for Structured Programs," Intl. Symp. on Fault Tolerant Computing, New York: IEEE, June 1977, p. 212.
21. M. R. Paige, "An Analytical Approach to Software Testing," Proc. Computer

Software and Applications Conf., *COMPSAC 78*, November 13–16, 1978, Chicago, IL, New York: IEEE, 1978, pp. 527–31.
22. M. R. Paige, "Program Graphs, an Algebra, and Their Implication for Programming," IEEE Trans. Software Eng., *SE-1*, No. 3 (September 1975), pp. 286–91.
23. T. J. McCabe, "A Complexity Measure," IEEE Trans. Software Eng., *SE-2*, No. 4 (December 1976), pp. 308–19.
24. L. A. Clarke and D. J. Richardson, "Symbolic Evaluation Methods for Program Analysis," in *Program Flow Analysis*, Muchnick and Jones (eds.), Englewood Cliffs, NJ: Prentice Hall, 1981, pp. 264–300.
25. L. A. Clarke, "A System to Generate Test Data and Symbolically Execute Programs," IEEE Trans. Software Eng., *SE-2*, No. 3 (September 1976), pp. 215–22.
26. C. V. Ramamoorthy, S. B. Ho, and W. T. Chen, "On the Automated Generation of Program Test Data," IEEE Trans. Software Eng., *SE-2*, No. 4 (December 1976), pp. 293–300.
27. S. Kundu, "SETAR—A New Approach to Test Case Generation," INFOTECH State of the Art Report, Software Testing, Infotech Intl. Ltd., 1979, pp. 163–86.

## AUTHOR

**Ronald E. Prather,** B.S. and M.S. (Electrical Engineering), 1955 and 1958, respectively; M.A. (Mathematics), 1966, University of California, Berkeley; Ph.D. (Mathematics), 1969, Syracuse University. Dr. Prather is a Professor of Mathematics and Computer Science at the University of Denver. He spent the 1982–1983 academic year on sabbatical leave with the Software Quality Analysis group at Bell Laboratories in Denver. He is the author of *Discrete Mathematical Structures for Computer Science* (Houghton Mifflin, 1976) and *Problem Solving Principles: Programming With Pascal* (Prentice Hall, 1982).

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Parallel Fault Simulation Using Distributed Processing*

By Y. H. LEVENDEL,† P. R. MENON,† and S. H. PATEL†

(Manuscript received June 3, 1983)

This paper presents a method of performing fault simulation of digital logic circuits using a special-purpose computer with distributed processing. The architecture for true value simulation presented in an earlier paper can also be used for parallel fault simulation. The special-purpose computer consists of inexpensive microprocessors interconnected by either a time-shared parallel bus or a cross-point matrix. The cross-point matrix provides higher performance than the time-shared parallel bus. The performance of the proposed simulator is better by over two orders of magnitude than traditional logic fault simulation performed on a general-purpose computer. The power of the simulator is proportional to the number of microprocessors over a certain range.

## I. INTRODUCTION

Fault simulation is an important part of the logic circuit design process. It is a means of determining the behavior of a logic circuit in the presence of each one of a predefined set of faults.

One of the most common uses of fault simulation is in determining the set of faults detected by a proposed test sequence, i.e., its fault coverage. Adequate fault coverage (usually greater than 90 percent of single stuck faults) is necessary to guarantee that the test sequence will detect most of the manufacturing defects.

* This paper is based upon material to be submitted by S. H. Patel in partial fulfillment of the requirements for the Ph.D. in Electrical Engineering at the Illinois Institute of Technology. † Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

In test pattern generation, fault simulation is used to determine the faults that are detected by the tests already generated so they can be removed from consideration. A yet undetected fault is then selected as a target for the next test. Additional faults detected by a newly generated test may be determined by simulation. Thus, fault simulation is used frequently as a part of the test generation process.

Fault simulation is also used to construct fault dictionaries for fault isolation. Other uses of fault simulation include the evaluation of test point effectiveness and the evaluation of self-checking circuitry. Effective test points are essential for factory testing. It is much cheaper and easier to locate and repair failures during manufacture than it is in the field. Also, good self-checking circuitry makes it easier to isolate faults in the field.

Currently, fault simulation is carried out on large general-purpose computers. This method has seen some use in large-scale integrated (LSI)* designs, but suffers from excessive run time at current levels of integration. Its applicability to very large-scale integration (VLSI) is doubtful, at least in the manner that it is currently used.[1] There is a need for more sophisticated and cost-effective fault simulators as very large simulation time and costs will result when dealing with circuits of VLSI complexity (more than 100,000 gates on a single chip).

## II. PARALLEL FAULT SIMULATION

A number of different algorithms have been developed for performing fault simulation efficiently on general-purpose computers. Among these the best known and widely used are the parallel,[2] deductive,[3] and concurrent[4] methods. All these methods attempt to simulate the effects of a number of individual faults simultaneously. This paper will consider the use of the parallel fault simulation algorithm in the special-purpose simulation hardware architecture developed in Ref. 5.

In parallel fault simulation the fault-free circuit and several different faulty circuits are processed simultaneously. The number of faulty circuits simulated in parallel is normally constrained by the number of bits in the host computer word. One bit of the computer word represents the signal value on a line in the fault-free circuit, while the remaining bits represent values on the same line in the presence of different single faults. Word-oriented operations performed on the host computer imply that the fault-free and faulty circuits are handled simultaneously and in exactly the same manner.

The output fault word of a gate is computed by simple word-oriented logic operations on the input fault words. The logic operations performed on the fault words correspond to the logic operation performed

* Acronyms and abbreviations are defined in the Glossary at the back of this article.

by the gate. Faults are injected using predefined masks. A stuck at 1 fault on a lead is injected by ORing the fault word with a mask containing a 1 in the bit position for the faulty value and 0's elsewhere. Similarly, a stuck at 0 fault can be injected by ANDing the fault word with a mask containing a 0 in the bit position for the faulty value and 1's elsewhere.

Two logic values are not sufficient for accurate logic simulation. Since each bit position in the computer word can represent only a logical 0 or 1, more than one bit per signal is required for multiple-value simulation.[6] In this case more than one word is required. For three-value representation two bits are required to represent each signal and, therefore, two computer words are required for representing a set of fault-free and faulty values. Since a pair of words are used to represent a signal, some sort of coding method is required to implement parallel simulation. A commonly used method of coding denotes one of the words as the 0-word and the other word as a 1-word. Let $i_0$ represent the $i$th bit in the 0-word and $i_1$ represent the $i$th bit in the 1-word. Then the $i_0i_1 = 01$ combination represents a logical 1, the $i_0i_1 = 10$ combination represents a logical 0, the $i_0i_1 = 00$ combination represents an unknown, and the $i_0i_1 = 11$ combination is unused. Simple word-oriented operations are still sufficient for performing the parallel simulation. This coding is the same as that in Ref. 7. Faults are injected in the 0-word and the 1-word using a 0-mask and a 1-mask, respectively. This injection is also done using simple logic operations. The method for simulating three logic values can be extended to any number of logic values by coding them using a sufficient number of bits.[8] Only three-valued simulation is considered in this paper.

The most widely used method of parallel simulation is the *event-driven* method. Event-driven simulation means that an element is not simulated unless there is a change in one of its input fault words. The main operations performed in an event-driven simulation are processing of active elements and scheduling changes to occur in future. The scheduling is done on a timing wheel. A *timing wheel* is a list structure in which events are chained together in the order they are to occur. In parallel fault simulation an element is considered to be *active* if the fault word associated with its output changes. The fault word is considered changed even if only one of its bits changes. All the values (i.e., the whole fault word) are propagated even if only one of the values changes.

Since the number of faults simulated at one time is restricted by the length of the host computer word, multiple passes through the simulator are necessary to simulate a large number of faults. It is possible to reduce the number of passes by using extra computer words

and simulating more faults during one pass. For example, 64 computer words can be used to simulate (two-value simulation) 1024 faults on a computer with a word size of 16 bits. The string of values in the set of computer words representing the fault-free and faulty signal values on a line will be called the *value vector.* The configuration of the value vector is shown in Fig. 1. The value vector consists of $L_p$ word pairs for three-value simulation. Two bits at the same position in the two words of a pair represent the signal value of one faulty circuit. Simulating $L_p$ word pairs at a time is better than making $L_p$ passes through the simulator since the overhead involved in the fanout search associated with each pass is saved. (However, as we will see in Section 6.1, some of these savings are lost due to increased activity as the number of faults per pass increases.) The number of words used in the value vector is usually constrained by the space requirements of the computer.

During simulation, operations are performed on value vectors by considering word pairs. Thus, the time required to perform an operation on the value vector is proportional to the number of word pairs used in the value vector. For example, if there are several word pairs in a value vector, then faults are injected one word pair at a time.

The whole value vector is considered active if any of the values in the vector changes. Furthermore, all the word pairs are propagated even if only one word pair changes. An element is considered *active* if the value vector associated with its output is active.

## III. CONCURRENCY IN FAULT SIMULATION

At least three types of concurrencies exist in fault simulation of logic circuits. The first type of concurrency occurs in the actual simulated hardware, the second type occurs in the simulation algo-

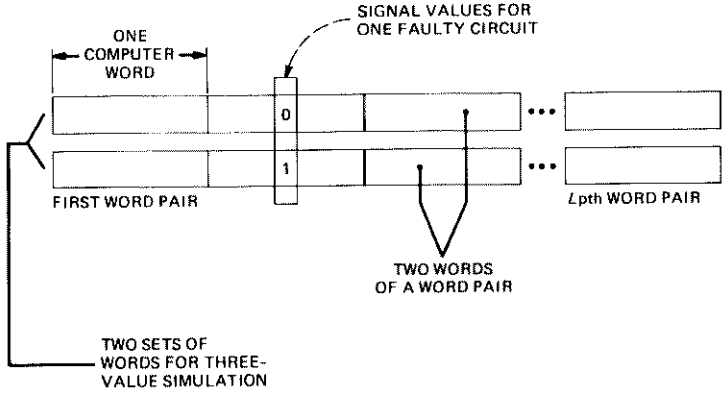



Fig. 1—Value vector configuration.

rithm, and the third type occurs in the form of fault activity. The first two types of concurrencies also occur in true value simulation of logic circuits and have been discussed in an earlier paper.[5]

The concurrency occurring in the actual simulated hardware can be called *logic circuit concurrency.* Utilizing this type of concurrency leads to distributed processing with identical processors performing identical tasks. The architecture for true-value simulation developed by Levendel et al.[5] and Denneau et al.[9–11] takes advantage of this type of concurrency.

The concurrency occurring in the simulation algorithm can be called *algorithm concurrency.* This concurrency is indirectly due to the concurrency occurring in the actual simulated hardware. Since several elements can be simultaneously active and a sequence of steps is to be performed for each active element, they can be processed in a pipeline fashion. Utilizing this type of concurrency leads to functional partitioning of tasks among several processors and a pipelined architecture. The architecture for true-value simulation developed by Barto and Szygenda[12] and Abramovici et al.[13] takes advantage of this type of concurrency.

For efficient fault simulation, a number of faults are simulated simultaneously in software-based simulators. This leads to *fault activity concurrency,* which can be utilized in special-purpose hardware for fault simulation.

This paper extends the architecture for true-value simulation described in Ref. 5 to fault simulation using the parallel method. The architecture takes advantage of the parallelism due to logic circuit concurrency and fault activity concurrency. The main difference between true-value simulation and fault simulation is in the algorithm executed by the individual processing units.

## IV. SPECIAL-PURPOSE ARCHITECTURE

The simulator consists of one master and a number of slaves (processors) interconnected by a communication structure (Fig. 2). The communication structure is used as a medium for transferring data between the slaves and between the slaves and the master. The communication structure can be either a time-shared parallel bus or a cross-point matrix. The circuit to be simulated is partitioned into subcircuits and each subcircuit is simulated in a separate processor. Subcircuits in different processors become active as signal values proceed from the primary inputs to primary outputs. As simulation progresses, data are transferred between subcircuits as the logic values on the signal connections between two subcircuits change. These data are transported across the communication structure. Typical data sent across the data path consist of element information and changed value

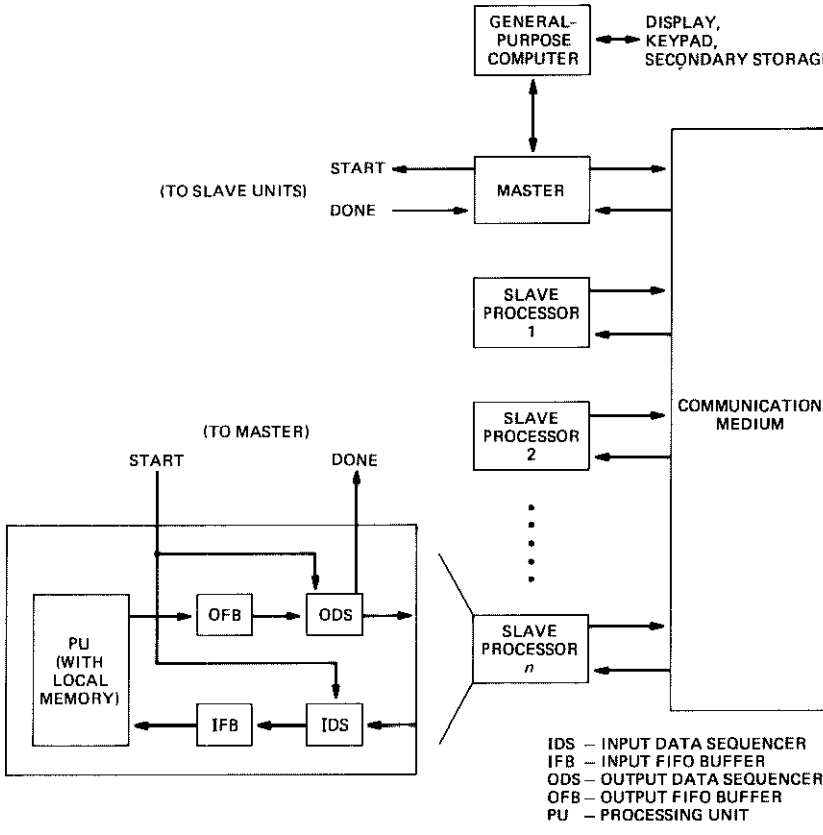



Fig. 2—Multiprocessor-based digital logic simulator.

vectors. The architecture of the simulator has been described in detail in the previous paper.[5] A summary of the architecture description is presented in Appendix A for completeness.

The overall architecture for true-value simulation is applicable to fault simulation since the algorithms for the two types of simulations are the same except that fault simulation requires:

1. Carrying of faulty signal values in addition to true-value signals (more than one word may be used to allow representation of a larger number of faulty signals)
2. Fault injection using masks
3. Multiple passes if the fault set is large.

The processing time per pass will be higher for fault simulation due to the extra processing required for injecting faults and manipulating multiple-word pairs in a value vector. The latter requirement applies only when the value vector contains more than one word pair. As far as communication between the processors is concerned, the data

transferred between the processors are greater than those transferred for true-value simulation, since the signal values require one or more complete words in addition to a word for the element number.

## V. ARCHITECTURE EVALUATION

The processing time per simulation cycle and the communication times using a time-shared parallel bus and a cross-point matrix are estimated first. These results are then used for selecting the communication structure. A multiple-bus communication structure is also discussed in this section.

### *5.1 Processing time $t_p$*

The average number of active elements per processor during a simulation cycle for true-value simulation is given by $kN/n$, where $N$ is the average number of active elements per simulation cycle during true-value simulation, $n$ is the number of processors in the multiprocessor simulator, and $k$ is the average unbalance factor representing the extra active elements per processor during a simulation cycle due to nonideal partitioning.[5] Ideal partitioning will cause an equal number of elements to be active in all the processors during all simulation cycles. However, because of some imbalance such as the fanout of all active elements during one simulation cycle not feeding equally into all the processors, some processors will have more active elements than the others during some simulation cycles. The average number of active elements per processor during a fault simulation cycle can be written as $kN_f/n$, where $N_f$ is the average number of active elements per simulation cycle in one pass during fault simulation. The value of $N_f$ is expected to be larger than $N$. Indeed, experimental runs on the Logic Analyzer for Maintenance Planning (LAMP) simulator[14] show that the overall activity (total number of active elements) during parallel fault simulation increases by a factor of about 2 for 16 faults per pass and by about 3.5 for 1024 faults per pass compared to true-value simulation. These results are averaged over several runs made using both combinational and sequential circuits with sizes ranging from 420 gates to 1912 gates. The number of faults simulated ranged from 1020 faults for the 420-gate circuit to 5046 faults for the 1912-gate circuit. The LAMP simulator is based on the deductive method. A mapping mechanism from deductive simulation activity to parallel simulation activity was implemented to predict the results for parallel simulation.

During one simulation cycle the following major operations occur in the given order:

1. Using the current list of events, list $L_t$ of the timing wheel (Fig.

11 in Appendix B), update, and find fanout of the elements whose outputs changed during the current simulation cycle.

2. Using the next list $L_{t+1}$ of the timing wheel, prepare external events to be transmitted to other processors for the next time interval.

3. From data in the Input FIFO Buffer (IFB) (sent by other processors), update and find fanout of the elements active during the current simulation cycle.

4. Evaluate the fanout of active elements (this includes fault injection).

5. Schedule on timing wheel elements whose output changes.

The detailed algorithm is given in Appendix C.

Let $t_a$ be the time required to process one active element. The average processing time per processor during one simulation cycle is then given by:

$$t_p = \frac{kN_f}{n} t_a.$$

Assume a microprogrammable microprocessor (e.g., Am2900 series) for each slave unit processing unit (PU) and the following operation times (150 ns cycle time): memory-to-memory move = 1.2 $\mu$s, memory-to-register move = 0.6 $\mu$s, and memory-to-memory logical AND/OR operation = 1.5 $\mu$s. Using these major microprocessor operations, the execution time for each operation in the parallel simulation algorithm described in Appendix C can be estimated. For example, obtaining each fanout of an updated element (after the fanout list has been accessed) takes one memory-to-register instruction and moving the element number to the Output FIFO Buffer (OFB) takes one memory-to-memory instruction. The processing times per active element for the various steps of the algorithm are shown in Table I, where $f_i$ is the average fan-in, $f_o$ is the average fan-out, and $L_p$ is the number of word pairs in the value vector.

The total processing time per element during one simulation cycle is the sum of all the expressions in Table I:

$$t_a = 9.6 + 8.7L_p + 3f_o + 3f_iL_p - \frac{(2.4 + 7.2L_p)}{f_o} \mu s.$$

Taking typical values of $f_i$ and $f_o$ to be 2 and an unbalance factor of $k = 1.1$, the processing time for a simulation cycle becomes:

$$t_p = (15.8 + 12.2L_p) \frac{N_f}{n} \mu s. \quad (1)$$

### *5.2 Communication time $t_c$*

The value of $t_c$ will depend on the type of communication structure.

Table I—Simulation cycle timings for parallel simulation

| Step | Expression ($\mu$s) |
|---|---|
| Update data from timing wheel | $\frac{(1.8 + 2.4f_o)}{f_o}$ |
| Prepare external events for next time interval | $\frac{(f_o - 1)(6.6 + 3.6L_p)}{f_o}$ |
| Update data from IFB | $\frac{(f_o - 1)(1.2 + 3f_o + 3.6L_p)}{f_o}$ |
| Evaluate schedule | $2.4 + 1.5L_p + 3f_iL_p + \left(\frac{3.6}{f_o}\right)$ |

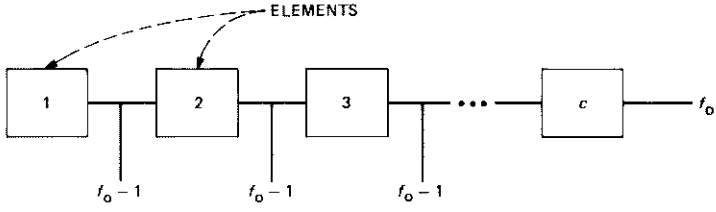



Fig. 3—An element string.

The two types of communication structures discussed in Ref. 5, namely the time-shared parallel bus and the cross-point matrix, will be considered here also. The partitioning algorithm discussed in the previous paper[5] partitions a circuit along its depth rather than its breadth. Since the signals in a circuit propagate in parallel, this places concurrent activities in different blocks. The same partitioning algorithm will be assumed here, since during fault simulation the signals still propagate in the same manner. The logic circuit to be simulated is partitioned into elements strings (see Fig. 3). The average number of communication events generated by one active element during a simulation cycle that have to be sent over the communication structure is:

$$e = \frac{[f_o + (c - 1)(f_o - 1)]}{c},$$

where $f_o$ is the average fanout and $c$ is the average number of elements in one element string. The typical value of $f_o$ can be taken as 2, and for large circuits $c$ is expected to be greater than 10. For $f_o = 2$ and $c = 10$, $e$ will be equal to 1.1.

#### *5.2.1 Time-shared parallel bus*

The total communication time during a simulation cycle for true-value simulation is given by[5]:

$$t_{c(bus)} = (n + 200)Ne \text{ ns}$$

This expression assumes one word of data to be transferred per active element. For fault simulation with $w$ words of data to be transferred per active element the expression becomes:

$$t_{c(bus)} = (n + 200)N_f ew \text{ ns.}$$

The number of words to be transferred is given by:

$$w = 1 + 2L_p,$$

where $L_p$ is the number of word pairs per value vector used in simulation. One word is required to carry the element number and the $2L_p$ words carry the value vector. Taking the value of $e = 1.1$:

$$t_{c(bus)} = (1.1n + 220)(1 + 2L_p)N_f \text{ ns.} \quad (2)$$

#### *5.2.2 Cross-point matrix*

In a cross-point matrix-based communication structure several processors can be simultaneously sending data to other processors. The total communication time during a simulation cycle for true value simulation is given by[5]:

$$t_{c(matrix)} = \frac{200Nke}{n} + 50j \text{ ns,}$$

where $j$ is the number of events for which the destination processor is found busy, i.e., the destination processor is communicating with some other processor. Once again this expression assumes one word of data transferred per active element. For $w$ words of data to be transferred, the expression for fault simulation becomes:

$$t_{c(matrix)} = \frac{200(N_f)kew}{n} + 50jw \text{ ns.}$$

For $k = 1.1$, $e = 1.1$, $w = 1 + 2L_p$, and $j = 0.1N_f/n$ (the channel is found busy for 10 percent of the transfer requests), the above expression can be rewritten as:

$$t_{c(matrix)} = (247 + 494L_p)\frac{N_f}{n} \text{ ns.} \quad (3)$$

### *5.3 Choice of communication structure*

The expressions for the processing time per simulation cycle per active element and the communication times per simulation cycle per active element for the bus and matrix structures are plotted in Fig. 4. The expressions are plotted for value vector length of one word (16 bits/word), i.e., $L_p = 1$.

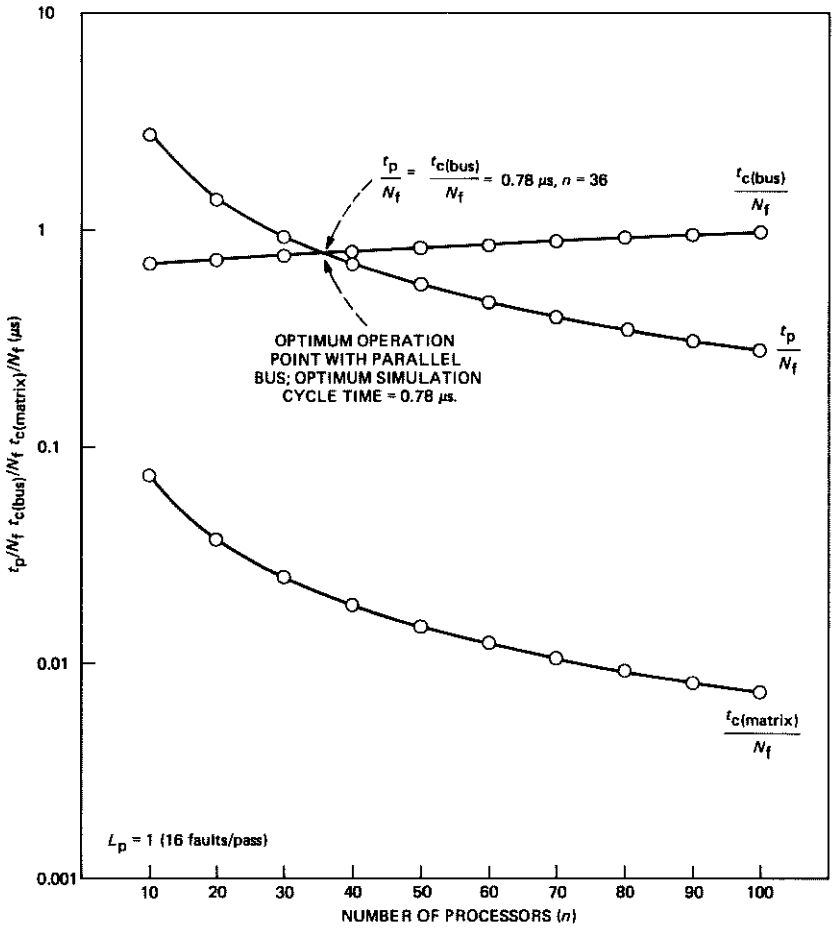



Fig. 4—Variation in processing and communication time.

The curves for the processing time and bus communication time for the parallel bus intersect at $n = 36$. The processing time is greater than the bus communication time for $n < 36$. Thus the processing time is the bottleneck. The processing time decreases as the value of $n$ increases. For $n > 36$ the bus communication time becomes larger than the processing time and the bus communication time becomes the bottleneck. Therefore, using more processors than $n = 36$ will not speed up the simulation. For optimum performance $n = 36$, and the length of the simulation cycle per active element becomes $t_m = 0.78$ μs. Based on the parallel simulation algorithm given in Appendix C, the actual simulation cycle length per active element for a single processor can be estimated as $t_1 = 24$ μs. The multiprocessor fault

simulator with a bus-based communication structure provides a speedup of 31 over the traditional single-processor logic fault simulator.

For further speedup a faster communication structure must be used. Figure 4 also shows the curve for the matrix communication time. This curve does not intersect with the curve for the processing time, and the communication time is always less than the processing time. The communication time will therefore never be a bottleneck. More processors can be added to speed up the simulator. For example, for $n = 100$ the speedup compared to the traditional single-processor logic fault simulator is 86 and for $n = 256$ the speedup is 220. The speedup of simulation is expected to be greater than two orders of magnitude for $n > 120$.

### *5.4 Multiple-bus communication structure*

The results of the previous section show that for a given number of faults per pass, the time-shared parallel bus is useful only for up to a fixed value of $n$. For further speedup the cross-point matrix has to be used. However, the cross-point is not used up to its maximum capability. For example, at 16 faults per pass and $n = 100$, the simulation cycle length is $280N_f$ ns while the communication cycle length is $7.4N_f$ ns, i.e., the communication structure is used less than 3 percent of the time. A communication structure that is slower and cheaper than the cross-point matrix might prove more cost-effective. This will be true especially since the control for the cross-point is very complex and thus expensive.

A communication structure that provides a capacity in between that of the time-shared parallel bus and the cross-point matrix is the multiple-bus structure. It consists of a bus arbitrator and several parallel buses. The configuration of the multiple-bus structure and its interface to the Output Data Sequencers (ODSs) and Input Data Sequencers (IDSs) of the slave units are given in Fig. 5. When the ODS needs to send data, it sets the Request To Send (RTS) line high and puts the destination address on the address lines. The requesting ODS keeps the RTS line high until granted a bus. The bus arbitrator grants it the use of the communication medium when it finds an unused bus and determines that the requested destination is not busy. The bus arbitrator grants the bus by setting the Bus Grant line high. The data are switched through the bus selector switch to the available bus. At the other end of the bus there is a line selector from which the data are sent to the destination processor. The ODS sends out all the data present in its Output FIFO Buffer (OFB). The data received by the IDS of the destination unit are put in its Input FIFO Buffer (IFB). The ODS then sets the RTS line low. This releases the bus, which is

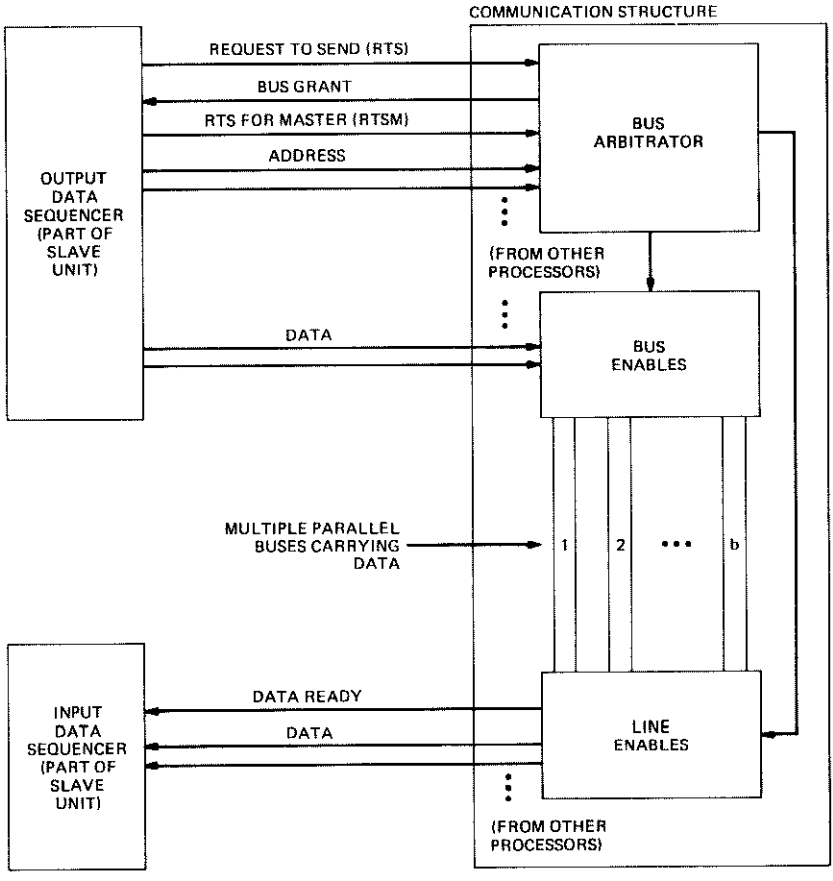



Fig. 5—Configuration of multiple-bus structure.

then granted by the bus control to another requesting slave or the master. All units have equal priority. The ODS will set the Request To Send line high again if it gets more data to transfer in the OFB.

The data sent out to a slave unit from another slave unit or the master consist of element information and changed value vectors. The data sent to a master consist of the address of the sending slave, element number (primary output or monitored point), and value vector. A separate line Request to Send to Master (RTSM) is used to address the master. When the destination is the master, the address lines from the ODS contain the sending slave unit address. This address together with the element number and value vector is stored in the master IFB by the master IDS.

To obtain an expression for the communication time for the multiple-bus structure, consider first of all the bus structure with only a

single bus. For this case, the communication time will consist of the same components as that for time-shared parallel bus[5]:

$$t_{c(mbus)} = (t_{brg} + t_{ds} + t_{da} + t_{br})(1 + 2L_p)(N_f)e,$$

where $t_{brg}$ is the bus request and grant time, $t_{ds}$ is the address and data setup time, $t_{da}$ is the data acknowledge, and $t_{br}$ is the bus release time. For the multiple-bus structure the bus request and grant time, $t_{brg}$, will be greater than in the parallel bus since extra checking has to be done before a bus is granted. The bus arbitrator will have to determine if a bus is available and if available then it has to further determine if the requested destination is busy. Assuming these extra actions double the time required for the bus request and grant time and the other times remain the same: $t_{brg} = 200$ ns, $t_{ds} = n$ ns, $t_{da} = 50$ ns, $t_{br} = 50$ ns, and $e = 1.1$. Each transaction across the multiple-bus has to wait for a bus to be granted. As more buses are added, the transactions can occur in parallel. Assuming the number of parallel buses is much smaller than the number of processors, the probability of the destination processor being busy will be small. The decrease in the total communication time will then be proportional to the number of buses. The expression for the total communication time for the multiple-bus communication structure becomes:

$$t_{c(mbus)} = \frac{(1.1n + 330)(1 + 2L_p)N_f}{b}, \tag{4}$$

where $b$ is the number of parallel buses.

Let $n_o$ be the number of processors at the optimum operation point, where $t_p = t_{c(mbus)}$. If the simulator operates with the number of processors not equal to $n_o$, then the speed of simulation will be lower. An expression for $n_o$ can be derived by equating eqs. (1) and (4):

$$n_o = -150 + 150\left(1 + b\left(\frac{0.49L_p + 0.64}{2L_p + 1}\right)\right)^{0.5}.$$

This expression is plotted for various values of $b$ with $L_p = 1$ in Fig. 6. It can be seen that higher performance is available with the multiple-bus structure compared to the single time-shared parallel bus for $b \geq 2$.

The implementation of the multiple-bus structure is expected to be less expensive than that of the cross-point matrix. The complexity of the communication structures and thus their cost is proportional to the number of switch points. For the cross-point matrix, the number of cross-points is proportional to the square of the inputs, i.e., $n^2$. For the multiple-bus structure two points have to be connected to establish a connection and thus the number of switch points is $2bn$. The cost of the multiple-bus structure will be less than that of the cross-point

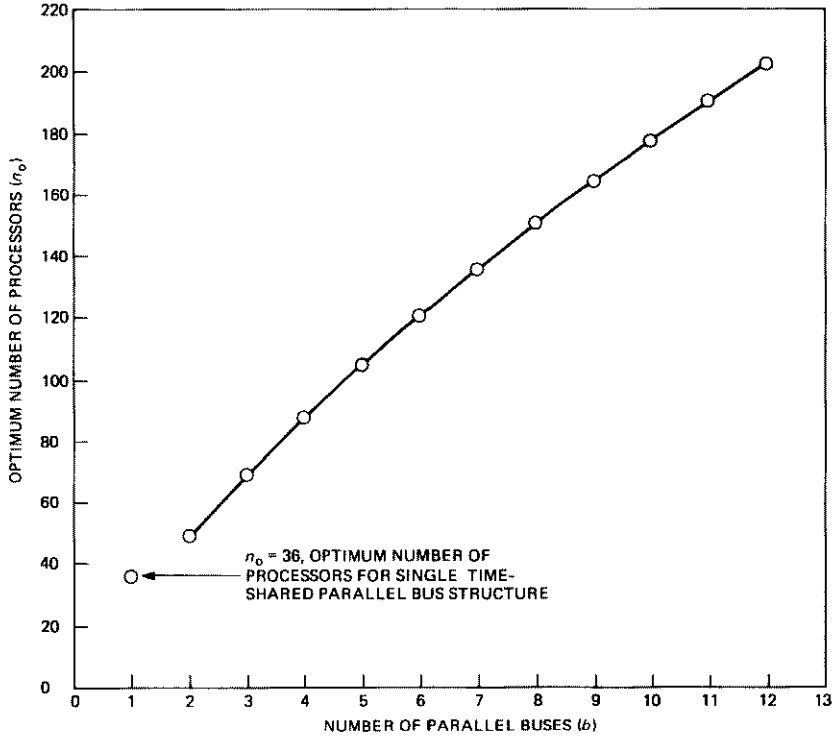



Fig. 6—Variation in the optimum number of processors with the number of parallel buses.

matrix as long as $2b < n$. As we saw in the analysis done earlier this will always be the case. For $b = 5$ and $n_o = 105$, the cost of the multiple-bus structure will be an order of magnitude lower than that of the cross-point matrix. Furthermore, it must also be noted that physical switching is not necessary in the case of the multiple-bus structure; it is cheaper to use logic enables for connecting an ODS and an IDS to a bus. This will result in even lower cost when compared with the cross-point matrix.

## VI. EFFECT OF NUMBER OF FAULTS PER PASS

When simulating a large number of faults per pass in parallel simulation, several pairs of words carrying the true and faulty values ($L_p$) must be manipulated per active element. All the faulty values in the words can be considered together as a vector. All the faulty values are evaluated even if only one faulty value is active.

The behavior of the multiprocessor fault simulator, using the parallel fault simulation algorithm, will be investigated for variations in the number of faults simulated per pass. Comparisons will be made

between 16, 32, 64, 256, and 1024 faults per pass, assuming a processor word length of 16 in all the cases.

Only the time-shared parallel bus and the cross-point matrix are discussed. The communication structure with multiple buses is not considered since the results for the time-shared parallel bus will apply, except for a scaling factor.

### *6.1 Simulator with time-shared parallel bus*

The expressions for processing time per simulation cycle, $t_p$, and the parallel bus communication time per simulation cycle, $t_{c(bus)}$, derived earlier in eqs. (1) and (2) are used to analyze the effects of variations in the number of faults per pass. It was seen earlier that the optimum operation point for the simulator occurs when the processing time and communication time are equal, i.e., $t_p = t_{c(bus)}$. An expression for the number of processors required to meet this condition for a given length of value vector ($L_p$) can be derived by equating the expressions for $t_p$ and $t_{c(bus)}$. Let $n_o$ be the number of processors at the optimum operation point and $t_o$ be the length of the processing and communication cycles at the optimum operation point. Values of $n$ smaller than the optimum $n_o$ cause the processing to be a bottleneck, while larger values of $n$ cause communication to be a bottleneck. Equating eqs. (1) and (2) yields the following expression for $n_o$, the number of processors required for optimum operation:

$$n_o = -100 + 100\left(\frac{3.11L_p + 2.44}{2L_p + 1}\right)^{0.5}.$$

Let $\alpha$ be the number of faults per pass. Then $\alpha = 16L_p$ assuming 16 bits per word. The above expression for $n_o$ can be rewritten as a function of the number of the faults $\alpha$, as:

$$n_o(\alpha) = -100 + 100\left(\frac{0.19\alpha + 2.44}{0.125\alpha + 1}\right)^{0.5}. \tag{5}$$

For $n = n_o(\alpha)$, the processing time per simulation cycle and the bus communication time per simulation cycle both reduce to:

$$t_o(\alpha) = (15.8 + 0.76\alpha)\frac{N_f(\alpha)}{n_o(\alpha)}. \tag{6}$$

For a simulator with optimum number of processors, $n_o(\alpha)$, the total time required to simulate a fixed number of faults is obtained by multiplying the time required for processing one active element ($t_o(\alpha)/N_f(\alpha)$) by the total number of active elements during the simulation ($N_T(\alpha)$):

$$T(\alpha) = \frac{t_o(\alpha)}{N_f(\alpha)} N_T(\alpha).$$

The total number of active elements during simulation is given by:

$N_T(\alpha) = N_f(\alpha) \times$ (number of simulation cycles per pass)

$\times$ (number of passes).

Substituting the expression for $t_o(\alpha)$ given in eq. (6):

$$T(\alpha) = (15.8 + 0.76\alpha)\frac{N_T(\alpha)}{n_o(\alpha)}. \tag{7}$$

Define the *simulation time ratio* as the ratio of the total time required to simulate a set of faults with $\alpha$ faults per pass to the total time required to simulate the given set of faults with 16 faults per pass, i.e., $T(\alpha)/T(16)$. Using a value of $n_o(16) = 36$ (eq. 5), the expression for the simulation time ratio is given by:

$$\frac{T(\alpha)}{T(16)} = \frac{(20.3 + 0.98\alpha)}{n_o(\alpha)} \frac{N_T(\alpha)}{N_T(16)}. \tag{8}$$

The values of the simulation ratio are first calculated theoretically and then compared with the experimental results.

#### *6.1.1 Theoretical simulation time ratio*

Let *simulation activity*, $N_T(\alpha)$, refer to the number of active elements during all passes of a simulation. The simulation activity can be expected to be inversely proportional to the number of faults per pass:

$$\frac{N_T(\alpha)}{N_T(16)} = \frac{16}{\alpha}.$$

For example, the expected simulation activity at 32 faults per pass will be half the simulation activity of 16 faults per pass since the number of passes needed will be halved. The theoretical expression for the simulation time ratio becomes:

$$\frac{T(\alpha)}{T(16)} = \frac{(20.3 + 0.98\alpha)}{n_o(\alpha)} \frac{16}{\alpha}. \tag{9}$$

Note that the theoretical simulation time ratio is independent of the simulation activity. Table II gives the variation of the optimum number of processors, $n_o$, and the variation of the simulation time ratio as a function of the number of faults per pass, $\alpha$. The theoretical results show that the simulation time ratio, and thus the total simulation time, decreases as the number of faults per pass increases.

Also, it is interesting to note that for 16 faults per pass the operation

Table II—Theoretical simulation time ratio

| Faults per Pass, $\alpha$ | Optimum Number of Processors, $n_o$ | Simulation Time Ratio, $\frac{T(\alpha)}{T(16)}$ |
|---|---|---|
| 16 | 36 | 1.0 |
| 32 | 32 | 0.81 |
| 64 | 29 | 0.72 |
| 256 | 26 | 0.65 |
| 1024 | 25 | 0.64 |

Table III—Experimental simulation time ratio

| Faults per Pass, $\alpha$ | Simulation Activity, $N_T(\alpha)$ | Optimum Number of Processors, $n_o$ | Simulation Time Ratio, $\frac{T(\alpha)}{T(16)}$ |
|---|---|---|---|
| 16 | 17586 | 36 | 1.0 |
| 32 | 9374 | 32 | 0.86 |
| 64 | 5204 | 29 | 0.84 |
| 256 | 1787 | 26 | 1.06 |
| 1024 | 689 | 25 | 1.6 |

peaks at 36 processors, while for 1024 faults per pass the operation peaks at 25 processors. As the number faults per pass increases, the point of optimum operation occurs for a slightly smaller number of processors. This is because the time required to transfer the increased data is more than the time required to process the increased data.

#### *6.1.2 Experimental simulation time ratio*

The values of $N_T(\alpha)$ averaged over several experimental runs made on the LAMP simulator[13] (with a mapping algorithm from deductive simulation to parallel simulation as discussed in Section 5.1) are given in Table III. Also shown in Table III are values for the simulation time ratio derived from eq. (8).

The experimental simulation time ratio, $T(\alpha)/T(16)$, is plotted together with the theoretical simulation time ratio in Fig. 7.

It is interesting to note that as the number of faults per pass increases, the experimental simulation time ratio falls below 1.0 initially, i.e., the simulation speeds up. After a certain point, the simulation time ratio then starts increasing and goes above 1.0. The fastest simulator is a 64 faults per pass simulator with 29 processors. On the other hand, the theoretical simulation time ratio always stays below 1.0 and keeps decreasing as the number of faults per pass increases. The difference in the experimental and theoretical results can be

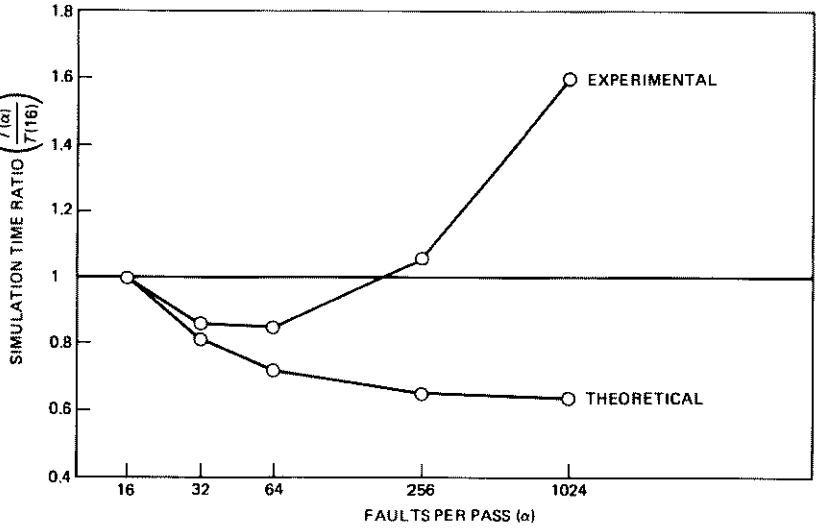



Fig. 7—Experimental and theoretical simulation time ratio for variation in number of faults per pass—a time-shared parallel bus.

explained by examining the variation in simulation activity as the number of faults per pass changes.

The curve for the theoretical simulation time ratio in Fig. 7 shows that the simulation speed increases as the number of faults per pass increases. This is as expected, since increasing the number of faults per pass decreases the number of passes and thus the fanout search and related processing time. In practice, however, there is extra simulation activity due to longer value vectors and this tends to increase the processing time. As more faults are simulated per vector, i.e., as the value of $\alpha$ gets larger, the simulation activity during the simulation will be higher than the theoretical. For example, as shown earlier, the theoretical activity at 32 faults per pass will be half the simulation activity at 16 faults per pass. However, the experimental simulation activity at 32 faults per pass will be more than the expected half. This is because the active faults in two value vectors at 16 faults per pass will not always directly map into one value vector at 32 faults per pass. Any active fault in the value vector will cause simulation activity even if the good value does not change. The effect of this is to cause extra schedulings. This increase in schedulings can be represented by an effective increase in the length of the simulation cycle and the number of simulation cycles. The runs made on the LAMP simulator show that most of the increase is in the length of the simulation cycle (i.e., more computation during the simulation cycle). The expected simulation activity and the actual simulation activity

obtained from runs made using the LAMP simulator are given in Table IV.

For 32 faults per pass, the simulation activity is only 1.07 times that theoretically expected. Thus the increase in processing time due to this extra activity is not substantial and the overall speed of simulation is higher due to the greater savings in the fanout search processing. For 1024 faults per pass, the simulation activity is 2.51 times that theoretically expected. In this case, the increase in processing time due to the extra activity is substantial compared to the savings obtained in the fanout search processing. This results in lowering the overall speed of simulation when compared with 16 faults per pass.

In summary, for the multiprocessor fault simulator with a parallel-bus-based communication structure, the fastest simulation speed occurs for 64 faults per pass and 29 processors. Note, however, that the number of processors required for 64 faults per pass is greater than the number of processors for 1024 faults per pass. Decreasing the number of processors for 64 faults per pass to 25 yields the total simulation time of 13,413 $\mu$s. This still favors the 64 fault per pass simulator over the 1024 fault per pass simulator.

### *6.2 Simulator with cross-point matrix*

The expressions for the processing time per simulation cycle, $t_p$, and the cross-point matrix communication time per simulation cycle, $t_{c(\text{matrix})}$, derived earlier in eqs. (1) and (3) are applicable to variations in the number of faults per pass. The increase in simulation activity caused by simulating more faults per pass will increase both the processing time and the communication time for the cross-point matrix. However, adding one word pair to the value vector will cause an increase in the communication time that is only 4 percent of the increase in the processing time. Thus, the communication time will always be less than the processing time. The cross-point matrix provides sufficient communication capacity for the parallel fault sim-

Table IV—Effect of multiple passes on simulation activity

| Faults per Pass, $\alpha$ | Experimental Simulation Activity, $N_T(\alpha)$ | Theoretical Simulation Activity, $N_T(\alpha)$ |
|---|---|---|
| 16 | 17,586 | 17,586 |
| 32 | 9374 | 8793 |
| 64 | 5204 | 4396 |
| 256 | 1787 | 1099 |
| 1024 | 689 | 275 |

ulator and does not cause a communication bottleneck. For faster fault simulation, more processors can be added.

The variation of the simulation time ratio as a function of the number of faults per pass will be investigated to obtain the optimum number of faults per pass. Since the processing time dominates, the total time required to simulate a set of faults with $\alpha$ faults will be equal to the total processing time. Using eq. (1):

$$T(\alpha) = (15.8 + 0.76\alpha)\frac{N_T(\alpha)}{n}.$$

The simulation time ratio is given by:

$$\frac{T(\alpha)}{T(16)} = (0.56 + 0.027\alpha)\,\frac{N_T(\alpha)}{N_T(16)}. \tag{10}$$

The experimental and theoretical simulation time ratios are plotted in Fig. 8. As was the case for the time-shared parallel bus, the experimental simulation time ratio increases after 64 faults per pass. This is because the time required to process the increased simulation activity is greater than the time saved in the fanout search overhead associated with each pass. For the multiprocessor fault simulator with a cross-point-matrix-based communication structure, the optimum number of faults per pass is 64. Using a different value for the number of faults per pass will decrease the speed of simulation. Note that the

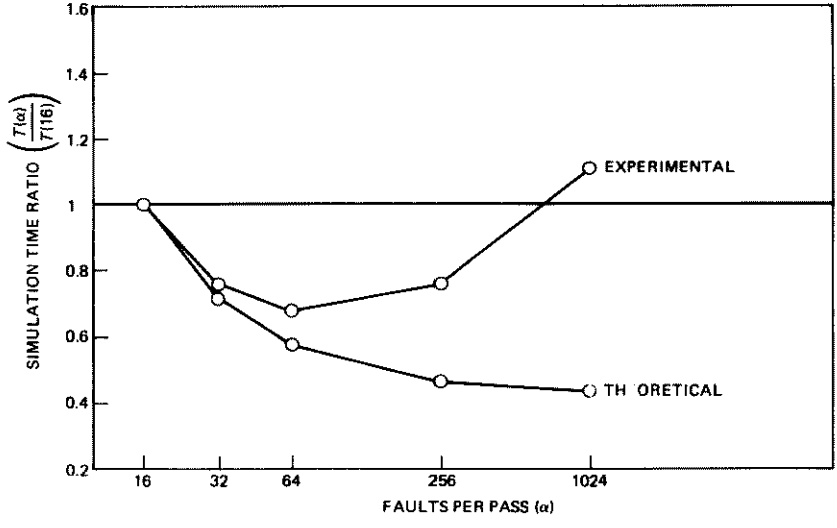



Fig. 8—Experimental and theoretical simulation time ratio for variation in number of faults per pass—cross-point matrix.

number of processors, $n$, does not affect the simulation time ratio. More processors can be added to obtain greater speed.

When compared with the parallel bus the cross-point matrix provides greater speed for any value of $n$ greater than the optimum $n$ for the parallel bus. For example, for 64 faults per pass, the cross-point matrix can be made faster than the parallel bus by selecting $n > 29$.

## VII. SUMMARY

In this paper we presented a special-purpose logic fault simulator based on the parallel simulation method. The simulator is expected to be two orders of magnitude faster than traditional logic fault simulators implemented on general-purpose computers. For both the time-shared parallel bus and the cross-point matrix, the simulator performs the best at 64 faults per pass. Decreasing the number of faults per pass slows down the simulation due to fanout search and other overhead required for every pass. Increasing the number of faults per pass also slows down the simulation due to the increase of simulation activity.

When the parallel bus is used, the power of the simulator can be increased over a certain range by increasing the number of slaves. The power of the simulator can be further increased by using the cross-point matrix.

The application of the special-purpose simulator to other fault simulation methods is being investigated currently.

## REFERENCES


1. D. F. Barbe (ed.), "Very Large Scale Integration (VLSI), Fundamentals and Applications," Berlin, Germany: Springer-Verlag, 1980.
2. E. W. Thompson and S. A. Szygenda, "Digital Logic Simulation in a Time-Based, Table-Driven Environment: Part 2. Parallel Fault Simulation," Computer, *8-3* (March 1975), pp. 38–49.
3. D. B. Armstrong, "A Deductive Method for Simulating Faults in Logic Circuits," IEEE Trans. Computers, *C-21*, No. 5 (May 1972), pp. 464–71.
4. E. G. Ulrich and T. G. Baker, "Concurrent Simulation of Nearly Identical Networks," Computer, *7-4* (April 1974), pp. 39–44.
5. Y. H. Levendel, P. R. Menon, and S. H. Patel, "Special-Purpose Logic Simulator Using Distributed Processing," B.S.T.J., *61*, No. 10, Part 1 (December 1982), pp. 2873–2909.
6. M. A. Breuer, "A Note on Three Valued Logic Simulation," IEEE Trans. Computers, *C-21*, No. 4 (April 1972), pp. 399–402.
7. S. G. Chappell, "Automatic Test Generation for Asynchronous Digital Circuits," B.S.T.J., *53*, No. 8 (October 1974), pp. 1477–1503.
8. Y. H. Levendel and P. R. Menon, "Fault Simulation Methods—Extensions and Comparison," B.S.T.J., *60*, No. 9 (November 1981), pp. 2235–58.
9. M. M. Denneau, "The Yorktown Simulation Engine," 19th Design Automation Conference, Las Vegas, Nevada, June 14–16, 1982, pp. 55–9.
10. E. Kronstadt and G. Pfister, "Software Support for the Yorktown Simulation Engine," 19th Design Automation Conference, Las Vegas, Nevada, June 14–16, 1982, pp. 60–4.
11. G. Pfister, "The Yorktown Simulation Engine: Introduction," 19th Design Automation Conference, Las Vegas, Nevada, June 14–16, 1982, pp. 51–4.

12. R. Barto and S. A. Szygenda, "A Computer Architecture for Digital Logic Simulation," Electronic Engineering, September 1980, pp. 35–66.
13. M. Abramovici, Y. H. Levendel, and P. R. Menon, "A Logic Simulation Machine," 19th Design Automation Conference, Las Vegas, Nevada, June 14–16, 1982, pp. 65–73.
14. S. G Chappell, C. H. Elemendorf, and L. D. Schmidt, "LAMP: Logic Circuit Simulators" B.S.T.J., *53*, No. 8 (October 1974), pp. 1451–76.
15. N. D. Phillips and J. G. Tellier, "Efficient Event Manipulation, A Key to Large Scale Simulation," IEEE 1978 Semiconductor Test Conference, Cherry Hill, New Jersey, October 31–November 2, 1978, pp. 266–73.
16. J. G. Vaucher and P. Duval, "A Comparison of Simulation Event List Algorithms," Comm. ACM, *18-4* (April 1975), pp. 233–30.

## APPENDIX A

### *Architecture Description*

#### *A.1 Introduction*

The simulator consists of one master and a multiplicity of slaves (processors) interconnected by a communication structure (see Fig. 2). The communication structure can be either shared or dedicated. The circuit to be simulated is partitioned into subcircuits and each subcircuit is simulated in a separate processor. The partitioning is such that the number of simultaneously active subcircuits (processors) is maximum and the number of simultaneously active elements in each subcircuit (processor) is minimum while keeping the communication from being a bottleneck.

The circuit to be simulated is initially modeled in the general-purpose computer, partitioned and loaded into the slave memories. The general-purpose computer performs all functions of input, output, and user interactions. The simulation is carried out in the multiprocessor simulator.

The simulator can be programmed to output intermediate results to the general-purpose computer. It also can be interrupted by the general-purpose computer for intermediate results. The user can ask for information about a simulation run while it is in progress (e.g., the status of a variable) and make certain run-time decisions like continuing simulation, applying extra input vectors, or stopping. After simulating each vector input, the simulation results and any other user requested information are sent to the general-purpose computer. User-requested information typically includes output values of elements (monitored points) at specific simulated times or under some other specified conditions. The general-purpose computer formats this information for suitable presentation to the user.

#### *A.2 Multiprocessor operation*

At the beginning of each simulation cycle the master sends primary input values (if any) to the appropriate slaves using the communication structure. The master then issues a start signal to the slaves. This

signal informs the slaves to start processing for the next simulation cycle. During the processing of a simulation cycle a slave unit may generate data for the other slaves or the master. The data are sent to the destination slave or the master using the communication structure. Only data for the subsequent time interval are transferred between the slaves to reduce the amount of information sent over the communication structure, thus minimizing the communication overhead. Therefore, the scheduled time is not sent.

Each slave informs the master when it has finished processing and transferring data for the current simulation cycle. When all slaves have informed the master about their completion of processing for the current simulated time interval, and also the master has finished transferring any primary input values scheduled for the next simulated time interval to the slaves, the master issues a start signal to the slaves for the next simulation cycle.

There are two signal lines between each slave unit and the master. The master signals the slaves using a START signal and the slaves signal the master using the DONE line. The START line from the master initiates processing for the next simulation cycle. The DONE line will become one when all the slaves have finished processing for the current simulation cycle.

### *A.3 Slave unit*

Each slave unit consists of a processing unit (PU), an input FIFO buffer (IFB), an output FIFO buffer (OFB), an input data sequencer (IDS), and an output data sequencer (ODS).

The slave unit PUs perform the actual fault simulation. The PU stores any data it has for other PUs or the master in the OFB. The ODS makes a request for the communication structure if there are any data to be transferred from the OFB. The ODS of the slave, if granted the use of the communication structure, takes data from the OFB and sends it over the communication structure to other slaves or the master. The data are received by the IDS of the destination slave or the master. Any data received by an IDS are put in its IFB. End of Data (EOD) flags are used to separate data streams since a PU can be writing new data to the OFB before its ODS has finished transferring current data, and similarly, an IDS can be receiving new data in the IFB before its PU has finished reading current data.

The slave unit PU operation can be described in terms of two essentially concurrent processes, namely the simulation cycle (execution of simulation algorithm) and the communication cycle (communication of events). Data generated during a simulation cycle are transferred as they are generated in a corresponding communication cycle (see Fig. 5). The algorithm executed by the Slave Unit PU is

similar to the traditional parallel simulation algorithm used on general-purpose computers except that it is modified to operate on a data structure distributed over several processors (see Appendix C).

A communication cycle is the period in between two START commands issued from the master. This cycle is phased with respect to the simulation cycle as shown in Fig. 9.

### *A.4 Master processor*

The master processor is the interface between the general-purpose computer and the simulator. Its main functions are to keep track of simulated time, keep the slaves in synchronism, supply the slaves with primary input values, and gather the primary output values from the slaves. The configuration of the master is similar to that of a slave unit and is shown in Fig. 10. It consists of a central processing unit (CPU) with some local memory, an IFB, an OFB, an IDS, and an ODS.

### *A.5 Communication structure*

The communication structure is used as a medium for transferring data between the slaves and between the slaves and the master. Either a shared or a dedicated structure can be used for the multiprocessor

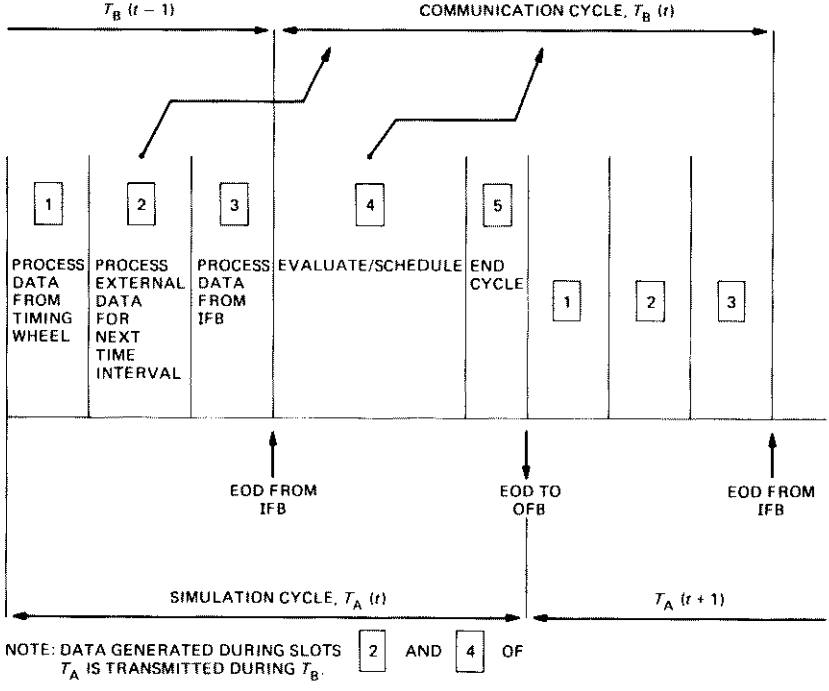



Fig. 9—Relationship between a simulation cycle and communication cycle.

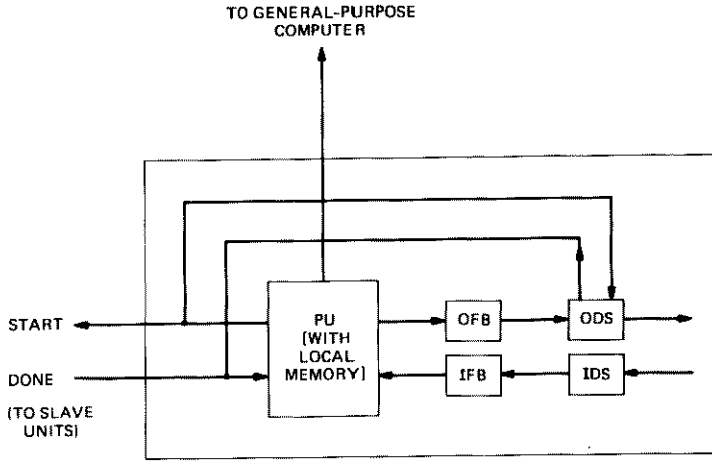



Fig. 10—Master configuration.

simulator. The details of a communication structure based on time-shared parallel bus and the cross-point matrix are discussed in detail in Ref. 5.

## APPENDIX B

### *Data Structure for Parallel Method*

The following data tables are used by each PU for its operation (see also Fig. 11):

1. Element table—This table contains the interconnection data and signal values for the circuit. For each element it contains the value vector, type, delay, fan-in list pointer and corresponding fan-in list, internal fanout list pointer and fanout list, external fanout list pointer and external fanout list. The value vector consists of a word pair. The first bit of each word of the word pair represents the good machine, while the remaining bits represent faulty machines. Three values can be represented: 0, 1, and X. If multiword parallel simulation is required, then additional word pairs are used to represent more faulty machines. The internal fanout list pointer and corresponding fanout lists give the fanout that remains in the subcircuit. The external fanout list pointer and corresponding fanout lists give fanout that goes to subcircuits located in other slaves. An element may have only internal fanout, only external fanout, or both internal and external fanout. Note that storing the external fanout takes up more space than storing an internal fanout, since both the destination processor address and element index have to be stored for the external fanout.

2. Activity list—This list is used to keep track of active elements during a simulation time interval. These elements are to be evaluated.

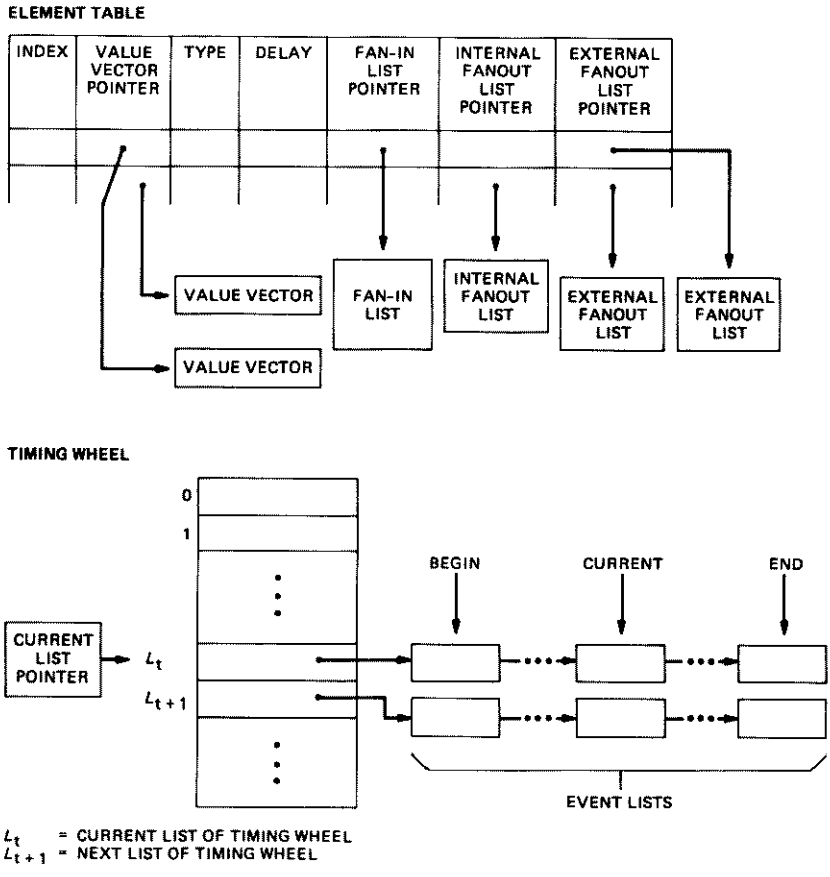



Fig. 11—Data tables for PU operation (parallel simulation).

3. Timing wheel—This data area contains the events that are scheduled in the future. A large amount of work has been done in this area.[15,16]

## APPENDIX C

### *Parallel Simulation Algorithm*

1. Update data from timing wheel

```
for each event in list L_t of timing wheel
  begin
    update value vector pointer in Element Table;
    for each fanout f of updated element
      if activity flag of f not set {
        set activity flag;
```

```
        put f on activity list;
      }
    end
  deallocate space on Timing Wheel;
```

2. Prepare external events for next time interval

```
  for each event in list L(t+1) of Timing Wheel
  begin
      if element has external fanout {
        for each external fanout {
          move destination processor address to OFB;
          move element number to OFB;
          for each word of value vector associated with event
            move word to OFB;
        }
        if element does not have internal fanout also
          deallocate space on Timing Wheel;
      }
    end
```

3. Update data from IFB

```
  enable interrupts from IFB;
  for each interrupt received from IFB
    begin
      if event is not EOD flag {
        for each word of value vector associated with event
          move word to Element Table;
        for each fanout f of updated element
          if activity flag of f not set {
            set activity flag;
            put f on activity list;
          }
      }
      else {
        disable interrupts from IFB;
        reset activity flags in Element Table;
        go to step 4 (Evaluate/Schedule);
      }
    end
```

(The EOD flag signifies end of data present in the IFB for the current simulation cycle. The EOD flag is loaded by the IDS when it receives a START signal from the master.)

4. Evaluate/schedule

```
  for each entry in activity list
    begin
```

```
            inject faults, if any, on inputs;
            for each set of words associated with input value vectors
            of element
              begin
                calculate corresponding output word;
                if output word changed from previous value
                  set change flag;
              end
            inject faults, if any, on outputs;
            if change flag set or faults injected {
              schedule event on Timing wheel;
              if (element delay is 1 && element has external fanout)
                for each external fanout {
                      move destination processor address to OFB;
                  move element number to OFB;
                  for each word of value vector associated with event
                    move word to OFB;
            }
        }
      end
```

5. End cycle

   store EOD flag in OFB;
   increment $t$;
   go to *STEP 1.*

## APPENDIX D

### Glossary

BIU—bus interface unit
CPU—central processing unit (part of a master unit)
EOD—end of data (flag)
IDS—input data sequencer
IFB—input first in first out (FIFO) buffer
$L_p$—number of word pairs used in parallel simulation
LAMP—logic analyzer for maintenance planning
LSI—large-scale integration
$L_t$—current list of timing wheel
$N$—average number of active elements per simulation cycle in true value simulation
$N_T$—total number of active elements during all passes of a simulation
$N_f$—average number of active elements per simulation cycle in fault simulation with 16 faults per pass
ODS—output data sequencer

OFB—output FIFO buffer
PU—processing unit (part of a slave unit)
T—time required to simulate a given set of faults
$c$—average number of elements in an element string
$f_i$—average fan-in of an element
$f_o$—average fanout of an element
$j$—number of events for which the channel is found busy in cross-point matrix
$k$—imbalance factor due to nonideal partitioning
$n$—number of processors in the multiprocessor simulator
$n_o$—number of processors required for optimum operation of multiprocessor simulator
$t_l$—length of simulation cycle for a single processor simulator
$t_a$—time required to process one active element
$t_{br}$—bus release time for a parallel bus structure
$t_{brg}$—bus request and grant time for a parallel bus structure
$t_c$—total communication time during one simulation cycle
$t_{c(bus)}$—total communication time during one simulation cycle for a single parallel bus structure
$t_{c(matrix)}$—total communication time during one simulation cycle for a matrix structure
$t_{c(mbus)}$—total communication time during one simulation cycle for a multiple parallel bus structure
$t_{da}$—data acknowledge time for a parallel bus structure
$t_{ds}$—address and data setup time for a parallel bus structure
$t_m$—length of simulation cycle for a multiprocessor simulator
$t_o$—average processing time, $t_p$, for number of processors $n = n_o$
$t_p$—average processing time per processor during one simulation cycle, where average processing time consists of the time required to process all active elements and schedule resulting events
VLSI—very large-scale integration
$w$—number of words to be transferred across the communication structure for one active event

## AUTHORS

**Ytzhak H. Levendel,** B.S.E.E., 1971, Technion-Israel; M.S.C.S., 1974, The Weitzman Institute of Science; Ph.D., 1976, University of Southern California; Bell Laboratories, 1976—. Mr. Levendel has done research in fault diagnosis and until lately was involved in the development of a logic and test design aid system. He is now a Supervisor in 5*ESS*.™ Member, IEEE, Eta Kappa Nu.

**Premachandran R. Menon,** B.S. (Electrical Engineering), 1954, Banaras Hindu University; Ph.D. (Electrical Engineering), 1962, University of Washington; Bell Laboratories, 1963—. Mr. Menon has done research in switching

theory and fault diagnosis and is currently involved in the development of a logic simulation system. Recipient of the Distinguished Technical Staff Award; member, IEEE.

**Suresh H. Patel,** B.E. (Electrical), 1975, University of Zambia; M.S.E.E., 1976, Illinois Institute of Technology; Association of American Railroads, 1977–1981; Bell Laboratories, 1981—. At the Association of American Railroads Mr. Patel was involved in hardware/software design, development and testing of a multi-microcomputer-based instrumentation system for freight trains. He was also involved in development of analytical and computer models on track circuit. At Bell Laboratories he is a member of the Processor Design Department. Mr. Patel was a part-time instructor at Illinois Institute of Technology during 1977–1978.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Two New Kinds of Biased Search Trees

By J. FEIGENBAUM* and R. E. TARJAN†

(Manuscript received May 20, 1983)

In this paper, we introduce two new kinds of biased search trees: biased, $a$, $b$ trees and pseudo-weight-balanced trees. A biased search tree is a data structure for storing a sorted set in which the access time for an item depends on its estimated access frequency in such a way that the average access time is small. Bent, Sleator, and Tarjan were the first to describe classes of biased search trees that are easy to update; such trees have applications not only in efficient table storage but also in various network optimization algorithms. Our biased $a$, $b$ trees generalize the biased 2, $b$ trees of Bent, Sleator, and Tarjan. They provide a biased generalization of $B$-trees and are suitable for use in paged external memory, whereas previous kinds of biased trees are suitable for internal memory. Our pseudo-weight-balanced trees are a biased version of weight-balanced trees much simpler than Bent's version. Weight balance is the natural kind of balance to use in designing biased trees; pseudo-weight-balanced trees are especially easy to implement and analyze.

## I. INTRODUCTION

The following problem, which we shall call the *dictionary problem*, occurs frequently in computer science. Given a totally ordered universe $U$, we wish to maintain one or more subsets of $U$ under the following operations, where $R$ and $S$ denote any subsets of $U$ and $i$ denotes any item in $U$:

access $(i, S)$—If item $i$ is in $S$, return a pointer to its location. Otherwise, return a special **null** pointer.

* Research done partly while a summer employee of Bell Laboratories and partly while a graduate student supported by Air Force grant AFOSR-80-042.

† Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

insert $(i, S)$—Insert $i$ in $S$, assuming it is not previously there.

delete $(i, S)$—Delete $i$ from $S$.

join $(R, S)$ (two-way join)—Return the set consisting of the union of $R$ and $S$, assuming that every item in $R$ precedes every item in $S$. This operation destroys $R$ and $S$, and can be regarded as a concatenation of $R$ and $S$.

split $(i, S)$—Split $S$ into three sets $L$, $I$, and $R$, where $L$ and $R$ are the sets of items strictly smaller and strictly larger than $i$, respectively, and $I = \{i\}$ if $i$ is in $S$ (three-way split), $I = \emptyset$ if $i$ is not in $S$ (two-way split).

One way to solve the dictionary problem is to store the items of each set in the external nodes of a search tree in left-to-right order, one item per external node. To guide the operations, the search tree also contains auxiliary items, called *keys*, in the internal nodes. The worst-case access time in a search tree is proportional to the depth of the tree. By imposing any one of a number of well-known *balance conditions* on the tree, we can guarantee that its depth is $O(\log n)$, where $n$ is the number of items it contains. Such a balance condition can be maintained during update operations by performing appropriate local rearrangements of the tree. With balanced search trees, each of the dictionary operations has an $O(\log n)$ time bound. Examples of balanced trees include height-balanced (AVL) trees,[1] 2, 3 trees,[2] $B$-trees,[3] and trees of bounded balance.[4]

In many applications of search trees the access frequencies of different items are different, and we would like our data structure to take this into account. To deal with this problem formally we assume that each item $i$ has a known weight $w_i$ providing an estimate of the access frequency. The *biased dictionary problem* is that of implementing the dictionary operations so that operations on heavier items are faster than those on lighter items. In particular, when representing a set $S$ as a search tree, we wish to bias the tree so as to approximately minimize its total weighted depth $\sum_{i \in S} w_i d_i$, where $d_i$ is the depth of the external node containing item $i$, while preserving the ability to update the tree rapidly. In addition to the five dictionary operations, we allow the following operation for changing the weight of an item:

*reweight* $(i, w, S)$—Redefine the weight of item $i$ in set $S$ to be $w$.

The following theorem, due to Shannon, gives a lower bound on the total weighted depth of a search tree:

*Theorem 1:*[5] *If $T$ is a search tree for a set $S$ and every node of $T$ has no more than $b$ children, then*

$$\sum_{i \in S} w_i d_i \geq \sum_{i \in S} \frac{w_i}{W} \log_b \frac{W}{w_i},$$

*where $W = \sum_{i \in S} w_i$ is the total weight of the items in $S$.*

In light of Theorem 1, our goal is to devise classes of search trees that are easy to update and have $d_i = O(\log_b W/w_i)$ for every item $i$. We call $O(\log_b W/w_i)$ the ideal access time of item $i$.

Bent, Sleator, and Tarjan[6–8] have devised several kinds of such biased search trees. Our work is an extension of theirs. A thorough discussion of previous work by others on the biased dictionary problem may be found in Ref. 8.

In their running time analyses, Bent, Sleator, and Tarjan used a technique called *amortization*, which we shall also use. The idea of amortization is to average the running time of individual operations over a (worst-case) sequence of operations. As a tool in deriving amortized time bounds we use *credits*. A credit will pay for one unit of computing time. To each operation we allocate a certain number of credits, called the *credit time* or *amortized time* of the operation. If a given operation does not need all its credits, we can save them for use in later operations; if an operation needs more than its share of credits we can use those previously saved. In any analysis using credits, the objective is to prove that an arbitrary sequence of operations can be performed without running out of credits.

Three points about amortization using credits are worth making. First, credits are a way of charging earlier operations for later ones. If a credit analysis is successful, we can assert that any sequence of operations requires an amount of actual time that is at most a constant multiplied by the sum of the credit times of the individual operations; slow operations are only possible if there are corresponding earlier fast ones. Second, although the word "average" appears in the description of the technique, it is not the usual kind of average-case analysis, and in particular we make no probabilistic assumptions; we obtain worst-case bounds holding for *any* sequence of operations. Third, credits serve as a kind of "potential energy": we place them in regions of search trees that may cause abnormally long update operations. This idea may illuminate the credit invariants we define below.

The remainder of the paper consists of three sections. In Sections II and III, we define and analyze biased $a$, $b$ trees, which generalize the biased 2, $b$ trees of Ref. 8 and provide a biased version of $B$-trees. Biased $a$, $b$ trees are a form of biased tree appropriate for paged external memory; earlier forms of biased trees are more appropriate for internal memory. In Section IV we introduce pseudo-weight-balanced trees, which give a biased version of weight-balanced trees much simpler than Bent's earlier version.[6] Weight balance is the natural kind of balance to use in designing a biased search tree; pseudo-weight-balanced trees are especially easy to implement and analyze.

We shall assume that the reader is familiar with search trees. In particular we shall not discuss the use and updating of keys, and we

shall draw freely on the results and techniques of Ref. 8. We shall use the terminology of Ref. 8 except that we use "external node" in place of "leaf". When appropriate we shall regard a node $x$ of a search tree as denoting the entire subtree rooted at $x$, with the context resolving whether a node or a tree is meant. The **null** node denotes the empty search tree. We denote the parent of a node $x$ by $p(x)$; $p(x) =$ **null** if $x$ is a tree root.

## II. LOCALLY BIASED *a, b* TREES

Our first class of biased trees uses height balance to guarantee fast access and variable node size to allow easy updating. The class is parameterized by two positive integer constants $a$ and $b$ such that $2 \leq a \leq \lceil b/2 \rceil$. The integer $b$ specifies the maximum allowed number of children of an internal node. If an internal node has at least $a$ children, we say it is *filled*; otherwise it is *underfilled*. (An external node is always filled.) Ideally, we would like every node to have at least $a$ children; since in our scheme this is impossible to achieve, we allow underfilled nodes but treat them specially.

If $x$ is a node in a search tree, we define its *weight* $w(x)$ to be the sum of the weights of all items in descendants of $x$. (We use the convention that every node is a descendant of itself.) We define the *rank* $s(x)$ of $x$ recursively by $s(x) = \lfloor \log_a w(x) \rfloor$ if $x$ is an external node, $s(x) = 1 + \max\{s(y) | p(y) = x\}$ if $x$ is an internal node. A node $x$ is *minor* if $x$ is not a root and either $s(x) < s(p(x)) - 1$ or $x$ is underfilled. All other nodes are *major*. A *locally biased a, b tree* is a search tree with the following properties (see Fig. 1):

1. Every internal node has at least two and at most $b$ children;

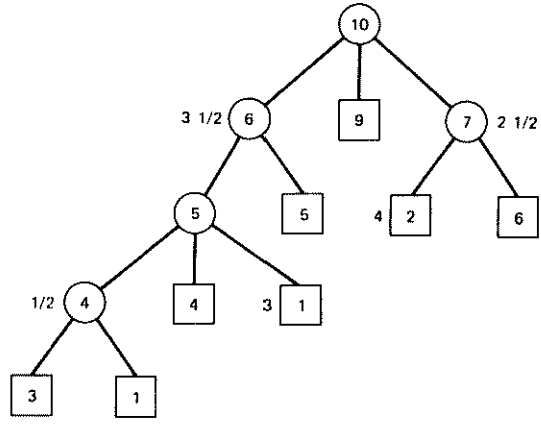



Fig. 1—A biased 3, $b$ tree. The numbers in nodes are ranks, and the numbers beside nodes are credits for the credit invariant.

2. If $x$ is a minor node, any adjacent sibling of $x$ is both major and external. When a node $x$ has this property, we say the tree is *locally biased* at $x$.

We call two nodes $r$-compatible if they can be adjacent children of a node of rank $r$ in a biased $a$, $b$ tree. That is, two nodes are $r$-compatible if both have rank at most $r-1$ and either at least one has rank $r-1$ and is external or both have rank $r-1$ and at least $a$ children each.

If $a = 2$ we obtain exactly the biased 2, $b$ trees of Bent, Sleator, and Tarjan. Our main new idea is in the definition and handling of underfilled nodes. Our first theorem guarantees that $a$, $b$ trees have ideal access time if $b$ is chosen appropriately.

*Lemma 1*: *Consider any biased $a$, $b$ tree. If $x$ is an external node, $a^{s(x)} \le w(x) < a^{s(x)+1}$. If $x$ is an internal node with at least one minor child, $a^{s(x)-1} \le w(x)$. If $x$ is an internal node with $k$ children, $k'a^{s(x)-2} \le w(x)$, where $k' = m\{k, a\}$.*

*Proof*: The first part of the lemma is immediate from the definition of rank. The second part follows from the first part and property 2: if $x$ has a minor child, it must have another child that is major and external. We prove the third part by induction on the height of $x$. If $x$ has a minor child, then $k'a^{s(x)-2} \le a^{s(x)-1} \le w(x)$ by the second part of the lemma. Otherwise, all children of $x$ are major. Let $y$ be a child of $x$. If $y$ is external, or internal with a minor child, then $a^{s(x)-2} = a^{s(y)-1} \le w(y)$. Otherwise, $y$ has at least $a$ major children, and by the induction hypothesis $a^{s(x)-2} = a \cdot a^{s(y)-2} \le w(y)$. Summing over the $k$ children of $x$ gives $ka^{s(x)-2} \le w(x)$. □

*Lemma 2*: *If $x$ is an external node in a biased $a$, $b$ tree of total weight $W$, the depth of $x$ is at most* $\log_a W/(w(x)) + 3$.

*Proof*: Let $r$ be the root of the tree and $d$ the depth of $x$. Since the rank increases by at least one from child to parent, $d \le s(r) - s(x)$. Lemma 1 implies $\log_a w(x) \le s(x) + 1$ and $\log_a W = \log_a w(r) \ge s(r) - 2$. Combining inequalities gives the lemma. □

*Theorem 2*: *A biased $a$, $b$ tree has ideal access time, with a constant factor proportional to* $\log_a b$.

*Proof*: Immediate from Lemma 2. □

According to Theorem 2, to minimize the access time in $a$, $b$ trees, we should choose $b$ as small as possible. The requirement $b \ge 2a - 1$ is necessary to allow efficient updating. The best choice of $b$ seems to be $2a - 1$ or $2a$. The choice $b = 2a$ allows purely up-down updates (see the end of Section III). The choice $b = 2a - 1$ gives a biased version of ordinary $B$-trees.[3] Any other choice gives a biased version of "hysterical" or "weak" $B$-trees.[9-11] Note also that Theorem 2 gives a worst-case example, not an amortized bound on the access time.

As Ref. 8 shows, all the update operations on search trees can be carried out using one or more joins. Our next task is thus to define a join algorithm for biased $a$, $b$ trees.

*Algorithm 1*: *local join* $(x, y)$. Join two locally biased $a$, $b$ trees with roots $x$ and $y$, assuming that all items in tree $x$ precede all items in tree $y$.

Case 0—$x =$ **null** or $y =$ **null.** If $x =$ **null,** return $y$; if $y =$ **null,** return $x$.

Case 1—$s(x) \geq s(y)$ and $x$ and $y$ are $(s(x) + 1)$-compatible, or $s(x) \leq s(y)$ and $x$ and $y$ are $(s(y) + 1)$-compatible. Create a new node with $x$ and $y$ as its children and return the new node.

Case 2—$s(x) = s(y)$ and $x$ and $y$ are not $(s(x) + 1)$ compatible. Let $u$ be the rightmost child of $x$ and $v$ the leftmost child of $y$ (see Fig. 2a). Perform *join*$(u, v)$, letting $w$ be the root of the resulting tree. If $s(w) < s(x)$, construct a node $z$ whose children are those of $x$ not including $u$, the node $w$, and the children of $y$ not including $v$. If $s(w) = s(x)$, construct a node $z$ whose children are those of $x$ not including $u$, those of $w$, and those of $y$ not including $v$ (see Fig. 2b). In either case, if $z$ has more than $b$ children, split it into two nodes $z'$ and $z''$ with $z$ as parent, dividing the old children of $z$ as evenly as possible between $z'$ and $z''$ (see Fig. 2c). Return $z$.

Case 3—$s(x) > s(y)$ and $x$ and $y$ are not $(s(x) + 1)$-compatible. Let $u$ be the rightmost child of $x$ (see Fig. 2d). Perform *join* $(u, y)$, letting $v$ be the root of the resulting tree. If $s(v) < s(x)$, replace $u$ as a child of $x$ by $v$. If $s(v) = s(x)$, replace $u$ as a child of $x$ by the set of children of $v$. If $x$ now has more than $b$ children, split it into two nodes $x'$ and $x''$ with $x$ as parent, dividing the old children of $x$ as evenly as possible between $x'$ and $x''$. Return $x$.

Case 4—$s(x) < s(y)$ and $x$ and $y$ are $(s(y) + 1)$-compatible. Symmetric to Case 3.

*Theorem 3*: *Given two biased a, b trees with roots x and y, the join algorithm produces a biased a, b tree whose root has rank* $\max\{s(x), s(y)\}$ *or* $\max\{s(x), s(y)\} + 1$. *In the latter case, the root of the new tree has exactly two children.*

*Proof*: An easy induction on rank shows that the root of the tree produced by the join has rank $\max\{s(x), s(y)\}$ or $\max\{s(x), s(y)\} + 1$ and that in the latter case the root has exactly two children. Furthermore, it is clear that every internal node in the new tree has at least two and at most $b$ children. All that remains is to show that any two adjacent siblings in the new tree are allowed to be adjacent by property 2. We prove this by induction on rank, using the same cases as in the algorithm.

Case 1—Assume without loss of generality that $s(x) \geq s(y)$. Since

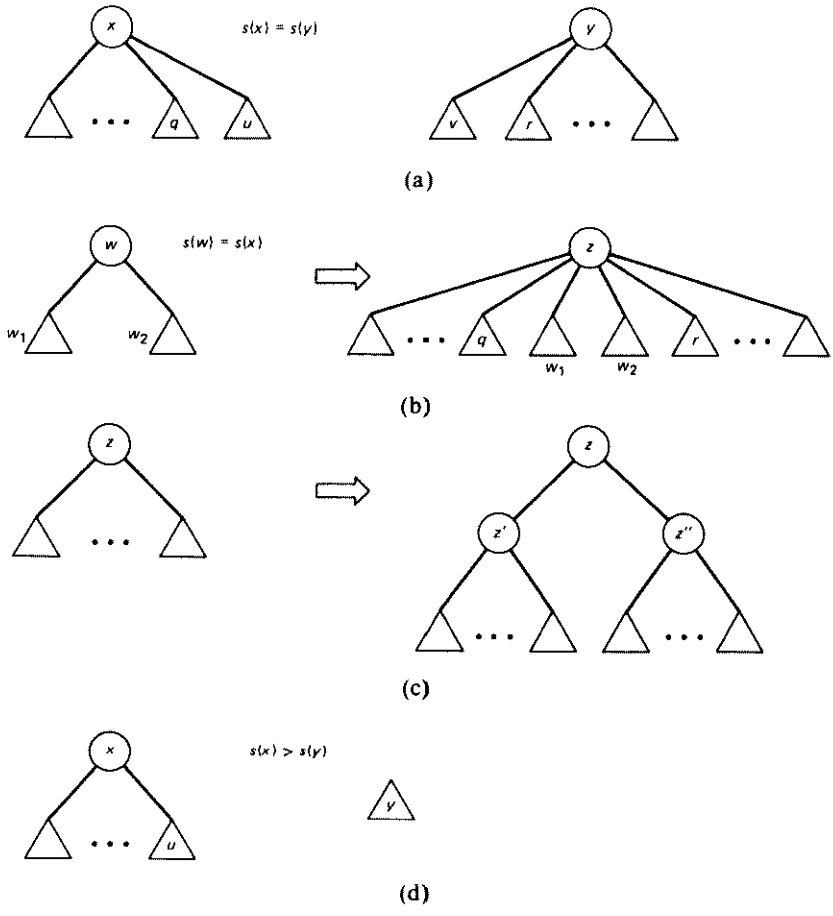



Fig. 2—Cases of the join algorithm. (a) Situation at the beginning of Case 2. (b) Recursive call *join* $(u, v)$ produces a tree of rank $s(x)$ with root $w$. (c) Division of an overfilled root. (d) Case 3.

$x$ and $y$ are $(s(x) + 1)$-compatible, the new tree is locally biased at $x$ and $y$.

Case 2—For the moment, ignore the split of $z$ if it takes place. Since $x$ and $y$ are not $(s(x) + 1)$-compatible, neither $x$ nor $y$ is external, and $u$ and $v$ exist. Suppose the left sibling of $u$, say $q$, is minor. Then $u$ is major and external, which means that the join of $u$ and $v$ will immediately terminate in Case 1, and the new right sibling of $q$ will be $u$. The symmetric statement holds for the right sibling of $v$. Finally, suppose $w$ is minor, i.e., $s(w) < s(x) - 1$ or $w$ has fewer than $a$ children. Then both $u$ and $v$ must be minor as children of $x$ and $y$, respectively, and both adjacent siblings of $w$ in the new tree will be major and external. Thus the tree before the split has property 2.

Splitting $z$ preserves property 2 since both new children of $z$ will have at least $a$ children of their own (this is where we use $b > 2a - 1$) and each will have rank $s(x)$. (Property 2 implies that before the split, of any two adjacent children of $z$, at least one has rank $s(z) - 1$).

Case 3—Similar to but simpler than Case 2. Case 4 is symmetric. □

To make our timing analysis as similar as possible to the one in Ref. 8 for biased 2, $b$ trees, we shall assume that credits can be divided in half, and that half a credit will pay for the work in one call of *join*, not counting the work in the recursive call. We use the following credit invariant: A nonroot node $x$ holds $s(p(x)) - s(x) - 1$ credits, plus an additional half credit if $x$ is underfilled.

*Theorem 4: The join algorithm runs in $O(|s(x) - s(y)|)$ amortized time. Specifically, carrying out the join while preserving the invariant takes $|s(x) - s(y)| + 1$ credits in Case 1 or 2, $|s(x) - s(y)| + 1/2$ credits in Case 3 or 4.*

*Proof*: We use induction on rank and a surprisingly complicated case analysis.

Case 1—Assume without loss of generality that $s(x) \geq s(y)$ and that if $s(x) = s(y)$, then $x$ is external or filled. We need half a credit for the work in the join and at most $s(x) - s(y) + 1/2$ credits to place on $y$, for a total of $s(x) - s(y) + 1$.

Case 2—The split of $z$, if it occurs, does not affect the credit invariant; therefore we ignore it. Assume without loss of generality that $s(u) \geq s(v)$. There are three subcases:

Subcase 2a—$s(w) = s(x)$ and *join* $(u, v)$ is a Case 1 join. The credits originally on $u$ and $v$ suffice to maintain the credit invariant on $u$ and $v$ after the join of $x$ and $y$ is completed. We have one credit on hand to join $x$ and $y$; we spend half for the work in the outer call and half for the work in the inner call.

Subcase 2b—$s(w) = s(x)$ and *join* $(u, v)$ is not a Case 1 join. To perform the join of $x$ and $y$ we are given one credit and can obtain at least $2s(x) - s(u) - s(v) - 2$ from $u$ and $v$, for a total of $2s(x) - s(u) - s(v) - 1$. We need half a credit for the work in the outermost call and at most $s(u) - s(v) + 1$ for the recursive call, for a total of at most $s(u) - s(v) + 3/2$. The difference between what we have and what we need is $2(s(x) - s(u)) - 5/2$. Since $s(x) - s(u) \geq 1$, we must find at most an additional half credit to spend.

If *join* $(u, v)$ is a Case 2 join, and $s(x) - s(u) = 1$, then either $u$ or $v$ is underfilled and yields an extra half credit. If *join* $(u, v)$ is a Case 3 join, we save half a credit on the join. Thus, in any case we can obtain the needed half credit.

Subcase 2c—$s(w) < s(x)$. As in Subcase 2b we have at least $2s(x) - s(u) - s(v) - 1$ credits to perform the join of $x$ and $y$. We need

one half for the work in the outermost call, at most $s(u) - s(v) + 1$ for the recursive call, and at most $s(x) - s(w) - 1/2$ to place on $w$, for a total of at most $s(x) + s(u) - s(v) - s(w) + 1$. The difference between what we have and what we need is $s(x) - s(u) + s(w) - s(u) - 2$. Since $s(x) - s(u) \geq 1$ and $s(w) \geq s(u)$, we have enough credits unless $s(x) - s(u) = 1$ and $s(w) = s(u)$, in which case we must find an extra credit.

Suppose $s(x) - s(u) = 1$ and $s(w) = s(u)$. Since $s(w) = s(u)$, *join* $(u, v)$ is not a Case 1 join, and $u$ is internal. If *join* $(u, v)$ is a Case 2 join, then both $u$ and $v$ are internal. Either both $u$ and $v$ are underfilled, giving us two additional half credits, or one of $u$ and $v$ is underfilled and $w$ is filled, giving us an extra half credit from $u$ or $v$ and saving a half credit that does not need to go on $w$. The only other possibility is that *join* $(u, v)$ is a Case 3 join, which saves us half a credit.

Furthermore in this case either $u$ is underfilled or $w$ is filled, either giving us an extra half credit from $u$ or saving us a half credit on $w$. Thus in all cases we obtain the necessary extra credit.

Case 3—We ignore the possible split of $x$, which does not affect the credit invariant. There are two subcases:

Subcase 3a—$s(v) = s(x)$. To perform the join we are given $s(x) - s(y) + 1/2$ credits and can obtain at least $s(x) - s(u) - 1$ more from $u$, for a total of $2s(x) - s(u) - s(y) - 1/2$. We need one half for the outermost call and at most $|s(u) - s(y)| + 1$ for the recursive call, for a total of $|s(u) - s(y)| + 3/2$. The difference between what we need and what we have is $2(s(x) - \max\{s(u), s(y)\}) - 2 \geq 0$.

Subcase 3b—$s(v) < s(x)$. To perform the join we have at least $2s(x) - s(u) - s(y) - 1/2$ credits. We need $|s(u) - s(y)| + 3/2$ for the outermost and recursive calls, plus at most $s(x) - s(v) - 1/2$ to place on $v$, for a total of $s(x) + |s(u) - s(y)| - s(v) + 1$. The difference between what we have and what we need is $s(x) - \max\{s(u), s(y)\} + s(v) - \max\{s(u), s(y)\} - 3/2$. We have enough credits unless $s(x) - \max\{s(u), s(y)\} = 1$ and $s(v) = \max\{s(u), s(y)\}$, in which case we need an extra half credit.

Suppose $s(x) - \max\{s(u), s(y)\} = 1$ and $s(v) = \max\{s(u), s(y)\}$. Then *join* $(u, y)$ is not a Case 1 join. If it is a Case 2 join, then either $u$ is underfilled, giving us an extra half credit, or $v$ is filled, saving us a half credit on $v$. If it is a Case 3 join, we save half a credit on the join. Thus in all cases we obtain the necessary half credit.

Case 4—Symmetric to Case 3. □

It is useful to restate Theorem 4 as follows. We say a biased $a, b$ tree with root $x$ is *cast to rank* $k$ if it satisfies the credit invariant and has $k - s(x)$ credits on its root. Theorem 4 implies that if $x$ and $y$ are

the roots of two biased $a, b$ trees cast to a rank $k > \max\{s(x), s(y)\}$, then they can be joined, without using additional credits, to produce a tree cast to rank $k$.

We can implement a split as a sequence of joins, exactly as described in Ref. 8. The following algorithm splits at an item $i$ in the tree:

*Algorithm 2*: *split* $(i, r)$. Split the biased $a, b$ tree with root $r$ at item $i$, assumed to be in the tree.

Locate the node $x$ containing item $i$. Initialize the current node to be $p(x)$ and the previous node to be $x$. Initialize the left and right trees to empty; they will contain the items smaller than and larger than $i$, respectively. Repeat the following step until the current node is **null** (the previous node is the root of the tree):

Split Step—Delete every child of the current node to the left of the previous node. If there is one such child, join it to the left tree; if there are two or more such children, give them a common parent and join the resulting tree to the left tree. Repeat this process with the children to the right of the previous node, joining the resulting tree to the right tree. Remove the previous node as a child of the current node and destroy it if it is not $x$. Make the current node the new previous node and its parent the new current node.

*Theorem 5*: *The split algorithm is correct and takes* $O(s(r) - s(x))$ *credit time, where* $x$ *is the node containing item* $i$.

*Proof*: Correctness follows immediately from the correctness of the join algorithm. The time bound follows as in Ref. 8; we shall sketch the idea. Let *cur*, *prev*, *left*, and *right* be the current node, the previous node, and the roots of the left and right trees, respectively. An induction shows that $s(prev) \geq \max\{s(left), s(right)\}$ just before each split step. Another induction shows that by allocating $O(s(cur) - s(prev))$ credits to a split step, we can carry out the step while preserving the invariant that the left and right trees are cast to a rank of $s(prev) + 1$. That is, the amortized time associated with a single step of the split is proportional to the rank difference of two consecutive nodes along the split path. If we sum over all split steps, then the sum telescopes, and we obtain the time bound in the statement of the theorem. □

Splitting at an item not in the tree is just like splitting at an item in the tree, except that the initialization is different. Let $r$ be the root of the tree, $i$ the item at which the split is to take place, $i^-$ and $i^+$ the largest item in the tree less than $i$ and the smallest item in the tree greater than $i$, respectively, and $x$ the *handle* of $i$, defined to be the nearest common ancestor of the external nodes containing $i^-$ and $i^+$. We can locate $x$ by searching down the path from $r$ to $x$ if appropriate keys are stored in the tree (see Ref. 8). To carry out the split, we combine all children of $x$ whose descendants contain items less than $i$

to form the initial left tree and all other children of $i$ to form the initial right tree. Then we initialize the previous and current nodes to be $x$ and $p(x)$, respectively, and repeat the split step until the current node is **null**. Such a split also runs in $O(s(r) - s(x))$ credit time.

We can implement insert, delete, and reweight as combinations of splits and joins: an insertion is a two-way split followed by two joins, a deletion is a three-way split followed by one join, and a weight change is a three-way split followed by two joins. Using the same kind of analysis as in Ref. 8, we obtain the following credit times for these three operations (we have stated the bounds in terms of weights rather than ranks):

$$\text{insert } (i, x)\text{: } O\left(\log_a\left(\frac{w(x) + w_i}{\min\{w_{i^-} + w_{i^+}, w_i\}}\right)\right),$$

where $i^-$ and $i^+$ are as defined above.

$$\text{delete } (i, x)\text{: } O\left(\log_a\left(\frac{w(x)}{w_i}\right)\right).$$

$$\text{reweight } (i, w, x)\text{: } O\left(\log_a\left(\frac{\max\{w(x), w\}}{\min\{w_i, w\}}\right)\right),$$

where $w(x)$ and $w_i$ are as defined before the weight change.

*Remark*: In all the time bounds derived in this section and the next the constant factor is proportional to $b$. □

## III. GLOBALLY BIASED *a, b* TREES

Local bias is sufficient to guarantee good amortized but not good worst-case running times for the dictionary operations. If it is important that every single operation be fast, we need a stronger balance condition. Figure 3 shows how a split can take more actual time than

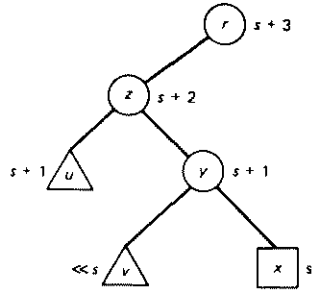



Fig. 3—Locally biased $a$, $b$ tree that cannot be split at $x$ in actual time $O(s(r) - s(x))$. Not all children of $y$, $z$, and $v$ are shown. Join of $u$ and $v$ can take an unbounded amount of time.

its credit time, and illustrates why local bias is not enough: a minor node that is the leftmost child of its parent has a constraint on its right but not on its left side, and symmetrically for a minor node that is the rightmost child of its parent. To overcome this problem we introduce globally biased $a$, $b$ trees.

If $x$ is a node in a search tree, we define $x^+$ to be the external node containing the smallest item greater than the largest item in a descendant of $x$. Symmetrically, $x^-$ is the external node containing the largest item less than the smallest item in a descendant of $x$. If $x$ is on the rightmost path of the tree, $x^+$ is undefined; if $x$ is on the leftmost path, $x^-$ is undefined. A *globally biased a, b tree* is a search tree with two properties (see Fig. 4):

1. Every internal node has at least two and at most $b$ children.

2. If $x$ is a minor nonroot node and $x^+$ is defined, $s(x^+) \geq s(p(x)) - 1$; if $x^-$ is defined, $s(x^-) \geq s(p(x)) - 1$. When a node $x$ has this property, we say the tree is *globally biased* at $x$.

Global bias implies local bias. Thus globally biased $a$, $b$ trees have ideal access time. We can join two globally biased $a$, $b$ trees using almost the same join algorithm as in Section II; the only difference is that the conditions determininng the cases are different. We shall call the algorithm in Section II *local join* to distinguish it from the following *global join* algorithm:

*Algorithm 3: global join* $(x, y)$. Join two globally biased $a$, $b$ trees with roots $x$ and $y$, assuming that all items in tree $x$ precede all items in tree $y$.

In each case, proceed as in the corresponding case of the local join algorithm:

Case 0—$x =$ **null** or $y =$ **null.**

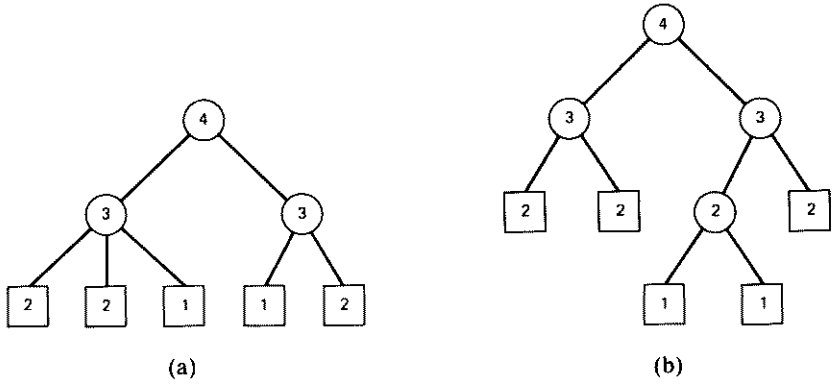



Fig. 4—Two biased 2, 3 trees with the same external nodes. Numbers in nodes are ranks. (a) Locally biased tree. (b) Globally biased tree.

Case 1—$s(x) \geq s(y)$ and $x$ is external, or $s(x) \leq s(y)$ and $y$ is external.

Case 2—$s(x) = s(y)$ and both $x$ and $y$ are internal.

Case 3—$s(x) > s(y)$ and $x$ is internal.

Case 4—$s(x) < s(y)$ and $y$ is internal.

The only difference between this and the local join algorithm is that if $x$ and $y$ have the same rank and both are internal with at least $a$ children, we apply Case 2 instead of Case 1. The algorithm is identical to the join algorithm for globally biased 2, $b$ trees given in Ref. 8.

*Theorem 6: The global join algorithm is correct.*

*Proof*: As in the proof of Theorem 3, we use induction on rank and a case analysis.

Case 1—Immediate.

Case 2—Ignore the split of $z$ if it takes place. Let $q$ be the left sibling of $u$ (recall that $u$ is the rightmost child of $x$). Suppose $q$ or one of the nodes on the rightmost path descending from $q$ is minor. Let this minor node be $r$. The join changes neither $p(r)$ nor $r^+$ and thus preserves global bias at $r$. The symmetric statement holds for the right sibling of $v$ (recall that $v$ is the leftmost child of $y$). Suppose $w$ (the join of $u$ and $v$) or one of the nodes on the leftmost path descending from $w$ is minor. Let this minor node be $r$. There must be a corresponding minor node $r'$ on the leftmost path descending from $u$, such that $s(p(r)) \leq s(p(r'))$. Since the original tree is globally regular at $r'$, the new tree must be globally regular at $r$. The symmetric statement holds for the rightmost path descending from $w$. Thus the new tree is globally biased before the split. The split preserves global bias.

Case 3—Similar to but simpler than Case 2. Case 4 is symmetric. □

*Theorem 7: The worst-case running time of the global join algorithm is* $O(\max\{s(x), s(y)\} - \max\{s(u), s(v)\})$, *where $u$ is the rightmost external descendant of $x$ and $v$ is the leftmost external descendant of $y$.*

*Proof*: The global join algorithm descends rank by rank concurrently along the rightmost path descending from $x$ and the leftmost path descending from $y$, until reaching a leaf; then it ascends. The theorem follows. □

We split a globally biased $a$, $b$ tree in exactly the same way as a locally biased $a$, $b$ tree, *using local rather than global joins.* This method not only produces globally biased trees, it has a worst-case time bound equal to the amortized bound given in Theorem 5.

*Theorem 8: Algorithm 2 (or its variant for an item not in the tree) correctly splits a globally biased $a$, $b$ tree with root $r$ at a node $x$ in* $O(s(r) - s(x))$ *worst-case time.*

*Proof*: The proof is the same as the corresponding proof for globally

biased 2, $b$ trees given in Ref. 8. For completeness, we sketch it here. Let $x_1, x_2, \cdots, x_k$ be the roots of the trees joined to form the final left tree. Let $y_1 = x_1$ and for $i = 2, \cdots, k$ let $y_i$ be the root of the tree formed by executing join $(x_i, y_{i-1})$. With this definition $y_k$ is the final left tree. Each node $x_i$ is either a child of an ancestor of $x$, say $a_i$, in the original tree, or the newly constructed parent of a set of children of such a node $a_i$; furthermore $a_{i+1}$ is a proper ancestor of $a_i$ for $i = 1, \cdots, k-1$. Consider a node $x_i$ for $i \geq 2$. If $x_i$ is a child of $a_i$, global bias implies $x_i$ is a major child, for otherwise its right sibling is external, which is impossible since this right sibling has $x_1$ or its children as ancestors. Thus $s(x_i) = s(a_i) - 1$. If $x_i$ is the new parent of children of $a_i$, then $s(x_i) = s(a_i)$. It follows that $s(x_i) \leq s(x_{i+1})$ for $i = 1, \cdots, k-1$. As noted in the proof of Theorem 5, an induction shows that $s(y_i) \leq s(a_i)$ for $i = 1, \cdots, k$; if $i \geq 2$ and $s(x_i) = s(a_i)$, $x_i$ has at most $b-1$ children, and the join of $x_i$ and $y_{i-1}$ cannot split $x_i$. Thus if $i \geq 2$, $s(a_i) - 1 \leq s(x_i) \leq s(y_i) \leq s(a_i)$.

Consider the join of $x_{i+1}$ and $y_i$. The join will descend rank by rank along the rightmost path from $x_{i+1}$ and the leftmost path from $y_i$. Global bias in the original tree implies that if the descent from $x_{i+1}$ encounters a minor node $z$ (other than the root of $x_{i+1}$), the leftmost external node of $y_i$ will have rank at least $s(p(z)) - 1$ and the join will immediately terminate. Thus the join either terminates by reaching an external descendant of $x_{i+1}$ of rank at least $s(y_i)$, in which case the global bias of the joined tree is immediate, or it reaches an internal descendant, say $z$, of $x_{i+1}$ of rank exactly $s(y_i)$. Now we need a similar but more complicated statement about the leftmost path from $y_i$. There are several cases.

Case 1—$s(x_i) < s(a_i)$ and $x_i$ is minor in the original tree. This can only happen if $i = 1$, i.e., $x_i = y_i$. Global bias implies that the rightmost external descendant of $x_{i+1}$ has rank at least $s(a_i) - 1$. The join descends along the path from $x_{i+1}$ to this external node and then ascends.

Case 2—$s(x_i) < s(a_i)$, $x_i$ is major, and $s(x_i) < s(y_i)$. In this case $y_i$ is also major and the leftmost paths descending from $x_i$ and $y_i$, not including $x_i$ and $y_i$ themselves, are identical. If $z$ is underfilled, $y_i$ is external, and the join terminates in Case 1. If $z$ is filled, $y_i$ is either external or filled, and the join also terminates in Case 1.

Case 3—$s(x_i) < s(a_i)$, $x_i$ is major, and $s(x_i) < s(y_i)$. In this case the left child of $y_i$ is major and the join stops descending either at rank $s(y_i)$ (if $y_i$ is external or filled) or at rank $s(y_i) - 1$ (if $y$ is internal).

Case 4—$s(x_i) = s(a_i)$. In this case the leftmost paths descending from $x_i$ and $y_i$, not including $x_i$ and $y_i$ themselves, are identical. If $y_i$ is external or filled, the join will stop descending at rank $s(y_i)$. If $y_i$ is underfilled, the join will stop at rank $s(y_{i-1})$; either the rightmost child

of $z$ or the leftmost child of $y_i$ is external of rank $s(y_{i-1})$, or both are filled and of rank $s(y_{i-1})$.

In all cases the global bias of the original tree implies that the joined tree is globally biased. This means that the final left tree is globally biased. Furthermore, the time required for joining $x_{i+1}$ and $y_i$ is $O(s(a_{i+1}) - s(a_i))$ in all cases. Summing over all joins gives a time bound of $O(s(r) - s(x))$ to form the final left tree. A symmetric argument applies to the right tree. □

If we implement insertion, deletion, and weight change as described in Section II, we obtain the following worst-case time bounds for the various operations on globally biased $a$, $b$ trees (see Ref. 8):

$$\text{split } (i, r)\text{: } O\left(\log_a \left(\frac{w(r)}{w_i}\right)\right)$$

if $i$ is in the tree, or

$$O\left(\log_a \left(\frac{w(r)}{w_{i^-} + w_{i^+}}\right)\right)$$

if $i$ is not in the tree, where $i^-$ and $i^+$ are the items before and after $i$ in the tree, respectively.

$$\text{insert } (i, x)\text{: } O\left(\log_a \frac{w(x)}{w_{i^-} + w_{i^+}} + \log_a \frac{w(x) + w_i}{w_i}\right).$$

$$\text{delete } (i, x)\text{: } O\left(\log_a \frac{w(x)}{w_i} + \log_a \frac{W'}{w_{i^-} + w_{i^+}}\right),$$

where $W'$ is the weight of the tree root after the deletion.

$$\text{reweight } (i, w, x)\text{: } O\left(\log_a \frac{w(x)}{w_i} + \log_a \frac{W'}{w}\right),$$

where $w(x)$ and $w_i$ are as defined before the weight change, and $W'$ is the weight of the tree root after the change.

As compared with the amortized time bounds for locally biased $a$, $b$ trees, the worst-case bounds for globally biased $a$, $b$ trees are larger for *join* and *delete* and the same for the other operations.

We conclude our discussion of biased $a$, $b$ trees with two remarks. First, if $b \geq 2a$ we can implement either local or global join in an iterative, purely top-down fashion by preemptively splitting nodes with $b$ children as they are encountered. By extending this idea we can implement all the operations top-down. This is a reason to choose $b = 2a$ over $b = 2a - 1$.

Second, for appropriate large values of $a$ and $b$, biased $a$, $b$ trees offer a biased alternative to $B$-trees. One of the advantages of $B$-trees is that there are no underfilled nodes except tree roots; thus if nodes

are stored one node per page in fixed-size pages, the storage efficiency is at least 50 percent, not counting root pages. Biased $a, b$ trees do not share this property. We leave open the problem of devising a space-efficient version of biased $a, b$ trees.

## IV. PSEUDO-WEIGHT-BALANCED TREES

In biased $a, b$ trees, we maintain balance through a height constraint. However, there are other possible balance constraints, such as weight balance. Nievergelt and Reingold[4] defined trees of bounded balance by imposing upper and lower bounds on the ratio *leftweight/rightweight* at each internal node, where the left and right weights count the number of items in the left and right subtrees, respectively. Bent developed a biased version of weight-balanced trees.[6] However, his data structure suffers from a complicated seven-case join algorithm that needs up to three recursive calls and also uses rebalancing rotations more complicated than standard single and double rotations. In this section we introduce a form of biased weight-balanced trees much simpler than Bent's. Our simplification comes from two new ideas: we discretize the weights and allow arbitrarily bad imbalance in some situations where balancing is possible. We call our trees pseudo-weight-balanced.

We consider binary trees, in which each internal node $x$ has exactly two children, a left child $l(x)$ and a right child $r(x)$. As in Section II we define the *weight* $w(x)$ of a node $x$ by $w(x) = w_i$ if $x$ is an external node containing item $i$, $w(x) = w(l(x)) + w(r(x))$ if $x$ is an internal node. We define the *rank* $s(x)$ of a node $x$ differently: $s(x) = \lfloor \lg w(x) \rfloor$. We call a binary search tree *pseudo-weight-balanced* (pwb) if it has two properties:

1. If three nodes in a row, say $x$, $p(x)$, and $p(p(x))$, have the same rank, then $x$ is external and either $x$ is a left child and $p(x)$ a right child or vice-versa (see Fig. 5a).

2. If $x$ and $l(x)$ (symmetrically $x$ and $r(x)$) are internal nodes of the same rank, then $w(r(l(x))) + w(r(x))$ (symmetrically $w(l(r(x))) + w(l(x))$) is at least $2^{s(x)}$ (see Figure 5b). (This property allows us to do single rotations when necessary to maintain property 1.)

The following result bounds the access time in pwb trees.

*Theorem 9: A pwb tree has ideal access time for all items. Specifically, if $x$ is an external node containing item $i$ in a tree of total weight $W$, then the depth of $x$ is at most* $2 \lg(W/w_i) + 3$.

*Proof:* Let $r$ be the root of the tree and $d$ the depth of $x$. According to property 1, after the first step up from $x$, every two steps taken along the path from $x$ to $r$ must cause a rank increase. Thus $\lfloor (d-1)/2 \rfloor \leq s(r) - s(x) = \lfloor \lg W \rfloor - \lfloor \lg w(x) \rfloor$, which implies $d \leq 2\lfloor (d-1)/2 \rfloor + 1 \leq 2(\lg W - \lg w(x) + 1) + 1 \leq 2 \lg (W/w(x)) + 3$. □

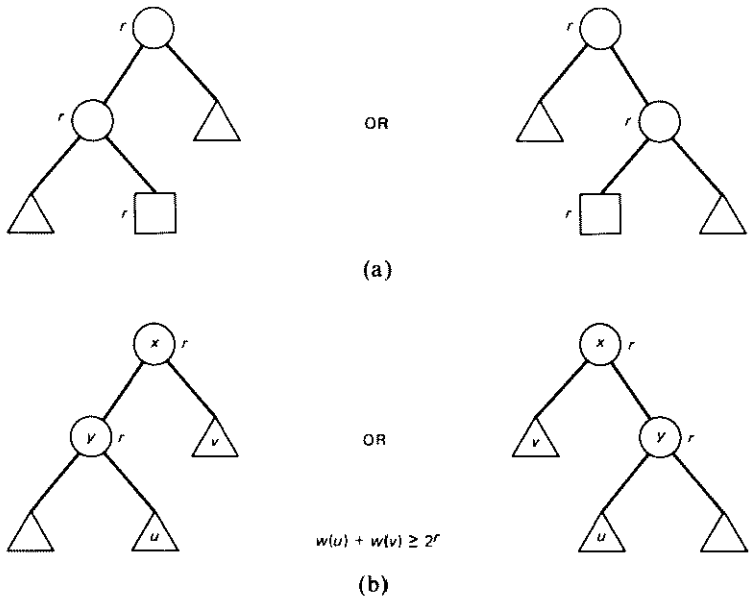



Fig. 5—Legal configurations in a pseudo-weight-balanced tree. (a) Three nodes of the same rank in a row. (b) Two internal nodes of the same rank in a row.

We join two pwb trees using the following algorithm:

*Algorithm 4: join* $(x, y)$. Join two pwb trees with roots $x$ and $y$, assuming that all items in tree $x$ precede all items in tree $y$.

Case 0—$x =$ **null** or $y =$ **null.** Return $y$ if $x =$ **null** or $x$ if $y =$ **null.**

Case 1—$\lg(w(x) + w(y)) \geq 1 + \max\{s(x), s(y)\}$ or the heavier of $x$ and $y$ is an external node. Return a new root with left child $x$ and right child $y$.

Case 2—$s(x) > s(y)$ and $\lg(w(x) + w(y)) < 1 + s(x)$ and $x$ is not external. If $r(x)$ is external, or internal of rank at most $s(x) - 1$, replace $r(x)$ by *join* $(r(x), y)$. Otherwise, perform a single left rotation at $x$ and replace the right child $r(u)$ of the new root $u$ by *join* $(r(u), y)$ (see Fig. 6).

Case 3—$s(x) < s(y)$ and $\lg(w(x) + w(y)) < 1 + s(y)$. Symmetric to Case 2.

*Remark*: Although the join algorithm is stated recursively, it is easy to implement it in an iterative, purely top-down fashion, since the rank of a node depends only on the total weight of its descendants and not on their arrangement.

*Theorem 10*: *Algorithm 4 produces a pwb tree of rank* $\max\{s(x), s(y)\}$ *or* $1 + \max\{s(x), s(y)\}$.

*Proof*: By induction on the depth of the recursion. The definition of rank implies that the rank of the new tree is $\max\{s(x), s(y)\}$ or $1 +$

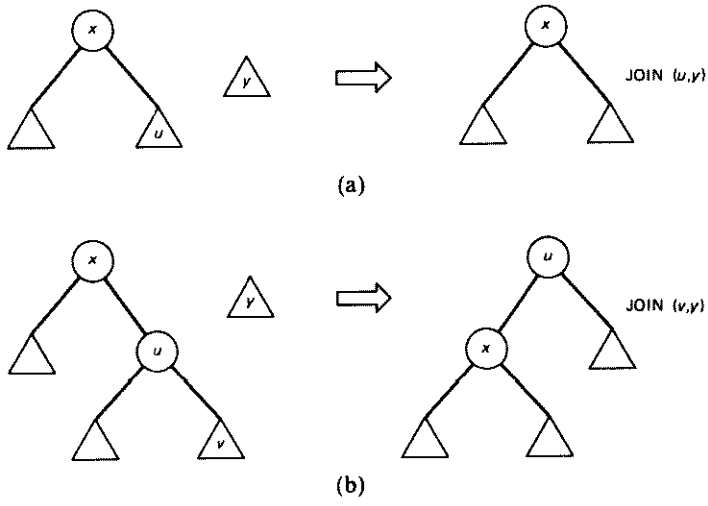



Fig. 6—Case 2 of the join algorithm for pwb trees. (a) Node $u = r(x)$ external or $s(u) < s(x)$: no rotation. (b) Node $u$ internal and $s(u) = s(x)$: rotation.

$\max\{s(x), s(y)\}$. In the latter case, the join must have executed Case 1 and both children of the new root must have rank smaller than the root's rank. Case 1 obviously constructs a tree with properties 1 and 2. In Case 2, property 2 guarantees that the single rotation, if it occurs, creates a pwb tree. If the tree produced by the recursive join has rank less than $s(x)$, the overall joined tree clearly has properties 1 and 2. This is also true if the tree produced by the recursive join has rank $s(x)$, by the observation above. □

In analyzing the running time of Algorithm 4, we use the following credit invariant: Any node $x$ contains $\max\{0, s(p(x)) - s(x) - 1\}$ credits.

*Theorem 11*: *Algorithm 4 runs in* $O(|s(x) - s(y)|) = O(\lg (w(x) + w(y))/\min\{w(x), w(y)\})$ *amortized time. Specifically, if we assume without loss of generality that* $s(x) \geq s(y)$, *performing the join while maintaining the credit invariant takes at most* $s(x) - s(y) + 1$ *credits.*

*Proof*: We consider the same cases as in the algorithm.

Case 1—We need one credit to build the new tree and either $s(x) - s(y)$ or $s(x) - s(y) - 1$ to establish the invariant on $y$, for a total of at most $s(x) - s(y) + 1$.

Case 2—Suppose the single rotation does not take place. We have on hand $s(x) - s(y) + 1$ tokens for performing the join plus $s(x) - s(r(x)) - 1$ from $r(x)$, for a total of $2s(x) - s(y) - s(r(x))$. We need one token for performing the outermost call of *join* plus $\max\{s(r(x)), s(y)\} - \min\{s(r(x)), s(y)\} + 1$ for the recursive call plus $s(x) - \max\{s(r(x)), s(y)\} - 1$ to establish the invariant on the root of the

tree returned by the recursive call, for a total of $s(x) - \min\{s(r(x)), s(y)\} + 1 \leq 2s(x) - s(y) - s(r(x))$, since $s(x) > \max(s(r(x)), s(y))$. Exactly the same argument applies if the rotation does take place, since the rotation preserves the credit invariant. □

The algorithm for splitting a pwb tree is almost identical to but simpler than the algorithm for splitting a biased $a, b$ tree.

*Algorithm 5*: *split* $(x, r)$. Split a pwb tree with root $r$ at a node $x$.

Initialize the current node *cur*, the previous node *prev*, and the left and right nodes *left* and *right* to be $p(x)$, $x$, $l(x)$, and $r(x)$, respectively. Repeat the following step until *cur* = **null:**

Split step—If *prev* = $l(cur)$, replace *right* by *join* (*right*, $r(cur)$); otherwise, replace *left* by *join* ($l(cur)$, *left*). Remove *prev* as a child of *cur* and destroy it if it is not $x$. Replace *prev* and *cur* by *cur* and $p(cur)$, respectively.

This algorithm is the same as that described by Bent, Sleator, and Tarjan[8] for splitting biased binary trees, and indeed will work for any class of binary search trees for which a join algorithm is known.

*Theorem 12*: The amortized time of *split* $(x, r)$ is

$$O\left(\lg\left(\frac{w(r)}{w(x)}\right)\right).$$

More precisely, performing the split while maintaining the credit invariant takes $O(s(r) - s(x))$ credits, where $x$ is the node containing item $i$.

*Proof*: The definition of ranks implies that $s(prev) \geq \max\{s(left), s(right)\}$ before each split step. An easy induction shows that if we allocate $O(s(cur) - s(prev) + 1)$ credits to each split step, we can carry out the step while maintaining the credit invariant on the trees *left* and *right* and in addition maintaining $2s(prev) - s(left) - s(right)$ credits on hand. Summing over all split steps and using property 2 gives the theorem. □

Using the appropriate combinations of *join* and *split*, we obtain the same amortized time bounds as in Section II (with binary logarithms) for insertion, deletion, and weight change in pwb trees. Pseudo-weight-balanced trees are a very simple version of locally biased trees, competitive with the biased binary trees presented in Ref. 8. We have been unable to devise a globally biased version of pwb trees and leave this as an open problem.

## REFERENCES


1. G. M. Adelson-Vel'skii and Y. M. Landis, "An algorithm for the organization of information," Soviet Math. Dokl., *3*, No. 5 (September 1962), pp. 1259–62.
2. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Reading, MA: Addison-Wesley, 1974.

3. R. Bayer and E. M. McCreight, "Organization and maintenance of large ordered indexes," Acta Info., *1*, No. 3 (1972), pp. 173–89.
4. J. Nievergelt and E. M. Reingold, "Binary search trees of bounded balance," SIAM J. Comput., *2* (1973), pp. 33–43.
5. N. Abramson, *Information Theory and Coding*, New York: McGraw-Hill, 1963.
6. S. W. Bent, "Dynamic weighted data structures," Ph.D. thesis, Computer Science Department, Stanford University, Stanford, CA, 1982.
7. S. W. Bent, D. D. Sleator, and R. E. Tarjan, "Biased 2–3 trees," Proc. Twenty-First Annual IEEE Symp. on Foundations of Computer Science, October 13–15, 1980, pp. 248–54.
8. S. W. Bent, D. D. Sleator, and R. E. Tarjan, "Biased search trees," unpublished work.
9. S. Huddleston and K. Mehlhorn, "Robust balancing in B-trees," Lecture Notes in Computer Science, *104* (1981), Berlin: Springer-Verlag, pp. 234–44.
10. S. Huddleston and K. Mehlhorn, "A new data Structure for representing sorted lists," Acta Info., *17*, No. 2 (1982), pp. 157–84.
11. D. Maier and S. C. Salveter, "Hysterical B-trees," Info. Proc. Letters, *12*, No. 4 (August 1981), pp. 199–205.

## AUTHORS

**Joan Feigenbaum,** B.A. (Mathematics), 1981, Harvard University. Ms. Feigenbaum is a graduate student in the Computer Science Department of Stanford University and holds a grant from the Bell Laboratories Graduate Research Program for Women. She spent the summers of 1980, 1981, and 1982 at Bell Laboratories. During the summer of 1982, she was a member of the Mathematical Foundations of Computing Department. Her current research is in the areas of data structures and cryptography. Member, Phi Beta Kappa.

**Robert E. Tarjan,** B.S. (Mathematics), 1969, California Institute of Technology; M.S., 1971, and Ph.D., 1972 (Computer Science), Stanford University; Cornell University, 1972–1973; University of California, Berkeley, 1973–1975; Stanford University, 1975–1980; Bell Laboratories, 1980—. Mr. Tarjan is a member of the Mathematical Foundations of Computing Department, where he has been studying the design and analysis of efficient data structures and combinatorial algorithms. Member, ACM, SIAM, AAAS, Tau Beta Pi, and Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# An Algebraic Theory of Relational Databases

By T. T. LEE*

(Manuscript received December 10, 1982)

In this paper we present a theory of relational database systems based on the partition lattice, which represents a new mathematical approach to the structure of relational database systems. A partition lattice can be defined for any given relation. This partition lattice is shown to be a meet-morphic image of the Boolean algebra of subsets of the attribute set. The partial ordering in the lattice is proved to be equivalent to the concept of functional dependency, and thus Armstrong's axioms for functional dependencies are proved. We solve the problem of finding the list of all keys by seeking the prime implicants of the Boolean function associated with the principal ideals generated by the attributes. We demonstrate the properties of the Boyce-Codd Normal Form (BCNF), and give a modified algorithm for synthesizing an information-lossless BCNF based on the principal filter. The necessary and sufficient conditions for multivalued dependency (MVD) are given in terms of a lattice equation, and the inference rules of MVD are proved. The necessary and sufficient conditions for join dependency (JD) are given; consequently, we can prove the known result that acyclic join dependency (AJD) is equivalent to a set of MVDs. The concept of data independence is introduced, and is extended to conditional independence and mutual independence. We established this algebraic theory of relational databases in the same spirit that the theory of probability was constructed. We present a comparison that demonstrates the similarities.

## I. INTRODUCTION

The existing theory of relational databases is based on Codd's relational model of data.[1,2] This relational database theory can be considered to be the study of data dependencies (or independencies).

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

The theory was initiated by Codd with the introduction of the concept of functional dependency; Codd observed that this concept can be used to design better, normalized, database schemes. The advantage of normalized database schemes is that they remove the possibility of updating anomalies caused by undesirable data dependencies.[2-5]

In the existing theory of logical database design, functional dependencies are input constraints that must always hold in the relation.[6] In the present paper, however, we take a different approach. We assume that for a particular database designer, there exists a (finite) universal relation $R[\Omega]$ for a given set of attributes $\Omega$, such that any relation $T$ on $\Omega$ is a subset of $R[\Omega]$. Furthermore, each subset $X$ of $\Omega$ corresponds to an equivalence relation (partition) on the set of tuples of $R[\Omega]$. That is, if two tuples in $R[\Omega]$ have the same $X$ value, then they are in the same equivalence class. With this approach, the concept of functional dependency becomes equivalent to the refinement partial ordering of the partition lattice. The partitions on the (finite) set of tuples of the universal relation $R[\Omega]$ can then be considered as the fundamental constraints, from which the functional dependencies (partial ordering) can be derived. Consequently, with our approach, the functional dependencies are inherent properties of the universal relation $R[\Omega]$. The input constraints of course must be consistent with the inherent properties within the database.

Another kind of data dependency, proposed by Fagin[7] and Zaniolo,[8] is the multivalued dependency, which includes functional dependency as a special case. Multivalued dependency is the necessary and sufficient condition for the lossless-join decomposition of a relation into two subrelations, such that the original relation can be regenerated by the (natural) join operation.[7-11] Using the partition lattice we propose, we can formulate multivalued dependency as a lattice equation (see Section VI). We show that the axioms for functional dependencies[12] and the inference rules of multivalued dependencies[13] can all be proved as theorems within the framework of partition lattice theory. We show how the concept of join dependency[10,11,14] is connected to multivalued dependency. We give the necessary and sufficient condition for join dependency and, consequently, we can prove the known result that the acyclic join dependency is equivalent to a set of MVDs.[15,16] We also introduce the concept of data independence, and the extension to conditional independence and mutual independence of sets of attributes.

The problem of listing all the keys of a relation is solved by using the concept of principal ideals in lattice theory. One form of a relation having desirable properties is the Boyce-Codd Normal Form (BCNF); we show that the concept of the principal filter (dual ideal) can be used to produce information-lossless Boyce-Codd Normal Forms.

Both the theoretical foundation and the practical application of the existing theory of relational databases appear to be fragmented. This paper shows that all the diverse kinds of data dependencies can be formulated within the lattice theory, which has the important advantage of unifying the theory of relational databases into a coordinated whole. Because of this, it would appear that future work in relational databases should be conducted using lattice theory as the basic framework.

The establishment of this algebraic theory of relational databases is done in the same spirit as the construction of the theory of probability, although probability theory is of course unrelated to database theory. We are convinced that the lattice theory could play a role in the theory of relational databases similar to the role that measure theory plays in the theory of probability.[17]

The basic notion of relational databases is defined in Section II, and the partition lattice of the relation is introduced in Section III. The problem of listing all keys is solved in Section IV, where the Boolean functions associated with the principal ideals are defined. The properties of the Boyce-Codd Normal Form are studied in Section V, where we present a modified algorithm for synthesizing information-lossless BCNFs based on the principal filters. Section VI is devoted to the proof of equivalence between multivalued dependency and a lattice equation. Section VII discusses join dependency and acyclic join dependency. Finally, in Section VIII we outline a possible direction for future research, as well as a comparison that shows the similarities between probability theory and the algebraic theory of relational databases. In Appendix A we list the laws of lattice theory for reference. The proofs of the axioms for functional and multivalued dependencies are listed in Appendix B.

Unless otherwise stated, we refer to the universal relation as simply "the relation" in the remainder of this paper.

## II. RELATIONS

An *attribute* is a symbol taken from a finite set $\Omega = \{A_1, A_2, \cdots, A_n\}$. For each attribute $A$ there is a set of possible values called its *domain*, denoted DOM($A$). We will use capital letters from the beginning of the alphabet ($A, B, \cdots$) for single attributes, and capital letters from the end of the alphabet ($X, Y, \cdots$) for sets of attributes. For a set of attributes $X \subseteq \Omega$, an $X$-value $x$ is an assignment of values to the attributes of $X = \{A_{i_1}, A_{2_2}, \cdots, a_{i_k}\}$ from their respective domains. The notation $XY$ will be used to represent the union of two arbitrary sets of attributes $X, Y \subseteq \Omega$.

A *relation* $R$ on the set of attributes $\Omega = \{A_1, \cdots, A_n\}$ is a subset of the Cartesian product DOM($A_1$) $\times \cdots \times$ DOM($A_n$). The elements

(rows) of $R$ are called *tuples*. A relation $R$ on $\{A_1, \cdots, A_n\}$ will be denoted by $R[A_1 \cdots A_n]$. Similarly, if $R$ is defined on the union of sets $(X_1, X_2, \cdots, X_m)$, then the notation $R[X_1 \cdots X_m]$ will be used. A relation can be visualized as a table whose columns are labeled with attributes and whose rows depict tuples. The ordering of the rows and columns is immaterial. The *cardinal* of $R$ is the total number of tuples in $R$ and is denoted by $|R|$.

Let $t$ be a tuple in $R[\Omega]$. For $X \subseteq \Omega$, $t[X]$ denotes the tuple that contains the components of $t$ corresponding to the attributes of $X$. The *projection* of $R$ on $X$, denoted by $R[X]$, is defined as follows:

$$R[X] = \{t[X] \mid t \in R\}.$$

Similarly, the *conditional projection* of $R$ on $X$ by a $Y$-value $y$, where $Y \subseteq \Omega$, is defined as follows:

$$R_y[X] = \{t[X] \mid t \in R \quad \text{and} \quad t[Y] = y\}.$$

Let $R[XZ]$ and $S[XZ]$ be relations where $X$, $Y$, and $Z$, are disjoint sets of attributes. The *join* (natural join) of $R$ and $S$, denoted by $R \mid \times \mid S$, is the relation $T[XYZ]$ whose attributes are $XYZ$, and is defined as follows:

$$\begin{aligned} T[XYZ] &= R[XZ] \mid \times \mid S[YZ] \\ &= \{(x, y, z) \mid (x, z) \in R \quad \text{and} \quad (y, z) \in S\}. \end{aligned}$$

The join can also be defined as the union of a collection of Cartesian products:

$$\begin{aligned} T[XYZ] &= R[XZ] \mid \times \mid S[YZ] \\ &= \{R_z[X] \times S_z[Y] \times (z) \mid (z) \in R[Z] \cap S[Z]\}. \end{aligned}$$

Let $R$ be a relation on the set of attributes $\Omega$. We may have two sets of attributes $X$, $Y$, $\subseteq \Omega$, such that for any two tuples $t_1$, $t_2 \in R$, $t_1[X] = t_2[X]$ implies $t_1[Y] = t_2[Y]$. We say then that $X$ *functionally determines* $Y$ in $R$, and denote this fact by $X \rightarrow Y$. A functional dependency (FD) $X \rightarrow Y$ is *trivial*, meaning it holds in all relations, if $Y \subseteq X$. Note that FDs enjoy the projectivity and inverse projectivity properties.[3,4] For sets $X$, $Y \subseteq \Omega' \subseteq \Omega$, the FD: $X \rightarrow Y$ is valid in $R[\Omega]$ iff it is valid in $R[\Omega']$.

We say that a set of relations $\{R[\Omega_1], \cdots, R[\Omega_n]\}$ has the *information-lossless join* property if $\Omega = \Omega_1 \cdots \Omega_n$ and

$$R[\Omega] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n].$$

If the set $\{R[\Omega_1], \cdots, R[\Omega_n]\}$ does not have this property, we say that it has a *lossy join*.[14] An important property of functional dependency[10] is that if FD: $X \rightarrow Y$ is valid in $R[\Omega]$ then

$$R[\Omega] = R[(\Omega - Y)X] \mid\times\mid R[XY].$$

This property will be discussed in more detail in Section VI.

## III. THE RELATION LATTICE

If $S$ is a nonempty set, then a subset $\rho$ of $S \times S$ is called a *binary relation* on $S$. The *product* of two binary relations $\rho$, $\rho' \subseteq S \times S$ is defined as:

$$\rho \circ \rho' = \{(a, b\} \in S \times S \mid \exists c \in S \text{ such that } (a, c) \in \rho, (c, b) \in \rho'\}.$$

We say that a relation $\rho$ on $S$ is *reflexive* if $(a, a) \in \rho$ for every $a$ in $S$; that $\rho$ is *symmetric* if $\rho^{-1} = \rho$, i.e., if

$$(\forall a, b \in S), \qquad (a, b) \in \rho \quad \text{implies} \quad (b, a) \in \rho;$$

and that $\rho$ is *transitive* if $\rho \circ \rho \subseteq \rho$, *i.e.*, if

$$(\forall a, b, c \in S), \qquad (a, b) \in \rho \quad \text{and} \quad (b, c) \in \rho \quad \text{imply} \quad (a, c) \in \rho.$$

A binary relation is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

A family $\pi = \{B_i \mid i \in I\}$ of subsets, called *blocks* of $S$, is said to form a *partition* of $S$ if the following conditions hold:

1. Each $B_i$ is nonempty
2. For all $i \neq j$ in $I$, $B_i \cap B_j = \varnothing$
3. $\cup\{B_i \mid i \in I\} = S$.

The two apparently different notions of "equivalence relation" and "partition" are interchangeable: Let $\rho$ be an equivalence relation on a set $S$. Then the family $a_\rho = \{b \mid (a, b) \in \rho\}$ of subsets of $S$ is a partition of $S$. Conversely, if $\pi = \{B_i \mid i \in I\}$ is a partition of $S$, then the relation $\{(a, b) \mid (\exists i \in I), (a, b) \in B_i\}$ is an equivalence relation on $S$.

If $\rho$ is an equivalence relation (partition) on $S$, we shall sometimes write $a\rho b$ as an alternative to $(a, b) \in \rho$. The sets $a_\rho$ that form the associated partition of the equivalence relation are called *$\rho$-classes.* The set of $\rho$-classes is called the *quotient set* of $S$ by $\rho$ and is denoted by $S/\rho$.

A binary relation $\leqq$ on the set $S$ is a *partial ordering* of $S$ if and only if $\leqq$ is reflexive; antisymmetric, i.e., if

$$(\forall a, b \in S), \qquad a \leqq b \quad \text{and} \quad b \leqq a \quad \text{imply} \quad a = b;$$

and transitive. A set $S$ with a partial ordering $\leqq$ is called a *partially ordered set* (poset) and it is denoted by the pair $(S, \leqq)$.

Let $(S, \leqq)$ be a poset and let $T$ be a subset of $S$. Then, $a \in S$ is the *greatest lower bound* (g.l.b.) of $T$ iff

1. $(\forall t \in T)$, $a \leqq t$.

2. $(\forall t \in T)$, $a' \leqq t$ implies $a' \leqq a$.

Similarly, $a \in S$ is the *least upper bound* (l.u.b.) of $T$ iff

1. $(\forall t \in T)$, $t \leqq a$.
2. $(\forall t \in T)$, $t \leqq a'$ implies $a \leqq a'$.

A *lattice* is a poset in which any two elements $a$ and $b$ have a g.l.b., called a *meet* and denoted by $a \cdot b$, and a l.u.b., called a *join* and denoted by $a + b$. We sometimes write the meet $a \cdot b$ as $ab$ if no confusion is created. The properties of the meet and join operations of a lattice[18] are listed in Appendix A.

Let the set of all partitions $\pi_i$ on $S$ be denoted by $\prod(S)$, and define the partial ordering on $\prod(S)$ as follows:

$$\text{If } (\forall a, b \in S), \quad a\pi_1 b \text{ implies } a\pi_2 b, \quad \text{then } \pi_1 \leqq \pi_2.$$

The poset $(\prod(S), \leqq)$ is seen to be a lattice $(\prod(S), \cdot, +)$ with a universal lower bound $\mathbf{0} = \{B_i | i \in I\}$ such that every block $B_i$ is a singleton, and an universal upper bound $\mathbf{1} = \{S\}$. To specify a particular partition, we list the elements, and distinguish blocks with bars and semicolons. For example, if $S = \{1, 2, 3, 4, 5\}$ and partition $\pi$ on $S$ has blocks $\{1, 3, 4\}$, $\{2, 5\}$, then we write $\pi = \{1, 3, 4; 2, 5\}$. The meet and join of any two partitions $\pi_1, \pi_2 \in \prod(S)$ can be determined as follows:

1. $(\forall a, b \in S)$, $a\pi_1 \cdot \pi_2 b$ iff $a\pi_1 b$ and $a\pi_2 b$.
2. $(\forall a, b \in S)$, $a\pi_1 + \pi_2 b$ iff $\exists n \in N$ and $c_0, \cdots, c_n \in S$ such that $a = c_0$, $b = c_n$ and $c_i \pi_1 c_{i+1}$ or $c_i \pi_2 c_{i+1}$ for each $i$, $0 \leqq i \leqq n - 1$.

A complemented distributive lattice is called a *Boolean algebra* (see Appendix A). The set of all subsets of $S$, called the *power set* of $S$, and denoted by $\mathbf{2}^S$, with the partial ordering $(\forall S_1, S_2 \in \mathbf{2}^S)$, $S_1 \leqq S_2$ iff $S_1 \supseteq S_2$, is a Boolean algebra $(\mathbf{2}^S, \cdot, +, ^-)$ with the universal bounds $\mathbf{0} = S$ and $\mathbf{1} = \varnothing$. The *dual* of a poset is the poset with the converse partial ordering on the same elements. The Boolean algebra defined above is the dual of the conventional Boolean algebra of the power set. The operations of meet and join are defined by

1. Meet (g.l.b.) $S_1 \cdot S_2 = S_1 \cup S_2$,
2. Join (l.u.b.) $S_1 + S_2 = S_1 \cap S_2$,

and the complement of $S_1 \in \mathbf{2}^S$ is $\bar{S}_1 = S - S_1$.

Let $\psi$: $L \to M$ be a function from a lattice $L$ into a lattice $M$. We say $\psi$ is a *meet-morphism* if

$$(\forall a, b \in L), \qquad \psi(a \cdot b) = \psi(a) \cdot \psi(b),$$

and $\psi$ is a *join-morphism* if

$$(\forall a, b \in L), \qquad \psi(a + b) = \psi(a) + \psi(b).$$

Meet-morphisms and join-morphisms are both *isotone* (order-preserving); i.e.,

$$(\forall a, b \in L), \qquad a \leqq b \quad \text{implies} \quad \psi(a) \leqq \psi(b),$$

and any order-preserving one-to-one mapping with an inverse is an *isomorphism.*[18]

Let $R$ be a relation on the set of attributes $\Omega$. The set of all subsets of $\Omega$, denoted $\mathbf{2}^{\Omega}$, with the partial ordering defined by set-containment, is a Boolean algebra $(\mathbf{2}^{\Omega}, \cdot, +, ^{-})$,[18] where the meet, join, and complement operations are defined as above. For every $X \in \mathbf{2}^{\Omega}$, there is an equivalence relation (partition) on the set of tuples in $R[\Omega]$ defined as follows:

Definition 1: Let $R$ be a relation on the set of attributes $\Omega$. Each subset of $\Omega$ is associated with a partition of the set of tuples of $R$. We define the function $\theta$: $\mathbf{2}^{\Omega} \rightarrow \prod(R[\Omega])$, which we call the *partition function* (associated with $R[\Omega]$), by

$$\theta{:}X \rightarrow \theta(X) = \{(t_1, t_2) \in R[\Omega] \times R[\Omega] \mid t_1[X] = t_2[X]\}. \quad \blacksquare$$

In general, the image set $Im(\theta)$ of $\theta$ is not a sublattice of $\prod(R[\Omega])$. Since $\pi_1, \pi_2 \in Im(\theta)$ implies $\pi_1 \cdot \pi_2 \in Im(\theta)$, $Im(\theta)$ is a complete lattice in its own right,[20] and it will be called the *relation lattice* of $R[\Omega]$, and denoted by $L(R[\Omega])$. Note that there are no duplicated tuples in $R[\Omega]$, so that $\theta(\Omega) = \mathbf{0}$. Since the tuples cannot be "differentiated" by the empty set of attributes, we define $\theta(\varnothing) = \mathbf{1}$. The universal bounds of $L(R[\Omega])$ are the same as those in $\prod[R)\Omega])$. We immediately recognize the concept of functional dependency to be equivalent to the refinement partial ordering of the partitions.

*Lemma 1: Let $R[\Omega]$ be a relation on the set of attributes $\Omega$, and let $\theta$: $\mathbf{2}^{\Omega} \rightarrow \prod(R[\Omega])$ be the partition function associated with $R[\Omega]$, defined above. Then*

$$X \rightarrow Y \quad \text{iff} \quad \theta(X) \leqq \theta(Y). \quad \blacksquare$$

An immediate consequence of the above lemma is that the projection $R[X]$ of $R[\Omega]$ on $X$ is simply the quotient of $R[\Omega]$ by $\theta(X)$, i.e., $R[X] = R[\Omega]/\theta(X)$. Thus each tuple in $R[X]$ corresponds to a $\theta(x)$-class in $R[\Omega]/\theta(X)$ and it takes the $X$-value only. Note that $\theta(X) = \theta(Y)$ does not imply $R[X] = R[Y]$ because the attributes $X$ and $Y$ may have different sets of values.

*Theorem 1: Let $R$ be a relation on the set of attributes $\Omega$, and let $L(R[\Omega])$ be the relation lattice of $R$. Then the partition function $\theta{:}\mathbf{2}^{\Omega} \rightarrow L(R[\Omega])$ is a meet-morphism.*

*Proof:* We want to show that

$$\theta(XY) = \theta(X)\theta(Y), \qquad \forall X, Y \in \mathbf{2}^{\Omega}.$$

Suppose $t_1\theta(XY)t_2$. Then, $t_1[XY] = t_2[XY]$, which implies

$$t_1[X] = t_2[X] \quad \text{and} \quad t_1[Y] = t_2[Y].$$

Hence,

$$t_1\theta(X)t_2 \quad \text{and} \quad t_1\theta(Y)t_2.$$

By the definition of the meet operation, we have

$$t_1\theta(X)\theta(Y)t_2,$$

so that

$$\theta(XY) \leqq \theta(X)\theta(Y).$$

Suppose $t_1\theta(X)\theta(Y)t_2$. Then,

$$t_1\theta(X)t_2 \quad \text{and} \quad t_1\theta(Y)t_2,$$

so that

$$t_1[X] = t_2[X] \quad \text{and} \quad t_1[Y] = t_2[Y],$$

and thus

$$t_1[XY] = t_2[XY].$$

Consequently,

$$t_1\theta(XY)t_2,$$

so that

$$\theta(X)\theta(Y) \leqq \theta(XY).$$

Hence

$$\theta(XY) = \theta(X)\theta(Y). \quad \blacksquare$$

Note that the partition function $\theta$ is order-preserving, but it is in general not a join-morphism.* However, if $\theta(X + Y) = \theta(X) + \theta(Y)$ holds in $L[R]$, the pair $(X, Y)$ has a special property in the relation. This is discussed further in Section VI.

It is clear now that Armstrong's axioms for functional dependencies become theorems within the framework of lattice theory. The proofs of the axioms for functional dependencies are given in Appendix B.

Let $R$ be a relation on the set of attributes $\Omega$, and let $\theta$: $\mathbf{2}^{\Omega} \rightarrow L[R(\Omega)]$ be the partition function associated with $R[\Omega]$. Then the relation $\theta \circ \theta^{-1}$ on $\mathbf{2}^{\Omega}$ defined by

$$\theta \circ \theta^{-1} = \{(X, Y) \in \mathbf{2}^{\Omega} \times \mathbf{2}^{\Omega} \mid \theta(X) = \theta(Y)\}$$

is obviously an equivalence relation. Sets in the quotient set $\mathbf{2}^{\Omega}/\theta \circ \theta^{-1}$ will be called $\theta$ *classes.*

* The join of $\pi_1$ and $\pi_2$ in $L(R[\Omega])$ may be different from their join in $\prod(R[\Omega])$. We will use the notation $\pi_1 \oplus \pi_2$ to denote the join of $\pi_1$ and $\pi_2$ in $\prod(R[\Omega])$, while $\pi_1 + \pi_2$ will denote the join of $\pi_1$ and $\pi_2$ in $L(R[\Omega])$; e.g., in Example 1 below, $\theta(E) \oplus \theta(S) = \{1, 2, 3, 4, 5, 6; 7, 8\}$, and $\theta(E) + \theta(S) = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Table I—Relation $R[ECSY]$

| | Employee | Child | Salary | Year |
|---|---|---|---|---|
| 1 | Hilbert | Hubert | \$35K | 1975 |
| 2 | Hilbert | Hubert | \$40K | 1976 |
| 3 | Gauss | Gwendolyn | \$40K | 1975 |
| 4 | Gauss | Gwendolyn | \$50K | 1976 |
| 5 | Gauss | Greta | \$40K | 1975 |
| 6 | Gauss | Greta | \$50K | 1976 |
| 7 | Pythagoras | Peter | \$15K | 1975 |
| 8 | Pythagoras | Peter | \$20K | 1976 |

Example 1: Consider the relation $R$ in Table I (see Ref. 7). Let $\Omega = \{E, C, S, Y\}$ be the set of attributes, where $E$ = employee, $C$ = child, $S$ = salary, $Y$ = year. Then

$$\mathbf{2}^{\Omega} = \{\varnothing, E, C, S, Y, EC, ES, EY, CS, CY, SY, ECS, ECY, ESY, CYS, CESY\},$$

and

$$\theta(\varnothing) = \{\overline{1, 2, 3, 4, 5, 6, 7, 8}\} = \mathbf{1},$$

$$\theta(E) = \{\overline{1, 2}; \overline{3, 4, 5, 6}; \overline{7, 8}\} = \pi_1,$$

$$\theta(C) = \theta(EC) = \{\overline{1, 2}; \overline{3, 4}; \overline{5, 6}; \overline{7, 8}\} = \pi_2,$$

$$\theta(S) = \{\overline{1}; \overline{2, 3, 5}; \overline{4, 6}; \overline{7}; \overline{8}\} = \pi_3,$$

$$\theta(Y) = \{\overline{1, 3, 5, 7}; \overline{2, 4, 6, 8}\} = \pi_4,$$

$$\theta(ES) = \theta(EY) = \theta(SY) = \theta(ESY) = \{\overline{1}; \overline{2}; \overline{3, 5}; \overline{4, 6}; \overline{7}; \overline{8}\} = \pi_5,$$

$$\theta(CS) = \theta(CY) = \theta(ECY) = \theta(ECS) = \theta(CSY) = \theta(ECSY) = \{\overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}; \overline{6}; \overline{7}; \overline{8}\} = \mathbf{0}.$$

The Hasse diagram[18,21] of the relation lattice is illustrated in Fig. 1. ■

## IV. LIST OF KEYS

Let $R$ be a relation on the set of attributes $\Omega$. We say that $X \subseteq \Omega$ is a *superkey* of $R$ if $X \rightarrow A$, $\forall A \in \Omega$. If $X$ is a superkey and no proper subset of $X$ is a superkey, $X$ is said to be a *key* of $R$.[1,2,5]

*Lemma 2:* $X \subseteq \Omega$ *is a superkey of* $R$ *iff* $\theta(X) = \mathbf{0}$.

*Proof:* (Necessity) Let $\Omega = \{A_1, \cdots, A_n\}$, and $X \rightarrow A_i$, $\forall A_i \in \Omega$. Then,

$$\theta(X) \leqq \theta(A_i), \qquad \forall A_i \in \Omega.$$

By the definition of the meet operation, we have

$$\theta(X) \leqq \theta(A_1)\theta(A_2) \cdots \theta(A_n).$$

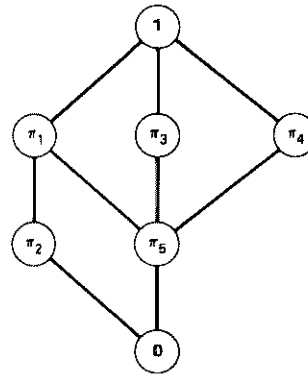



Fig. 1—Relation lattice $L(R[\Omega])$.

It follows from Theorem 1 that

$$\theta(X) \leqq \theta(A_1A_2 \cdots A_n) = \theta(\Omega) = \mathbf{0}.$$

Hence,

$$\theta(X) = \mathbf{0}.$$

(Sufficiency) Suppose $\theta(X) = \mathbf{0}$. Then

$$\theta(X) \leqq \theta(A_i), \qquad \forall A_i \in \Omega.$$

Hence,

$$X \rightarrow A_i, \qquad \forall A_i \in \Omega. \quad \blacksquare$$

An *ideal* is a subset $J$ of a lattice $L$ with the properties[18]

1. $a \in J$, $x \in L$, and $x \leqq a$, imply $x \in J$,
2. $a, b \in J$ implies $a + b \in J$.

For every $a \in L$, the subset of all elements "less than or equal to" $a$ is evidently an ideal; it is called the *principal ideal* of $L$ generated by $a$, and is denoted by $(a]$, i.e.,

$$(a] = \{x \in L \mid x \leqq a\}.$$

Definition 2: Let $R$ be a relation on the set of attributes $\Omega = \{A_1, \cdots, A_n\}$. For each $A_i \in \Omega$, $J_i = (\theta(A_i)]$ is the principal ideal of the relation lattice $L(R[\Omega])$ generated by $\theta(A_i)$. A Boolean function

$$f_i(A_1, \cdots, A_n) = \sum_{\theta(X) \in J_i} X$$

defined on $\mathbf{2}^\Omega$ is the Boolean sum of all $X \in \mathbf{2}^\Omega$ such that $\theta(X) \leqq \theta(A_i)$. We will call $f_i$ the *principal ideal function* (generated by $A_i$). ■

This function plays a role similar to the Boolean function used in Ref. 16.

*Theorem 2: Let $R$ be a relation on $\Omega = \{A_1, \cdots, A_n\}$. $X \subseteq \Omega$ is a superkey of $R$ iff $X$ is a product term in the expansion of the Boolean function*

$$F(A_1, \cdots, A_n) = \prod_{i=1}^{n} f_i(A_1, \cdots, A_n),$$

*where $f_i$ is the principal ideal function generated by $A_i$.*

*Proof:* The Boolean function $F(A_1, \cdots, A_n)$ has the expansion

$$F(A_1, \cdots, A_n) = \prod_{i=1}^{n} f_i = \sum_{\theta(X_i) \in J_i} X_1 \cdots X_n.$$

We want to show that every term $K = X_1 \cdots X_n$ is a superkey. Since $\theta(X_i) \in J_i = (\theta(A_i)]$, it follows that

$$\theta(X_i) \leqq \theta(A_i), \qquad 1 \leqslant i \leqslant n.$$

From L6 in Appendix A, we have

$$\theta(X_1)\theta(X_2) \cdots \theta(X_n) \leqq \theta(A_1)\theta(A_2) \cdots \theta(A_n).$$

It follows from Theorem 1 that

$$\theta(X_1X_2 \cdots X_n) \leqq \theta(A_1 \cdots A_n) = \theta(\Omega) = \mathbf{0},$$

and thus

$$\theta(X_1X_2 \cdots X_n) = \mathbf{0}.$$

Hence, $K = X_1X_2 \cdots X_n$ is a superkey of $R$.

Conversely, suppose $X$ is a superkey of $R$. Then

$$X \rightarrow A_i, \qquad \forall A_i \in \Omega.$$

Thus,

$$\theta(X) \leqq \theta(A_i), \qquad 1 \leqslant i \leqslant n.$$

By the definition of the principal ideal $J_i$, we must have

$$\theta(X) \in J_i = (\theta(A_i)], \qquad 1 \leqslant i \leqslant n.$$

It follows that $X = X \cdots X$ ($n$ times) is a product term in the expansion of $F(A_1, \cdots, A_n)$. ■

It is natural to call $F(A_1, \cdots, A_n)$ the *key Boolean function* of the relation $R[A_1 \cdots A_n]$. Since any key $X$ is a superkey of $R$, $X$ must be a product term of the key Boolean function $F(A_1, \cdots, A_n)$. Since no proper subset of $X$ is a superkey, then by the definition of the *prime implicant* of a Boolean function,[22] we have

*Corollary 1: Let $R$ be a relation on the set of attributes $\Omega = \{A_1, \cdots, A_n\}$. $X \subseteq \Omega$ is a key of $R$ iff $X$ is a prime implicant of the key Boolean function $F(A_1, \cdots, A_n)$.* ■

An attribute $A \in \Omega$ is *prime* in $R[\Omega]$ if $A$ is in any key of $R$; otherwise $A$ is *nonprime*. $A \subseteq \Omega$ is a nonprime attribute if and only if the key Boolean function is independent of $A$.

*Theorem 3: Let $R$ be a relation on $\Omega$. $A \in \Omega$ is a nonprime attribute iff there exists $X \subseteq \Omega$ such that*

1. $A \notin X$, $X \rightarrow A$,
2. $AZ \rightarrow X$ implies $Z \rightarrow X$.

*Proof:* (Necessity) Let $A \in \Omega$ be a nonprime attribute, and let $X$ be any key of $R$. Then

$$A \notin X, \quad \text{and} \quad X \rightarrow A.$$

Suppose $AZ \rightarrow X$. Then $\theta(AZ) \leqq \theta(X) = \mathbf{0}$. It follows that $\theta(AZ) = \mathbf{0}$ and thus $AZ$ is a superkey; it contains a key $K \subseteq AZ$ and $A \notin K$. We have $K \subseteq Z$, so that

$$\theta(Z) \leqq \theta(K) = \mathbf{0} = \theta(X).$$

Hence,

$$Z \rightarrow X.$$

(Sufficiency) Let $\Omega = \{A_1, \cdots, A_n\}$, $n \geqslant 2$. Assume there is an $X = A_2, \cdots, A_m$, such that (1) $X \rightarrow A_1$, and (2) $A_1Z \rightarrow X$, implies $Z \rightarrow X$. We want to show that $A_1$ must be a nonprime attribute. The key Boolean function $F(A_1, \cdots, A_n)$ of $R[\Omega]$ can be written in the form

$$F(A_1, \cdots, A_n) = \prod_{i=1}^{n} f_i = f_1 f_2 \cdots f_m f_{m+1} \cdots f_n$$

$$= (f_1 f_X f_{m+1}) \cdots (f_1 f_X f_n),$$

where $f_X = f_2 \cdots f_m$. For any product term $Y$ in $f_X$ we have

$$\theta(Y) \leqq \theta(X) \leqq \theta(A_1).$$

Therefore, $Y$ must be a term in $f_1$. It follows that $f_1$ has the form

$$f_1 = f_X + g$$

for some Boolean function $g$. Since $\theta(A_1Z) \leqq \theta(X)$ implies $\theta(Z) \leqq \theta(X)$, $f_X$ can be written in the form

$$f_X = A_1 h + h + p = h + p$$

for some Boolean functions $h$ and $p$ which are independent of $A_1$. Also, every $f_j$, $j = m + 1, \cdots, n$, can be written in the form

$$f_j = f_1 e + f_X e + q$$

for some Boolean functions $e$ and $q$, which are independent of $A_1$. It follows that

$$
\begin{aligned}
f_1 f_X f_j &= f_1 f_X (f_1 e + f_X e + q) \\
&= f_X (f_1 e + f_1 f_X e + f_1 q) \\
&= f_X (f_1 e + f_1 q) \\
&= f_1 f_X (e + q) = (f_X + g) f_X (e + q) \\
&= f_X (e + q) = (h + p)(e + q).
\end{aligned}
$$

Since $h$, $p$, $e$, and $q$ are all independent of $A_1$, we know that $f_1 f_X f_j$ is independent of $A_1$ for all $j = m + 1, \cdots, n$. Clearly, no prime implicant of $F(A_1, \cdots, A_n)$ contains $A_1$, and therefore $A_1$ is a nonprime attribute. ■

Example 2: Consider the relation $R$ in Example 1. To obtain the prime implicants of the key Boolean function $F$, we can first simplify each principal ideal function. The principal ideal functions of the relation $R[ECSY]$ are

$$
\begin{aligned}
f_E &= E + C + SY, \\
f_C &= C, \\
f_S &= S + EY + CY, \\
f_Y &= Y + ES + CE,
\end{aligned}
$$

and the key Boolean function is

$$
\begin{aligned}
F(E, C, S, Y) &= (E + C + SY) \cdot C \cdot (S + EY + CY) \\
&\qquad \cdot (Y + ES + CE) \\
&= CS + CY.
\end{aligned}
$$

The sets $CS$ and $CY$ are the keys, and $E$ is the only nonprime attribute. ■

## V. BOYCE-CODD NORMAL FORM

Normalization is a logical database design process that can be viewed as the decomposition of a relation into a set of subrelations, such that the original relation can be regenerated by the joins of the subrelations. The purpose of decomposition is to separate the independent components into distinct relations, to avoid updating anomalies.[2] It is claimed in Ref. 4 that the Boyce-Codd Normal Form is one that is free of insertion and deletion anomalies. This section is devoted to the BCNF and its relation lattice. A modified algorithm for synthesizing an information-lossless BCNF[6] is included, based on the concept of the principal filter of the relation lattice.

Recall that a functional dependency $X \rightarrow Y$ is trivial if $Y \subseteq X$. A

relation $R[\Omega]$ is said to be in Boyce-Codd Normal Form if, for all nontrivial FDs $X \rightarrow Y$, $X$ is a superkey.[2,4]

Definition 3: Relation $R[\Omega]$ is in BCNF if $X \rightarrow Y$ implies either

1. $X$ is a superkey, i.e., $\theta(X) = \mathbf{0}$,

or

2. $Y \subseteq X$. ■

If a relation is in BCNF, we will show that its relation lattice has some special properties. To analyze these properties we need the concept of the principal filter.[18]

An ideal of the dual of the lattice $L$ is called a *filter* of $L$. A subset $M$ of $L$ is a filter of $L$ if

1. $a \in M$, $x \in L$, and $x \geqq a$, imply $x \in M$,
2. $a, b, \in M$ implies $a \cdot b \in M$.

For every $a \in L$, the subset of all elements "greater than or equal to" $a$ is a filter; it is called the *principal filter* of $L$ generated by $a$, and is denoted by $[a)$, i.e.,

$$[a) = \{x \in L \mid x \geqq a\}.$$

If $a$ and $b$ are elements of a lattice $L$, where $a < b$, and there is no $c \in L$ such that $a < c < b$, then we say that $a$ is *covered* by $b$ (or $b$ covers $a$).[18] An element that covers the universal lower bound $\mathbf{0}$ of $L$ is referred to as an *atom* of $L$.[18]

Definition 4: Let $R$ be a relation on the set of attributes $\Omega$, and let $\pi$ be an atom of the relation lattice $L(R[\Omega])$. Let $\Omega_\pi = \{A \mid A \in \Omega, \theta(A) \geqq \pi\} \subseteq \Omega$. Then the projection $R[\Omega_\pi]$ of $R$ and $\Omega_\pi$ is called an *atomic projection*, and $[\pi)$ is called an *atomic filter*. ■

It is easy to verify that the relation lattice of the atomic projection $R[\Omega_\pi]$ is isomorphic to the principal filter $[\pi)$ of $L(R[\Omega])$ generated by $\pi$.

Definition 5: Let $R$ be a relation on the set of attributes $\Omega$, and let $\pi \in L(R[\Omega])$ be an atom. The principal filter $[\pi)$ of $L(R[\Omega])$ is called *normal* iff whenever $X \rightarrow Y$ is valid in the atomic projection $R[\Omega_\pi]$ then $Y \subseteq X$; otherwise, it is called *abnormal*. ■

*Lemma 3: A relation $R[\Omega]$ is in BCNF iff every atomic filter of $L(R[\Omega])$ is normal.*

*Proof*: (Necessity) Trivial.

(Sufficiency) Suppose $X \rightarrow Y$ and $X$ is not a superkey, i.e., $\theta(X) \neq \mathbf{0}$. Then there must exist an atom $\pi$, such that

$$\mathbf{0} < \pi \leqq \theta(X) \leqq \theta(Y).$$

It follows that $X$, $Y \subseteq \Omega_\pi$ and that $X \rightarrow Y$ is valid in the atomic projection $R[\Omega_\pi]$, which is assumed normal. Therefore $Y \subseteq X$. ■

The join operation in the Boolean algebra $(\mathbf{2}^{\Omega}, \cdot, +, ^{-})$ is not always preserved by the $\theta$ mapping. But for a relation $R[\Omega]$ in BCNF, if $X, Y \subseteq \Omega$ and neither $X$ nor $Y$ is a superkey of $R[\Omega]$, then the join $X + Y$ is preserved by $\theta$. We have

*Corollary 2: If $R[\Omega]$ is in BCNF, $X, Y \subseteq \Omega$, $\theta(X) \neq \mathbf{0}$, and $\theta(Y) \neq \mathbf{0}$, then*

$$\theta(X + Y) = \theta(X) + \theta(Y).$$

*Proof*: Since $X + Y \subseteq X$ and $X + Y \subseteq Y$, we have

$$\theta(X) \leqq \theta(X + Y) \quad \text{and} \quad \theta(Y) \leqq \theta(X + Y).$$

By definition of the join operation, we have

$$\theta(X) + \theta(Y) \leqq \theta(X + Y).$$

Suppose there is a $Z \subseteq \Omega$ such that

$$\theta(X) \leqq \theta(Z) \quad \text{and} \quad \theta(Y) \leqq \theta(Z).$$

Given $\theta(X) \neq \mathbf{0}$ and $\theta(Y) \neq \mathbf{0}$, we have

$$Z \subseteq X \quad \text{and} \quad Z \subseteq Y.$$

Thus,

$$Z \subseteq X + Y,$$

so that

$$\theta(X + Y) \leqq \theta(Z).$$

By the definition of least upper bound, we have

$$\theta(X + Y) = \theta(X) + \theta(Y). \quad \blacksquare$$

The most important characteristic of the BCNF is given in the following theorem.

*Theorem 4: The relation $R[\Omega]$ is in BCNF iff every atomic filter $[\pi)$ of $L(R[\Omega])$ is isomorphic to the Boolean algebra $(\mathbf{2}^{\Omega_\pi}, \cdot, +, ^{-})$.*

*Proof*: (Necessity) Since $[\pi)$ is a meet-morphic image of $\theta$ restricted to $\mathbf{2}^{\Omega_\pi}$, it is sufficient to show that $\theta$ is a one-to-one mapping on $\mathbf{2}^{\Omega_\pi}$. Let $X, Y \in \mathbf{2}^{\Omega_\pi}$, and $\theta(X) = \theta(Y)$. It follows that

$$\theta(X) = \theta(Y - X)\theta(Y + X) \leqq \theta(Y - X),$$

which implies

$$X \rightarrow Y - X.$$

Since $\mathbf{0} < \pi \leqq \theta(X)$ and $[\pi)$ is normal, we have

$$Y - X \subseteq X.$$

Hence,

$$Y \subseteq X.$$

Similarly,

$$X \subseteq Y.$$

Therefore, $X = Y$, and $\theta$ is a one-to-one mapping on $\mathbf{2}^{\Omega_\pi}$.

(Sufficiency) Suppose $X \rightarrow Y$ is valid in $R[\Omega_\pi]$. Then $\theta(X) \leqq \theta(Y)$. Since the inverse of an isomorphism is also order-preserving, it follows that $X \supseteq Y$. Therefore, $[\pi)$ is normal and $R[\Omega]$ is in BCNF. ■

The above theorem implies that if $[\pi)$ is normal, the only key of $R[\Omega_\pi]$ is $\theta^{-1}(\pi) = \Omega_\pi$.

It is known that any relation has a lossless-join decomposition into Boyce-Codd Normal Form, and an algorithm for determining the decomposition is given in Ref. 6. We will show how the concept of the principal filter can be used to modify this algorithm. In the algorithm for synthesizing the Third Normal Form,[5] a concept similar to the principal filter is used implicitly by Bernstein when he partitions the functional dependencies (Step 2). Before describing the improved algorithm, we need the following:

*Lemma 4: Let $R$ be a relation on $\Omega$. Let $\pi \in L(R[\Omega])$ be an atom of the relation lattice, and let $K$ be a key of the atomic projection $R[\Omega_\pi]$. Then,*

$$R[\Omega] = R[(\Omega - \Omega_\pi)K] \,|\times|\, R[\Omega_\pi].$$

*Proof*: $K \subseteq \Omega_\pi$ and $K \rightarrow \Omega_\pi$. ■

The algorithm for determining the lossless-join decomposition into BCNF is simply to construct a sequence of decompositions $D_i = (R_1, \cdots, R_m)$ of $R$, each with lossless join: Initially, let $D_0$ consist of $R$ alone. If $T[\Omega]$ is a relation in $D_i$, and $T[\Omega]$ is not in BCNF, let $\pi$ be an atom of $L(T[\Omega])$ for which the principal filter $[\pi)$ is abnormal. Let $K$

Table II—Relation $R[MSPCNY]$

| | Model Number | Serial Number | Price | Color | Name | Year |
|---|---|---|---|---|---|---|
| 1 | 1234 | 342 | 13.25 | blue | pot | 1974 |
| 2 | 1234 | 347 | 13.25 | red | pot | 1974 |
| 3 | 1234 | 410 | 14.23 | red | pot | 1975 |
| 4 | 1465 | 347 | 9.45 | black | pan | 1974 |
| 5 | 1465 | 390 | 9.82 | black | pan | 1976 |
| 6 | 1465 | 392 | 9.82 | red | pan | 1976 |
| 7 | 1465 | 401 | 9.82 | red | pan | 1976 |
| 8 | 1465 | 409 | 9.82 | blue | pan | 1976 |
| 9 | 1623 | 311 | 22.34 | blue | kettle | 1973 |
| 10 | 1623 | 390 | 30.21 | blue | kettle | 1976 |
| 11 | 1623 | 410 | 28.55 | black | kettle | 1975 |
| 12 | 1623 | 423 | 28.55 | black | kettle | 1975 |
| 13 | 1623 | 428 | 28.55 | blue | kettle | 1975 |
| 14 | 1654 | 435 | 28.55 | red | kettle | 1975 |

be a key of the atomic projection $T[\Omega_\pi]$. Now replace $T[\Omega]$ in $D_i$ by $T[\Omega - \Omega_\pi)K]$ and $T[\Omega_\pi]$ to obtain $D_{i+1}$. Continue the process until all the relations in the decomposition $D_k$ are in BCNF.

Example 3: Let us consider the relation $R[MSPCNY]$ from Ref. 23, where $M$ = model number, $S$ = serial number, $P$ = price, $C$ = color, $N$ = name, and $Y$ = year. The tuples of the relation $R[MSPCNY]$ are shown in Table II.

The Hasse diagram of the relation lattice $L(R[\Omega])$ is illustrated in Fig. 2, where

$$\pi_1 = \{\overline{1, 8, 9, 10, 13}; \overline{2, 3, 6, 7, 14}; \overline{4, 5, 11, 12}\},$$

$$\pi_2 = \{\overline{1, 2, 3}; \overline{4, 5, 6, 7, 8}; \overline{9, 10, 11, 12, 13, 14}\},$$

$$\pi_3 = \{\overline{1, 2, 4}; \overline{3, 11, 12, 13, 14}; \overline{5, 6, 7, 8, 10}; \overline{9}\},$$

$$\pi_4 = \{\overline{1, 2, 3}; \overline{4, 5, 6, 7, 8}; \overline{9, 10, 11, 12, 13}; \overline{14}\},$$

$$\pi_5 = \{\overline{1, 2}; \overline{3}; \overline{4}; \overline{5, 6, 7, 8}; \overline{9}; \overline{10}; \overline{11, 12, 13, 14}\},$$

$$\pi_6 = \{\overline{1}; \overline{2, 3}; \overline{4, 5}; \overline{6, 7}; \overline{8}; \overline{9, 10, 13}; \overline{11, 12}; \overline{14}\},$$

$$\pi_7 = \{\overline{1, 2}; \overline{3}; \overline{4}; \overline{5, 6, 7, 8}; \overline{9}; \overline{10}; \overline{11, 12, 13}; \overline{14}\},$$

$$\pi_8 = \{\overline{1}; \overline{2, 4}; \overline{3, 11}; \overline{5, 10}; \overline{6}; \overline{7}; \overline{8}; \overline{9}; \overline{12}; \overline{13}; \overline{14}\},$$

$$\pi_9 = \{\overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}; \overline{6, 7}; \overline{8}; \overline{9}; \overline{10}; \overline{11, 12}; \overline{13}; \overline{14}\},$$

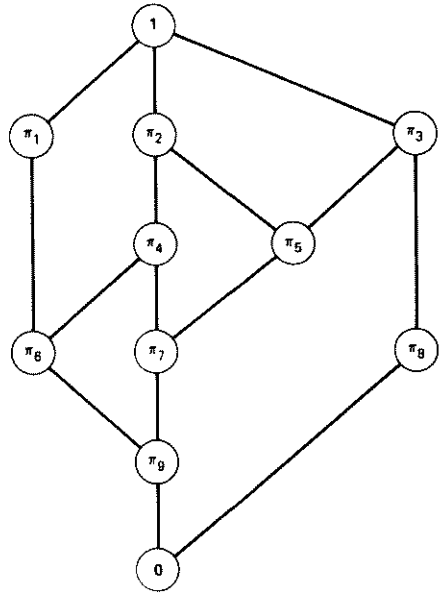



Fig. 2—Relation lattice $L(R[MSPCNY])$.

and $\theta(C) = \pi_1$, $\theta(N) = \pi_2$, $\theta(Y) = \pi_3$, $\theta(M) = \pi_4$, $\theta(P) = \pi_5$, $\theta(S) = \pi_8$. For $X \subseteq \Omega$, $\theta(X)$ can be obtained easily by carrying out the meet operations on the attributes in $X$.

The principal ideal functions of $R[MSPCNY]$ are

$$f_C(M, S, P, C, N, Y) = C + MS + NS + PS,$$

$$f_N(M, S, P, C, N, Y) = N + M + P + CY + CS,$$

$$f_P(M, S, P, C, N, Y) = P + MY + CS + MS + NS,$$

$$f_M(M, S, P, C, N, Y) = M + CN + CP + CY + NS + PS + CS,$$

$$f_Y(M, S, P, C, N, Y) = Y + P + S,$$

$$f_S(M, S, P, C, N, Y) = S,$$

and the key Boolean function is

$$\begin{aligned} F(M, S, P, C, N, Y) &= (C + MS + NS + PS)\cdot(N + M + P + CY \\ &\quad + CS) \\ &\quad \cdot(P + MY + CS + MS + NS) \\ &\quad \cdot(M + CN + CD + CY + NS + PS + CS) \\ &\quad \cdot(Y + P + S)\cdot S \\ &= CS + MS + NS + PS. \end{aligned}$$

The keys of $R[MSPCNY]$ are $\{CS, MS, NS, PS\}$, and $Y$ is the only nonprime attribute.

Initially, let $D_0 = \{R(MSPCNY)\}$. Since both atomic filters $[\pi_8)$ and $[\pi_9)$ are abnormal, we arbitrarily choose $\pi_9$, and let $\Sigma = \Omega_{\pi_9} = MPCNY$. The relation lattice of $R[\Sigma]$ is isomorphic to $[\pi_9)$. The principal ideal functions of $R[\Sigma]$ are

$$g_C(M, P, C, N, Y) = f_C(M, \mathbf{0}, P, C, N, Y) = C,$$

$$g_N(M, P, C, N, Y) = f_N(M, \mathbf{0}, P, C, N, Y) = N + M + P + CY,$$

$$g_P(M, P, C, N, Y) = f_P(M, \mathbf{0}, P, C, N, Y) = P + MY,$$

$$g_M(M, P, C, N, Y) = f_M(M, \mathbf{0}, P, C, N, Y) = M + CN + CP + CY,$$

$$g_Y(M, P, C, N, Y) = f_Y(M, \mathbf{0}, P, C, N, Y) = Y + P,$$

and the key Boolean function is

$$\begin{aligned} G(M, P, C, N, Y) &= C\cdot(N + M + P + CY)\cdot(P + MY) \\ &\quad \cdot(M + CN + CP + CY)\cdot(Y + P) \\ &= CP + CMY. \end{aligned}$$

We choose the key $CP$ and replace $R[MSPCNY]$ in $D_0$ by $R[(\Omega - \Sigma)K] = R[SPC]$ and $R[\Sigma] = R[MPCNY]$ to obtain $D_1 = \{R[SPC], R[MPCNY]\}$. The relation $R[SPC]$ and its lattice are shown in Table III and Fig. 3, respectively.

The relation $R[SPC]$ is in BCNF, but the relation $R[MPCNY]$ is not. The relation lattice of $R[MPCNY]$ is isomorphic to the filter $[\pi_9)$. We will not duplicate the figure. Both "atoms" $\pi_6$ and $\pi_7$ of $[\pi_9)$ are abnormal. We choose the filter $[\pi_7)$. The principal ideal functions of $R[\Sigma_{\pi_7}] = R[MPNY]$ are

$$h_M(M, P, N, Y) = g_M(M, P, \mathbf{0}, N, Y) = M,$$
$$h_N(M, P, N, Y) = g_N(M, P, \mathbf{0}, N, Y) = N + M + P,$$
$$h_P(M, P, N, Y) = g_P(M, P, \mathbf{0}, N, Y) = P + NY,$$
$$h_Y(M, P, N, Y) = g_Y(M, P, \mathbf{0}, N, Y) = Y + P,$$

and the key Boolean function of $R[MPNY]$ is given by

$$H(M, P, N, Y) = M \cdot (N + M + P) \cdot (P + MY) \cdot (Y + P)$$
$$= MP + MY.$$

We choose the key $K' = MP$ and replace $R[MPCNY]$ in $D_1$ by $R[(\Sigma - \Sigma_{\pi_7})K'] = R[MPC]$ and $R[MPNY]$ to obtain $D_2 = \{R[SPC], R[MPC], R[MPNY]\}$. The relation $R[MPC]$ and its relation lattice are illustrated in Table IV and Fig. 4, respectively.

Now we have to decompose the relation $R[MPNY]$ in $D_2$. The relation lattice of $R[MPNY]$ is isomorphic to $[\pi_7)$ of $L(R[MSPCNY])$. We choose the abnormal filter that is isomorphic to $[\pi_5)$. Since $\Sigma_{\pi_5} = PNY$ and the only key is $P$, we can replace $R[MPNY]$ in $D_2$ by $R[MP]$ and $R[PNY]$ to obtain $D_3 = \{R[SPC], R[MPC], R[MP], R[PNY]\}$. All the relations in $D_3$ are in BCNF. The relations $R[MP]$, $R[PNY]$ and

Table III—Relation $R[SPC]$

| | Serial Number | Price | Color |
|---|---|---|---|
| 1 | 342 | 13.25 | blue |
| 2 | 347 | 13.25 | red |
| 3 | 410 | 14.23 | red |
| 4 | 347 | 9.45 | black |
| 5 | 390 | 9.82 | black |
| 6 | 392 | 9.82 | red |
| 7 | 401 | 9.82 | red |
| 8 | 409 | 9.82 | blue |
| 9 | 311 | 22.34 | blue |
| 10 | 390 | 30.21 | blue |
| 11 | 410 | 28.55 | black |
| 12 | 423 | 28.55 | black |
| 13 | 428 | 28.55 | blue |
| 14 | 435 | 28.55 | red |

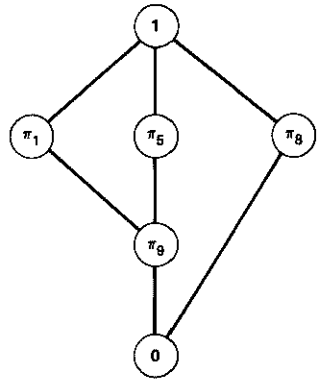



Fig. 3—Relation lattice $L(R[SPC])$.

Table IV—Relation $R[MPC]$

| | Model Number | Price | Color |
|---|---|---|---|
| 1 | 1234 | 13.25 | blue |
| 2 | 1234 | 13.25 | red |
| 3 | 1234 | 14.23 | red |
| 4 | 1465 | 9.45 | black |
| 5 | 1465 | 9.82 | black |
| (6, 7) | 1465 | 9.82 | red |
| 8 | 1465 | 9.82 | blue |
| 9 | 1623 | 22.34 | blue |
| 10 | 1623 | 30.21 | blue |
| (11, 12) | 1623 | 28.55 | black |
| 13 | 1623 | 28.55 | blue |
| 14 | 1654 | 28.55 | red |

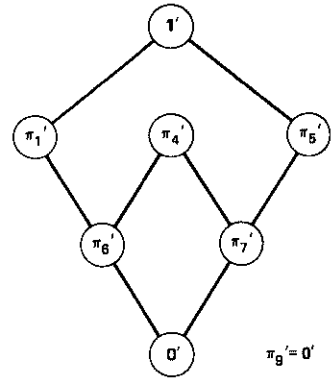



Fig. 4—Relation lattice $L(R[MPC])$.

their respective lattices are shown in Tables V and VI, and Figs. 5 and 6. ■

## VI. MULTIVALUED DEPENDENCIES

Multivalued dependency (MVD) proposed by Fagin[7] and Zaniolo[8] is the necessary and sufficient condition for a (binary) lossless-join decomposition. A similar concept, called hierarchical dependency, was defined by Delobel.[24] A bit later, the concept of multivalued dependency was generalized to join dependency by Rissanen.[10,11] A set of "axioms" or inference rules for multivalued dependencies was given by Beeri, Fagin, and Howard.[25] We know from our previous discussion that functional dependency is equivalent to partial ordering in the partition lattice. In this section we show that multivalued dependency

Table V—Relation $R[MP]$

| | Model Number | Price |
|---|---|---|
| (1, 2) | 1234 | 13.25 |
| 3 | 1234 | 14.23 |
| 4 | 1465 | 9.45 |
| (5, 6, 7, 8) | 1465 | 9.82 |
| 9 | 1623 | 22.34 |
| 10 | 1623 | 30.21 |
| (11, 12, 13) | 1623 | 28.55 |
| 14 | 1654 | 28.55 |

Table VI—Relation $R[PNY]$

| | Price | Name | Year |
|---|---|---|---|
| (1, 2) | 13.25 | pot | 1974 |
| 3 | 14.23 | pot | 1975 |
| 4 | 9.45 | pan | 1974 |
| (5, 6, 7, 8) | 9.82 | pan | 1976 |
| 9 | 22.34 | kettle | 1973 |
| 10 | 30.21 | kettle | 1976 |
| (11, 12, 13, 14) | 28.55 | kettle | 1975 |

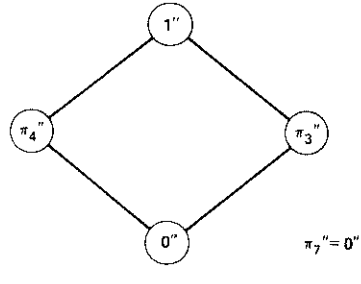



Fig. 5—Relation lattice $L(R[MP])$.

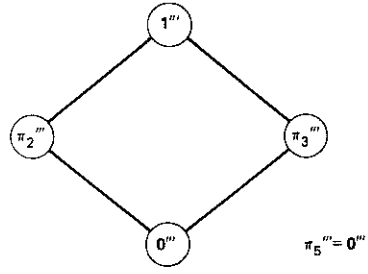



Fig. 6—Relation lattice $L(R[PNY])$.

is equivalent to a lattice equation. First, however, we state the definition of MVD and show that MVD guarantees information-lossless join decomposition.

Definition 5: Let $R$ be a relation on the set of attributes $\Omega = XYZ$, where $X$, $Y$, and $Z$ are disjoint subsets of $\Omega$. We say there is a multivalued dependency $X \longrightarrow\!\!\!\!\!\rightarrow Y$ if

$$R_{xz}[Y] = R_x[Y], \qquad \forall (x) \in R[X], \qquad (z) \in R[Z]. \quad \blacksquare$$

*Lemma 5: Let $R$ be a relation on $\Omega = XYZ$, where $X$, $Y$, and $Z$ are disjoint subsets. Then,*

$$R[XYZ] = R[XY] \mid \times \mid R[XZ]$$

*iff*

$$R_x[YZ] = |R_x[Y]| \cdot |R_x[Z]|, \qquad \forall (x) \in R[X].$$

*Proof*: (Necessity) $R[XYZ] = R[YX] \mid \times \mid R[XZ]$ implies

$$R_x[YZ] = R_x[Y] \times R_x[Z], \qquad \forall (x) \in R[X].$$

Hence,

$$|R_x[YZ]| = |R_x[Y]| \cdot |R_x[Z]|.$$

(Sufficiency) It is easy to verify that

$$R_x[YZ] \subseteq R_x[Y] \times R_x[Z], \qquad \forall (x) \in R[X].$$

The given cardinal identity assures that

$$R_x[YZ] = R_x[Y] \times R_x[Z], \qquad \forall (x) \in R[X]. \quad \blacksquare$$

*Theorem 5: Let $R$ be a relation on the set of attributes $\Omega = XYZ$, where $X$, $Y$, and $Z$ are disjoint subsets.* Then,*

$$R[XYZ] = R[XY] \mid \times \mid R[XZ] \textit{ iff } X \longrightarrow\!\!\!\!\!\rightarrow Y.$$

---

* For convenience, we assume $X$, $Y$, and $Z$ to be disjoint. It will later become clear that this assumption is not necessary.

*Proof*: (Necessity) From Lemma 5, it is sufficient to show that

$$|R_x[YZ]| = |R_x[Y]| \cdot |R_x[Z]|, \qquad \forall (x) \in R(X)$$

iff

$$R_{xx}[Y] = R_x[Y], \qquad \forall (x) \in R(X), \qquad (z) \in R(Z).$$

Since

$$R[XYZ] = R[XY] \mid \times \mid R[XZ]$$

implies

$$R_{xz}[Y] \times (x, z) = R_x[Y] \times (x, z), \qquad \forall (x) \in R[X], \qquad (z) \in R[Z].$$

Hence,

$$R_{xz}[Y] = R_x[Y].$$

(Sufficiency) For every $(x) \in R[X]$, we have

$$\begin{aligned}(x) \times R_x[ZY] &= \{(x, z_i) \times R_{xz_i}[Y] \mid (x, z_i) \in R[XZ]\} \\ &= \{(x, z_i) \times R_x[Y] \mid (x, z_i) \in R[XZ]\} \\ &= (x) \times R_x[Z] \times R_x[Y].\end{aligned}$$

Since $|x| = 1$, it follows that

$$|R_x[YZ]| = |R_x[Y]| \cdot |R_x[Z]|, \qquad \forall (x) \in R[X]. \quad \blacksquare$$

We need the commutative property of the product of two equivalence relations (partitions) to establish the lattice equation of multivalued dependency. The product of two equivalence relations may not be an equivalence relation; if it is an equivalence relation then the product must be commutative and vice versa.

Definition 6: Two binary relations $\rho$ and $\rho'$ and $S$ are *permutable* (commute) if and only if $\rho \circ \rho' = \rho' \circ \rho$. This means that if $a \; \rho \; x \; \rho' \; b$ for some $x \in S$, then $a \; \rho' \; y \; \rho \; b$ for some $y \in S$, and conversely.[18] ■

*Lemma 6*: *Let $\rho$ and $\rho'$ be equivalence relations (partitions) on $S$. Then the following are equivalent*:

1. $\rho \circ \rho' = \rho \circ \rho$
2. $\rho \circ \rho' = \rho \oplus \rho'$
3. $\rho \circ \rho'$ is an equivalence relation
4. $\rho \circ \rho'$ is symmetric.

*Proof*: The proof of Lemma 6 is given in Ref. 21. ■

*Lemma 7*: *Let $R$ be a relation on the set of attributes $\Omega$ and let $X, Y, Z \subseteq \Omega$. Then,*

$$\theta(X) = \theta(XY) + \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY)$$

*iff*

$$\theta(X) \subseteq \theta(XY) \circ \theta(XZ).$$

*Proof*: (Necessity) Trivial.

(Sufficiency) Suppose $t_1\theta(XY) \circ \theta(XZ)t_2$. Then there exists $t_3 \in R[\Omega]$ such that

$$t_1\theta(XT)t_3\theta(XZ)t_2,$$

which implies

$$t_1\theta(X)t_3\theta(X)t_2.$$

Therefore,

$$t_1\theta(X)t_2.$$

Hence,

$$\theta(XY) \circ \theta(XZ) \subseteq \theta(X).$$

It follows that

$$\theta(X) = \theta(XY) \circ \theta(XZ).$$

From Lemma 6, we have

$$\theta(X) = \theta(XY) \oplus \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY).$$

Since

$$\theta(XY) \leqq \theta(XY) + \theta(XZ) \quad \text{and} \quad \theta(XZ) \leqq \theta(XY) + \theta(XZ),$$

by the definition of the join operation $\oplus$ in $\prod(R[\Omega])$, we must have

$$\theta(X) = \theta(XY) \oplus \theta(XZ) \leqq \theta(XY) + \theta(XZ).$$

But,

$$\theta(XY) + \theta(XZ) \leqq \theta(X) + \theta(X) = \theta(X),$$

so it follows that

$$\theta(X) = \theta(XY) + \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY). \quad \blacksquare$$

The following theorem shows that the multivalued dependency can be formulated as a lattice equation.

*Theorem 6*: *Let $R$ be a relation on the set of attributes $\Omega = XYZ$, where $X$, $Y$, and $Z$ are disjoint subsets. Then, $R[XYZ] = R[XY] \mid\times\mid R[XZ]$ iff*

$$\theta(X) = \theta(XY) + \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY).$$

*Proof*: (Necessity) Since $R[XYZ] = R[XY] \mid\times\mid R[XZ]$ implies

$$R_x[YZ] = R_x[Y] \times R_x[Z], \qquad \forall (x) \in R[X],$$

there is a one-to-one and onto mapping $\phi_x$: $R_x[YZ] \rightarrow R_x[Y] \times R_x[Z]$, which takes every tuple $(y, z) \in R_x[YZ]$ into $\phi_x((y, z)) = ((y), (z)) \in R_x[Y] \times R_x[Z]$, $\forall(x) \in R[X]$. Suppose $t_1, t_2 \in R[XYZ]$ and $t_1\theta[X]t_2$, and assume $t_1 = (x, y_1, z_1)$ and $t_2 = (x, y_2, z_2)$. As

$$(y_1, z_1), (y_2, z_2) \in R_x[YZ] = R_x[Y] \times R_x[Z],$$

we have

$$(y_1), (y_2) \in R_x[Y] \quad \text{and} \quad (z_1), (z_2) \in R_x[Z].$$

Since $\phi_x$ is an onto mapping, there must exist two tuples $t_3 = (x, y_1, z_2)$, and $t_4 = (x, y_2, z_1) \in R[XYZ]$. Hence,

$$t_1\theta(XY)t_3\theta(XZ)t_2,$$

which means

$$t_1\theta(XY) \circ \theta(XZ)t_2.$$

It follows that

$$\theta(X) \subseteq \theta(XY) \circ (XZ).$$

From Lemma 7, we have

$$\theta(X) = \theta(XY) + \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY).$$

(Sufficiency) We know $R[XYZ] \subseteq R[XY] \mid \times \mid R[XZ]$. Suppose $t = (x, y, z) \in R\,[XY] \mid \times \mid R[XZ]$. Then there exist $t_1 = (x, y, z')$, $t_2 = (x, y', z) \in R[XYZ]$. Thus,

$$t_1\theta(X)t_2,$$

which implies

$$t_1\theta(XY) \circ \theta(XZ)t_2.$$

There must exist $t_3 \in R[XYZ]$ such that

$$t_1\theta(XY)t_3\theta(XZ)t_2.$$

Therefore,

$$t_3 = (x, y, z) = t \in R[XYZ],$$

and thus

$$R[XY] \mid \times \mid R[XZ] \subseteq R[XYZ].$$

Hence,

$$R[XYZ] = R[XY] \mid \times \mid R[XZ]. \quad \blacksquare$$

It should be noted that in the above proof we use the fact that $\Omega = XYZ$ and $\theta(\Omega) = \theta(XYZ) = \mathbf{0}$, i.e., there are no duplicated tuples in

$R[\Omega]$. The inference rules of MVD are given and proved in Appendix B.

Example 4: Consider the relation $R[ECSY]$ of Example 1. We have the MVD: $E \twoheadrightarrow SY$, where $\theta(E) = \pi_1 = \{1, 2; 3, 4, 5, 6; 7, 8\}$, $\theta(EC) = \pi_2 = \{1, 2; 3, 4; 5, 6; 7, 8\}$, $\theta(ESY) = \pi_5 = \{1; 2; 3, 5; 4, 6; 7; 8\}$. It is easy to verify that

$$\pi_1 = \pi_2 + \pi_5 = \pi_2 \circ \pi_5 = \pi_5 \circ \pi_2. \quad \blacksquare$$

It is known that if $R$ is a relation on $\Omega = XYZ$, and $X \twoheadrightarrow Y$ then $X \twoheadrightarrow Z$. The symmetricity of the MVD can easily be seen in the lattice equation of Lemma 7.

If $XYZ \subset \Omega$ and $\theta(X) = \theta(XY) + \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY)$ holds, then $X \twoheadrightarrow Y|Z$ is called an embedded multivalued dependency (EMVD)[7]; this is simply a multivalued dependency in the projection $R[XYZ]$ of $R[\Omega]$.

Theorem 6 clearly indicates that the MVD is actually a condition pertaining to data *independency* rather than data dependency. For this reason, we introduce the notion of decomposition of two sets of attributes in a relation as follows.

Definition 7: Let $R$ be a relation on the set of attributes $\Omega$. The two sets of attributes $\Omega_1, \Omega_2 \subseteq \Omega$ are decomposable in $R$ if

$$\theta(\Omega_1 + \Omega_2) = \theta(\Omega_1) + \theta(\Omega_2) = \theta(\Omega_1) \circ \theta(\Omega_2) = \theta(\Omega_2) \circ \theta(\Omega_1). \quad \blacksquare$$

It is easy to see that $\Omega_1$ and $\Omega_2$ are decomposable in $\Omega$ iff $\Omega_1 + \Omega_2 \rightarrow \rightarrow \Omega_1 - \Omega_2 | \Omega_2 - \Omega_1$ is an EMVD in $R$. Furthermore, if $\Omega_1\Omega_2 = \Omega$ then $\Omega_1 + \Omega_2 \twoheadrightarrow \Omega_1 - \Omega_2$ (or $\Omega_1 + \Omega_2 \twoheadrightarrow \Omega_2 - \Omega_1$) is an MVD in $R$. In the latter case, $(\Omega_1, \Omega_2)$ is called a *decomposition pair* by Armstrong and Delobel.[26]

We feel that decomposition is a basic concept in the study of the structure of databases. It can be naturally generalized to the concepts of projective decomposition and mutual decomposition. Projective composition concerns the data independence of two sets of attributes on the projection of a relation. Mutual decomposition extends the concept of decomposition to more than two sets of attributes.

Let $\rho$ be a partition on the set $S$, the function $\rho^* = S \rightarrow S/\rho$ maps $a \in S$ into $(a)\rho^* = a_\rho$ is called the *canonical function* of $\rho$. For $S = \{a, b, \ldots, e\}$, we will use the notation

$$\rho^* = \begin{pmatrix} a & b & \cdots & e \\ a_\rho & b_\rho & \cdots & e_\rho \end{pmatrix}$$

to illustrate the canonical function $\rho^*$. The equivalence relation

$$\ker \rho^* = \rho^* \circ \rho^{*-1} = \{(a, b) \in S \times S | \rho^*(a) = \rho^*(b)\}.$$

is called the *kernel* of $\rho^*$. Notice that $\ker \rho^* = \rho$.

Let $\rho$ and $\sigma$ be partitions on $S = \rho \leqq \sigma$; then there is a unique function $f$ from $S/\rho$ onto $S/\sigma$ such that $(a_\rho)f = a_\sigma$. The kernal of $f$,

$$\ker f = f \circ f^{-1} = \{(a_\rho, b_\rho) \in S/\rho \times S/\rho \mid a \; \sigma \; b\},$$

is an equivalence on $S/\rho$. It is usual to write ker $f$ as $\sigma/\rho$, the quotient of $\sigma$ and $\rho$. Note that $a_\rho(\sigma/\rho)b_\rho$ if and only if $a \; \sigma \; b$ and the mapping $g$: $(S/\rho)/(\sigma/\rho) \rightarrow S/\sigma$ defined by $((a_\rho)_{\sigma/\rho})g = a_\sigma$ is one-to-one and onto. Thus the function $f$ defined above is in fact the canonical function of $\sigma/\rho$, i.e., $f = (\sigma/\rho)^*$. It is easy to see the diagram in Fig. 7 commutes, that is $\rho^* \circ (\sigma/\rho)^* = \sigma^*$.

Example 5: Let $\rho$, $\sigma$ be partitions on the set $S = \{1, 2, 3, 4, 5, 6, 7, 8\}$, such that $\rho = \{1, 2; 3, 4; 5, 6; 7, 8\}$ and $\sigma = \{1, 2, 3, 4; 5, 6, 7, 8\}$ with $\rho \leqq \sigma$. Then $S/\rho = \{I, II, III, IV\}$, where $I = 1, 2$, $II = 3, 4$, $III = 5, 6$, $IV = 7, 8$, and $S/\sigma = \{\alpha, \beta\}$, where $\alpha = 1, 2, 3, 4$, $\beta = 5, 6, 7, 8$. The canonical functions of $\rho$ and $\sigma$ are

$$\rho^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ I & I & II & II & III & III & IV & IV \end{pmatrix}$$

and

$$\sigma^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \alpha & \alpha & \alpha & \alpha & \beta & \beta & \beta & \beta \end{pmatrix}.$$

It follows that

$$\sigma/\rho = \{\overline{I, II}; \overline{III, IV}\}$$

and

$$(\sigma/\rho)^* = \begin{pmatrix} I & II & III & IV \\ \alpha & \alpha & \beta & \beta \end{pmatrix}. \quad \blacksquare$$

*Lemma 8: Let $\rho$, $\sigma_1$, $\sigma_2$ be partitions on $S$ such that $\rho \leqq \sigma_1$, $\rho \leqq \sigma_2$, and $\sigma_1 \circ \sigma_2 = \sigma_2 \circ \sigma_1$. Then*

$$(\sigma_1 \circ \sigma_2)/\rho = (\sigma_1/\rho) \circ (\sigma_2/\rho).$$

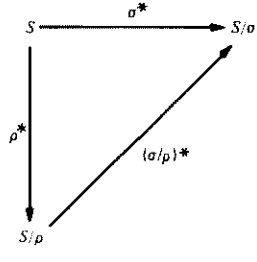



Fig. 7—Canonical function of quotient partition.

*Proof:* It is clear that $\sigma_1 \circ \sigma_2$ is a partition on $S$ and $\rho \leqq \sigma_1 \circ \sigma_2 = \sigma_2 \circ \sigma_1$. It follows from the definition of quotient partition that the lemma is true. ■

*Lemma 9: Let $\rho$, $\sigma_1$, $\sigma_2$ be partitions on $S$, such that $\rho \leqq \sigma_1$, $\rho \leqq \sigma_2$. Then*

$$\sigma_1 \circ \sigma_2 = \sigma_2 \circ \sigma_1$$

*iff*

$$(\sigma_1/\rho) \circ (_2/\rho) = (\sigma_2/\rho) \circ (\sigma_1/\rho).$$

*Proof:* (Necessity)

$$(\sigma_1/\rho) \circ (\sigma_2/\rho) = (\sigma_1 \circ \sigma_2)/\rho = (\sigma_2 \circ \sigma_1)/\rho = (\sigma_2/\rho) \circ (\sigma_1/\rho).$$

(Sufficiency) Suppose $a\sigma_1 \circ \sigma_2 b$. Then there is a $c \in S$ such that $a\sigma_1 c\sigma_2 b$. It follows that

$$a_\rho(\sigma_1/\rho)c_\rho(\sigma_2/\rho)b_\rho.$$

There must be a $d \in S$ such that

$$a_\rho(\sigma_2/\rho)d_\rho(\sigma_1/\rho)b_\rho.$$

Thus

$$a\sigma_2 d\sigma_1 b,$$

and

$$a\sigma_2 \circ \sigma_1 b.$$

Hence

$$\sigma_1 \circ \sigma_2 \subseteq \sigma_2 \circ \sigma_1.$$

Similarly, we have $\sigma_2 \circ \sigma_1 \subseteq \sigma_1 \circ \sigma_2$. Then $\sigma_1 \circ \sigma_2 = \sigma_2 \circ \sigma_1$. ■

Definition 8: Let $R$ be a relation on the set of attributes $\Omega$. For $\Omega_1$, $\Omega_2 \subseteq \Omega$, the *projective partition* defined by

$$\theta(\Omega_1|\,\Omega_2) = \theta(\Omega_1 + \Omega_2)/\theta(\Omega_2)$$

is a partition on the set of tuples of $R[\Omega]/\theta(\Omega_2) = R[\Omega_2]$. The canonical function of $\theta(\Omega_1|\,\Omega_2)$ is denoted by $\theta^*(\Omega_1|\,\Omega_2) = (\theta(\Omega_1 + \Omega_2)/\theta(\Omega_2))^*$, which satisfies $\theta^*(\Omega_2) \circ \theta^*(\Omega_1|\,\Omega_2) = \theta^*(\Omega_1 + \Omega_2)$. ■

Certain properties of the projective partition are demonstrated in the following theorems and their proof directly follows from the definition of projective partition.

*Theorem 7: Let $R$ be a relation on the set of attributes $\Omega$ and $\Omega_1, \cdots, \Omega_n \subseteq \Omega$. Then*

$$\theta^*\left(\bigcap_{k=1}^{n} \Omega_k\right) = \theta^*(\Omega_1) \circ \theta^*(\Omega_2|\,\Omega_1) \circ \cdots \circ \theta^*(\Omega_n|\bigcap_{k=1}^{n-1} \Omega_k). \quad ■$$

*Theorem 8: Let $R$ be a relation on the set of attributes $\Omega$ and $X$, $\Omega_1$, $\cdots$, $\Omega_n \subseteq \Omega$, such that*

$$\bigcup_{k=1}^{n} \Omega_k = \Omega.$$

*Then*

1. $\theta(X) = \prod_{k=1}^{n} \ker(\theta^*(\Omega_k) \circ \theta^*(X|\Omega_k))$
2. $\theta(\Omega_m|X) = \ker(\theta^*(\Omega_m) \circ \theta^*(X|\Omega_m))/\prod_{k=1}^{n} \ker(\theta^*(\Omega_k) \circ \theta^*(X|\Omega_k))$. ■

Definition 9: Let $R$ be a relation on the set of attributes $\Omega$ and $\Omega_1$, $\Omega_2$, $\Sigma \subseteq \Omega$. We say $\Omega_1$ and $\Omega_2$ are projectively decomposable on $\Sigma$ if

$$\begin{aligned}\theta(\Omega_1 + \Omega_2|\Sigma) &= \theta(\Omega_1|\Sigma) + \theta(\Omega_2|\Sigma) \\ &= \theta(\Omega_1|\Sigma) \circ \theta(\Omega_2|\Sigma) \\ &= \theta(\Omega_2|\Sigma) \circ \theta(\Omega_1|\Sigma). \quad ■\end{aligned}$$

The EMVD is a special case of projective decomposition, which can be seen from the following theorem.

*Theorem 9: Let $R$ be a relation on $\Omega$, and let $\Omega_1$, $\Omega_2$, $\Sigma \subseteq \Omega$. Then $\Omega_1$ and $\Omega_2$ are projectively decomposable on $\Sigma$ iff*

$$\begin{aligned}\theta(\Omega_1 + \Omega_2 + \Sigma) &= \theta(\Omega_1 + \Sigma) + \theta(\Omega_2 + \Sigma) \\ &= \theta(\Omega_1 + \Sigma) \circ \theta(\Omega_2 + \Sigma) = \theta(\Omega_2 + \Sigma) \circ \theta(\Omega_1 + \Sigma).\end{aligned}$$

*Proof:* The proof follows from Lemma 8 and 9. ■

Example 6: Consider the relation $R$ on $\Omega = ABCDE$ in Table VII. The Hasse diagram of the relation lattice $L(R[\Omega])$ is shown in Fig. 8, where

$$\begin{aligned}\pi_1 &= \{\overline{1, 2, 3, 5, 6, 7};\ \overline{4}\} = \theta(A), \\ \pi_2 &= \{\overline{1, 3, 4};\ \overline{2, 5, 6, 7}\} = \theta(B), \\ \pi_3 &= \{\overline{1, 6, 7};\ \overline{2, 3, 4, 5}\} = \theta(C), \\ \pi_4 &= \{\overline{1, 3};\ \overline{2, 5, 6, 7};\ \overline{4}\} = \theta(AB), \\ \pi_5 &= \{\overline{1, 6, 7};\ \overline{2, 3, 5};\ \overline{4}\} = \theta(AC), \\ \pi_6 &= \{\overline{1};\ \overline{2, 5};\ \overline{3, 4};\ \overline{6, 7}\} = \theta(BC), \\ \pi_7 &= \{\overline{1, 6, 7};\ \overline{2, 5};\ \overline{3};\ \overline{4}\} = \theta(D), \\ \pi_8 &= \{\overline{1, 3};\ \overline{2, 6};\ \overline{4};\ \overline{5, 7}\} = \theta(E), \\ \pi_9 &= \{\overline{1};\ \overline{2, 5};\ \overline{3};\ \overline{4};\ \overline{6, 7}\} = \theta(ABC) = \theta(ABD).\end{aligned}$$

Let $\Omega_1 = ABD$, $\Omega_2 = ACE$, $\Sigma = ABC$. We find that

Table VII—Relation lattice $L(R[\Omega])$

| | $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|---|
| 1 | $a_1$ | $b_1$ | $c_1$ | $d_1$ | $e_1$ |
| 2 | $a_1$ | $b_2$ | $c_2$ | $d_2$ | $e_2$ |
| 3 | $a_1$ | $b_1$ | $c_2$ | $d_3$ | $e_1$ |
| 4 | $a_2$ | $b_1$ | $c_2$ | $d_4$ | $e_3$ |
| 5 | $a_1$ | $b_2$ | $c_2$ | $d_2$ | $e_4$ |
| 6 | $a_1$ | $b_2$ | $c_1$ | $d_1$ | $e_2$ |
| 7 | $a_1$ | $b_2$ | $c_1$ | $d_1$ | $e_4$ |

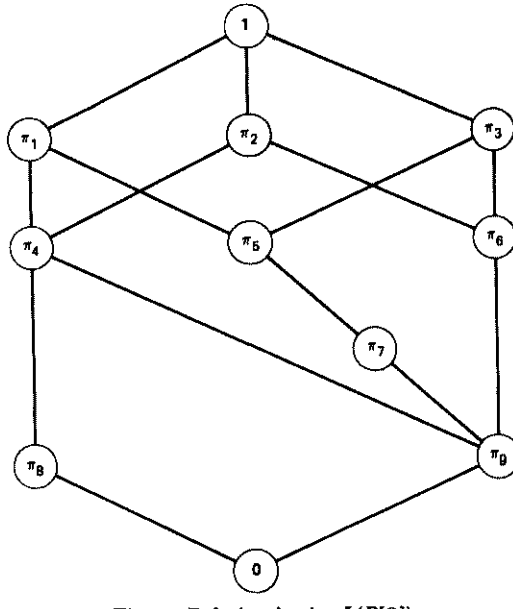



Fig. 8—Relation lattice $L(R[\Omega])$.

$$\theta(\Omega_1 + \Omega_2 | \Sigma) = \theta(A | ABC) = \theta(A)/\theta(ABC) = \{\overline{I, II, III, V}; \overline{IV}\},$$

$$\theta(\Omega_1 | \Sigma) = \theta(ABD | ABC) = \theta(AB)/\theta(ABC) = \{\overline{I, III}; \overline{II, V}; \overline{IV}\},$$

$$\theta(\Omega_2 | \Sigma) = \theta(ACE | ABC) = \theta(AC)/\theta(ABC) = \{\overline{I, V}; \overline{II, III}; \overline{IV}\},$$

and

$$\theta(ABC) = \{I, II, III, IV, V\},$$

where $I = \overline{1}$, $II = \overline{2, 5}$, $III = \overline{3}$, $IV = \overline{4}$, $V = \overline{6, 7}$.

It is easy to see that $ABD$ and $ACE$ are projectively decomposable on $ABC$, i.e.,

$$\begin{aligned}\theta(A | ABC) &= \theta(ABD | ABC) + \theta(ACE | ABC) \\ &= \theta(ABD | ABC) \circ \theta(ACE | ABC) \\ &= \theta(ACE | ABC) \circ \theta(ABD | ABC).\end{aligned}$$

But the MVD: $A \twoheadrightarrow BD$ (or $A \twoheadrightarrow CE$) does not hold in $R[\Omega]$. Nevertheless, the EMVD: $A \rightarrow B \mid C$ does hold in $R[\Omega]$. ■

So far we have discussed the properties of decomposition of two sets of attributes. The concept of decomposition certainly can be extended to any $n > 2$ sets of attributes. We define the notion of mutual decomposition as follows:

Definition 10: Let $R$ be a relation on the set of attributes $\Omega$. The sets of attributes $\Omega_1, \Omega_2, \cdots, \Omega_n \subseteq \Omega$ are mutually decomposable, if for any $I \subseteq N = \{1, \cdots, n\}$ and $J \subseteq N - I$, the two sets of attributes $\Omega_I = \cup_{i \in I} \Omega_i$ and $\Omega_J = \cup_{j \in J} \Omega_j$ are decomposable. ■

*Theorem 10: Let $R$ be a relation on the set of attributes $\Omega$ and $\Omega_1 \cdots \Omega_n = \Omega$. Suppose $\Omega_1, \cdots, \Omega_n$ are mutually decomposable. Then*

$$R[\Omega] = R[\Omega_1 \cdots \Omega_n] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n].$$

*Proof:* It follows from the definition of mutual decomposition that

$$R[\Omega_1 \cdots \Omega_m] = R[\Omega_1 \cdots \Omega_{m-1}] \mid \times \mid R[\Omega_m], \qquad m = 2, \cdots n.$$

Therefore the assertion is true by induction. ■

The above theorem states that mutual decomposition implies an information-lossless join. The converse is not true in general. The necessary and sufficient condition of an information-lossless join is called join dependency, which will be discussed in the next section.

## VII. JOIN DEPENDENCIES

*Join dependency* (JD)[10,11,14] is a generalization of MVD. It refers to a collection $\{\Omega_1, \cdots, \Omega_n\}$ of subsets of $\Omega$ such that

$$\Omega = \Omega_1 \cdots \Omega_n$$

and

$$R[\Omega] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n].$$

Join dependency can be considered as a "set of coordinates" of the relation. The connection between join dependencies and multivalued dependencies is given by the following lemma:

*Lemma 10: Let $R[\Omega] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n]$, let $N_0$ be a subset of $\{1, \cdots, n\}$, and let $N_1 = \{1, \cdots, n\} - N_0$. Then $(\Omega_{N_0}, \Omega_{N_1})$ is a decomposition pair, where*

$$\Omega = \Omega_1 \cdots \Omega_n, \qquad \Omega_{N_0} = \bigcup_{i \in N_0} \Omega_i, \qquad \text{and} \qquad \Omega_{N_1} = \bigcup_{i \in N_1} \Omega_i.$$

*Proof:* Since

$$R\left[\bigcup_{i \in N_0} \Omega_i\right] \subseteq \underset{i \in N_0}{|\times|} R[\Omega_i],$$

and

$$R\left[\bigcup_{i\in N_1}\Omega_i\right] \subseteq \bigl|\times\bigr|_{i\in N_1} R[\Omega_i],$$

it follows that

$$R[\Omega_{N_0}] \,|\times|\, R[\Omega_{N_1}] \subseteq \left(\bigl|\times\bigr|_{i\in N_0} R[\Omega_i]\right) |\times| \left(\bigl|\times\bigr|_{i\in N_1} R[\Omega_i]\right).$$

Since the natural join operation is commutative and associative,[6] we have

$$R[\Omega_{N_0}] \,|\times|\, R[\Omega_{N_1}] \subseteq R[\Omega_1] \,|\times|\, \cdots \,|\times|\, R[\Omega_n] = R(\Omega).$$

But we know

$$R[\Omega] \subseteq R[\Omega_{N_0}] \,|\times|\, R[\Omega_{N_1}].$$

Hence,

$$R[\Omega] = R[\Omega_{N_0}] \,|\times|\, R[\Omega_{N_1}]. \quad \blacksquare$$

Let $x$ be an $X$-value, and assume $Y \subseteq X$. We shall denote the $Y$-value in $x$ as $x[Y]$. Let $t \in R[\Omega]$ be a tuple and let $\Omega = \Omega_1 \cdots \Omega_n$. The notation $t \triangleq (w_1, \cdots, w_n)$ will be used to indicate that $t[\Omega_i] = w_i$, $\forall i \in N$, where $N = \{1, \cdots, n\}$ denotes the index set.

Before we state the necessary and sufficient conditions for join dependency, we first introduce the concepts of *a set of consistent values* and an *indexed family of tuples.*

Definition 11: Let $R$ be a relation on the set of attributes $\Omega$, and let $\{X_i \mid i \in N\}$ be a collection of subsets of $\Omega$. The set of values $\{x_i \mid x_i$ is an $X_i$-value, $i \in N\}$ is called *a set of consistent values* of $\{X_i \mid i \in N\}$ if the values of $X_i \cap X_j$ in $x_i$ and $x_j$ agree, i.e., if

$$x_i[X_i \cap X_j] = x_j[X_i \cap X_j], \qquad \forall i, j \in N.$$

The set of tuples $\{t_i \mid i \in N\}$ of $R[\Omega]$ is called an *indexed family of tuples* with respect to $\{X_i \mid i \in N\}$ if $\{x_i \mid t_i[X_i] = x_i, i \in N\}$ is a set of consistent values. $\blacksquare$

*Theorem 11: Let $R$ be a relation on the set of attributes $\Omega$, and let $\Omega = \Omega_1 \cdots \Omega_n$. Then*

$$R[\Omega] = R[\Omega_1] \,|\times|\, \cdots \,|\times|\, R[\Omega_n]$$

*iff for every indexed family of tuples $\{t_i \mid i \in N\}$ with respect to $\{X_i \mid i \in N\}$ there is a tuple $t \in R[\Omega]$ such that $t[\Omega_i] = t_i[\Omega_i]$, $\forall i \in N$, where $X_i = \Omega_i \cap \hat{\Omega}_i$, and $\hat{\Omega}_i = \cup_{j\neq i}\Omega_j$.*

*Proof:* (Necessity) Let $\{t_i \mid i \in N\}$ be an indexed family of tuples of $R[\Omega]$ with respect to $\{X_i \mid i \in N\}$. Thus, $\{x_i \mid t_i[X_i] = x_i, i \in N\}$ is a set of consistent values. Suppose $t_i[\Omega_i] = w_i$, $i \in N$. We want to show that

there exists a tuple $t \triangleq (w_1, \cdots, w_n) \in R[\Omega]$. We will prove this by mathematical induction. We know that $s_1 = (w_1) \in R[\Omega_1]$ and $(w_2) \in R[\Omega_2]$. Thus,

$$w_1[X_1] = x_1 \quad \text{and} \quad w_2[X_2] = x_2.$$

Since $\{x_i \mid i \in N\}$ is consistent, it follows that

$$w_1[X_1 \cap X_2] = x_1[X_1 \cap X_2] = x_2[X_1 \cap X_2] = w_2[X_1 \cap X_2].$$

It is known that

$$X_i \cap X_j = (\Omega_i \cap \hat{\Omega}_i) \cap (\Omega_j \cap \hat{\Omega}_j) = \Omega_i \cap \Omega_j, \qquad i \neq j.$$

Therefore,

$$w_1[\Omega_1 \cap \Omega_2] = w_2[\Omega_1 \cap \Omega_2].$$

By the definition of natural join, we know that there exists a tuple $s_2 \triangleq (w_1, w_2) \in R[\Omega_1] \mid\times\mid R[\Omega_2]$.

Suppose there is a tuple $s_{n-1} \triangleq (w_1, \cdots, w_{n-1}) \in R[\Omega_1] \mid\times\mid \cdots \mid\times\mid R[\Omega_{n-1}]$. Then

$$\begin{aligned} s_{n-1}[X_i \cap X_n] &= w_i[X_i \cap X_n] = x_i[X_i \cap X_n] \\ &= x_n[X_i \cap X_n] = w_n[X_i \cap X_n], \qquad i = 1, \cdots, n-1. \end{aligned}$$

Hence,

$$\begin{aligned} s_{n-1}[\Omega_n \cap \hat{\Omega}_n] &= s_{n-1}[\Omega_n \cap (\Omega_1 \cup \cdots \cup \Omega_{n-1})] \\ &= s_{n-1}[(\Omega_1 \cap \Omega_n) \cup \cdots \cup (\Omega_{n-1} \cap \Omega_n)] \\ &= s_{n-1}[(X_1 \cap X_n) \cup \cdots \cup (X_{n-1} \cap X_n)] \\ &= w_n[(X_1 \cap X_n) \cup \cdots \cup (X_{n-1} \cap X_n)] \\ &= w_n[\Omega_n \cap \hat{\Omega}_n]. \end{aligned}$$

It follows that there exists a tuple $t$ such that

$$t = s_n \triangleq (w_1, \cdots, w_n) \in R[\Omega_1] \mid\times\mid \cdots \mid\times\mid R[\Omega_n] = R[\Omega].$$

(Sufficiency) We know that

$$R[\Omega] \subseteq R[\Omega_1] \mid\times\mid \cdots \mid\times\mid R[\Omega_n].$$

For any $t \triangleq (w_1, \cdots, w_n) \in R[\Omega_1] \mid\times\mid \cdots \mid\times\mid R[\Omega_n]$, there exists an indexed family of tuples $\{t_i \mid t_i[\Omega_i] = w_i, i \in N\}$ of $R[\Omega]$ with respect to $\{X_i \mid i \in N\}$ that has a set of consistent values $\{x_i \mid w_i[X_i] = x_i, i \in N\}$. It follows that $t \triangleq (w_1, \cdots, w_n) \in R[\Omega]$. Hence,

$$R[\Omega] = R[\Omega_1] \mid\times\mid \cdots \mid\times\mid R[\Omega_n]. \quad \blacksquare$$

The necessary and sufficient conditions for JD given in the above

theorem are similar to the notion of template dependency introduced by Sadri and Ullman.[27] The following condition can be considered as an extension of the binary natural join operation.

*Corollary 3: Let $R$ be a relation on the set of attributes $\Omega$, and $\Omega = \Omega_1 \cdots \Omega_n$. Then,*

$$R[\Omega] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n]$$

*iff*

$$R_{x_1,\ldots,x_n}[\Omega] = R_{x_1}[\Omega_1] \mid \times \mid \cdots \mid \times \mid R_{x_n}[X_n]$$

*for every set of consistent values $\{x_i \mid i \in N\}$ of $\{X_i \mid i \in N\}$, where $X_i = \Omega_i \cup \hat{\Omega}_i$, $\forall i \in N$, and $R_{x_1,\ldots,x_n}[\Omega] = \{t \mid t \in R[\Omega], t[X_i] = x_i, \forall i \in N\}$.*

*Proof:* The proof follows from Theorem 11. ■

Clearly, for any $t \in R[\Omega] = R[\Omega_1 \cdots \Omega_n]$, the set of values $\{x_i \mid t[X_i] = x_i, X_i = \Omega_i \cap \hat{\Omega}_i, i \in N\}$ is always consistent. The converse is not necessarily true. Suppose for any set of consistent values $\{x_i \mid x_i$ is an $X_i$-value, $i \in N\}$ there is a tuple $t \in R[\Omega]$ such that $t[X_i] = x_i$, $\forall i \in N$; in this case we say $\{\Omega_i \mid i \in N\}$ is *complete*.

*Corollary 4: Let $R$ be a relation on the set of attributes $\Omega$, and $\Omega = \Omega_1 \cdots \Omega_n$. Then $\{\Omega_i \mid i \in N\}$ is complete iff*

$$R[X_1 \cdots X_n] = R[X_1] \mid \times \mid \cdots \mid \times \mid R[X_n],$$

*where $X_i = \Omega_i \cap \hat{\Omega}_i$, $i \in N$.*

*Proof:* The proof follows directly from Theorem 11. ■

The necessary and sufficient conditions for JD may be stated in a different form, as follows:

*Theorem 12: Let $R$ be a relation on the set of attributes $\Omega$, and $\Omega = \Omega_1 \cdots \Omega_n$. Then*

$$R[\Omega] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n]$$

*iff*

1. *$\{\Omega_i, \hat{\Omega}_i\}$ is a decomposition pair, $i \in N$,*
2. *$\{\Omega_i \mid i \in N\}$ is complete, i.e.,*

$$R[X_1 \cdots X_n] = R[X_1] \mid \times \mid \cdots \mid \times \mid R[X_n], \quad X_i = \Omega_i \cap \hat{\Omega}_i, \quad i \in N.$$

*Proof:* (Necessity) Condition 1 follows from Lemma 10. Condition 2 is a consequence of Theorem 11.

(Sufficiency) We know that

$$R[\Omega] \subseteq R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n].$$

Suppose $t \triangleq (w_1, \cdots, w_n) \in R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n]$. Then there is an indexed family of tuples $\{t_i \mid t_i[\Omega_i] = w_i, i \in N\}$ of $R[\Omega]$ with respect

to $\{X_i \mid i \in N\}$ and the set of consistent values $\{x_i \mid w_i[X_i] = x_i, X_i = \Omega_i \cap \hat{\Omega}_i, i \in N\}$. We will prove by mathematical induction that $t \triangleq (w_1, \cdots, w_n) \in R[\Omega]$.

Since $\{\Omega_i \mid i \in N\}$ is complete, there exists a tuple $s \triangleq (y_1, \cdots, y_n) \in R[\Omega]$ such that

$$s[X_i] = x_i, \qquad \forall i \in N.$$

We know that

$$t_1[\Omega_1 \cap \hat{\Omega}_1] = t_1[X_1] = w_1[X_1] = x_1 = s[X_1] = s[\Omega_1 \cap \hat{\Omega}_1],$$

which means

$$t_1 \theta(\Omega_1 \cap \hat{\Omega}_1) s.$$

Since $(\Omega_1, \hat{\Omega}_1)$ is a decomposition pair, there exists a tuple $s_1 \in R[\Omega]$ such that

$$t_1 \theta(\Omega_1) s_1 \theta(\hat{\Omega}_1) s.$$

Hence,

$$s_1 \triangleq (w_1, y_2, \cdots, y_n) \in R[\Omega].$$

Suppose there is a tuple $s_{n-1} \triangleq (w_1, \cdots, w_{n-1}, y_n) \in R[\Omega]$. It follows that

$$t_n[\Omega_n \cap \hat{\Omega}_n] = t_n[X_n] = w_n[X_n] = x_n = s_{n-1}[X_n] = s_{n-1}[\Omega_n \cap \hat{\Omega}_n].$$

Thus there is a tuple $s_n \in R[\Omega]$ such that

$$t_n \theta(\Omega_n) s_n \theta(\hat{\Omega}_n) s_{n-1}.$$

Hence

$$t = s_n \triangleq (w_1, \cdots, w_n) \in R[\Omega]. \quad \blacksquare$$

It is known that a special class of JD, called *acyclic join dependency*, has many desirable properties; this class makes operations like updates and joins especially easy.[15, 28] A collection of subsets $\{\Omega_i \mid i \in N\}$ of the set of attributes $\Omega$ is called *acyclic* if all the attributes can be deleted by repeatedly applying the following two operations:[15,28]

1. Delete from some $\Omega_i$ an attribute $A$ that appears in no other $\Omega_j$
2. Delete one $\Omega_i$ if there is an $\Omega_j$, $i \neq j$, such that $\Omega_i \subseteq \Omega_j$.

A reduction $\{Y_j \mid j \in J \subseteq N$, and $\forall i \in N - J \exists j \in J$ such that $Y_i \subseteq Y_j\}$ is obtained by removing from $\{Y_i \mid i \in N\}$ each $Y_i$ that is contained in another $Y_j$.

Definition 12: Let $\mathbf{S} = \{\Omega_i \mid i \in N\}$ be a collection of subsets of $\Omega$. The *core* of $\mathbf{S}$, denoted by $\dot{\mathbf{S}}$, is defined as follows:

1. $\dot{\mathbf{S}} = \varnothing$, for $|\mathbf{S}| = N = 1$
2. $\dot{\mathbf{S}}$ is the reduction of $\{\Omega_i \cap \hat{\Omega}_i \mid i \in N\}$, for $|\mathbf{S}| = N > 1$. $\blacksquare$

There are many different but equivalent conditions that characterize a collection of subsets as acyclic.[15] We will use the following one:

*Lemma 11: A collection* $\mathbf{S} = \{\Omega_i \mid i \in N\}$ *of subsets of* $\Omega$ *is acyclic iff its core* $\dot{\mathbf{S}}$ *is acyclic.*

*Proof:* $\dot{\mathbf{S}}$ can be obtained from $\mathbf{S}$ by performing the operations 1 and 2 defined above. It follows that if $\mathbf{S}$ is acyclic then $\dot{\mathbf{S}}$ is acyclic and vice versa. ■

*Corollary 5: Let* $\mathbf{S} = \{\Omega_i \mid i \in N\}$ *be an acyclic collection of subsets of* $\Omega$. *Then* $|\mathbf{S}| > |\dot{\mathbf{S}}|$.

*Proof:* For $|\mathbf{S}| = 1$, $|\dot{\mathbf{S}}| = |\varnothing| = 0$. For $|\mathbf{S}| \geqslant 2$, we know that $|\mathbf{S}| \geqslant |\dot{\mathbf{S}}|$. Suppose $|\mathbf{S}| = |\dot{\mathbf{S}}|$. Then any attribute $A$ in $\dot{\mathbf{S}}$ must be contained in at least two distinct subsets of $\dot{\mathbf{S}}$. Let $A \in \Omega_i \cap \hat{\Omega}_i$. Then $A \in \Omega_i$ and $A \in \hat{\Omega}_i$. There is a $j \neq i$ such that $A \in \Omega_j$. Since $\Omega_i \subseteq \hat{\Omega}_j = \cup_{k \neq j} \Omega_k$ it follows that

$$A \in \Omega_j \cap \hat{\Omega}_j \in \dot{\mathbf{S}}.$$

Since $|\dot{\mathbf{S}}| = |\mathbf{S}| \geqslant 2$, $\dot{\mathbf{S}}$ is not empty. Now, neither operation 1 nor 2 can be applied to reduce $\dot{\mathbf{S}}$. From Lemma 11 we know this contradicts the assumption that $\mathbf{S}$ is acyclic. Thus, $|\mathbf{S}| > |\dot{\mathbf{S}}|$. ■

A JD $R[\Omega] = R[\Omega_1] |\times| \cdots |\times| R[\Omega_n]$, $\Omega = \Omega_1 \cdots \Omega_n$, is an acyclic join dependency if $\{\Omega_i \mid i \in N\}$ is acyclic. A recursive condition for acyclic join dependency is as follows:

*Corollary 6: Let R be a relation on the set of attributes* $\Omega = \Omega_1 \cdots \Omega_n$. *Then*

$$R[\Omega] = R[\Omega_1] |\times| \cdots |\times| R[\Omega_n]$$

*is an acyclic join dependency iff*

1. $(\Omega_i, \hat{\Omega}_i)$ *is a decomposition pair of* $R[\Omega]$, $i = 1, \cdots, n$,
2. $R[X_1 \cdots X_m] = R[X_1] |\times| \cdots |\times| R[X_m]$ *is an acyclic join dependency over the set* $X_1 \cdots X_m \subseteq \Omega$, *where* $\{X_i \mid i = 1, \cdots, m\}$ *is the core of* $\{\Omega_i \mid i \in N\}$.

*Proof:* The join dependency of a collection of sets and the join dependency of its reduction are equivalent.[25] The proof easily follows from Theorem 12 and Lemma 11. ■

The above corollary simply states that acyclic join dependency is equivalent to a set of MVDs and EMVDs, i.e., a set of simultaneous lattice equations that can be derived recursively. It has been shown by hypergraph theory that an acyclic join dependency is equivalent to a set of MVDs.[15,16] That is, the converse of Lemma 10 is true for acyclic join dependency; we will prove that the converse of Lemma 10 is a consequence of Corollary 6.

*Theorem 10: Let R be a relation on the set of attributes* $\Omega = \Omega_1 \cdots \Omega_n$ *such that* $\{\Omega_i \mid i \in N\}$ *is acyclic. Suppose for any* $N_0 \subseteq N = \{1, \cdots, n\}$, $N_1 = N - N_0$, *and*

$$R[\Omega] = R[\Omega_{N_0}] \mid \times \mid R[\Omega_{N_1}].$$

*Then*

$$R[\Omega] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n]$$

*is an acyclic join dependency.*

*Proof:* This theorem will be proved by mathematical induction on $n$. For the smallest nontrivial case $n = 3$, let the core set $\{X_i \mid i = 1, \cdots, m\}$ of $\{\Omega_i \mid i = 1, 2, 3\}$ be the reduction of $\{Y_i = \Omega_i \cap \hat{\Omega}_i \mid i = 1, 2, 3\}$. First we want to show that

$$R[X_1 \cdots X_m] = R[X_1] \mid \times \mid \cdots \mid \times \mid R[X_m].$$

We know $m < 3$ from Corollary 5. There is nothing to be proved if $m < 2$. For $m = 2$, without loss of generality, let $X_1 = Y_1$, $X_2 = Y_2$, and $Y_3 \subseteq Y_2 = X_2$. Then

$$X_1 \cap X_2 = Y_1 \cap Y_2 = Y_1 \cap Y_2Y_3 = (Y_1 \cap Y_2) \cup (Y_1 \cap Y_3)$$

$$= (\Omega_1 \cap \Omega_2) \cup (\Omega_1 \cap \Omega_3) = \Omega_1 \cap \Omega_2\Omega_3.$$

Since $(\Omega_1, \Omega_2\Omega_3)$ is a decomposition pair,

$$\theta(X_1 \cap X_2) = \theta(\Omega_1 \cap \Omega_2\Omega_3) \subseteq \theta(\Omega_1) \circ \theta(\Omega_2\Omega_3).$$

Also we have

$$\Omega_1 \supseteq Y_1 = X_1,$$

and

$$\Omega_2\Omega_3 \supseteq Y_2Y_3 = Y_2 = X_2.$$

Thus

$$\theta(\Omega_1) \leqq \theta(X_1),$$

$$\theta(\Omega_2\Omega_3) \leqq \theta(X_2),$$

and

$$\theta(\Omega_1) \circ \theta(\Omega_2\Omega_3) \subseteq \theta(X_1) \circ \theta(X_2).$$

It follows that

$$\theta(X_1 \cap X_2) \subseteq \theta(X_1) \circ \theta(X_2).$$

Hence

$$R[X_1X_2] = R[X_1] \mid \times \mid R[X_2].$$

It follows from Corollary 6 that

$$R[\Omega] = R[\Omega_1] \mid \times \mid R[\Omega_2] \mid \times \mid R[\Omega_3].$$

Suppose the theorem is true for all $k < n$. Let the core set $\{X_i \mid i = 1,$

$\cdots, m\}$ of $\{\Omega_i \mid i \in N\}$ be the reduction of $\{Y_i = \Omega_i \cap \hat{\Omega}_i \mid i \in N\}$. We know $m < n$ and for any $M_0 \subseteq M = \{1, \cdots, m\}$ and $M_1 = M - M_0$ there is an $N_0 \subseteq N$ and $N_1 = N - N_0$ such that

$$M_0 \subseteq N_0, \qquad M_1 \subseteq N_1,$$

and

$$X_{M_0} = Y_{N_0}, \qquad X_{M_1} = Y_{N_1}.$$

Then,

$$X_{M_0} \cap X_{M_1} = Y_{N_0} \cap Y_{N_1} = \bigcup_{i \in N_0, j \in N_1} (Y_i \cap Y_j)$$

$$= \bigcup_{i \in N_0, j \in N_1} (\Omega_i \cap \Omega_j) = \Omega_{N_0} \cap \Omega_{N_1}.$$

Since $(\Omega_{N_0}, \Omega_{N_1})$ is a decomposition pair, we have

$$\theta(X_{M_0} \cap X_{M_1}) = \theta(\Omega_{N_0} \cap \Omega_{N_1}) \subseteq \theta(\Omega_{N_0}) \circ \theta(\Omega_{N_1}).$$

Also, we know

$$\Omega_{N_0} \supseteq Y_{N_0} = X_{M_0},$$

and

$$\Omega_{N_1} \supseteq Y_{N_1} = X_{M_1}.$$

Thus

$$\theta(\Omega_{N_0}) \leqq \theta(X_{M_0}),$$

$$\theta(\Omega_{N_1}) \leqq \theta(X_{M_1}),$$

and

$$\theta(\Omega_{N_0}) \circ \theta(\Omega_{N_1}) \subseteq \theta(X_{M_0}) \circ \theta(X_{M_1}).$$

It follows that

$$\theta(X_{M_0} \cap X_{M_1}) \subseteq \theta(X_{M_0}) \circ \theta(X_{M_1}).$$

Hence

$$R[X_1 \cdots X_m] = R[X_{M_0}] \mid \times \mid R[X_{M_1}]$$

for any $M_0 \subseteq M$, $M_1 = M - M_0$.

Since the theorem is true for $m < n$, we have

$$R[X_1 \cdots X_m] = R[X_1] \mid \times \mid \cdots \mid \times \mid R[X_m].$$

It follows from Corollary 6 that

$$R[\Omega_1 \cdots \Omega_n] = R[\Omega_1] \mid \times \mid \cdots \mid \times \mid R[\Omega_n]. \quad \blacksquare$$

Further discussion of the properties of acyclic join dependencies can be found in Refs. 15 and 16. A linear-time algorithm for testing acyclicity is given in Ref. 28.

## VIII. CONCLUSIONS

We have shown that lattice theory is a powerful tool in the analysis of the structure of relational database systems. Using this tool, we have established a unified theory of relations. As we have seen, almost every concept in the existing relational database theory has a counterpart in the lattice theory. This suggests that further study of relations should be carried out within the framework of lattice theory. The independency theory of lattices, which is a generalization of the familiar notion of independency in the geometries,[18,21] is especially important and relevant to the structure of relational database systems if its relation lattice is modular. This approach may lead to a geometric interpretation of data dependencies and independencies, which would make the theory more intuitive and also more useful for practical application.

The establishment of this algebraic theory of relational databases is done in the same spirit as the construction of probability theory. A probability space is a triple $(\Omega, \Sigma, P)$, where $\Omega$ is the sample space, $\Sigma$ is a $\sigma$-algebra of the subsets of $\Omega$, and $P$ is a real-valued function, called a *probability measure*, defined on the $\sigma$-algebra $\Sigma$.[17,29] The notion

Table VIII—Comparison of probability theory and the theory of relational databases

| Probability Theory | Theory of Relational Databases |
|---|---|
| Sample space $\Omega$ | Set of attributes $\Omega$ |
| $\Sigma$, the $\sigma$-Algebra of subsets of $\Omega$ | $\mathbf{2}^{\Omega}$, the Boolean algebra of subsets of $\Omega$ |
| Probability measure<br>$P\colon \Sigma \to R[0, 1]$ | Partition function<br>$\theta\colon \mathbf{2}^{\Omega} \to \Pi[R(\Omega)]$ |
| $\sigma$-additivity:<br>$\{X_k\}$ is an denumerable union of disjoint events<br>$P\left(\bigcup_{k=1}^{\infty} X_k\right) = \sum_{k=1}^{\infty} P(X_k)$ | Meet-morphism:<br>$\{X_k\}$ is a finite collection of sets of attributes<br>$\theta\left(\bigcup_{k=1}^{n} X_k\right) = \theta(X_1) \cdots \theta(X_n)$ |
| $P(\Omega) = 1$<br>$P(\varnothing) = 0$ | $\theta(\Omega) = \mathbf{0}$<br>$\theta(\varnothing) = \mathbf{1}$ |
| $0 \leqslant P(X) \leqslant 1, \forall X \in \Sigma$ | $\mathbf{0} \leqq \theta(X) \leqq \mathbf{1}, \forall X \in \mathbf{2}^{\Omega}$ |
| If $X \subseteq Y$,<br>$P(X) \leqslant P(Y)$ | If $X \supseteq Y$,<br>$\theta(X) \leqq \theta(Y)$ |
| If $\Omega_1$ and $\Omega_2$ are independent,<br>$P(\Omega_1 \cap \Omega_2) = P(\Omega_1)P(\Omega_2)$ | If $\Omega_1$ and $\Omega_2$ are decomposable,<br>$\theta(\Omega_1 \cap \Omega_2) = \theta(\Omega_1) + \theta(\Omega_2)$<br>$= \theta(\Omega_1) \cdot \theta(\Omega_2) = \theta(\Omega_2) \cdot \theta(\Omega_1)$ |

of a $\sigma$-algebra of sets also has an abstract generalizaton, namely it is a particular case of a Boolean $\sigma$-algebra.[30] A comparison of the algebraic theory of relational databases and probability theory is shown in Table VIII.

We feel that this theory of relational databases can be used to analyze the nonquantitative aspects of data dependencies (or independencies), whereas probability theory is the basis of quantitative data analysis, namely statistics. This comparison is not meant to imply that there is a one-to-one correspondence between the theory of relational databases and the theory of probability. Nevertheless, we are convinced that the lattice theory could play a role in the theory of relational databases similar to the role measure theory plays in the theory of probability.[17]

The computational algorithms for meet and join operations of partitions are given in Ref. 31, which provides the basic tools for future development of algorithms for relations.

## IX. ACKNOWLEDGMENT

I am grateful to Dr. M. Eisenberg for his careful review of the manuscript and his helpful comments and suggestions.

## REFERENCES

1. E. F. Codd, "A relational model of data for large shared data banks," Comm. ACM, *3*, No. 6 (June 1970), pp. 377–87.
2. E. F. Codd, "Further normalization of the database relational model," in *Database Systems*, R. Rustin, ed., Englewood Cliffs, NJ: Prentice Hall, 1972, pp. 33–64.
3. C. Beeri, P. A. Bernstein, and N. Goodman, "A sophisticated introduction to database normalization theory," Proc. Int. Conf. on Very Large Databases, West Berlin, Germany (September 1978), pp. 113–24.
4. P. A. Bernstein and N. Goodman, What does Boyce-Codd normal form do? Proc. 6th Int. Conf. on Very Large Databases, Montreal, Canada (1980), pp. 245–59.
5. P. A. Bernstein, "Synthesizing third normal form relations from functional dependencies," ACM Trans. Database Syst., *1*, No. 4 (December 1976), pp. 277–98.
6. J. D. Ullman, *Principles of Database Systems*, Rockville, MD: Computer Science Press, Inc., 1980.
7. R. Fagin, "Multivalued dependencies and a new normal form for relational databases," ACM Trans. Database Syst., 2, No. 3 (September 1977), pp. 262–78.
8. C. Zaniolo, "Analysis and design of relational schemata for database systems," Ph.D. Diss., Tech. Rep. UCLA-ENG-7661, U. of California, Los Angeles, CA, July, 1976.
9. A. K. Anora, and C. R. Carlson, "The information preserving properties of relational database transformations," Proc. 4th Int. Conf. on Very Large Databases, West Berlin, Germany (September 1978), pp. 352–9.
10. J. Rissanen, "Independent components of relations," ACM Trans. Database Syst., 2, No. 4 (December 1977), pp. 317–25.
11. J. Rissanen, "Theory of relations for databases—a tutorial survey, Proc. 7th Symp Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 64, J. Winkowski, ed., Berlin, Heidelberg: Springer-Verlag, 1978, pp. 537–51.
12. W. W. Armstrong, "Dependency structure of database relationships," Proc. IFIP, *74*, Amsterdam: North-Holland Publ. Co. (1979), pp. 580–3.
13. C. Beeri, R. Fagin, and J. H. Howard, "A complete axiomization for functional and multivalued dependencies in database relations," Proc. ACM SIGMOD. Int. Conf. on Management of Data, Toronto, Canada (1977), pp. 47–61.

14. A. V. Aho, C. Beeri, and J. D. Ullman, "The theory of joins in relational databases," ACM Trans. Database Syst., *4*, No. 3 (September 1979), pp. 297–314.
15. C. Beeri, R. Fagin, D. Maier, A. Mendelzon, J. D. Ullman, and M. Yannakakis, "Properties of acyclic database schemes, Proc. Thirteenth Annual ACM Symp. on Theory of Computing (Milwaukee 1981), pp. 355–62.
16. R. Fagin, A. O. Mendelzon, and J. D. Ullman, "A simplified universal relation assumption and its properties," ACM Trans. Database Syst., *7*, No. 3 (September 1982), pp. 343–360.
17. M. Loeve, *Probability Theory*, 3rd ed., Princeton, NJ: Van Nostrand, 1963.
18. G. Birkhoff, *Lattice Theory*, 3rd ed., Providence, RI: American Mathematical Society Colloquium Publ. XXV, 1967.
19. C. Delobel and R. G. Casey, "Decomposition of a database and the theory of Boolean switching functions," IBM J. Res. and Develop., *17*, No. 5 (September 1973), pp. 374–86.
20. P. M. Cohen, *Algebra*, Vol. 2, London: John Wiley, 1977.
21. A. Rosenfeld, *An Introduction to Algebraic Structures*, San Francisco, CA: Holden-Day, 1968.
22. H. C. Torng, *Switching Circuits Theory and Logic Design*, Reading, MA: Addison-Wesley Publ. Co., 1972.
23. T. W. Lin, F. W. Tompa, and T. Kameda, "An improved third normal form for relational databases," ACM Trans. Database Syst., *6*, No. 2 (June 1981), pp. 329–46.
24. C. Delobel, "Normalization and hierarchical dependencies in the relational data model," ACM Trans. Database Syst., *3*, No. 3 (September 1978), pp. 201–222.
25. C. Berri, A. O. Mendelzon, Y. Sagiv, and J. D. Ullman, "Equivalence of relational database schemes, Proc. Eleventh Annual ACM Symposium on the Theory of Computing (1979), pp. 319–29.
26. W. W. Armstrong, and C. Delobel, "Decompositions and functional dependencies in relations," ACM Trans. Database Syst., *5*, No. 4 (December 1980), pp. 404–30.
27. F. Sadri and J. D. Ullman, "Template dependency: a large class of dependencies is relational databases and its complete axiomation," J. ACM, *29*, No. 2 (April 1982), pp. 363–72.
28. R. E. Tarjan and M. Yannakakis, unpublished work.
29. A. N. Kolmogorov, *Foundation of Probability*, New York: Chelsea, 1950.
30. P. R. Halmos, *Lectures on Boolean Algebra*, New York: Springer-Verlag, 1974.
31. T. T. Lee, "Order-preserving representations of the partitions on the finite set," J. Combinatorial Theory, Series A.31, No. 2 (September 1981), pp. 136–45.

## APPENDIX A

### *Properties of Meet and Join Operations*

In any lattice $(L, \cdot, +)$, the operations of meet and join satisfy the following laws:

L1—$a \cdot a = a$, $a + a = a$; (*Idempotent*)

L2—$a \cdot b = b \cdot a$, $a + b = b + a$; (*Commutative*)

L3—$a \cdot (b \cdot c) = (a \cdot b) \cdot c$,
$a + (b + c) = (a + b) + c$; (*Associative*)

L4—$a \cdot (a + b) = a + (a \cdot b) = a$; (*Absorption*)

L5—$a \leqq b$ iff $a \cdot b = a$,
$a \leqq b$ iff $a + b = b$; (*Consistency*)

L6—$b \leqq c$ implies $a \cdot b \leqq a \cdot c$
$b \leqq c$ implies $a + b \leqq a + c$; (*Isotone*)

L7—$a \cdot (b + c) \geqq (a \cdot b) + (a \cdot c)$
$a + (b \cdot c) \leqq (a + b) \cdot (a + c)$; (*Distributive Inequalities*)

L8—$a \leqq c$ implies $a + (b \cdot c) \leqq (a + b) \cdot c$. (*Modular Inequality*)

A lattice is called *distributive* if equality holds in L7 and is called

*modular* if equality holds in L8. A Boolean algebra is a lattice $(L, \cdot, +, \bar{\ })$ with the following additional properties:[30]

L9—$a\cdot(b+c) = (a\cdot b) + (a\cdot c)$,
$a + (b\cdot c) = (a + b)\cdot(a + c)$; (*Distributive Identities*)

L10—$a \leqq c$ implies $a + (b\cdot c) = (a + b)\cdot c$; (*Modular Identity*)

L11—$L$ contains universal bounds $\mathbf{0}$, $\mathbf{1}$, which satisfy
$\mathbf{0}\cdot a = \mathbf{0}$, $\mathbf{0} + a = a$,
$\mathbf{1}\cdot a = a$, $\mathbf{1} + a = \mathbf{1}$;

L12—$\forall a \in L$, $\exists \bar{a} \in L$ such that
$a\cdot\bar{a} = \mathbf{0}$, $a + \bar{a} = \mathbf{1}$, $\bar{\bar{a}} = a$,
$\overline{(a\cdot b)} = \bar{a} + \bar{b}$, $\overline{(a + b)} = \bar{a}\cdot\bar{b}$.

## APPENDIX B

### *The Proofs of Axioms for Functional and Multivalued Dependencies*

The first three of the following are Armstrong's axioms for functional dependencies:[12]

B1. (Reflexivity for functional dependencies)
If $Y \subseteq X \subseteq \Omega$, then $X \rightarrow Y$.

*Proof:* $\theta(X) = \theta(Y(X - Y)) = \theta(Y)\theta(X - Y) \leqq \theta(Y)$. ■

B2. (Augmentation for functional dependencies)
If $X \rightarrow Y$ and $Z \subseteq \Omega$, then $XZ \rightarrow YZ$.

*Proof:* $\theta(XZ) = \theta(X)\theta(Z) \leqq \theta(Y)\theta(Z) = \theta(YZ)$. ■

B3. (Transitivity for functional dependencies)
If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

*Proof:* $\theta(X) \leqq \theta(Y)$ and $\theta(Y) \leqq \theta(Z)$ imply $\theta(X) \leqq \theta(Z)$. ■

The next three axioms apply to multivalued dependencies:[13]

B4. (Complementation for multivalued dependencies)
If $X \twoheadrightarrow Y$ then $X \twoheadrightarrow \Omega - X - Y$.

*Proof:* $\theta(X) = \theta(XY) + \theta(XZ) = \theta(XY) \circ \theta(XZ) = \theta(XZ) \circ \theta(XY)$, where $Z = \Omega - X - Y$. ■

B5. (Augmentation for multivalued dependencies)
If $X \twoheadrightarrow Y$, and $V \subseteq W$, then $WX \twoheadrightarrow VY$.

*Proof:* Without loss of generality,* we can let $\Omega = ABCDEFGHIJKL$, $X = ABCDEF$, $Y = BCGHFI$, $W = CDEFHIJK$, $V = EFIJ$ (see Fig. 9). Then $\Omega - X - Y = JKL$ and $\Omega - WX - VY = L$.

We want to show that

$$\theta(ABCDEF) \subseteq \theta(ABCDEFGHI) \circ \theta(ABCDEFJKL)$$

* This proof is carried out in terms of equivalence relations (partitions). It is irrelevant here whether an equivalence relation is the image of a single attribute or the image of a set of attributes.

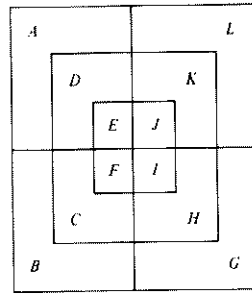



Fig. 9—Set of attributes Ω for B5.

implies

$$\theta(ABCDEFHIJK) \subseteq \theta(ABCDEFGHIJK) \circ \theta(ABCDEFHIJKL).$$

Suppose

$$t_1\theta(ABCDEFHIJK)t_2. \tag{1}$$

Then

$$t_1\theta(ABCDEF)t_2.$$

There exists $t_3$, such that

$$t_1\theta(ABCDEFGHI)t_3\theta(ABCDEFJKL)t_2. \tag{2}$$

From (1) and (2), we have

$$t_3\theta(JK)t_2\theta(JK)t_1.$$

It follows from (2) that

$$t_1\theta(JKG)t_3. \tag{3}$$

Combining (2) and (3), we have

$$t_1\theta(ABCDEFGHIJK)t_3. \tag{4}$$

Relation (2) also implies that

$$t_1\theta(HI)t_3.$$

From (1), we know

$$t_1\theta(HI)t_2,$$

and therefore

$$t_3\theta(HI)t_2. \tag{5}$$

It follows from (2) and (5) that

$$t_3\theta(ABCDEFHIJKL)t_2. \tag{6}$$

Combining (4) and (6), we have

$$t_1\theta(ABCDEFGHIJK)t_3\theta(ABCDEFHIJKL)t_2.$$

It follows that

$$\theta(ABCDEFHIJK) \subseteq \theta(ABCDEFGHIJK) \circ \theta(ABCDEFHIJKL). \quad \blacksquare$$

B6. (Transitivity for multivalued dependencies)

If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow Z - Y$.

*Proof:* Again, without loss of generality, we can let $\Omega = ABCDEFGH$, $X = AFGH$, $Y = BCFG$, $Z = CDGH$ (see Fig. 10). Then $Z - Y = DH$, $\Omega - XY = DE$, $\Omega - YZ = AE$, $\Omega - X(Z - Y) = BCE$.
We want to show that

$$\theta(AFGH) \subseteq \theta(ADEFGH) \circ \theta(ABCFGH)$$

and

$$\theta(BCFG) \subseteq \theta(ABCEFG) \circ \theta(BCDFGH)$$

imply

$$\theta(AFGH) \subseteq \theta(ADFGH) \circ \theta(ABCEFGH).$$

Suppose $t_1\theta(AFGH)t_2$. Then there exists $t_3$ such that

$$t_1\theta(ADEFGH)t_3\theta(ABCFGH)t_2. \tag{7}$$

Since $t_2\theta(BCFG)t_3$, there exists $t_4$ such that

$$t_2\theta(ABCEFG)t_4\theta(BCDFGH)t_3. \tag{8}$$

It follows that

$$t_1\theta(AFG)t_3\theta(AFG)t_2\theta(AFG)t_4. \tag{9}$$

From (7) and (8), we have

$$t_1\theta(DH)t_3\theta(DH)t_4. \tag{10}$$

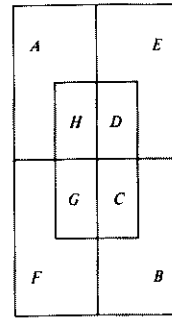



Fig. 10—Set of attributes Ω for B6.

Combining (9) and (10) yields

$$t_1\theta(ADFGH)t_4. \tag{11}$$

From (7) and (8), we have

$$t_4\theta(H)t_3\theta(H)t_2.$$

It follows from (8) that

$$t_4\theta(ABCEFGH)t_2. \tag{12}$$

Relations (11) and (12) yield

$$t_1\theta(ADFGH)t_4\theta(ABCEFGH)t_2.$$

Hence

$$\theta(AFGH) \subseteq \theta(ADFGH) \circ \theta(ABCEFGH). \quad \blacksquare$$

The last two axioms relate functional and multivalued dependencies.

B7. If $X \rightarrow Y$ then $X \twoheadrightarrow Y$.

*Proof:* Let $Z = \Omega - XY$. We want to show that

$$\theta(X) \leqq \theta(Y) \quad \text{implies} \quad \theta(X) \subseteq \theta(XY) \circ (XZ).$$

Suppose $t_1\theta(X)t_2$. Since $\theta(X) \leqq \theta(Y)$ implies $\theta(XY) = \theta(X)$, then $t_1\theta(XY)t_2$. It follows that

$$t_1\theta(XY)t_2\theta(XZ)t_2.$$

Hence

$$\theta(X) \subseteq \theta(XY) \circ \theta(XZ). \quad \blacksquare$$

B8. If $X \twoheadrightarrow Y$, $Z \subseteq Y$, and for some $W$ disjoint from $Y$, we have $W \rightarrow Z$, then $X \rightarrow Z$.

*Proof:* Again, without loss of generality, we can let $\Omega = ABCDEFGH$,

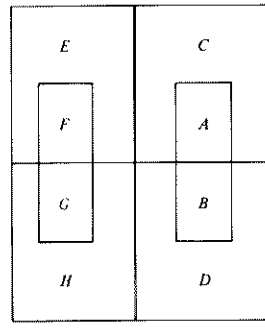



Fig. 11—Set of attributes $\Omega$ for B8.

$X = ACEF$, $Y = EFGH$, $Z = FG$, and $W = AB$ (see Fig. 11). Then $\Omega - XY = BD$.

We want to show that

$$\theta(ACEF) \subseteq \theta(ACEFGH) \circ \theta(ABCDEF)$$

and

$$\theta(AB) \leqq \theta(FG)$$

imply

$$\theta(ACEF) \leqq \theta(FG).$$

Suppose $t_1\theta(ACEF)T_2$. Then there exists $t_3$ such that

$$t_1\theta(ACEFGH)t_3\theta(ABCDEF)t_2.$$

Since

$$t_1\theta(FG)t_3 \quad \text{and} \quad t_3\theta(AB)t_2,$$

we have

$$t_1\theta(FG)t_3 \quad \text{and} \quad t_3\theta(FG)t_2,$$

and thus

$$t_1\theta(FG)t_2.$$

Hence

$$\theta(ACEF) \leqq \theta(FG). \quad \blacksquare$$

## AUTHOR

**Tony T. Lee,** B.S. (Electrical Engineering), 1971, National Cheng Kung University; M.S. (Mathematics), 1973, Cleveland State University, M.S., 1976, Ph.D. (Electrical Engineering), 1977, Polytechnic Institute of New York; Bell Laboratories, 1977—. At Bell Laboratories, Mr. Lee has worked on Common Control Interoffice Signaling network planning, teletraffic usage forecasting, 5*ESS*™ message switch performance analysis, and queueing modeling of computer and communication systems. He is currently working in the Teletraffic Theory and Application Department at Bell Laboratories. Member, Sigma Xi.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Generation of Syntax-Directed Editors With Text-Oriented Features

By B. A. BOTTOS* and C. M. R. KINTALA†

(Manuscript received March 16, 1983)

Often, syntax-directed editors rely solely on menu selection for program construction. We describe here the generation of *hybrid editors* that give a programmer the option of either (1) using menu selection and tree navigation as in a syntax-directed editor, or (2) entering text for parsing and navigating through the text as in a conventional editor at *any* stage during the expansion of a program. A prototype system, HEG (Hybrid Editor Generator), has been built to automatically generate such a hybrid editor from a high-level specification of a grammar for an application language. Each such generated hybrid editor is called an AGE (Automatically Generated Editor). We describe the HEG meta-language and briefly summarize the editing process in AGEs. We also describe possible extensions to the meta-language to describe program semantics, and the generation of the procedures to check those semantics during program construction.

## I. INTRODUCTION

In the past, there has been a dichotomy between the way a development tool such as a text editor would manipulate the text of a computer program and the way another development tool such as a parser would manipulate the same text. The advent of syntax-directed editing has removed this difference by introducing the use of editors that store and manipulate programs entirely as (partially) instantiated syntax trees.[1-8] The question, though, of whether a program should be manipulated *only* in terms of its syntax tree, or also in terms of its

*Carnegie-Mellon University. †Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

textual representation, is yet to be resolved.[8,9] Some editing systems purport to allow the programmer both points of view, but provide only one, predetermined, choice for the manipulation of each construct in the language.[7] We introduce here the concept of a *hybrid editor*, which integrates the tree-navigation and menu-selection capabilities of a syntax-directed editor with the text-navigation and the string-entry capabilities of a conventional editor. Either of these views of a program can be taken at *any* stage during the expansion of the program in a hybrid editor.

One of the features of the hybrid editors discussed herein is the fact that a complete editing system is automatically generated from a high-level description of a language. The concrete and abstract syntaxes of the language are both described by one context-free grammar; and both the syntax-directed editor, and the incremental parser for use with the editor, are generated from this grammar.

In this paper, we outline the overall design of a hybrid editor generator and describe briefly the interface presented to a programmer by the hybrid editor. We define the meta-language in which the application language is specified for the generator, and describe the operations performed on the language specification in the process of generating the editor. Based on this design, a prototype hybrid editor generator, called HEG, has been built and is being tested. We describe possible extensions to the meta-language to describe program semantics, and the generation of the procedures to check those semantics during program construction.

## II. BACKGROUND

Although syntax-directed editing can be very helpful to a programmer who is unfamiliar with the language he is using, one who knows the syntax of the language may easily become frustrated with the plethora of menu choices that must be made to "write" a program substructure as simple as an assignment statement. For this reason, allowing the programmer the option of giving the editor a string to parse and insert into the partially expanded program is a desirable feature to have in such a menu-driven editor. We define a hybrid editor to be a syntax-directed editor with the ability to parse and to integrate into the program tree a string given at any time during the expansion of a program.

There have been attempts to generate syntax-directed editors automatically for several programming languages from specifications of the syntactic (given by a context-free grammar)[5] and the semantic (given using attributes for the symbols in the grammar) aspects of a language.[10] There is also a system in use that requires only the addition of a context-free grammar for any new language for which it is to

operate.[4] (In this latter system, detailed knowledge of the structure of the grammar seems to be required in order to use the editor with any grace.) Usually, however, such editors are built individually for each programming language. This is because (1) high-level programming languages have many context-sensitive semantic constraints that cannot be expressed by a context-free grammar, (2) programmers using high-level languages usually need more local context-sensitive aid than syntactic help, and (3) the external interface that such an editor is expected to provide depends heavily upon the language it supports.

Little attention, however, has been paid to syntax-directed editors for special-purpose languages such as the input specification languages for Yet Another Compiler-Compiler (YACC)[11] and database interface languages such as HISEL.[12] Yet it is for these seldom used, but numerous, languages that a hybrid editor would be most useful. Users of such application programs often write their input in a file using a regular text editor and manually check conformity with the syntactic constraints imposed by the particular application program. Some of these application programs have parsers in their front ends to check the syntactic correctness of their input. (Sometimes these parsers are automatically generated from utilities such as YACC.[11]) Others just assume that their input is syntactically correct and abort when it is not. If, in place of the parser, a hybrid editor is available within the application program, the user can be guided by the program's editor during input construction.

A tool to automatically generate such editors for application languages can be quite useful for several reasons: (1) A special-purpose application language, unlike a programming language, is likely to change more rapidly with an evolving application program. (2) Many application programs are not frequently used and hence their idiosyncratic syntactic constraints are likely to be forgotten. (3) The amount of syntactic help from a hybrid editor can be determined by the programmer. (4) The derivation tree built by the hybrid editor may be used by the application program to process its input in a more structured manner than is often possible when only a textual view of the input is available to the program. It is for these types of languages that HEG-generated hybrid editors, called AGEs (Automatically Generated Editors), are most valuable.

## III. THE EDITING PROCESS

In a HEG-generated hybrid editor for a given language, a program is internally represented by a derivation tree of the program in that language, along with a symbol table containing the programmer-defined character strings. (See Appendix A for an example of a short session with an AGE.) When the programmer wishes to expand a

particular nonterminal in the tree by syntax-directed menu selection, and if there are two or more production rules for that nonterminal, then the AGE creates a one-item-at-a-time wraparound menu, on the bottom line of the screen, displaying the menu strings associated with those production rules. After a production rule is chosen for expansion, the internal node for the nonterminal is grown in such a way that the frontier of the subtree rooted at that internal node corresponds to the right-hand side of the production rule selected. If, however, the programmer wishes to forego menu selection, he/she may expand a nonterminal by entering a string that is derivable from the nonterminal. The editor will parse the string, build a subtree representing the string, and graft the subtree into the program tree at the node corresponding to the nonterminal. The string given by the programmer may be any combination of terminal strings and nonterminal names.

Even though the internal representation of a program is a tree, both tree and text interfaces are provided to the programmer for navigation through the program. At any moment, a tree cursor navigating around the internal nodes of the tree is available. The programmer sees the cursor spanning the entire frontier of the subtree rooted at the tree-cursor node. He can move the tree cursor along the internal nodes of the tree (using the commands up [^], down [V], left-sibling under the same parent [<], right-sibling under the same parent [>], next internal node of the same type under the same parent [N], etc). Or, he can navigate textually (using the commands n for next word, b for back word, CR for next line, - for previous line) through the program. The commands 'e' and 'u' provide syntax-driven expansion and unexpansion facilities. Commands 'p' and 'r' allow parser-driven nonterminal expansion and file-reading facilities.

When the programmer quits editing, both the text and the tree versions of the program are saved. The tree is saved by storing the leftmost derivation sequence of the production rules in the derivation tree and the symbol table. This sequence is used to reconstruct the tree for a later editing session on that program. The test is saved for possible use by other tools.

## IV. ARCHITECTURE OF HEG

HEG produces a generalized table-driven syntax-directed editor, linked with a parser, for a given application language. The application language is specified by an AGE grammar in the meta-language described in the next section. A parser generator front end, named 'GENPAR', generates a YACC file and a LEX file[13] from this specification of a language. The parser generated from these two files is capable of parsing strings, which may contain nonterminal names,

starting from any nonterminal of the grammar. The grammar tables used by the editor, the parser, and the generic editing routines are then linked together to generate an AGE for the given language.

Figure 1 shows the interactions between the different components of an AGE.

## V. THE META-LANGUAGE

The "meta-language" for HEG is the grammar specification language in which the production rules of any context-free grammar are specified, along with the pretty-printing information required by the display utilities of the editor (e.g., indentation and color codes, if available) and the strings for the menu lists. As an example of the use of the language,

```
query       : ⟨QUERY⟩
              | "select" list "where" conds
              ;
```

is a normal production rule. Here, ⟨QUERY⟩ is a "user-friendly" name for the nonterminal symbol 'query'. The sequence of strings following the character '|' specifies a production for the nonterminal 'query'. Strings within double quotes in the production specification are terminal characters and others are nonterminals. The terminal character strings may also contain the following special characters denoting pretty-printing information: '\n' for a new-line, '~' for a blank character, '\t' for tabbing one position to the right and increasing the tab count by one for the starting positions of the subsequent lines, and '\T' for decreasing the tab count by one for the current and the subsequent lines.

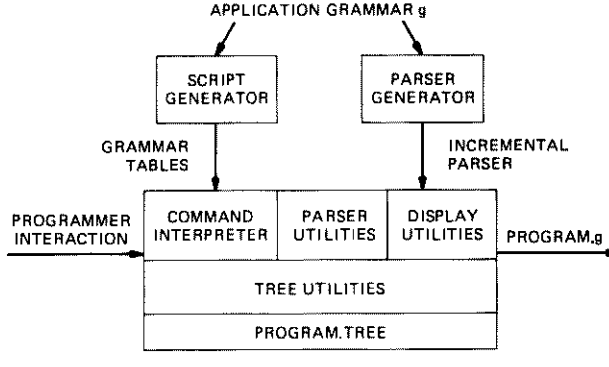



Fig. 1—Interactions between the components of an AGE.

Another example is

```
list        : ITEM_LIST
            | +item ","
            ;
```

This is an iterative production rule. It specifies that the nonterminal, 'list', can be expanded into one or more occurrences of the nonterminal, 'item', separated by the terminal character string ','. A '*', in place of the '+' above, would denote zero or more occurrences of the nonterminal.

There may be more than one production rule for a nonterminal; if so, all of the rules for the nonterminal must be specified in one rule set, each preceded by a '|'. For example, the following two sets of rules show several possible expansions for the corresponding nonterminals:

```
clause      : CLAUSE
            | field_name "~" op "~" constant
            | constant "~" op "~" field_name
            | field_name "~" op "~" field_name
            ;
op          : OPERATOR
            | "="
            | "!="
            | ">="
            | "<="
            | ">"
            | "<"
            ;
```

For each production rule, the grammar specification must include an associated "expansion character". This expansion character must be unique within the set of rules for that nonterminal. If there are two or more rules for the expansion of a nonterminal and if an instance of that nonterminal in the program is to be expanded, the programmer must indicate his choice by typing the expansion character associated with the desired rule. However, if the programmer wishes to examine all the choices for that nonterminal, the AGE creates a one-item-at-a-time wraparound menu on the bottom line of the screen, from which the programmer may choose a production. Therefore, the implementor (the person defining the grammar) must provide a "menu-item-string", which is displayed in the menu for each production rule. These menu items should be suggestive of the associated production rule for better communication with the programmer. When a particular rule is chosen for the expansion, the right-hand side of that rule replaces the left-hand side nonterminal node in the program's syntax tree.

To illustrate the provision of menu information in a grammar for HEG as above and to clarify the grammar specifications further, a complete set of rules for a database query language HISEL[12] is given below. Sentences (queries) in this language have the form:

select x1.fld1,x2.fld2, ··· where ⟨CLAUSES⟩ or ⟨CLAUSES⟩ ···

where 'x' is a cursor name, 'fld' is a field name, and ⟨CLAUSES⟩ is a conjunctive sequence of field conditions. The single characters following the menu items in the rules are the expansion characters. For the rule sets having just one production rule, meaningless menu-item strings (e.g., 'xxx') and expansion characters (e.g., 'x') are used for uniformity in the meta-syntax. Observe that there is no white space in the menu-item strings.

```
query      : ⟨QUERY⟩
           | xxx x "select" list "\nwhere~" conds
           ;
list       : ⟨ITEM_LIST⟩
           | xxx x + item ","
           ;
item       : ITEM
           | xxx x cursor "." field_name
           ;
conds      : ⟨CONDITIONS⟩
           | xxx x *disjunct "\n~~~or~"
           ;
disjunct   : ⟨CLAUSES⟩
           | xxx x +clause "~,~"
           ;
clause     : CLAUSE
           | field_op_const 1 field_name "~" op "~" constant
           | const_op_field 2 constant "~" op "~" field_name
           | field_op_field 3 field_name "~" op "~" field_name
           ;
op         : OPERATOR
           | equal = "="
           | not_equal ! "!="
           | greater_equal 1 ">="
           | less_equal 2 "<="
           | greater > ">"
           | less < "<"
           ;
```

The left-hand side nonterminal of the first rule in the grammar specification (i.e., 'query', above) is the starting nonterminal. If no

production rule is available for a nonterminal (e.g., 'field_name'), and that nonterminal appears during the derivation of a program, then it is assumed to expand into a character string to be provided by the programmer at the time of expansion. Such nonterminals are said to derive "identifiers". A regular expression specification for any such nonterminal can be used to restrict the format of the identifier strings that a programmer may supply at the time of expansion. HEG provides a default specification for all nonterminals deriving identifiers and having no regular expression specification.

As another example, the following is a grammar in the meta-language (meta-grammar) for the meta-language described in this section. In fact, one can invoke an AGE for this meta-language to enter a grammar specification for any user-defined language.

```
gram        : ⟨AN_AGE_GRAMMAR⟩
            | xxx x cfrules "\n%%\n" rerules
            ;
cfrules     : ⟨CONTEXT_FREE_RULES⟩
            | xxx x +ruleset "\n"
            ;
ruleset     : ⟨A_RULE_SET⟩
            | xxx x nonterm-name "\t:~" user_name "\n" rules
              "\n;\n\T"
            ;
rules       : ⟨RHS_OF_A_RULE_SET⟩
            | xxx x +rule "\n"
            ;
rule        : ⟨A_RULE⟩
            | normal-rule n "|~" menu_string "~" exp_char "~"
              tklist
            | nonempty-rec-rule + "|~xxx~x~+" nonterm_name
            "~\"" separator "\""
            | empty-rec-rule * "|~xxx~x~*" nonterm_name "~\""
              separator "\""
            ;
tklist      : ⟨A_SEQ_OF_TOKENS⟩
            | xxx x +token "~"
            ;
token       : TOKEN
            | nonterminal-token n nonterm_name
            | terminal-token t "\"" terminal_str "\""
            ;
rerules     : ⟨REGULAR_EXP_RULES⟩
            | xxx x *rerule "\n"
            ;
```

```
rerule      : ⟨REG_EXP_RULE⟩
            | xxx x nonterm-name "~~\~##!" reg_expr
            ;
%%
exp_char    ~##![a-zA-Z0-9+* ^<>&]
```

## VI. THE PARSER GENERATOR FOR HEG

The tools YACC and LEX are used to produce a parser and a scanner for an AGE system. The parser generator front end, GENPAR, takes, as input, the AGE grammar specification for the desired language and produces YACC and LEX specification files. (Note that the use of YACC implies that the input grammar must be LALR(1) in order to generate a parser for an AGE system.)

The generated parser can be invoked to parse an input string representing the expansion of any nonterminal node in the program derivation tree. The string to be parsed can be any combination of terminal strings and nonterminal names that is derivable from the nonterminal at the "current" position of the editing cursor.

### *6.1 The parsing of an input string*

When the programmer provides a string to be parsed, rather than making a menu selection, the AGE system opens a temporary file, writes the prefix *$$NONTERMINAL_NAME$$* to the file (where *NONTERMINAL_NAME* represents the nonterminal from which the programmer's input string is to be derived), and appends the input string. The parser is called to process the entire string contained in the file. The prefix is considered to be an integral part of the input string. If the portion of the string entered by the programmer cannot be derived from the nonterminal indicated by the prefix, the input will be considered syntactically incorrect. If a syntax error is found, the temporary file is saved in case the programmer should wish to edit the string and resubmit it to the parser.

The parser will parse its input in a "backtracking bottom-up" fashion, constructing a syntax tree to represent the derivation of the input string. After the tree is constructed, a preorder traversal of the tree is performed, producing a list of production numbers representing the leftmost derivation of the input string. This list is then passed back to the controlling program in AGE.

In an AGE, the YACC-generated parser is run subordinate to a "parser monitor" to provide a certain amount of parser backtracking, made necessary by the conflict resolution scheme of the LEX-generated scanner. For such a scanner, the lexical tokens desired are specified by regular expressions. When two or more of the given regular expressions match equal-length segments of the input (starting at the

"current input position" of the scanner), and these are the longest matches possible, the scanner will select the regular expression that had been listed first (textually) in the input specification file as the "correct" match, and return the corresponding token number. If the parser is expecting one of the other possible matches at the current input position, the parser will find a "syntax error" where no error may, in fact, exist.

The parsers produced by the YACC program are standard shift/reduce ("bottom-up") parsers. The required backtracking is accomplished through the interaction of code at two levels of the parsing scheme. At the lowest level, the handling of multiple matches in the scanner is slightly modified by forcing the scanner to "reject" an "identifier" match, after linking the token number associated with the match into a "token map." In this manner, all possible matches for an "identifier" are linked into the token map for the parse. At the highest level, the parser monitor runs the shift/reduce parser, handing the parser token numbers from either the scanner or the token map, depending upon the current state of the parse. Then, if the YACC-generated portion of the parser discovers a syntax error in the input, the monitor can rotate the last set of entries in the token map to (temporarily) "forget" the "preferred" token number for the last "choice position" in the input stream, allowing the use of another possible token number for that position.* In this manner, no syntactically correct input will be declared to contain a syntax error, and an incorrect input will only be rejected after trying the allowable combinations of token numbers for the "identifier" positions.

### *6.2 What the parser generator does*

Several transformations must be applied to the AGE grammar to produce a grammar specification that is acceptable to YACC, and that will specify a parser offering the features and enforcing the constraints desired. All the transformations are performed and the YACC grammar specification is produced in a single pass over the input grammar.

#### *6.2.1 Starting the derivations from arbitrary nonterminals*

For every nonterminal in the AGE grammar, two additional productions are generated. One such set of productions makes it possible to

---

*Due to the relative simplicity of most application languages (in terms of permissible "identifier" combinations and possible token map complexity), the erroneous "identifier" token number will usually be in the last set of entries in the token map. So, in most cases, only the last one or two "identifiers" need to be retried (if the input really is syntactically correct). In all cases, the LALR(1) property of the grammar should aid in the isolation of groups of points in the "identifier" cross-product space that could not possibly be characteristic of a correct parse. The points in these groups, then, need not be considered individually during parsing.

start a derivation from any nonterminal in the original grammar, even though YACC will allow the specification of only one start symbol for a grammar. Each such added production is of the form:

ppstart : AGE_TERM_20 clause

where "ppstart" is a newly defined start symbol for the grammar, "clause" is a nonterminal in the original grammar, and "AGE_TERM_20" is the token returned by the scanner when it reads the prefix (e.g., "$$clause$$") encoding the nonterminal from which the remainder of the input is to be derived. This type of production also allows the enforcement of the rule that the input string must be derivable from the nonterminal at the "current" node in the program tree.

#### *6.2.2 Use of nonterminal names*

The second set of productions generated for each nonterminal allows the acceptance of a "user-name" for a nonterminal, in place of a string that could be derived from that nonterminal, wherever the nonterminal may appear in an input sentence. Each of these productions is of the form:

clause : AGE_TERM_22

where "AGE_TERM_22" is the token returned by the scanner when it reads the user-name for the nonterminal on the left-hand side of the production (e.g., "clause").

#### *6.2.3 Iteration in the grammars*

The iterative specifications in the AGE grammar must be transformed into explicit left recursions, with the separators appropriately treated. Since the most general form of iteration in AGE grammars is that of a list item repeated zero or more times, with a nonwhite-space separator, that case is considered here. The AGE grammar specification for such a list might be:

clauses : *clause ","

where "clause" is the nonterminal to be repeated in the list structure and "," is the terminal separator to appear between list elements.

The YACC specification corresponding to the above production would be:

clauses : Xclause_2_0X

where Xclause_2_0X is a new nonterminal encoding the nonterminal to be repeated as the list elements (e.g. "clause"), the number of the terminal string to be used as the list element separator (e.g., "2"), and

the minimum number of times the nonterminal must appear in the list (e.g., "0"). Such encoding permits the use of the list item nonterminal, "clause," as the repeated element in other lists with different separators and/or different minimal numbers of occurrences.

To derive the required number of occurrences of "clause," sets of productions of the following form must be added to the YACC grammar:

```
Xclause_2_0X  : e
              | Xclause_2_1X
```

where "e" represents the empty string. The second new nonterminal, Xclause_2_1X, denotes one or more occurrences of "clause" separated by occurrences of terminal number 2. For this second new nonterminal, sets of productions of the following form are added to the YACC grammar:

```
Xclause_2_1X  : clause
              | Xclause_2_1X AGE_TERM_2 clause
```

where "AGE_TERM_2" denotes the required separator ("terminal number 2") between elements of the list.

Note that other iterative specifications are specializations of the above case. If the list must be nonempty, only the second new nonterminal, with its associated productions, is generated. If the list element separator is not significant (i.e., if it is any form of white space), then no terminal is encoded in the new nonterminal(s) or included in any of the new productions.

#### *6.2.4 Treatment of identifiers*

In an AGE input grammar, there are no explicit productions for the derivation of character strings representing identifiers. Any nonterminal that does not appear on the left-hand side of any production, in an AGE grammar specification, may produce an identifier. Since there are no such implied rules in a YACC grammar specification, GENPAR must add explicit productions to the grammar to allow the reduction of terminal identifier strings to appropriate nonterminals. These productions are of the form:

```
cursr : AGE_IDENTIFIER_5
```

where an identifier number (e.g., "5") is specified only if the default lexical specification is overridden for the nonterminal on the left-hand side of the production (see below).

To handle the lexical specification for each terminal identifier, the grammar designer has two choices. The designer may use GENPAR's default specification, which allows identifiers to be any combination

of letters, digits, and the characters "-", "_", and ".", which begins with a letter, and is not a reserved terminal string in the language. He may, instead, associate an arbitrary LEX specification with any nonterminal that may derive an identifier. The given specification would then be used to override the default identifier specification for that particular nonterminal.

#### *6.2.5 Treatment of white space*

In the generation process, all types of white space in the AGE grammar specifications of terminal strings are treated identically, and are considered insignificant to the specification of the generated grammar. For example, if a "PROGRAM" is specified as a "list of statements separated by new-lines", the YACC-generated parser will accept any "list of statements separated by any white space (e.g., blanks, tabs, or new-lines)" as a "PROGRAM".

### *6.3 Generation of YACC actions*

If the input string to the parser is valid, AGE requires the output of the parser to be a list of rule numbers, symbol table indices, and list item occurrence counts representing the leftmost derivation of the input string. Since the YACC-generated parser is a shift/reduce parser, the order in which that parser uses the grammar productions will not represent a leftmost derivation of the input. To produce the proper input for the AGE monitor, the parser builds a tree to represent the derivation of its input string as it parses the input string, and, as part of the last production applied (reduction to the start symbol), performs the preorder traversal of the tree, generating the required list of numbers.

The building of this tree is the main activity of the "actions" generated for each production in the YACC specification. To generate code to appropriately link nonterminal sub-trees, the positions of all the nonterminals in a given production rule must be saved as the production is processed by the generator. This is accomplished by counting the tokens on the right-hand side of the given production rule and stacking the position numbers that correspond to nonterminals. Actions are then generated that call precoded subroutines that link the tree nodes together. These subroutines, and their associated data and type declarations, are constant across all grammars, and are included in the appropriate sections of each generated YACC grammar.

### *6.4 The size of the processed grammar*

The growth of the grammar in the generation process is actually not as large as may be imagined. If $n$ = the total number of nonterminals and $r$ = the number of iterative nonterminals in the original

AGE grammar, at most $3n + 3r < 6n$ productions, $2n - 1$ nonterminals, and betweeen $2n + 1$ and $3n - r$ terminals are added to the grammar before YACC generates the parser.

## VII. ATTRIBUTES AND SEMANTIC PROCESSING

One possible extension to this work would involve the addition of attributes to the grammars described above. These attributes could allow a certain amount of semantic checking of user programs, in addition to allowing interpretation and/or code generation (for simple application languages) during program construction. The basic principles behind attribute grammars are described elsewhere;[14] we shall only discuss the required extensions to our meta-language and the generation of evaluation functions for the attributes.

### *7.1 Extensions to the meta-language*

The meta-language described in Section V can be enhanced to include attributes and their evaluation routines in each production. An example of the use of the enhanced meta-language is shown below.

```
clause      : CLAUSE
              inherited: int temp_loc;;
              synthesized: int next_temp;
            | field_op_const 1 field_name "~" op "~" constant
              {
              $0.next_temp = $0.temp_loc;
              }
            | const_op_field 2 constant "~" op "~" field_name
              {
              $0.next_temp = $0.temp_loc;
              }
            | field_op_field 3 field_name "~" op "~" field_name
              {
              $0.next_temp = $0.temp_loc + 2;
              }
            ;
```

In general, the attributes and their evaluation rules are specified in a pseudo-C language notation. We provide data typing facilities for the attribute variables.* The inherited attributes of a nonterminal and their data types are listed after the key word "inherited", which appears after the user-name of that nonterminal. The synthesized attributes

*As with the attributed grammars for programming languages, our experience in using attributed grammars for application languages suggests that facilities that include standard libraries of functions, user-defined types, and global attributes are needed.

of that nonterminal are then listed in a similar fashion. The attribute evaluation rules are listed after each production specification, within braces. "$0.attribute_name" in these rules refers to the correspondingly named attribute of the nonterminal in the left-hand side of the production. "$i.attribute_name" refers to the correspondingly named attribute of the $i$th nonterminal in the right-hand side of the production.

For each inherited attribute of each nonterminal appearing in the right-hand side of a production rule, there must be an evaluation rule (a function call or an arithmetic expression) assigning a value to that attribute associated with that production rule. Similarly, for each synthesized attribute of the left-hand side nonterminal of a production rule, there must be a rule (a function call or an arithmetic expression) assigning a value to that attribute. These rules may take as arguments the inherited attributes of the left-side nonterminal and/or synthesized attributes of the nonterminals in the right-hand side. These rules must not, however, violate the properties defining $L$-attributedness[14] of the grammar in order for the proposed evaluation scheme to work.

For iterative production rules, the semantic specifications appear as:

```
disjunct      : ⟨CLAUSES⟩
                inherited:;
                synthesized:;
                |xxx x +clause "~,~"
                 {
                 int current_temp;
                 init: {
                    current_temp = 0;
                 }
                 repeat: {
                    $1.temp_loc = current_temp;
                    current_temp = $1.next_temp;
                 }
                 }
                ;
```

The routine following the key word "init:" is executed when this production rule is initially chosen. Then, for each instance of the nonterminal "clause" under the parent "disjunct", the specification following the key word "repeat:" is used. The "$1" in the latter specification refers to the instance of the nonterminal "clause" whose attributes are being evaluated. These evaluation specifications must follow the same guidelines as those of the regular productions. Nonterminals deriving "identifiers" (i.e., those having no production rules)

are assumed to have only one synthesized attribute, "value", whose value is the string entered by the programmer at the time of expansion.

### *7.2 Generation of the attribute evaluators*

In any specification of an application language as above, each implementor-defined attribute evaluation function is associated with a specific production. For each distinct attribute in the grammar, the calls to these functions are collected into one generated evaluation function, which determines the appropriate implementor-defined function to call, based on the rule that produced the associated nonterminal. To make the required environment information accessible during the attribute evaluation, a pointer to the tree node at which the generated function is to be evaluated is passed as a parameter to the generated function. The evaluation functions for inherited attributes must be passed a pointer to the parent node of the nonterminal node with which the attribute is associated (i.e., inherited attributes are evaluated when the associated nonterminal appears on the right-hand side of a production). Synthesized attribute evaluation functions must be passed a pointer to the nonterminal node with which the attribute is associated (i.e., synthesized attributes are evaluated when the associated nonterminal is expanded).

Each generated evaluation function for an inherited attribute contains a section of code for each production in which the associated nonterminal can appear on the right-hand side. Similarly, each generated evaluation function for a synthesized attribute contains a section of code for each production in which the associated nonterminal appears on the left-hand side. Within each of these sections of code is the code for the appropriate implementor-defined function, along with the function calls required to evaluate the actual parameters of the implementor's function. The attributes are evaluated only as required, and the evaluation nesting is managed automatically by the C (target programming language) procedure calling mechanism.

## VIII. REMARKS

The editor generator HEG, as described here (excluding the semantic analysis), currently exists and has been used for a variety of application languages. The novel aspects of this system include: (1) the ability to give a programmer the option to use menu selection or to enter strings containing terminal characters and nonterminal names at *any* stage during the expansion of a program in the user-language, and (2) the ability to generate a hybrid editor from a high-level specification of a user grammar. This system was an experiment to investigate only these aspects of editing. Various other related issues such as building a robust syntax-sensitive editor for higher-level

languages (e.g., C language), experimenting with different ways of exhibiting the menus, and human-factors issues are being addressed separately.[5]

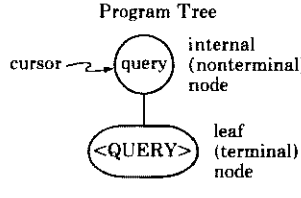
## REFERENCES


1. V. Donzeau-Gouge et al., "A Structure-Oriented Program Editor: a First Step Towards Computer-Assisted Progamming," Proc. of Int. Computing Symp., Antibes, June 1975.
2. P. H. Feiler and R. Medina-Mora, "An Incremental Programming Environment," Carnegie-Mellon University, Computer Science Department Report, December 1980.
3. C. N. Fischer, G. Johnson, and J. Mauney, "An Introduction to Release 1 of Editor Allan Poe," University of Wisconsin, Technical Report 453, 1981.
4. C. W. Fraser, "Syntax-Directed Editing of General Data Structures," Proc. ACM SIGPLAN-SIGOA Symp. Text Manipulation, Portland, Oregon, June 1981, pp. 17–21.
5. E. R. Gansner et al., "SYNED—A Language Based Editor for an Interactive Programming Environment," Spring COMPCON 83 San Francisco, California, February 1983, pp. 406–10.
6. M. R. Horton, "Design of a Multi-Language Editor with Static Error Detection Capabilities," Ph.D. Dissertation, Computer Science, University of California, Berkeley, July 1981, p. 158.
7. T. Teitelbaum et al., "The Why and Wherefore of the Cornell Program Synthesizer," Proc. ACM SIGPLAN-SIGOA Symp. on Text Manipulation, Portland, Oregon, June 1981, pp. 8–16.
8. S. R. Wood, "Z—The 95% Program Editor," Proc. ACM SIGPLAN-SIGOA Symp. on Text Manipulation, Portland, Oregon, June 1981, pp. 1–7.
9. R. C. Waters, "Program Editors Should Not Abandon Text Oriented Commands," ACM SIGPLAN Notices, *17*, No. 7 (July 1982), pp. 39–46.
10. T. Reps, "Optimal-time Incremental Semantic Analysis for Syntax-directed Editors," Proc. 9th Annual Symp. on Principles of Programming Languages, Albuquerque, New Mexico, January 1982, pp. 169–76.
11. Stephen C. Johnson, "Yacc: Yet Another Compiler-Compiler," Computer Science Technical Report #32, Bell Laboratories, Murray Hill, 1976.
12. E. R. Gansner et al., "Semantics and Correctness of a Query Language Translation," Ninth Symp. Principles of Programming Languages, Albuquerque, New Mexico, January 1982, pp. 289–98.
13. M. E. Lesk, "Lex—A Lexical Analyzer Generator," Computer Science Technical Report #39, Bell Laboratories, Murray Hill, 1975.
14. P. M. Lewis, D. J. Rosenkrantz, and R. E. Stearns, "Attributed Translations," J. Computer and System Sciences, *9* (December 1974), pp. 279–307.


## APPENDIX A

### *An Editing Session With 'Hisel.age'*

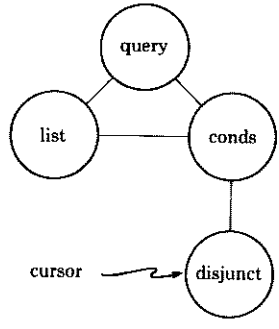
\$hisel.age
program? test
Initializing the tree/text for test..
Terminal Screen
1) ⟨QUERY⟩



:r <- user command - invisible
file name? temp

Assume that there is a file named 'temp' in the working directory and that it has the string "select ⟨ITEM_ LIST⟩ where ⟨CLAUSES⟩". AGE will read it, recognize the two nonterminal names, parse the string, create the subtree, and graft it to the root as shown below.

2) select ⟨ITEM_ LIST⟩
   where ⟨CLAUSES⟩

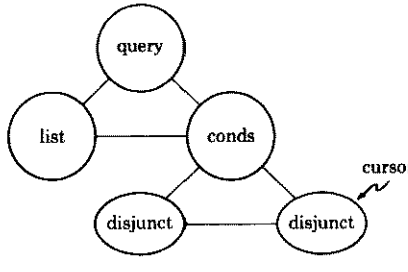



(Leaves are not shown)

:a

This will add one more instance of CLAUSES. The screen after this command will be:

3) select ⟨ITEM_ LIST⟩
   where ⟨CLAUSES⟩
      or ⟨CLAUSES⟩




:e
:e

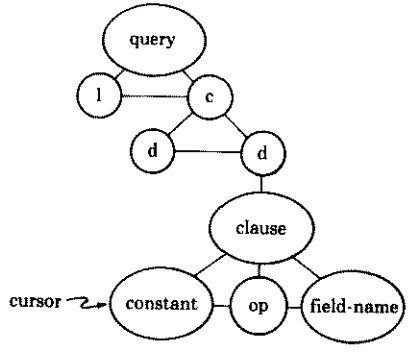
AGE will install a CLAUSE in place of ⟨CLAUSES⟩ after the first e command. Since there is a choice for the expansion corresponding to the next e command, the following menu items will appear one after

another if the user asks for a menu:

expansion character (type ? for menu): ? <- user types this
field_op-const 1 y(es), n(ext), q(uit)? n <- user types this
const-op-field 2 y(es), n(ext), q(uit)? y

The screen after this will be:

4) select ⟨ITEM_LIST⟩
   where ⟨CLAUSES⟩
     or constant OPERATOR field_name




:w
This command will write the above text into test.hisel file and the tree representation in test.lmd file.
:q
AGE will then print:
BYE! test hasn't been fully expanded; call me back later.

## APPENDIX B

### *AGE Command Summary*

***Text-navigation commands:***

cr - a carriage return : go to the next line
- - the character minus: go back one line
space-bar or n : go to the next word
b : go back one word

***Tree-navigation commands:***

^ : go to the parent of the current nonterminal
V : go to the first son of the current nonterminal
> : go to the right neighboring nonterminal under the same parent
< : go to the left neighboring nonterminal under the same parent
N : go the next unexpanded nonterminal
B : go backwards for the next unexpanded nonterminal

*Nonterminal expansion commands:*

p : parse text for the current nonterminal
r : read text from a file and parse it for the current nonterminal
e : expand the current nonterminal by menu selection
a : append an instance of a 'listy' (i.e., iterative) nonterminal
i : insert an instance of a 'listy' nonterminal
d : delete the current instance of the 'listy' nonterminal

*Other commands:*

U : unexpand nonterminal
w : write program
q : quit editing with AGE

## AUTHORS

**Beth A. Bottos,** B.S.E.E., 1979, Purdue University; M.S. (Electrical Engineering), 1980, Stanford University; M.S. (Computer Science), 1983, Carnegie-Mellon University. From 1976 to 1979, Ms. Bottos was a participant in the Bell Laboratories Engineering Scholarship Program (BLESP). During that time, her work included telephone ringer characterization and modification and hardware design for a home appliance control system. From 1979 to 1981, she was involved in the development of the 1AVSS (Voice Storage System) and LADT (Local Area Data Transport) systems. Since 1981, Ms. Bottos has been a student in Computer Science at Carnegie-Mellon University. During that time, she has also worked for Bell Laboratories in the Advanced Software Department and in the Advanced Software Technology Laboratory.

**Chandra M. R. Kintala,** B.Sc. (Honors), 1970, Regional Engineering College, Rourkela, India; M.Tech. (Electrical Engineering), 1973, Indian Institute of Technology, Kanpur, India; Ph.D. (Computer Science), 1977, Pennsylvania State University; Assistant Professor, Computer Science Department, University of Southern California, 1977–1980; Bell Laboratories, 1980—. Mr. Kintala is a member of the Advanced Software Department. He has worked on the complexity of nondeterminism in various classes of Automata and Turing machines. His current interests include programming environments, compiler techniques for database query language translators and software graphics. Member, ACM, IEEE Computer Society, Sigma Xi, Phi Kappa Phi, and Who's Who in the East.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# Performance Analysis of a Preemptive Priority Queue with Applications to Packet Communication Systems

By M. G. HLUCHYJ,* C. D. TSAO,* and R. R. BOORSTYN†

(Manuscript received May 6, 1983)

In this paper we analyze the performance of a preemptive priority queue. We give the model description in the context of a packet communication system where message sources, having different priorities, share a common communication channel. Each source generates, as an independent Poisson process, messages consisting of an arbitrarily distributed, random number of fixed-length packets. The channel server can only begin service at integer multiples of the packet transmission time (i.e., a time-slotted channel is assumed), and the server will preempt an ongoing message transmission at the next packet boundary whenever there is a message arrival from a higher-priority source. The average in-queue waiting time for each packet in any given source message and the average message delay are derived along with the corresponding moment-generating functions. Also, comparisons are made with the first-come first-served queueing discipline.

## I. INTRODUCTION

We analyze the performance of a preemptive priority queueing system. To make clear at the outset the importance of the particular queueing system studied, we describe the system model in a packet communication context. Specifically, as Fig. 1 illustrates, a number of data sources share a single communication channel. Each source generates, according to a Poisson process, messages consisting of a

*AT&T Information Systems. †Polytechnic Institute of New York.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

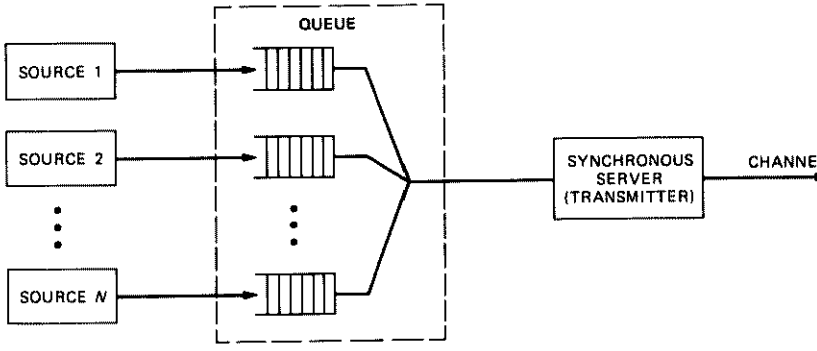



Fig. 1—Queueing model for a packet communication system.

random number of fixed-length data packets. The packets comprising a message arrive in bulk to be transmitted on the communication channel. For clarity, we view each source as having its own separate buffer to queue packets. Here, packets generated by the source wait for access to the channel.

Packet transmissions on the channel are synchronized. More precisely, time is divided into a sequence of fixed-length intervals or time slots. Each time slot is just large enough to allow the transmission of one packet, and packet transmissions must occur within time-slot boundaries. Hence, a packet arriving at the queue, at the very least, must wait until the start of the next time slot before its transmission can begin.

Packets from any given source are served (i.e., transmitted) on a first-come first-serve basis. The sources, however, are assigned fixed priorities: the first source has the highest priority, the last has the lowest. At the start of each time slot, the first packet queued from the highest-priority source is served. That is, a packet at the head of the source $k$ buffer is transmitted if and only if the buffers associated with sources 1 to $k-1$ are empty. Hence, an ongoing packet transmission cannot be preempted; however, an ongoing message transmission will be preempted (at the next slot boundary) whenever there is a message arrival from a higher-priority source.

Such a priority queueing discipline arises naturally in many packet communication systems. The channel might be a link in a data communication network, or may simply be a shared data bus. The use of priority may be required to give more urgent messages lower delay. For example, one might choose to give network control messages higher priority than interactive data messages, which in turn are given higher priority than long file transfers. In some situations, the priority structure is inherent in the mechanism for sharing the channel among the independent messages sources. This is the case with Datakit,[1]

where the module (i.e., the interface between the source and the channel) with the highest address always wins the channel contention. It is also true of some slotted ring systems, where the physical order of sources along the ring imposes a priority ordering for access to the channel.[2]

The first results on queues with preemptive priority appear to be due to White and Christie.[3] Shortly thereafter, others studied the problem using different assumptions about the service time distribution. A comprehensive treatment of some of the early work is given in Jaiswal,[4] and a more up-to-date, but less comprehensive, discussion may be found in Kleinrock.[5] The models examined, however, all assume an "asynchronous" server where service starting times and preemption times are not constrained to certain periodically recurring points. The use of a synchronous service facility in queueing models arises in the context of computer and data communication systems where there is a natural elementary unit of time such as the machine cycle of a processor, or the bit, byte, or packet transmission time on a channel. Many such models are reviewed, and references given, in Kobayashi and Konheim.[6] As we indicated, the model we have selected for study has applications to slotted ring systems, and it is here that one finds analysis of other models similar to ours. The model that seems to come closest is by Konheim and Meister,[2] where the main differences have to do with the arrival process. Konheim and Meister assume discrete arrivals (between slots) of packets, whereas we assume continuous arrivals of messages with each message containing an arbitrarily distributed number of packets. In this way, we are better able to examine message delays in the system.

In this paper we analyze the performance of the above preemptive priority queueing system. We begin in Section II by summarizing the queueing model and introducing performance measures that are of interest. In Section III we derive the average in-queue waiting time for each packet in any given source message. From this result we easily obtain the average delay in transporting a message. The corresponding moment-generating functions are derived in the appendix. Finally, in Section IV, we compare performance with the First-Come First-Served (FCFS) queueing discipline.

## II. QUEUEING MODEL

In this section we briefly summarize the important points of the queueing model, and indicate the steady-state statistics that are of interest. Notation established here is used in the performance analysis that follows.

The queueing system under study has the following properties:

1. $N$ sources of messages.

2. Priorities are assigned to sources in decreasing order (i.e., source $k$ has higher priority than source $k+1$, $k = 1, 2, \cdots, N-1$).

3. Source $k$ generates messages as an independent Poisson process with rate $\lambda_k$ messages per time slot. Each such message has its length (in packets) selected independently from the distribution $P_{m_k}(\cdot)$ with first and second moments, $\overline{m}_k$ and $\overline{m_k^2}$, respectively.

4. During busy periods, one packet is transmitted in each time slot and is always selected at the beginning of the time slot from the head of the highest-priority, nonempty source buffer.

5. Each source buffer is assumed infinite, and packets enter and are removed from the buffer on a first-in first-out basis.

We define $W_{kj}$ as the steady-state in-queue waiting time for the $j$th packet in a message from source $k$, $k = 1, 2, \cdots, N$. In addition, we define

$$\rho_k = \lambda_k \overline{m}_k,$$

where $\rho_k$ is interpreted as the fraction of time the server is busy with source $k$ packets. We also find it convenient to define

$$\sigma_k = \sum_{i=1}^{k} \rho_i.$$

Other notation is introduced as needed in the analysis.

## III. PERFORMANCE ANALYSIS

We begin this section by deriving $\overline{W}_{kj}$, the average in-queue waiting time for the $j$th packet in a message from source $k$. Using this result we then obtain the average delay in transporting a message from source $k$. Included in the discussion are specific numerical examples to illustrate the derived results.

### *3.1 Average waiting time analysis*

In Fig. 2, observe that we may express the waiting time for the $j$th packet in a source $k$ message as

$$W_{kj} = W_{k1} + \sum_{\ell=1}^{j-1} w_{k\ell}, \tag{1}$$

where the incremental waiting time $w_{k\ell}$ is defined by

$$w_{k\ell} = W_{k,\ell+1} - W_{k\ell}.$$

For a given message length, $m_k$, the random variables $\{w_{k1}, w_{k2}, \cdots, w_{k,m_k-1}\}$ are independent and identically distributed. We observe that at the beginning of a slot during which a packet from source $k$ is in service, there are no packets from sources 1 to $k-1$ in the system.

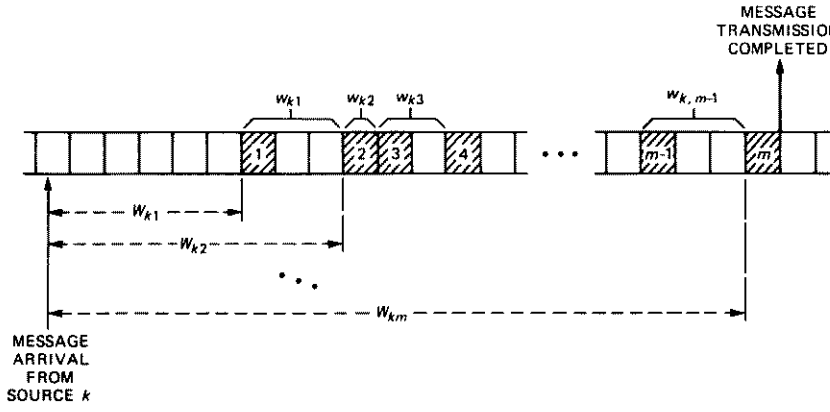



Fig. 2—Waiting times for packet transmissions.

Any messages that arrive from sources 1 to $k-1$ while this source $k$ packet is in service spawn a busy period of service, starting in the next slot, for sources 1 to $k-1$. All such busy periods are independent and identically distributed, and hence so are the random variables $\{w_{k1}, w_{k2}, \ldots, w_{k,m_k-1}\}$.

Note that the incremental waiting time $w_{k\ell}$ consists of one slot time to transmit the $\ell$th packet in the source $k$ message plus the time to serve all messages from sources 1 to $k-1$ that arrive in the interval $w_{k\ell}$. Hence, the average incremental waiting time $\overline{w}_{k\ell}$ satisfies

$$\overline{w}_{k\ell} = 1 + \sigma_{k-1}\overline{w}_{k\ell},$$

from which we obtain

$$\overline{w}_{k\ell} = \frac{1}{1-\sigma_{k-1}}.$$

It then follows from (1) that the average in-queue waiting time for the $j$th packet in a source $k$ message is given by

$$\overline{W}_{kj} = \overline{W}_{k1} + \frac{j-1}{1-\sigma_{k-1}}. \tag{2}$$

Hence we are left with having to determine $\overline{W}_{k1}$, the average waiting time for the first packet in the message.

By applying standard queueing arguments, we have

$$\overline{W}_{k1} = \frac{1}{2} + \sum_{i=1}^{k}\sum_{j=1}^{\infty}\rho_{ij}\overline{W}_{ij} + \sum_{i=1}^{k-1}\rho_i\overline{W}_{k1}, \tag{3}$$

where

$$\rho_{ij} \triangleq \lambda_i \Pr[m_i \geq j]. \tag{4}$$

The first term on the right-hand side of (3) is simply the average time between the arrival of a message and the start of the next slot. The second term is, by Little's result, the average number of packets of equal or higher priority awaiting transmission at the moment the message arrives. Finally, the last term corresponds to the average number of packets of higher priority that arrive while the first packet in the source $k$ message waits on queue.

Now substituting (2) into (3) yields

$$\overline{W}_{k1} = \frac{1}{2} + \sum_{i=1}^{k} \sum_{j=1}^{\infty} \rho_{ij} \left( \overline{W}_{i1} + \frac{j-1}{1-\sigma_{i-1}} \right) + \sum_{i=1}^{k-1} \rho_i \overline{W}_{k1}. \tag{5}$$

Note from the definition of $\rho_{ij}$ in (4) that

$$\begin{aligned} \sum_{j=1}^{\infty} \rho_{ij} &= \lambda_i \sum_{j=1}^{\infty} \Pr[m_i \geq j] = \lambda_i \sum_{j=1}^{\infty} \sum_{\ell=j}^{\infty} \Pr[m_i = \ell] \\ &= \lambda_i \sum_{\ell=1}^{\infty} \sum_{j=1}^{\ell} \Pr[m_i = \ell] = \lambda_i \sum_{\ell=1}^{\infty} \ell \Pr[m_i = \ell] \\ &= \lambda_i \overline{m}_i = \rho_i. \end{aligned} \tag{6}$$

Similarly, we have that

$$\sum_{j=1}^{\infty} (j-1)\rho_{ij} = \frac{\lambda_i}{2} (\overline{m_i^2} - \overline{m}_i). \tag{7}$$

Hence, using (6) and (7), we may rewrite (5) as

$$\overline{W}_{k1} = \frac{\frac{1}{2} + \sum_{i=1}^{k-1} \rho_i \overline{W}_{i1} + \sum_{i=1}^{k} \lambda_i (\overline{m_i^2} - \overline{m}_i)/2(1-\sigma_{i-1})}{(1-\sigma_k)}.$$

Solving recursively, we obtain

$$\overline{W}_{k1} = \frac{1 + \sum_{i=1}^{k} \lambda_i (\overline{m_i^2} - \overline{m}_i)}{2(1-\sigma_k)(1-\sigma_{k-1})}. \tag{8}$$

Finally, substituting (8) into (2) yields

$$\overline{W}_{kj} = \frac{1 + \sum_{i=1}^{k} \lambda_i (\overline{m_i^2} - \overline{m}_i)}{2(1-\sigma_k)(1-\sigma_{k-1})} + \frac{j-1}{1-\sigma_{k-1}}. \tag{9}$$

This concludes the derivation of the average in-queue waiting time $\overline{W}_{kj}$. The derivation of the moment-generating function for $W_{kj}$ (from which $\overline{W}_{kj}$ can be obtained directly) is given in the appendix.

To illustrate the performance, we begin by considering a homoge-

neous system where $\lambda_k = \lambda$, $\overline{m}_k = \overline{m}$, and $\overline{m_k^2} = \overline{m^2}$ for $k = 1, 2, \cdots, N$. For this case, (8) becomes

$$\overline{W}_{k1} = \frac{1 + \frac{k}{N}\rho\,(\overline{m^2}/\overline{m} - 1)}{2\left(1 - \frac{k}{N}\rho\right)\left(1 - \frac{(k-1)}{N}\rho\right)}, \tag{10}$$

where $\rho$ is the total system utilization (or load) defined by

$$\rho = \sum_{i=1}^{N} \rho_i$$

$$= N\lambda\overline{m} \qquad \text{(for a homogeneous system).}$$

If we take $N = 10$ and assume a constant message length of 10 packets (i.e., $\overline{m} = 10$, $\overline{m^2} = 100$), Fig. 3 is a plot of $\overline{W}_{k1}$ vs. $\rho$ for $k$ varying from

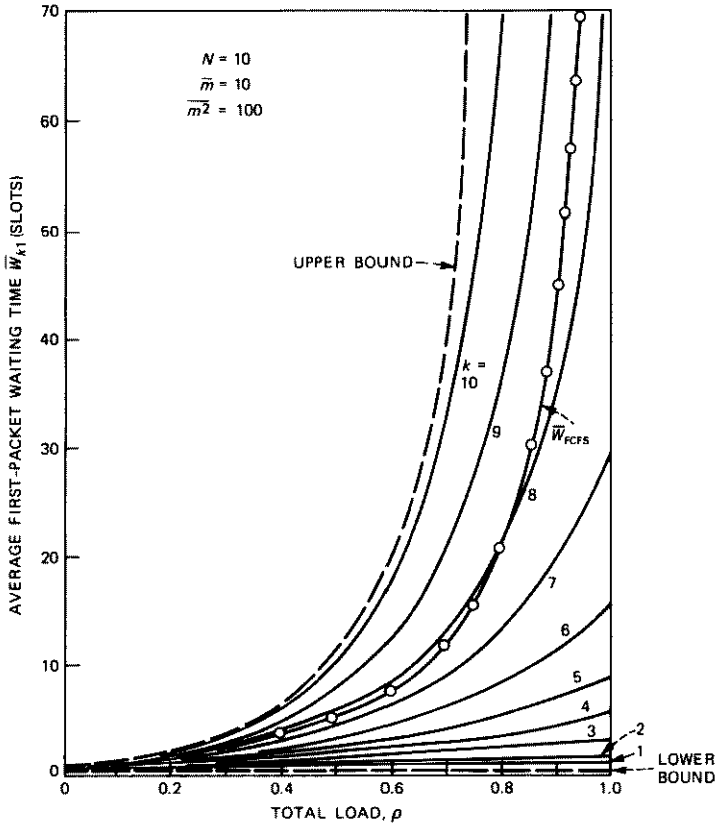



Fig. 3—Average first packet waiting time $\overline{W}_{k1}$ vs. total load $\rho$.

1 to 10. Also shown in Fig. 3 is the average waiting time for the first-come first-served (FCFS) queueing discipline, which is derived in Section IV. Note from (10) that if we allow $N \to \infty$, then

$$\overline{W}_{11} \to \frac{1}{2}$$

$$\overline{W}_{N1} \to \frac{1 + \rho(\overline{m^2}/\overline{m} - 1)}{2(1 - \rho)^2}.$$

These two expressions represent, respectively, lower and upper bounds on the average first packet waiting time for all sources and arbitrary $N$. These bounds are plotted as dashed lines in Fig. 3. Finally, if we assume the same values for $N$, $\overline{m}$, and $\overline{m^2}$ as in Fig. 3, Fig. 4 is a plot of the average incremental waiting time $\overline{w}_{k\prime}$ vs. $\rho$ for $k$ varying from 1 to 10. Also shown in Fig. 4 is the upper bound $1/(1 - \rho)$ on $\overline{w}_{k\prime}$, valid for all parameter values.

### *3.2 Average message delay analysis*

We now consider the average message delay. Defining $\overline{D}_k(m)$ as the average delay (in slots) from the arrival to the queue of an $m$-packet

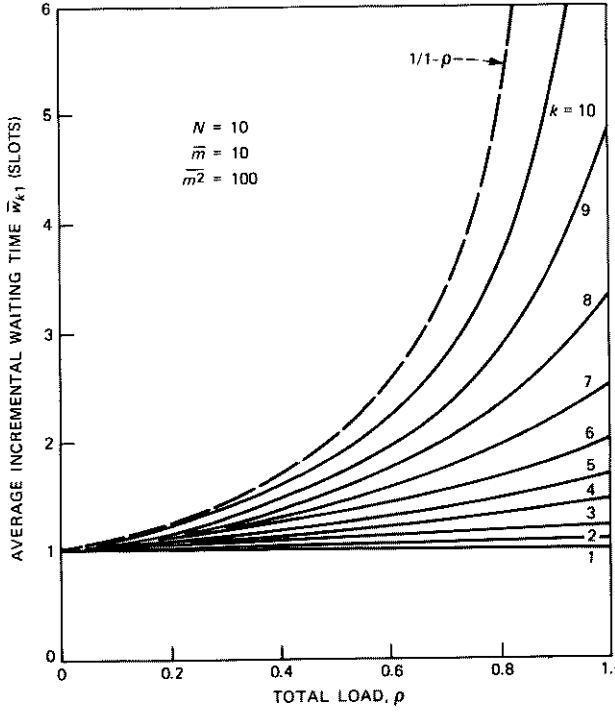



Fig. 4—Average incremental waiting time $\overline{w}_{k1}$ vs. total load $\rho$.

message from source $k$ until the end of its transmission, we have

$$\overline{D}_k(m) = \overline{W}_{km} + 1.$$

Letting $\overline{D}_k$ denote the average delay over all messages from source $k$, it follows, since $\overline{W}_{km}$ is linear in $m$, that

$$\begin{aligned}\overline{D}_k &= \sum_m \overline{D}_k(m) P_{m_k}(m) \\ &= \overline{W}_{k,\overline{m}_k} + 1. \end{aligned} \tag{11}$$

If we assume the same homogeneous system as represented in Figs. 3 and 4, Fig. 5 is a plot of $\overline{D}_k$ vs. $\rho$ for $k$ varying from 1 to 10. Also shown in Fig. 5 is an upper bound on $\overline{D}_k$, obtained from the upper bounds on $\overline{W}_{k1}$ and $\overline{w}_{k\ell}$. Specifically, we have

$$\overline{D}_k \leq \frac{1 + \rho[\overline{m^2}/\overline{m} - 1]}{2(1-\rho)^2} + \frac{\overline{m} - 1}{(1-\rho)} + 1,$$

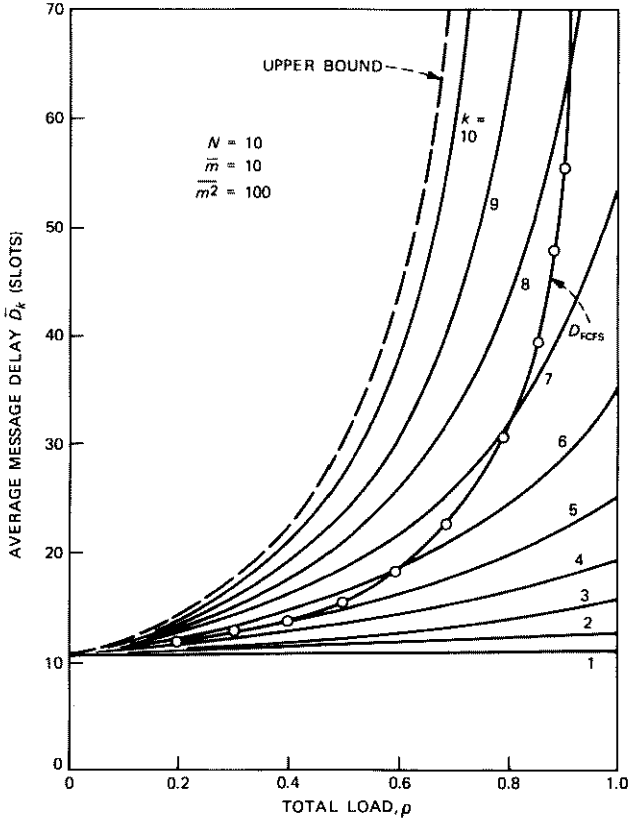



Fig. 5—Average message delay $\overline{D}_k$ vs. total load $\rho$.

which depends on $\overline{m}$ and $\overline{m^2}$, but is valid for all sources and arbitrary $N$.

To complete this section, we consider a nonhomogeneous system consisting of 10 host computers and 300 terminals. The terminals and hosts correspond to the message sources and may be viewed as sharing a common time-slotted bus. There is a priority ordering of the terminals and hosts, with terminals having priority over hosts (i.e., the terminals correspond to sources 1 to 300 and the hosts correspond to sources 301 to 310). Each host is assumed to generate two types of traffic: host-to-host file transfers consisting of fixed-length 32-packet messages, and host-to-terminal messages with an average message length of 2 packets and a standard deviation of 1. Each terminal, on the other hand, only generates messages that are one packet in length and destined to a host. The message generation rates for each of the two types of host traffic are the same for all hosts. Similarly, all terminals generate messages at the same rate. The specific generation rate of each traffic type is such that the total load on the channel is divided as follows: 30 percent host-to-host, 60 percent host-to-terminal, and 10 percent terminal-to-host. The average delay performance for this system is plotted in Fig. 6. Observe that the results obtained allow us to distinguish between different types of traffic generated by the same source. In particular, in Fig. 6, the average message delay performance for the host-to-host and host-to-terminal traffic are shown separately.

From the moment-generating function for $D_k$ derived in the appendix, one can obtain the second moment of the message delay. This, in turn, may be used to compute the message delay standard deviation. For hosts 1 and 10 (i.e., the two extremes), shown in Fig. 7 for the host-to-host messages and in Fig. 8 for the host-to-terminal messages, we see the mean delay and mean delay plus one, two, and three standard deviations (denoted by $1\sigma$, $2\sigma$, and $3\sigma$). The second-moment-of-message delay depends on the first three moments of message length, and in Fig. 8 we set $\overline{m^3} = 15$.

## IV. COMPARISONS WITH FCFS

In this section we compare the average delay performance of the priority queueing discipline studied in the previous section with that of the First-Come First-Served (FCFS) discipline. With the FCFS discipline, messages are served in the order in which they are generated, independent of the source from which they originate. In this way, the FCFS discipline allocates the communication channel more fairly than does the priority discipline. For simplicity, we assume in the analysis a homogeneous system where $\lambda_k = \lambda$, $\overline{m}_k = \overline{m}$, and $\overline{m_k^2} = \overline{m^2}$ for $k = 1, 2, \cdots, N$.

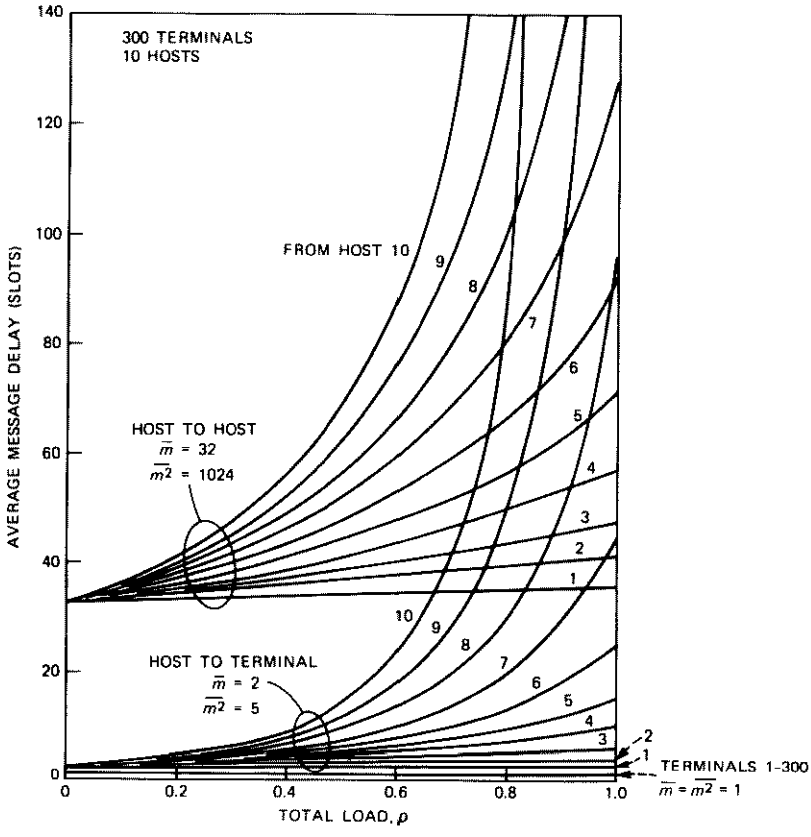



Fig. 6—Average message delay vs. total load $\rho$.

The performance analysis of the FCFS queueing discipline is a special case of the results obtained for the priority discipline. Specifically, we combine the $N$ independent Poisson streams into a single Poisson stream (using the well-known result that the sum of independent Poisson processes is a Poisson process) with rate $N\lambda$. From (10) we have that the average in-queue waiting time for a message generated by this combined (single) source is given by

$$\overline{W}_{\text{FCFS}} = \frac{\rho(\overline{m^2}/\overline{m})}{2(1-\rho)} + \frac{1}{2}, \tag{12}$$

where again $\rho = N\lambda\overline{m}$ is the total system utilization. The average message delay for the FCFS system is given by

$$\begin{aligned}\overline{D}_{\text{FCFS}} &= \overline{W}_{\text{FCFS}} + \overline{m} \\ &= \frac{\rho(\overline{m^2}/\overline{m})}{2(1-\rho)} + \frac{1}{2} + \overline{m}.\end{aligned} \tag{13}$$

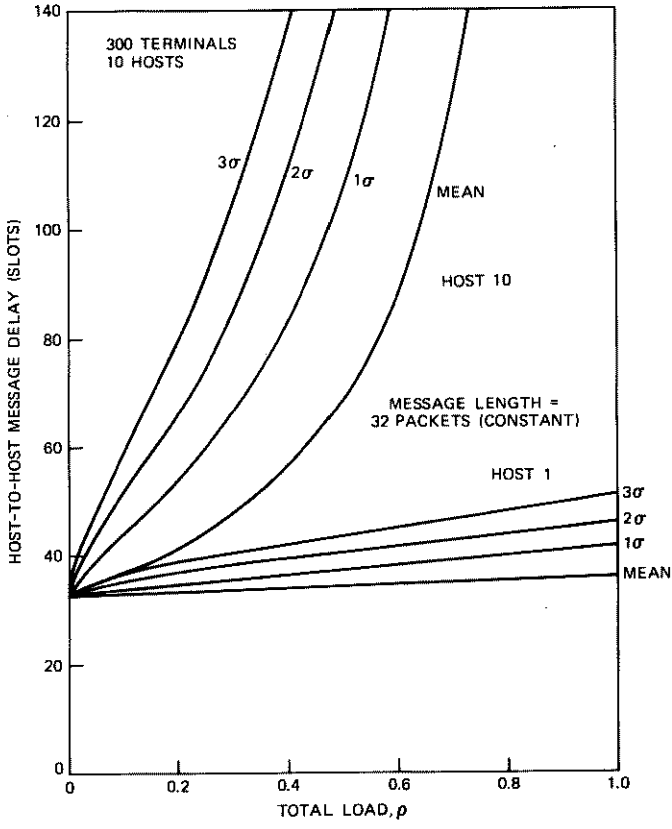



Fig. 7—Host-to-host message delay vs. total load $\rho$.

$\overline{W}_{\text{FCFS}}$ is plotted in Fig. 3 and $\overline{D}_{\text{FCFS}}$ is plotted in Fig. 5 for the assumed system parameter values.

It is worth noting that the waiting time and delay results given by (12) and (13), respectively, differ from those corresponding to the standard M/G/1 queueing system by the additional term 1/2. This added term results from the synchronous nature of the server and represents the average time an arriving message must wait before the start of the next time slot.

We continue the priority and FCFS comparison by focusing on the unfairness issue. Specifically, we consider the ratio of the average message delay for source $N$ to that of source 1, $\overline{D}_N/\overline{D}_1$. Since all sources encounter the same average delay in the FCFS discipline, $\overline{D}_N/\overline{D}_1 = 1$. With the priority discipline, source $N$ has the lowest priority and source 1 the highest; hence $\overline{D}_N/\overline{D}_1 > 1$ for $\rho > 0$. In particular, we have from (9) and (11) that

$$\overline{D}_k = \frac{1 + \frac{k}{N}\rho[(1 + c_m^2)\overline{m} - 1]}{2\left(1 - \frac{k}{N}\rho\right)\left(1 - \frac{(k-1)}{N}\rho\right)} + \frac{\overline{m} - 1}{\left(1 - \frac{(k-1)}{N}\rho\right)} + 1, \quad (14)$$

where $c_m^2$ is the squared coefficient of variation for the message-length distribution defined by

$$c_m^2 = \frac{\text{variance}(m)}{(\overline{m})^2}.$$

Hence, for large $N$ we have from (14) that

$$\overline{D}_1 = \frac{\frac{\rho}{N}(1 + c_m^2)\overline{m}}{2\left(1 - \frac{\rho}{N}\right)} + \frac{1}{2} + \overline{m}$$

$$\approx \frac{\overline{m}}{2}\left[\frac{1}{\overline{m}} + \frac{\rho}{N}(1 + c_m^2) + 2\right]$$

and

$$\overline{D}_N = \frac{1 + \rho[(1 + c_m^2)\overline{m} - 1]}{2(1 - \rho)\left(1 - \frac{(N-1)}{N}\rho\right)} + \frac{\overline{m} - 1}{\left(1 - \frac{(N-1)}{N}\rho\right)} + 1$$

$$\approx \frac{\overline{m}\left[(1 + c_m^2)\rho + \left(2 - \frac{1}{\overline{m}}\right)(1 - \rho) + \frac{2}{\overline{m}}(1 - \rho)^2\right]}{2(1 - \rho)^2}.$$

It follows then that

$$\frac{\overline{D}_N}{\overline{D}_1} \approx \begin{cases} \dfrac{1 + c_m^2\rho + 2(1 - \rho)^2}{3(1 - \rho)^2} & \text{for } \overline{m} = 1 \\ \dfrac{2 + (c_m^2 - 1)\rho}{2(1 - \rho)^2} & \text{for } \overline{m} \gg 1. \end{cases}$$

Observe that for large $N$ and fixed $\rho$, the increase in $\overline{D}_N/\overline{D}_1$ is approximately linear with the squared coefficient of variation $c_m^2$. In Fig. 9, the ratio $\overline{D}_N/\overline{D}_1$ is plotted against total utilization $\rho$ for the FCFS and priority disciplines with $c_m^2 = 0$ and 1.

To complete this section, we compare the average delay performance of the FCFS discipline with the overall average delay of the priority discipline. That is, we compare $\overline{D}_{\text{FCFS}}$ as given by (13) to the quantity

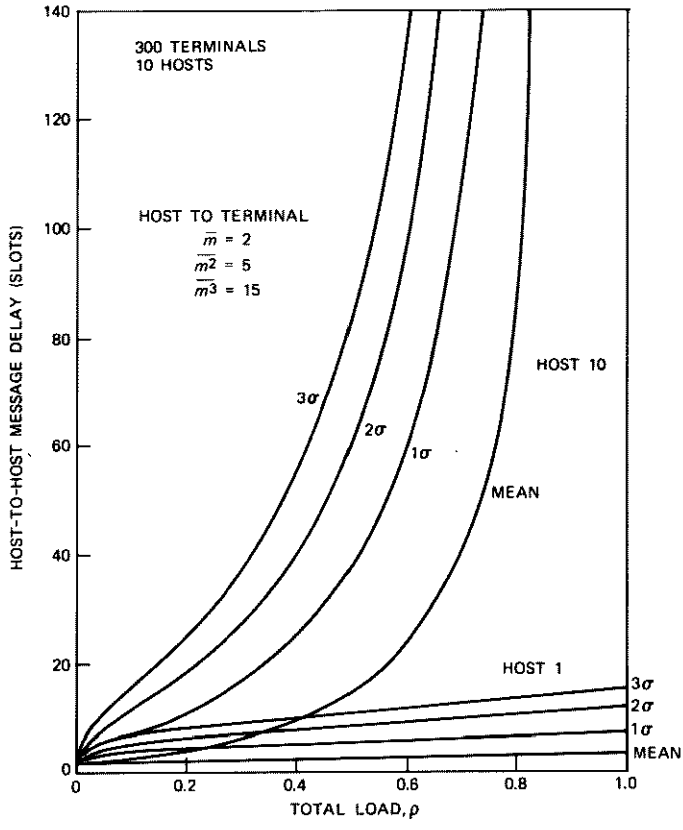



Fig. 8—Host-to-terminal message delay vs. total load $\rho$.

$$\overline{D} \triangleq \frac{1}{N} \sum_{k=1}^{N} \overline{D}_k.$$

Using the expression for $\overline{D}_k$ given in (14), we obtain after some manipulation

$$\overline{D} = \frac{(1 + c_m^2)\overline{m}}{2(1 - \rho)} + \frac{1}{2N} [2\overline{m} - 1 - (1 + c_m^2)\overline{m}]\gamma + 1,$$

where

$$\gamma = \sum_{k=1}^{N} \left[1 - \frac{(k-1)}{N} \rho\right]^{-1}.$$

From this, one may show that

$$\overline{D}_{\text{FCFS}} \leq \overline{D} \quad \text{for} \quad 0 \leq c_m^2 \leq \frac{\overline{m} - 1}{\overline{m}}$$

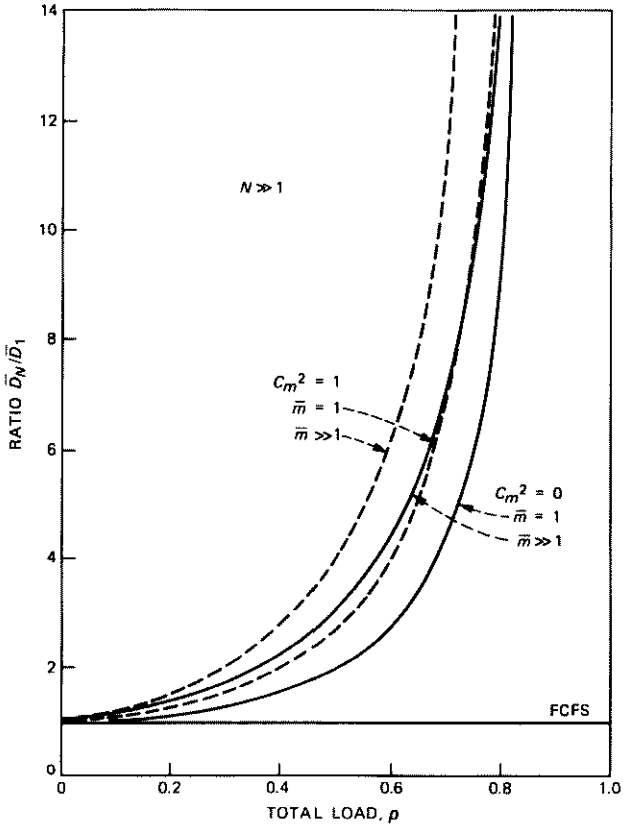



Fig. 9—Ratio $\overline{D}_N/\overline{D}_1$ vs. total load $\rho$.

and

$$\overline{D}_{\text{FCFS}} \geq \overline{D} \quad \text{for} \quad c_m^2 \geq \frac{\overline{m} - 1}{\overline{m}}.$$

Hence, for sufficiently large message-length coefficient of variation $c_m^2$, the overall average delay for the priority discipline is less than the average delay for the first-come first-served discipline. Of course, as we saw earlier, as $c_m^2$ increases so does the relative unfairness of the priority discipline over the FCFS discipline.

## V. CONCLUSIONS

We analyzed the performance of a preemptive priority queue, which has direct applications to packet communication systems. The main distinguishing feature of the system studied compared to the standard M/G/1 preemptive resume priority queue[5] is that the server can only

begin serving a "customer" (and preemptions take place) at integer multiples of time corresponding to packet slot boundaries in the communication context. Mean value formulas for in-queueing waiting time and average message delay were derived and comparisons made to the FCFS queueing discipline. A derivation of the waiting time and delay moment-generating functions is given in the appendix.

## REFERENCES


1. A. G. Fraser, "Datakit—A Modular Network for Synchronous and Asynchronous Traffic," Proc. ICC (June 1979), pp. 20.1.1–.3.
2. A. G. Konheim and B. Meister, "Service in a Loop System," JACM, *19*, No. 1 (January 1972), pp. 92–108.
3. H. White and L. S. Christie, "Queuing with Preemptive Priorities or with Breakdown," Oper. Res., *6*, No. 1 (January–February 1958), pp. 79–95.
4. N. K. Jaiswal, *Priority Queues*, New York, NY: Academic Press, 1968.
5. L. Kleinrock, *Queueing Systems, Vol. II: Computer Applications*, New York, NY: John Wiley and Sons, 1976.
6. H. Kobayashi and A. G. Konheim, "Queueing Models for Computer Communications System Analysis," IEEE Trans. Commun., *COM-25*, No. 1 (January 1977), pp. 2–29.


## APPENDIX

### *Derivation of the Waiting Time and Message Delay Moment-Generating Functions*

As we introduced in Section II, $W_{kj}$ is the steady-state in-queue waiting time for the $j$th packet in a message from source $k$, $k = 1, 2, \cdots, N$. Its Moment-Generating Function (MGF), defined as

$$G_{W_{kj}}(v) = E[e^{vW_{kj}}]$$

is derived in this appendix. From this result, the message delay MGF is easily obtained. The approach taken parallels in many respects the analysis given in Section III.

We begin the derivation by examining the duration of a busy period for sources 1 to $k-1$, denoted by $Y_k$. Such a busy period starting in a slot is initialized by one or more message arrivals from sources 1 to $k-1$ in the previous slot (which contains no packet from sources 1 to $k-1$). Let $A_k$ denote the total number of packets that arrive from sources 1 to $k-1$ in this previous slot. For the $i$th packet in this set, we define the "sub-busy" period $X_k(i)$ to consist of the duration of the "virtual" busy period (i.e., as if $i = A_k = 1$) initiated by the messages (if any) that arrive from sources 1 to $k-1$ while this $i$th packet is in service. In other words, we conceptually reorder the priorities so that each of the $A_k$ packets, and the sub-busy period it spawns, is served in turn. This does not change $Y_k$ and is a standard approach to busy-period analysis.

Due to the memoryless property of the arrival process, the sub-busy period random variables $X_k(i)$, $i = 1, 2, \cdots, A_k$, are independent and

identically distributed (*iid*). In addition, note that $Y_k$ has the same distribution as the generic random variable $X_k$, and satisifies the relation

$$Y_k = A_k + \sum_{i=1}^{A_k} X_k(i). \tag{15}$$

The Probability-Generating Function (PGF) for the discrete random variable $X_k$ is defined as

$$\Phi_{X_k}(z) = E[z^{X_k}] = \sum_{i=0}^{\infty} z^i \Pr[X_k = i].$$

Using (15) and the result that $X_k$ is distributed as $Y_k$, we obtain

$$\begin{aligned}\Phi_{X_k}(z) &= E[z^{Y_k}] \\ &= E\left[z^{A_k + \sum_{i=1}^{A_k} X_k(i)}\right] \\ &= E[(z\Phi_{X_k}(z))^{A_k}] \\ &= \Phi_{A_k}(z\Phi_{X_k}(z)),\end{aligned} \tag{16}$$

where $\Phi_{A_k}(z)$ is the PGF for the random variable $A_k$.

Now, $A_k$ is equal to the total number of packets arriving from sources 1 to $k-1$ in one time slot. Recall that each source $i$ generates messages as an independent Poisson process with rate $\lambda_i$; and each such message has its length selected independently from the distribution $P_{m_i}(\cdot)$, whose PGF we denote by $\Phi_{m_i}(z)$. It follows then that

$$\begin{aligned}\Phi_{A_k}(z) &= \prod_{i=1}^{k-1} \left\{ \sum_{r=0}^{\infty} \frac{\lambda_i^r e^{-\lambda_i}}{r!} \cdot [\Phi_{m_i}(z)]^r \right\} \\ &= \prod_{i=1}^{k-1} e^{\lambda_i[\Phi_{m_i}(z)-1]}.\end{aligned} \tag{17}$$

Hence, substituting (17) into (16), we obtain

$$\Phi_{X_k}(z) = \prod_{i=1}^{k-1} e^{\lambda_i[\Phi_{m_i}(z\Phi_{X_k}(z))-1]}. \tag{18}$$

As we shall see, $\Phi_{X_k}(z)$, the PGF for the duration of a busy period for sources 1 to $k-1$, plays an important role in the derivation of $G_{W_{kj}}(v)$, the waiting time MGF.

Returning to eq. (1) in Section III, we note that $W_{kj}$ is the sum of $W_{k1}$ and the $j-1$ *iid* random variables $w_{k1}, w_{k2}, \cdots, w_{k,j-1}$. Observe, however, that $w_{k\ell}$, $\ell = 1, 2, \cdots, j-1$, is distributed as $X_k + 1$. That is, $w_{k\ell}$ is composed of the service time for the $\ell$th packet plus the busy

period for sources 1 to $k-1$ initiated during this service time. In addition, it follows that the waiting time for the first packet in a source $k$ message, $W_{k1}$, is statistically independent of $w_{k\ell}$, $\ell = 1, 2, \cdots, j-1$. Hence we may write

$$G_{W_{kj}}(\nu) = G_{W_{k1}}(\nu) \cdot [e^{\nu} \Phi_{X_k}(e^{\nu})]^{j-1}. \tag{19}$$

This leaves us with having to determine the MGF for $W_{k1}$.

Let us for the moment consider the time-dependent behavior for the number of packets queued from sources 1 to $k$. For time slot $n$, we let $Q_k(n)$ denote the number of such packets queued just after the beginning of the slot and, to be consistent with our previous notation, we let $A_{k+1}(n)$ denote the number of packets that arrive from sources 1 to $k$ during the $n$th slot. It follows that

$$Q_k(n+1) = [Q_k(n) + A_{k+1}(n) - 1]^+, \tag{20}$$

where

$$[\epsilon]^+ = \begin{cases} \epsilon & \text{if } \epsilon \geq 0 \\ 0 & \text{if } \epsilon < 0. \end{cases}$$

From (20) we obtain the relation

$$E[z^{Q_k(n+1)}] = E[z^{[Q_k(n)+A_{k+1}(n)-1]^+}], \tag{21}$$

which may be rewritten as

$$\begin{aligned}
\Phi_{Q_k(n+1)}(z) &= E[z^{[Q_k(n)+A_{k+1}-1]^+}] \\
&= \Pr[Q_k(n) + A_{k+1} = 0] + \Pr[Q_k(n) + A_{k+1} > 0] \\
&\quad \cdot z^{-1} E[z^{Q_k(n)+A_{k+1}} \mid Q_k(n) + A_{k+1} > 0] \\
&= \Pr[Q_k(n) = 0]\Pr[A_{k+1} = 0] \\
&\quad + z^{-1} \sum_{i=1}^{\infty} z^i \Pr[Q_k(n) + A_{k+1} = i] \\
&= \Pr[Q_k(n) = 0]\Pr[A_{k+1} = 0] \\
&\quad + z^{-1}\{E[z^{Q_k(n)+A_{k+1}}] - \Pr[Q_k(n) = 0]\Pr[A_{k+1} = 0]\} \\
&= \Pr[Q_k(n) = 0]\Pr[A_{k+1} = 0](1 - z^{-1}) \\
&\quad + z^{-1} \Phi_{Q_k(n)}(z) \Phi_{A_{k+1}}(z),
\end{aligned} \tag{22}$$

where we have used the fact that $Q_k(n)$ and $A_{k+1}$ are statistically independent. Taking the limit as $n \to \infty$ on both sides of (22) (the limits exist for $\sigma_k < 1$) yields

$$\Phi_{Q_k}(z) = \Pr[Q_k = 0]\Pr[A_{k+1} = 0](1 - z^{-1}) + z^{-1} \Phi_{Q_k}(z) \Phi_{A_{k+1}}(z), \tag{23}$$

where $Q_k$ represents the steady-state number of packets queued from sources 1 to $k$ at the beginning of a slot. Rearranging the terms in (23), we obtain

$$\Phi_{Q_k}(z) = \frac{\Pr[Q_k = 0]\Pr[A_{k+1} = 0](z - 1)}{z - \Phi_{A_{k+1}}(z)}. \tag{24}$$

Taking the limit as $z \to 1$ on both sides of (24) yields

$$\Pr[Q_k = 0]\Pr[A_{k+1} = 0] = 1 - \frac{\partial}{\partial z}\Phi_{A_{k+1}}(z)\bigg|_{z=1}$$

$$= 1 - \sigma_k.$$

Hence, using this result and (17), (24) may be rewritten as

$$\Phi_{Q_k}(z) = \frac{(1 - \sigma_k)(z - 1)}{z - \prod_{i=1}^{k} e^{\lambda_i[\Phi_{m_i}(z) - 1]}}. \tag{25}$$

Now consider the end of the time slot during which a source $k$ message is generated. The number of packets of higher or equal priority that are queued and must be transmitted before the first packet in this source $k$ message is given by

$$Q_k + A_k + B_k,$$

where $Q_k$ is the number of queued packets from sources 1 to $k$ just after the beginning of the slot, $A_k$ is the number of packets from sources 1 to $k - 1$ that arrive during the slot, and the new random variable, $B_k$, represents the number of packets from source $k$ that arrive during the slot prior to the generation of the source $k$ message in question. The $i$th packet in this set of $(Q_k + A_k + B_k)$ packets initiates a sub-busy period of duration $X_k(i)$. Hence we may write

$$W_{k1} = U + \sum_{i=0}^{(Q_k+A_k+B_k)} [1 + X_k(i)], \tag{26}$$

where $U$ is a random variable, uniformly distributed over one slot time, that represents the time from when the source $k$ message is generated until the start of the next slot.

From (26) we may write

$$E[e^{\nu W_{k1}} \mid U = u,\ Q_k = q_k,\ A_k = a_k,\ B_k = b_k] = e^{\nu u}[e^{\nu}\Phi_{X_k}(e^{\nu})]^{(q_k+a_k+b_k)}.$$

Removing the conditioning on the independent random varaibles $Q_k$ and $A_k$ yields

$$E[\alpha^{W_{k1}} \mid U = u,\ B_k = b_k]$$

$$= \alpha^{u}[\alpha\Phi_{X_k}(\alpha)]^{b_k}\Phi_{Q_k}(\alpha\Phi_{X_k}(\alpha))\Phi_{A_k}(\alpha\Phi_{X_k}(\alpha)), \tag{27}$$

where, for simplicity, we have substituted $\alpha$ for $e^{\nu}$. Now, using the same approach as we did with $A_k$, we obtain

$$E[z^{B_k} \mid U = u] = e^{\lambda_k(1-u)[\Phi_{m_k}(z)-1]}.$$

Thus, removing the conditioning on $B_k$ in (27) yields

$$E[\alpha^{W_{k1}} \mid U = u] = \alpha^u e^{\lambda_k(1-u)[\Phi_{m_k}(\alpha\Phi_{X_k}(\alpha))-1]} \cdot \Phi_{Q_k}(\alpha\Phi_{X_k}(\alpha))\Phi_{A_k}(\alpha\Phi_{X_k}(\alpha))$$

$$= [e^{\nu-\lambda_k[\Phi_{m_k}(\alpha\Phi_{X_k}(\alpha))-1]}]^u \Phi_{Q_k}(\alpha\Phi_{X_k}(\alpha))\Phi_{A_{k+1}}(\alpha\Phi_{X_k}(\alpha)).$$

Now, removing the conditioning on $U$, we obtain

$$G_{W_{k1}}(\nu) = G_U\{\nu - \lambda_k[\Phi_{m_k}(\alpha\Phi_{X_k}(\alpha)) - 1]\}\Phi_{Q_k}(\alpha\Phi_{X_k}(\alpha))\Phi_{A_{k+1}}(\alpha\Phi_{X_k}(\alpha)),$$

where

$$G_U(\nu) = \int_0^1 e^{\nu u} du = \frac{1}{\nu}[e^{\nu} - 1]. \tag{28}$$

Finally using (19), we obtain

$$G_{W_{kj}}(\nu) = G_U\{\nu - \lambda_k[\Phi_{m_k}(\alpha\Phi_{X_k}(\alpha)) - 1]\}$$
$$\cdot \Phi_{Q_k}(\alpha\Phi_{X_k}(\alpha))\Phi_{A_{k+1}}(\alpha\Phi_{X_k}(\alpha)) \cdot [\alpha\Phi_{X_k}(\alpha)]^{j-1},$$

where $\alpha = e^{\nu}$, $G_U(\nu)$ is given by (28), $\Phi_{Q_k}(z)$ is given by (25), $\Phi_{A_{k+1}}(z)$ is given by (17), and $\Phi_{X_k}(z)$ is given by (18).

The delay in transmitting a source $k$ message of length $m$, $D_k(m)$, is given by

$$D_k(m) = W_{km} + 1.$$

Hence, the MGF for $D_k(m)$ is given by

$$G_{D_k(m)}(\nu) = e^{\nu} G_{W_{km}}(\nu).$$

It follows that the MGF for $D_k$, the delay in transmitting a randomly selected source $k$ message, is given by

$$G_{D_k}(\nu) = G_{W_{k1}}(\nu)\Phi_{m_k}(e^{\nu}\Phi_{X_k}(e^{\nu}))/\Phi_{X_k}(e^{\nu}).$$

## AUTHORS

**Robert R. Boorstyn,** B.E.E., 1958, City College of New York; M.S. (Electrical Engineering), and Ph.D. (Electrical Engineering), 1963 and 1966, respectively, Polytechnic Institute of Brooklyn; Sperry Gyroscope Company, 1958–1961; Polytechnic Institute of New York, 1961—; on leave at Bell Laboratories, 1977–1978; consultant at Bell Laboratories, 1981–1982. Mr. Boorstyn is engaged in research on computer communication networks, specializing in packet radio networks, routing, network design, and analysis. He is currently a Professor of Electrical Engineering and Computer Science at Polytechnic Institute of New York. He has been Editor for Computer Communications of the *IEEE Transactions on Communications*, Chairman of the Computer Com-

munications Committee of the IEEE Communications Society, and Secretary of the IEEE Information Theory Group. He is an Associate Editor of *Networks Journal.*

**Michael G. Hluchyj,** B.S. (Electrical Engineering), 1976, University of Massachusetts at Amherst; S.M., E.E., and Ph.D. (Electrical Engineering), Massachusetts Institute of Technology, 1978, 1978, and 1981, respectively; Bell Laboratories, 1981–1982; AT&T Information Systems, 1983—. Mr. Hluchyj's work at Bell Laboratories and AT&T Information Systems has centered around the architectural design and performance analysis of local area data communication networks.

**Chan David Tsao,** B.S. (Control Engineering), 1973, National Chiao Tung University (Taiwan); M.S. (Systems Engineering), 1978, Florida Institute of Technology; M.S. and Ph.D. (Electrical Engineering), Polytechnic Institute of New York, 1979 and 1982, respectively; Bell Laboratories, 1981–1982; AT&T Information Systems, 1983—. At Bell Laboratories, Mr. Tsao worked on performance evaluation and system design for local area networks. His current research interest is architectures and applications for local area networks.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Prologue

By R. E. MARKLE*

(Manuscript received January 27, 1983)

The AR6A Single-Sideband Amplitude Modulation (SSBAM) Transmission System makes very efficient use of the radio spectrum by transmitting 6000 message circuits in a 29.65-MHz radio channel. This capability is achieved by maintaining a tighter frequency spacing of the multiplexed mastergroup signals and by discarding redundant sideband signals before transmission over the radio path. The AR6A System can provide a maximum capacity of 42,000 message circuits for the 6-GHz portion of the radio network when all eight channels are utilized. This capability, combined with the alternate use of the 4-GHz spectrum for message or digital circuits, provides the proper mix of network capability choices. This introduction gives an overview of the AR6A System and a brief history of single-sideband evolvement in the Bell System. It also is a prologue to the papers in this volume, which describe some of the specifics of the system design.

## I. INTRODUCTION

On January 12, 1981, the first AR6A† Microwave Radio System was placed in service between Hillsboro, Missouri, and La Cygne, Kansas. This event marked the first application of single-sideband transmis-

* Bell Laboratories.

† The AR6A designation represents Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

sion to the Bell System long-haul microwave radio network. The AR6A System is designed for use in the 6-GHz common carrier band and can provide a route capacity of 42,000 high-quality, 4-kHz, two-way message circuits when all radio channels in the 6-GHz band have been equipped (i.e., one protection and seven service channels). If TD radio operating in the 4-GHz band is used on the same route at 1800 message circuits per radio channel, the combined route capacity is an impressive 61,800 message circuits.

## II. EVOLUTION OF LONG-HAUL RADIO

The evolution of the Bell System long-haul radio network is outlined in Table I; in 31 years the radio channel capacity has been increased by a factor of 12.5 and the route capacity by a factor of 25.75. The ubiquitous TD-2 System was placed in service in 1950 and had a channel capacity of 480 message circuits with a maximum capability of five service channels in the 4-GHz band. In 1954 the circuit loading was increased to 600 message circuits per radio channel, and in 1959 the route capacity was further extended through the use of interstitial radio channels to provide a total of ten service channels. The deployment of TH-1 in the 6-GHz band in 1961 expanded the route capacity by 10,800 circuits to a total of 16,800 message circuits; the advent of

Table I—Evolution of long-haul radio capacities

| System/Channel Capacity | Initial Availability | Protection Configuration (Protection × Service) | Route Capacity in Message Circuits |
|---|---|---|---|
| TD/480 | 1950 | 1 × 5* | 2400 |
| TD/600 | 1953 | 1 × 5 | 3000 |
| TD/600 | 1959 | 2 × 10 | 6000 |
| TD/600 + TH/1800 | 1961 | 2 × 10<br>2 × 6 | 16,800 |
| TD/1200 + TH/1800 | 1968 | 2 × 10<br>2 × 6 | 22,800 |
| TD/1500† + TH/1800† | 1973 | 2 × 18 | 28,800 |
| TD/1800†,‡ + TH/2400† | 1979 | 2 × 18 | 36,000 |
| TD/1800†,‡ + AR 6/6000 | 1981 | 1 × 11<br>1 × 7 | 61,800 |

* Automatic protection on the basis of one protection channel for five service channels (1 × 5) was not available until 1953; five service channels are used here for consistency in comparing route capacities.
† Each channel also can be equipped with a 1.544-Mb/s system.
‡ The channel alternately can be used to provide 20-Mb/s capability or (beginning in 1983) 45-Mb/s capability.

TD-3 in 1968, with a capacity of 1200 circuits per radio channel, increased the route capacity by an additional 6000 circuits. By 1973, the use of crossband protection switching systems and 1500-circuit loading on TD provided a radio route capacity of 28,800 circuits. Six years later, the circuit loadings of both TD and TH had been increased to 1800 and 2400 circuits, respectively, to give a capability of 36,000 circuits for a fully loaded radio route.

In parallel, the radio systems have grown in digital transmission capability to match network needs. In 1974 data-under-voice capability of 1.544 Mb/s was added to each FM radio channel. The alternate use of a TD-2 radio channel to provide 20-Mb/s capacity was first available in 1979, and an upgrade to 45-Mb/s capacity was provided in 1983.

Even with this capacity some radio routes were expected to be filled by the mid-1980s, and meeting additional growth needs would have required building parallel radio routes or providing some other new facility for relief. It was expected that some of the new radio routes would be difficult to engineer because of radio frequency interference encountered at radio station junctions in the existing radio network. The availability of AR6A in 1981 increased the long-haul radio route capacity to 61,800 circuits and postponed the time when new radio routes would need to be constructed, which in turn postponed the initial capital investment required for land, buildings, towers, and associated station equipment.

### III. AR6A SYSTEM FEATURES

The AR6A System allows for orderly growth in the existing radio network, since it uses the same frequency plan as TH-3 and can be used for conversion of TH routes as well as for additions to existing TD routes. The radio equipment is designed to be installed in existing radio stations and shares the use of the radio towers, antennas, common waveguide runs, power plant, and alarm telemetry systems. Additional space-diversity antennas are required over those used on existing FM routes because the single-sideband signal is more susceptible to the effects of multipath fading.

The AR6A System is designed to meet a two-way outage objective of 0.02 percent and a radio channel dynamic-amplitude-misalignment objective of 2 dB or less for 99.9 percent of the time. For a 4000-mile route length, the noise objective is 40 dBrnc0. The system meets these objectives by the appropriate use of space-diversity reception and dynamic equalization in conjunction with a single-channel frequency-diversity protection system.

In meeting intermodulation noise objectives the multiplex, protection, and radio equipment were designed to linearity requirements far more stringent than those of previous radio systems. Of particular

note is the AR6A transmitter-receiver (TR)* unit that has a linearity that is more than 30 dB better than the comparable TH-3 TR unit. This is achieved by the use of ultralinear semiconductor devices in the intermediate frequency circuits and predistortion in the transmitter to improve the linearity of the frequency converter and traveling-wave-tube amplifier combination.

The frequency stability required for AR6A is much tighter than that for FM systems (typically by a factor of about 25). This stability allows equalizer pilots to be selected by narrowband crystal filters and reduces the frequency range over which the receiving multiplex terminal must operate.

The AR6A System uses microprocessor-controlled maintenance and surveillance methods to monitor system performance and to isolate troubles to a specific station on a radio route. Automated testing is performed under the control of a central minicomputer at access points provided in both the multiplex and the protection switching equipment. Both in-service and out-of-service testing capabilities are provided with special stress-testing features implemented at radio repeater locations for use in trouble isolation and diagnosis. The computer-controlled measurement system is designated "Transmission Surveillance System—Radio" and is described in detail in this issue of the *Journal.*

## IV. SINGLE-SIDEBAND HISTORY

In the late 1960s, detailed system studies were performed by A. J. Giger to define linearity requirements for radio repeaters (TR units) for long-haul single-sideband transmission. Several methods were recommended for achieving the necessary transmitter performance. Subsequently, experiments were performed in the early 1970s with feedforward microwave techniques to improve the linearity of traveling-wave-tube amplifiers, the most difficult challenge in achieving the radio transmitter intermodulation noise requirement. The success of these experiments led to addressing the problems of overall repeater linearity, equalization for multipath fading effects, and the effects of interference when used in the existing FM radio network.

Studies were performed to obtain a better understanding of traveling-wave-tube distortion, and this resulted in the use of predistortion to meet transmitter linearity requirements. Additional field data were gathered on multipath fading, and these data were a key ingredient in achieving a simplified dynamic equalization design. Supporting studies of a subjective nature also were conducted to provide more detailed

* Acronyms and abbreviations used in the text are defined at the back of this *Journal.*

definitions of tone and crosstalk annoyance objectives when Single-Sideband Amplitude Modulation (SSBAM) was used in the radio network; the frequency control plan and the intermediate frequency (IF) filter design were chosen to meet these objectives. The adequacy of the methods for linearization, equalization, and interference control were established by a field test performed in the 4-GHz network in 1974 and early 1975.

Detailed system design of AR6A for the 6-GHz network started in 1975, and in 1979 a complete system was installed on a 6-hop radio route from Hillsboro, Missouri, to Windsor, Missouri. All subsystems of AR6A were tested and evaluated on this 177-mile route, and system verification testing was completed on June 14, 1980. This route then was extended by three radio hops (71 miles) to La Cygne, Kansas, and became the first single-sideband amplitude modulation route to be placed in commercial service on January 12, 1981.

## V. ACKNOWLEDGMENTS

The feasibility demonstration of SSBAM transmission on the long-haul 4-GHz radio network in 1974–1975 was the culmination of the efforts of many people in different organizations throughout Bell Laboratories. The realization of AR6A for the 6-GHz network was the result of the dedicated efforts of many organizations at Bell Laboratories working with Western Electric and in close concert with AT&T Communications. Technology foundations were provided by the TD, TH, and L5 Systems and these were augmented by advances in filter, network, and device designs, as well as the effective use of the many computational and measurement facilities available at Bell Laboratories. The authors of this volume express their appreciation to all those who have contributed to the ultimate attainment of SSBAM transmission on the long-haul radio network.

## AUTHOR

**Raymond E. Markle,** B.S., 1952 (Electrical Engineering), Pennsylvania State University; M.S., 1959 (Electrical Engineering), Newark College of Engineering; Bell Laboratories, 1952—. Since joining Bell Laboratories, Mr. Markle has worked in such areas as radar design, submarine cable communications, and single-sideband and digital microwave radio. He is presently Head of the Radio TR Systems Department. Member, IEEE-SM.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# System Design and Performance

by J. GAMMIE,* J. P. MOFFATT,* R. H. MOSELEY,* and W. A. ROBINSON*

(Manuscript received February 14, 1983)

This paper describes the overall architecture of the first long-distance microwave telephone transmission system to use Single-Sideband Amplitude Modulation in the microwave transmission path. Compared with commonly used frequency modulation systems, the AR6A Single-Sideband Microwave Radio System more than doubles the number of telephone circuits that can be carried per unit of microwave bandwidth. Included in the description are the system design model, performance objectives, and the allocation of performance impairments. Also discussed are the computer-controlled automated maintenance features and the results of laboratory and field tests.

## I. GENERAL DESCRIPTION

### *1.1 Introduction*

The AR6A† System is the first long-haul microwave transmission facility to employ Single-Sideband Amplitude Modulation (SSBAM)‡ in the microwave transmission path. To fully exploit this alternative to the widely used frequency-modulation (FM) system, it has been necessary to develop for SSBAM new subsystems to accomplish the

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

‡ Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

functions commonly associated with an integrated microwave transmission facility. In this sense, the use of SSBAM affects not only the design of the transmitter-receiver (TR) equipment but all ancillary equipment as well.

Broadly speaking, the main functional elements of the system are fairly conventional, but a number of SSBAM's basic requirements have had a major impact on the AR6A System design. The first and most obvious of these is the high level of linearity needed in many of the transmission elements to control intermodulation distortion. Although the most taxing task was making the transmitting traveling-wave-tube amplifier meet overall system linearity objectives, the intermodulation performance of many other elements required careful scrutiny and improvement. Frequently, the more stringent linearity requirements have been associated with higher quiescent current for transistors and, in general, with increased power requirements relative to comparable elements in an FM system. Nevertheless, when power requirements are normalized to a per-circuit or a per-mastergroup basis, power utilization efficiencies are very competitive.

In addition, the direct dependence of signal amplitude response on the response of the transmission medium is particularly stressing during multipath fading. This requires adding a capability for dynamic amplitude equalization. Design of the necessary adaptive equalizers required an extensive experimental effort aimed at quantitatively describing the statistics for and understanding the effects of multipath fading on channel amplitude shape. The dynamic dispersive channel resulting from multipath fading required not only adaptive equalizers but also the use of space diversity.

Unlike FM, in which demodulation is performed relative to a transmitted FM carrier, accumulated frequency shifts due to multiple modulation and remodulation at repeaters must be removed. This adds to the complexity of the terminal multiplex equipment that performs the frequency-correction function.

Finally, successful transmission of SSBAM requires new maintenance and equalization considerations. In FM systems, minor amounts of transmission slope and absolute amplitude changes do not greatly affect the quality of the recovered baseband signal. The greater sensitivity of SSBAM to these effects, particularly their potentially cumulative nature, requires much closer control of equalization and absolute amplitude levels. These considerations, combined with the trend towards centralized operational support systems, resulted in the integrated development of a minicomputer-based Transmission Surveillance System—Radio (TSS-R) with associated trouble isolation and stress test features.

During development, effort was initially focused on applications

employing frequency-diversity protection, the most commonly used system arrangement. This work was later supplemented by the development of hot-standby and space-diversity arrangements for applications where lower cross-sectional growth rates preclude the use of frequency-diversity protection.

Because of the many new system features, AR6A underwent an early but limited field trial between Ashburnham and Wendell, Massachusetts, in the period October 1977 to July 1978. This two-hop installation provided valuable initial field experience prior to a more comprehensive field evaluation between Hillsboro and Windsor, Missouri. The Missouri evaluation involving a six-hop radio section with frequency- and space-diversity protection switching encompassed the period from April 1979 to June 1980. The evaluation of hot-standby arrangements was performed later on a two-hop section between Colorado Springs and Cedarwood, Colorado. The shipment of standard AR6A equipment began in June 1980; first commercial service was placed on the AR6A route between Hillsboro, Missouri, and La Cygne, Kansas, in January 1981.

The following paragraphs are devoted to an overview of the AR6A System using the block diagram of Fig. 1. The figure and discussion relate to a system configuration using frequency-diversity protection switching which, besides being the most common application, will provide a satisfactory medium for describing the main features of the system.

### *1.2 Multiplex*

#### *1.2.1 Mastergroup translators*

The AR6A System interfaces with the standard Bell System analog multiplex hierarchy at the basic U600 mastergroup level. This is also the highest level in the multiplex hierarchy at which AR6A can be interconnected to other broadband transmission facilities. The U600 mastergroup contains 600 4-kHz circuits in the frequency band 564 to 3084 kHz. Figure 1 illustrates how the basic mastergroup is delivered from the Mastergroup Distributing Frame (MGDF) to the first frequency translation step provided by the Mastergroup Translator, Type B (MGTB). Five MGTBs translate and frequency multiplex five separate mastergroup signals into adjacent positions in a multimastergroup band extending from 8.628 to 21.900 MHz. With a fixed intermastergroup spacing of 168 kHz, the tight packing provided by the MGTB is an important factor in being able to place two multimastergroups (ten mastergroups) in each 30-MHz broadband radio channel.

In addition to modulation and demodulation, the MGTB modems supply the 2840-kHz mastergroup pilot which is used in the receiving

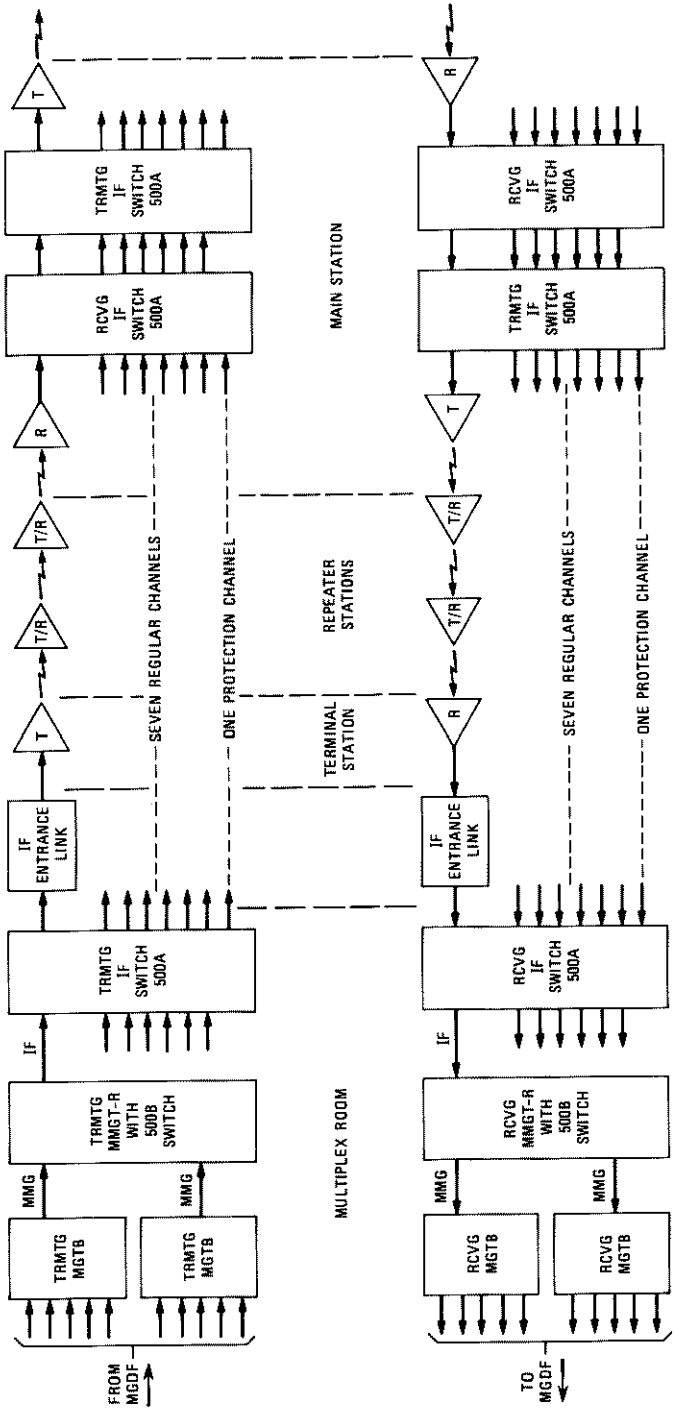



Fig. 1—Block diagram of AR6A System.

portion of the modem for gain regulation purposes. Other functions associated with the MGTB combining circuits are (1) means for inserting a 13.920-MHz multimastergroup continuity pilot for sectionalizing troubles, (2) a comb filter with narrowband band-stop sections which remove signals and noise from intermastergroup slots prior to the insertion of line pilots, and (3) test access points for TSS-R and other automated measuring facilities.

#### *1.2.2 Multimastergroup translators*

The Multimastergroup Translator—Radio (MMGT-R) performs the next modulation step by translating two multimastergroups into the intermediate frequency (IF) band centered at 74.13 MHz. The disposition of the multimastergroups and the previously described translation steps from basic mastergroup are illustrated in Fig. 2.

In the MMGT-R, the two multimastergroups are translated to IF by separate modulators with independent, unsynchronized, crystal-controlled local oscillators. On the receiving side, the demodulating carriers are supplied by Voltage-Controlled Oscillators (VCOs). Each VCO tracks the accumulated line frequency errors with a Phase-Locked Loop (PLL) to deliver a frequency-corrected multimastergroup signal at the receiving end of the terminal section. Essential to the frequency-correcting process is the addition of an intermastergroup recovery pilot to each multimastergroup signal before it is translated to IF. This pilot at 16.608 MHz is frequency locked to a 512-kHz synchronizing signal derived from the Bell System Reference Frequency (BSRF). In view of the recovery pilot's importance, the synchronizing source supplied to the MMGT-R is duplicated and automatically protected. At the output of the receiving MMGT-R, the demodulated recovery pilot is compared with a locally generated reference frequency to provide the PLL error signal for frequency correction. An important additional function of the receiving PLL is to reduce accumulated phase jitter on the radio line.

In addition to recovery pilots, the transmitting portion of the MMGT-R supplies three radio-line pilots from free-running crystal oscillators. When translated to IF these pilots are located near the center and band edges of the radio channel, as illustrated in Fig. 2. At radio receivers, the pilots provide a reference signal for Automatic Gain Control (AGC) and, in addition, provide information on channel amplitude shape for dynamic amplitude equalization. The radio pilots, recovery pilots, and multimastergroup pilots are each transmitted at a level of −10 dBm0.

#### *1.2.3 Office master frequency supply*

Carrier supplies for the Bell System analog multiplex hierarchy are

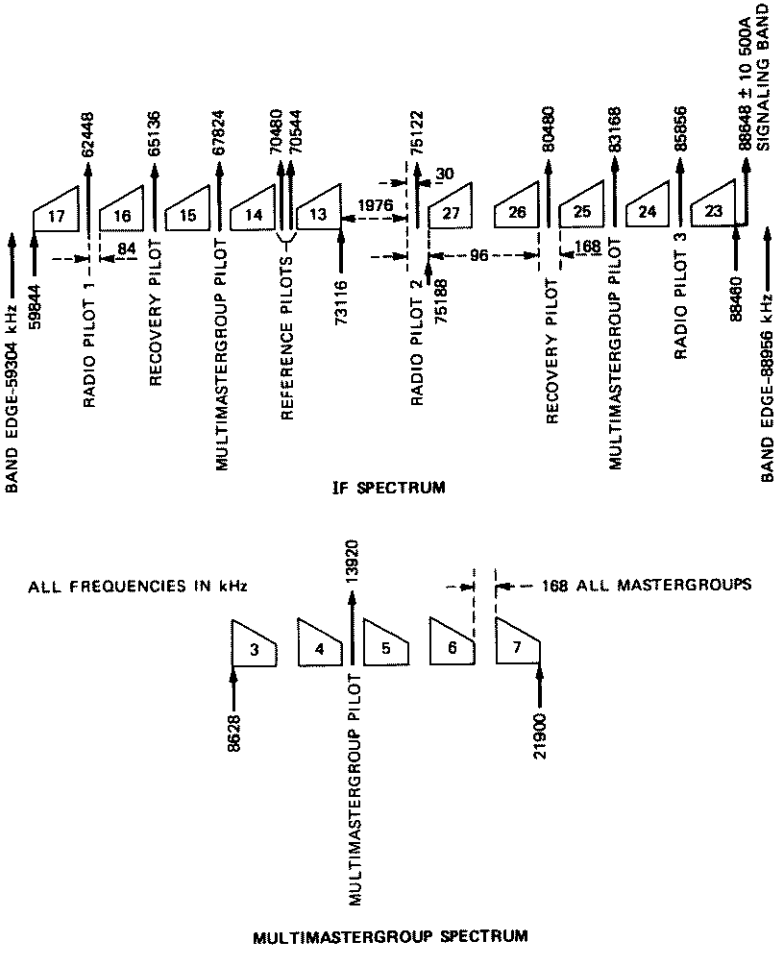



Fig. 2—Multimastergroup and IF spectrum.

locked to a 64- or 512-kHz synchronizing source derived from an office Primary Frequency Supply (PFS). The PFS is in turn synchronized to a signal with a frequency accuracy that is traceable within very narrow limits to that of the BSRF source at Hillsboro, Missouri. If the PFS temporarily loses its input synchronizing signal, the output frequency will suddenly change to the unit's natural or rest frequency. This frequency transient, although tolerable at lower levels in the multiplex hierarchy, was considered excessive in a synchronizing source for the AR6A recovery pilots. For this reason, a new Office Master Frequency Supply (OMFS) was developed which provides 64- and 512-kHz synchronizing signals that are quasi-frequency locked to the 2.048-MHz BSRF. With quasi-frequency lock, instead of being

synchronized to the incoming reference signal, the highly stable OMFS oscillators are free-running. Their frequency is compared with the reference frequency at intervals and smoothly updated if the error exceeds the prescribed threshold. If the reference frequency is temporarily lost, updating is inhibited and the oscillators continue to run at their most recently updated frequency. In this way the OMFS shields dependent multiplex equipment from transient effects on the synchronizing network.

#### *1.2.4 Reference frequency transmission*

The 2.048-MHz Bell System Reference Frequency must be available at all AR6A terminal stations for the purposes described in Section 1.2.3. The reference signal can be made available over a variety of transmission facilities, but there are instances where the only available transmission medium will be the AR6A System. To provide this capability, an optional reference frequency sending unit is available that inserts two pilots between MG13 and MG14 (Fig. 2) at a level of −15 dBm0. The pilot frequencies differ by 64 kHz with an accuracy determined by the sending end BSRF. At the receiving end of the terminal section, a reference frequency receiving unit recreates a BSRF at 2.048 MHz from the accurately defined 64-kHz difference frequency. In this way, reference frequency transmission accuracy is unaffected by inaccuracies in the multiple frequency translations experienced on the radio line. Sixty-four-kHz rather than a wider frequency separation was chosen to minimize transmission dispersion effects and to optimize pilot positions from the standpoint of noise and freedom from interaction with other system pilots. The sending and receiving frequency units are physically associated with the MMGT-R equipment.

### *1.3 The radio line*

#### *1.3.1 Entrance links*

Figure 1 shows that the IF signal from the MMGT-R is fed to the radio transmitting equipment via the 500A Protection Switching System, which will be discussed in Section 1.4. Since the radio transmitters may be at some physical distance from the preceding multiplex equipment, provision must be made for handling trunk losses and associated transmission shapes. In some instances, the trunks may be so long that they require amplification, necessitating automatic protection of the active circuits. In the AR6A System, the 500A protection switching equipment is located near the MGTB and MMGT-R multiplex equipment. In this way, the trunks or entrance links become extensions of the radio line and may, therefore, share the 500A protection facilities.

For distances up to 600 feet, the entrance link trunks are passive

and are equalized to within 25 feet. Greater distances require amplification by amplifiers located in the TR bays for both the transmitting and receiving trunks. This provides for entrance link distances of up to 1200 feet using 728-type office coaxial cable. Entrance link distances up to 3600 feet can be accommodated with 0.375-inch coaxial cable.

#### *1.3.2 Transmitter-receiver equipment*

The heterodyne-type transmitter-receiver (TR) bays are designed for overbuilding on existing 4-GHz radio routes with an average repeater spacing of 25.6 miles. The nominal repeater gain is 61.7 dB with up-fade and down-fade ranges of 15 and 46 dB, respectively. The full-load transmitter output power is +24.6 dBm, reflecting an average power per 4-kHz message circuit of −19.6 dBm0.

The TR bay equipment is designed to permit field installation by technicians into existing bay frameworks. This modular installation allows the bay frameworks to be installed and the station wiring to be done for all radio channels at one time. The TR modules are installed as required by the route traffic.

The transmitter uses IF predistortion to compensate for the nonlinearities of the up converter and the output traveling-wave-tube amplifier. Automatic gain control (AGC) at each receiver is based on maintaining the voltage average of the three pilots constant at the receiver output. In addition to AGC, the last receiver in each switch section is equipped with a dynamic amplitude equalizer.

A microwave preamplifier with a noise figure of approximately 3 dB is provided in the common receiving waveguide run on each polarization. If the hop length is less than 63 percent of nominal, the preamplifier is omitted.

To meet long-haul reliability objectives within the FCC-prescribed limit of only one frequency-diversity protection channel to serve up to seven service channels, it is estimated that approximately half the radio hops will require space diversity. The TR bay is, therefore, equipped with an optional space-diversity switch that is actuated by a 36-dB fade in any one of the three radio-line pilots.

In the event of a loss of received signal due to equipment trouble or other reasons, a pilot resupply signal is automatically inserted at the receiver output. In addition to supplying the three radio-line pilots, the resupply source provides a fourth pilot that is located in the center noise-monitoring slot of the 500A Protection Switching System. The channel with resupply is thereby marked noisy and unavailable for service. The pilot resupply signal at each station is provided from a common source and distribution network located in a radio support bay.

Networks in the IF shelf provide repeater bandpass shaping and

basic equalization for the nominal repeater transmission shape. Also, positions are provided for limited amounts of mop-up equalization for nonsystematic amplitude response variations.

Carrier supplies for up and down frequency conversion are provided by the TR bay microwave generator and its associated 252-MHz shift oscillator. To maintain pilot frequencies within the passband of the narrowband pilot pick-off filters at each repeater, the carrier frequencies must be accurate to better than one part in $10^7$. This is achieved by locking the microwave generators and shift oscillators to a common submultiple frequency, 308.87354 kHz. This synchronizing frequency is distributed to each TR bay from a common synchronizing source located in the radio support bay. If the synchronizing signal fails, both the microwave generator and shift oscillator have memory circuits that maintain their frequencies at the most recently updated values. Subsequent drifting is at a rate that allows a one- to two-day window for repairing the failed synchronizing source.

#### *1.3.3 Support bay*

The support bay contains the Microwave Carrier Synchronization Supply (MCSS) and the pilot resupply source, along with associated distributing circuits. In addition, it provides power distribution, fusing, and alarm arrangements for up to eight microwave preamplifiers. The support bay is designed to supply up to 16 TR bays corresponding to one fully loaded route passing through a repeater station. In hot-standby applications, the radio support bay also contains the hot-standby control unit.

#### *1.3.4 Antenna system*

Since a common application of AR6A will involve overbuilding on existing 4-GHz radio routes, the system is designed around the currently used horn antenna system with associated system combining and separation networks. The space-diversity path should essentially duplicate the main path for a number of reasons, including the anticipated later application of space-diversity combining. With nominal tower heights and hop lengths, the combined section loss, waveguide losses, network losses, and antenna gains give a net overall nominal section loss of 61.7 dB.

### *1.4 Protection switching*

#### *1.4.1 500B Switching System*

Since each MMGT-R carries 6000 message circuits, automatic transfer of the message load to a hot-standby protection unit is necessary for reliability and maintenance. The function is handled by the 500B System. One microprocessor-based controller oversees a 1 ×

14 protection arrangement for transmitting MMGT-Rs and a corresponding 1 × 14 protection arrangement for receiving MMGT-Rs. Since the protection function is local, multimastergroup translators from more than one route may be associated within the same 1 × 14 switching system.

In its automatic mode, a switch is initiated by a number of conditions including loss of the multimastergroup continuity pilot and loss of synchronization.

### *1.4.2 500A Protection Switching System*

The 500A System provides frequency-diversity protection switching on the radio line. Switching sections may be up to ten hops in length and the microprocessor-based control functions are located at the receiving end. The protection switching systems in the two directions of transmission are electrically independent.

Each protection channel can serve up to seven working channels. A frequency shift-keyed (FSK) tone transmitted over the oppositely directed radio channels signals between the receiving and head end of the switch section. The FSK tone is inserted at IF near the upper channel edge and is blocked at the receiving end of each switch section.

Switch initiators monitor each of the three radio-line pilots as well as the noise in three narrowband slots adjacent to the pilots. This multiple monitoring causes more switching activity than the single-point monitoring on FM systems and is partly responsible for the approximately 50-percent space-diversity requirement for AR6A. In addition to protecting against frequency-selective fading, particularly on hops without space diversity, 500A protects against equipment failure and provides out-of-service access to the radio channels for maintenance purposes.

Maintenance access to a radio channel via a maintenance switch command to the 500A System is an essential element in the implementation of the centralized Transmission Surveillance System—Radio (TSS-R). Besides providing switched access to a channel at the transmit and receive ends of the switch section, the 500A control logic is responsible for ensuring that accessing the channel is subordinate to maintaining service continuity. Even after out-of-service access to a channel is permitted, a preemption feature in the 500A logic will restore the channel to service automatically if this is necessary to prevent a service outage. When a channel is opened at the head end of the switch section to allow maintenance access, a fresh set of locally generated radio pilots must be inserted to provide proper control at subsequent radio receivers. These pilots are provided by the 500A System in the same way as resupply pilots are provided at repeaters.

### *1.5 Hot standby*

On routes where the cross-sectional growth rate is low, current FCC rules do not permit the use of frequency-diversity protection channels. In these circumstances, protection against outage due to multipath fading is generally provided by using space diversity. To protect against radio equipment outage, space diversity must be supplemented with hot-standby radio equipment that is automatically switched into service when the need arises. This hot-standby configuration is available on AR6A although the number of applications is expected to be small compared to the frequency-diversity configuration.

Since radio repeaters are frequency-dependent, each working TR bay generally has a separate hot-standby bay. In this sense, hot-standby configurations are equipment-intensive. To ease this situation in AR6A, the east-west and west-east repeaters on a given radio channel are identical, including the frequency-dependent components. By providing appropriate switching and control arrangements, a common TR bay can be used to protect equipment for both directions of transmission. Main stations do not lend themselves to this reduced configuration and are, therefore, arranged with conventional one-for-one hot standby.

Hot-standby manual and automatic control functions at repeaters are provided by a microprocessor-based control unit. The same unit with a subset of features is used at main stations to control hot-standby switching of transmitters. Main station receivers are protected by the 501A Protection Switching System, which also has a double feed to the regular and hot-standby transmitters. In addition to providing noise and pilot-level monitoring with 500A-type initiators, the 501A System also enables the TSS-R System to monitor in service.

On a hot-standby route, each repeater has its own protection against equipment outage and multipath fading. There is, therefore, no natural equivalent to the multihop switch section of the frequency-diversity configuration. However, since hot-standby routes will often be converted to frequency diversity when growth permits, main station TR bay and waveguide configurations are recommended at those stations which at a later time may become switching or main stations. Since the main station configuration implies the use of the 501A switch with surveillance access, this sectionalizing of hot-standby routes into surveillance sections subdivides the route naturally for purposes of dynamic equalization, transmission monitoring, and trouble isolation. The interface transmission levels on the 501A System are identical to those on the 500A switch to further simplify the transition from hot standby to frequency diversity.

### *1.6 Maintenance*

Compared with FM systems, the somewhat less rugged modulation

technique and the higher capacity of the AR6A System have emphasized the need for careful maintenance planning in the overall system design. Supplementing these considerations is the ongoing trend towards a demand maintenance scenario, improved trouble isolation procedures, and centralized transmission surveillance capabilities.

On AR6A, status reporting and command functions are handled conventionally utilizing the C1 or E2 systems in conjunction with the Surveillance and Control of Transmission Systems (SCOTS) or Telecommunication Alarm Surveillance and Control (TASC) central processors. However, in the 500A-to-E2 interface, which can involve a sizeable number of wire pairs, optional arrangements have been developed to provide a serial interface with only two wired pairs. This feature capitalizes on the 500A microprocessor that can communicate serially to microprocessor-based alarm remotes such as E2A.

For trouble-isolation purposes, a significant new feature is the ability to break the transmission path and insert resupply pilots at any selected station by remote command. This is supplemented by the capability to remotely modify the resupply signal to perform a variety of stress tests. With this feature, in conjunction with a measuring capability at the receiving end of a switch section, a variety of transmission troubles can be localized by making out-of-service measurements on successively shortened portions of the switch section. Of course, these capabilities must be circumscribed with protections against service interruption and loss of the 500A preemption feature. These matters are discussed more fully in Section 4.9.

Another important new maintenance feature is the Transmission Surveillance System—Radio (TSS-R). This system uses programmable measuring, signal source, and access equipment at radio main stations that are under the control of a local microprocessor. This microprocessor is, in turn, accessible and controlled via the Direct Distance Dialing (DDD) network from a central minicomputer. Data link dial-up and communication protocol, measuring, evaluation, and reporting routines residing in the central software provide for a variety of switch section transmission measurements that can be made either on an in-service or out-of-service basis. Furthermore, by allowing the TSS-R central minicomputer to talk to a SCOTS or TASC central, the trouble-isolation and stress test features described in the preceding paragraph can be integrated into automated programmable procedures. It should again be noted that access to a radio channel for in-service or out-of-service measurements is subject to approval by the 500A System control logic. It is anticipated that five TSS-R centrals will be able to cover all AR6A routes in the continental United States.

With respect to equipment maintenance, test equipment is, in general, conventional. However, transmitter linearization and precise

frequency adjustment of the MCSS require some special test equipment, which is described in a companion paper.[1]

## II. PERFORMANCE OBJECTIVES

### *2.1 Noise*

The 4000-mile, voice-circuit noise objective is 40 dBrnc0 under standard (free-space) propagation conditions. Attaining this level of performance requires state-of-the-art designs for many devices in the system and the recognition that the current per-circuit average power in the telephone network is lower than previous designs had assumed. The per-circuit average power used in the system design is −19.6 dBm0. This level has been found to be representative of both radio trunks[2] and the telephone network as a whole.[3] With the current increase in the use of common channel interoffice signaling, it is expected that this average power will decrease slightly.

### *2.2 Amplitude equalization*

#### *2.2.1 Static equalization*

The static amplitude equalization objective requires the amplitude shape developed across a mastergroup to be not greater than ±3 dB in 4000 miles. This limit is established to be within the available correction range of regulators in the receiving multiplex equipment. Typical regulators have a ±6 dB range and ±3 dB is chosen as the allocation for the radio system. A supplementary objective of ±0.5 dB is established for the total misalignment within a radio channel in any given switch section. This limit prevents excessive errors in voiceband data systems during frequency-diversity switching between the regular and protection radio channels.

#### *2.2.2 Dynamic equalization*

As a result of the time-variant nature of propagation over typical microwave radio paths, an objective for time-variant signal amplitude is established. The objective is that for a 4000-mile circuit, the amplitude error should be less than ±2 dB for 99.9 percent of the time. The ±2 dB limit is chosen since variations exceeding this level can cause errors in some voiceband data systems. Ordinary telephone service is more tolerant of this impairment.

### *2.3 Tone interference*

The tone interference objective uses the ratio of spurious tone power to background (masking) noise power in a voice circuit as the measure of impairment. This measure is directly related to the annoying effect perceived by the customer. It also has the desirable feature of being

straightforward to administer in a large system, since each subsystem produces its own background noise and the tone requirements for the individual subsystems follow directly. The objective is stated on a percent of time basis as follows. The tone-to-noise power ratio should be less than: (1) −10 dB for 99 percent of the time, (2) 0 dB for 99.9 percent of the time, and (3) 2 dB 100 percent of the time.

In applying the tone objective to the radio line, it is found that on long routes the radio-line noise is dominated by the noise of a faded hop for more than 1 percent of the time. For tone mechanisms that increase with fading, the receiver thermal noise is then used as the background noise for the per-repeater tone objective. For tones that do not increase in level, the total radio repeater noise is used.

### *2.4 Crosstalk*

Separate objectives have been established for intelligible and unintelligible crosstalk. The objective for the former is that the crosstalk index should be less than 0.5 percent. This corresponds approximately to a probability of 0.005 of hearing one or more syllables of intelligible crosstalk during an average duration call. The unintelligible crosstalk or babble objective is stated in terms of a babble-to-noise power ratio that must be met during certain percentages of the time. The babble-to-noise ratio must be less than −10 dB 99 percent of the time and less than 7 dB 100 percent of the time.

### *2.5 Phase jitter*

The objective for phase jitter (unwanted phase modulation) in 4000-mile circuits is 8 degrees peak to peak. The objective applies to jitter with either discrete or continuous frequency components. The jitter is measured by a standard jitter indicator[4] that is sensitive to spectral components in the range of 20 through 300 Hz.

### *2.6 Reliability*

The reliability objective for a 4000-mile circuit is that it should be available for two-way service with noise less than 55 dBrnc0 for 99.98 percent of the time. This objective is apportioned among three sources of outage as follows: (1) equipment outage, 0.005 percent; (2) flat fading, 0.005 percent; and (3) selective fading, 0.010 percent. To meet the equipment reliability objective, the system design requires either frequency-diversity or hot-standby protection switching. The application of space-diversity antennas to meet the selective fading objective is discussed in Section 4.7.

## III. SYSTEM MODEL AND PERFORMANCE ALLOCATIONS

### *3.1 The 4000-mile system model*

As a basis for calculating the expected performance of the radio

system, a 4000-mile system model is used. The model is derived from the characteristics of the existing Bell System microwave radio network. The network has a mean hop length of approximately 25.6 miles, resulting in 156 hops in 4000 miles. The network root-mean-square (rms) hop length is 27.1 miles and this latter value is used for thermal noise and received signal-level calculations. The section loss corresponding to the rms hop is assumed to be 61.7 dB with typical antenna systems. This value includes all losses and gains from the output monitor-shutter in the transmitter to the input of either the receiving bay lineup or the microwave preamplifier if it is installed. The 4000-mile route is assumed to have 52 switch sections averaging three hops in length. Multiplex and terminal equipment is assumed to be located at nine intermediate points on the route. A 4000-mile circuit is thus assumed to pass through ten sets of MMGT-R and MGTB equipment. In addition, the circuit is assumed to pass through a total of nine sets of supergroup multiplex, six sets of group multiplex, and three sets of channel banks. The mastergroups within the spectrum of a radio channel are assumed to be interchanged or frogged every 400 miles for six of the terminal sections and 800 miles for two of the terminal sections.

### *3.2 Performance allocations*

#### *3.2.1 Noise*

Figure 3 shows the noise allocation for the system. The terminal equipment allocation results from assuming two-thirds of the multiplex equipment below mastergroup level is LMX-2 and one-third is LMX-3. This mix is representative of the network during the initial introduction of the system. Thermal and intermodulation noise are the primary contributors to the terminal equipment and protection switching allocations.

Four major interference mechanisms contribute to the RF interference allocation. Cochannel same-route exposures occur at the nine intermediate multiplex stations in the system model. At other stations, the same-route mechanism produces talker echo as a result of the system frequency stability and Radio Frequency (RF) channel plan. At stations with intersecting routes, cochannel junction interference occurs. The allocation assumes a mixture of AR6A and FM intersecting routes that will exist during the initial introduction of AR6A into the radio network. The third mechanism is cochannel interference from other terrestrial radio systems. All stations in the system model are subject to this interference. The final mechanism is cochannel interference from geostationary satellite earth stations. An allocation of 30 dBrnc0 has been set for this source.

The major contributors to the radio repeater allocation are thermal

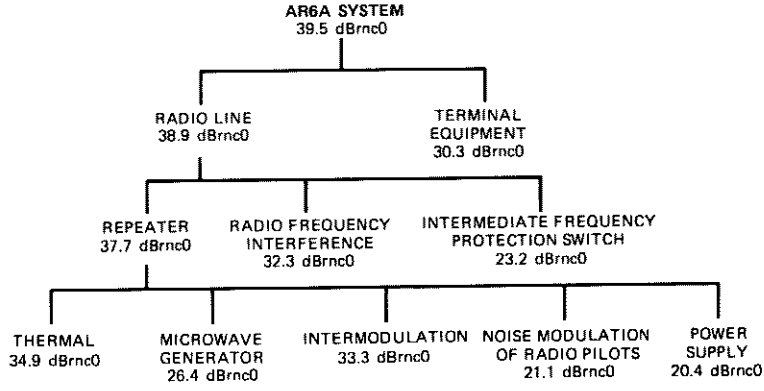



Fig. 3—AR6A noise allocation.

and intermodulation noise. The relative levels of these contributors result from an optimization of the overall system noise with respect to transmitter output power. During the development process, the allocations for these sources were lowered as a result of the use of the microwave preamplifier and the lower per-circuit average power of −19.6 dBm0.

The microwave generator allocation results from phase noise sidebands produced by the generator. In the modulation processes that use the generator output, these sidebands are convolved with the signal and produce an unwanted additive noise. In the system design, this noise is minimized at repeater stations by using one microwave generator to down-convert and up-convert each channel. The noise introduced in the two conversions nearly cancels. The predominant contributors to the allocation are the separate microwave generators used at main stations.

Since the system gain is controlled by the detected level of the three radio pilots, any spurious energy entering the passbands of the detection circuitry causes unwanted amplitude modulation of the signal. The modulation results in additive noise. The allocation for noise modulation of the radio pilots results from accumulation within a multiplex section of radio-line thermal and other noise near the radio pilots. At multiplex locations, the noise from previous paths, within the detection circuitry bandwidth, is removed and new pilots are inserted.

The power supply allocation results from a similar process. Ripple or other nonuniformity of the power supply voltages results in unwanted modulation of the signal. That portion of the modulation energy that is outside of the radio pilot detection circuitry bandwidth remains on the signal and produces noise. The predominant contributor to this allocation is the traveling-wave-tube (TWT) power supply.

#### *3.2.2 Amplitude equalization*

The static equalization objective for mastergroup shape is allocated among: (1) the residual shape of IF and MMG trunks, (2) the residual shape of an equalized radio bay, (3) echoes in the antenna and waveguide systems, (4) signal leakage through the space-diversity switch, and (5) tertiary interference. The predominant shape component is slope. The slopes introduced by all of the above sources are random and contribute to an estimated rms slope of 1.21 dB in 4000 miles. The slope distribution is approximately normal and the probability of exceeding a slope of 3 dB is then 1.3 percent. This probability is sufficiently small to satisfy the intent of the objective.

#### *3.2.3 Crosstalk*

The main sources of crosstalk in AR6A are interference exposures at junction stations and multiplex locations. The type of crosstalk produced by these exposures depends on the relative frequency stability of the interfering and desired signals. The calculated probabilities for the interference exposures producing intelligible crosstalk and babble are 0.45 and 0.55, respectively. Calculations using interference coupling levels typical of the radio network result in a crosstalk index of 0.12 percent for 4000-mile routes. The calculated probability of exceeding a −10 dB babble-to-noise ratio is 0.32 percent for 4000-mile routes. The remainder of the crosstalk and babble objectives are allocated to mechanisms producing discrete-frequency incidental modulation in the radio line.

## IV. DESIGN ASPECTS

### *4.1 Frequency plan*

#### *4.1.1 Channel assignments*

The AR6A System uses the standard 6-GHz frequency plan, which is shown in Fig. 4. This plan permits eight channels in each direction of transmission, which are designated the same as in the TH System: 11 through 18 in the lower portion of the frequency band and 21 through 28 in the upper portion of the frequency band. This plan places the receivers in one-half of the band and the transmitters in the other half. A station using the frequency plan shown in Fig. 4 is called a "low-high" station, because the receive channels are in the low-frequency portion of the band and the transmit channels are in the high-frequency portion of the band. Adjacent stations will be "high-low" stations. The microwave carrier frequencies, which are used in the receiver and transmitter modulators, are also shown in the figure, along with the channels served.

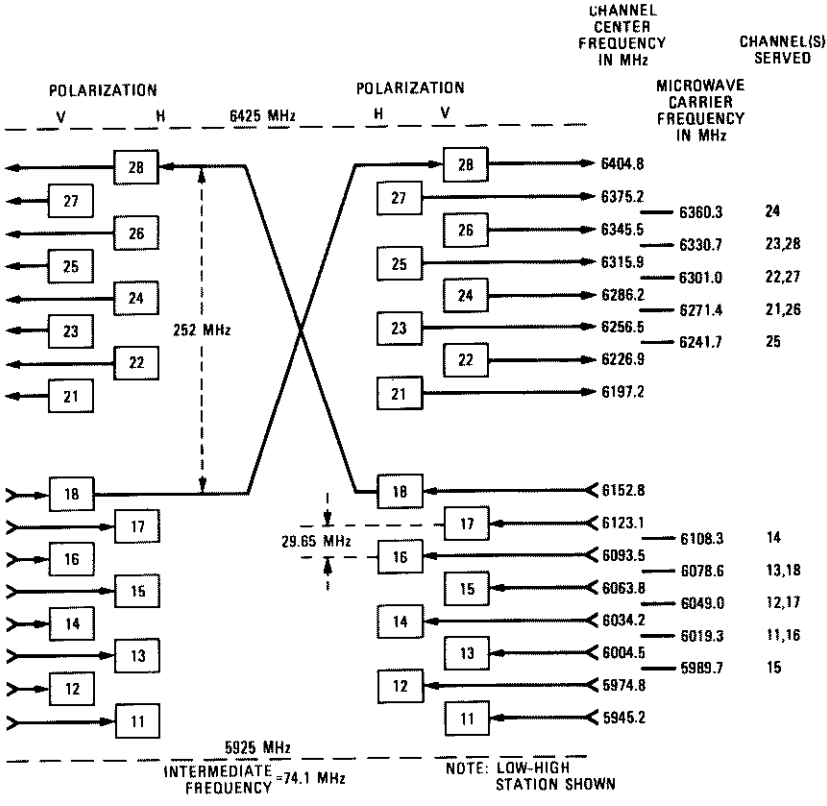



Fig. 4—AR6A frequency plan.

### *4.1.2 Choice of IF*

An IF band centered at a frequency that is an odd multiple of half the channel separation is the best choice for minimizing the effects of microwave carrier leakages. With this choice of IF center frequency, these leakages appear at the edge between two channels and do not need as much suppression as they would if they fell inside a channel. On this basis the best IF center frequency for the 6-GHz frequency plan is 74.12965 MHz (usually referred to as 74.1 MHz). In addition to the advantage of controlling the effect of the microwave carrier leakages, a 74.1-MHz center frequency permits a reduction from 16 to 10 of the microwave carrier and filter codes needed, since some microwave carrier frequencies can be used for two channels by using upper and lower sideband modulators. These design factors favor the choice of 74.1 MHz. Consideration was given to using a center frequency of 70.0 MHz, which is used in the TH-3 and TD Radio Systems. This choice would permit the direct IF interconnection of FM systems

onto an AR6A radio channel. The ability to handle both AM and FM signals would have excessively complicated the AR6A design; therefore, an IF center frequency of 74.1 MHz was used.

#### *4.1.3 IF band utilization*

At both IF and microwave frequencies, the SSBAM signal with suppressed carrier occupies no additional bandwidth over the original baseband signal. Thus, mastergroups can be translated into the channel bandwidth with a spacing determined by the filtering requirements needed for separation when shifted back to baseband. Carrier systems previously used a frequency spacing between mastergroups of approximately 4 percent of the frequency, where the mastergroup was to be located based on the desire to separate mastergroups (block and branch) at the multimastergroup level. The use of mastergroup translators makes it more convenient and economical to make these rearrangements at the basic mastergroup level, which permits the use of a fixed frequency separation at the multimastergroup level. Improvements in filter performance have made it possible to form the MMG spectrum with only 168-kHz spacing between mastergroups. The number of mastergroups that can be placed in the radio channel bandwidth is constrained by cochannel interference from FM systems. Clearly, the center of the channel, which may contain interfering FM carriers, cannot be used, and further allowance must be made for the close-in sidebands of the FM signal since these are strong enough to cause serious interference into AR6A voice circuits. The situation for TH-3 is shown in Fig. 5. Consequently, a center gap of 2.072 MHz has been provided which, with frogging of the mastergroups, keeps interference at an acceptable level, and achieves a loading of ten mastergroups in the 29.65-MHz channel bandwidth. The addition of pilot tones in the gaps between mastergroups for regulation and control purposes completes the IF spectrum as shown in Fig. 2.

### *4.2 Frequency stabilization*

#### *4.2.1 Need for stabilization*

An important aspect of the system design is the required frequency stability along a radio route. The radio pilots, which are transmitted as part of the IF spectrum, must be centered in their respective pick-off filters in each receiver. For best operation, the pilots should be within ±2 kHz of their nominal frequencies. The necessary frequency accuracy is achieved by locking the microwave generators and shift oscillators at each station to a high-stability synchronization oscillator located in each station. This will stabilize both the microwave and the IF frequencies of the system.

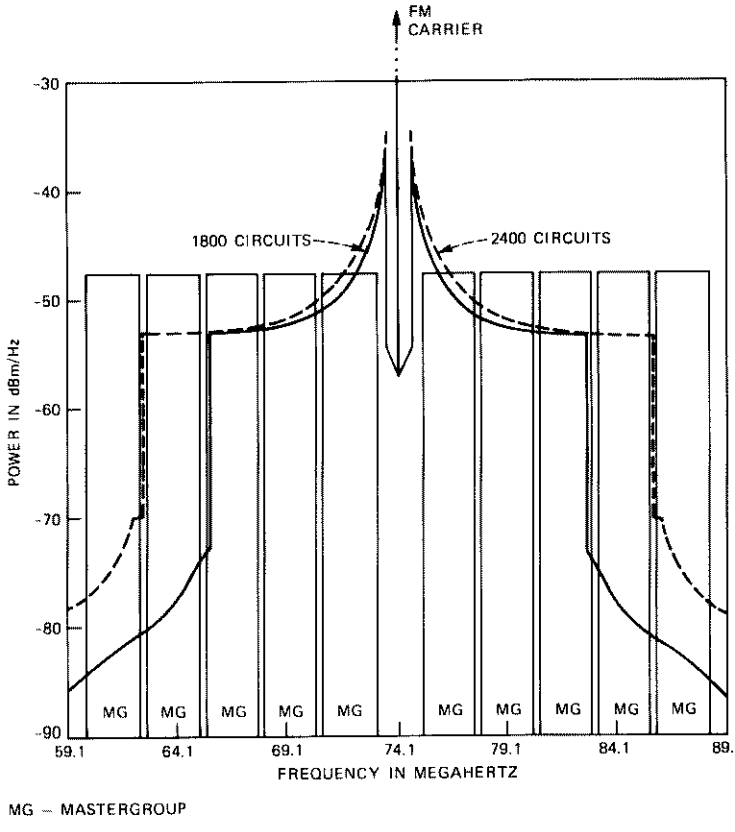



Fig. 5—AR6A and TH spectra.

### *4.2.2 Method of stabilization*

As a result of choosing the IF center frequency of 74.12965 MHz, all the microwave generator frequencies and the shift oscillator frequency are multiples of 14.825930 MHz. Because the best crystal frequency stability is obtained around 5 MHz, the AR6A synchronization supply oscillator was chosen to have a frequency of 4.94197666 MHz, which is one-third of 14.825930 MHz. This oscillator is oven controlled and has a stability of two parts in $10^8$ over the required temperature range and an aging rate of less than five parts in $10^{10}$ per day. In order to get all the harmonics needed for locking the microwave generators and for circuit convenience, the AR6A synchronization oscillator frequency is divided by 16 to obtain 0.30887354 MHz. This latter frequency is distributed to the phase-locked loops associated with microwave generators and the shift oscillators. The phase-locked loops contain circuitry that will hold the frequency at its last value

and prevent a frequency jump if the synchronization signal is lost. The AR6A synchronization oscillator supplies the synchronization for all 16 bays of a fully equipped repeater station and is protected by automatic switching to a second unit.

### *4.3 Signal recovery*

The customer-to-customer frequency offset objective of the Bell System is ±2 Hz. This means that carriers used in translating baseband signals up and down in frequency must be very accurate in frequency. This is accomplished by locking these carriers to a primary frequency supply (PFS) which itself is locked to the Bell System master oscillator in Hillsboro, Missouri. Frequency-division multiplex systems such as LMX, MMX, and JMX all operate in this manner. A newly developed OMFS is provided in AR6A terminal stations to meet the frequency offset objective.

In the AR6A System no carrier is transmitted along with the AM sidebands, so one must be accurately generated in the receiving MMGT-R. The IF signal into the receiving MMGT-R can be a few hundred hertz from nominal because of accumulated frequency errors from intermediate repeaters. The receiving carrier generated in the MMGT-R receiver is controlled in frequency to compensate for the accumulated frequency errors by using recovery pilots (one in each half of the IF spectrum). These recovery pilots at the input to the receiving MMGT-R have the same accumulated frequency offset as the rest of the IF signal. The receiving MMGT-R translates each half of the IF spectrum down to the MMG spectrum independently. In each case, the received recovery pilot is selected from the MMG spectrum and is compared in a phase-locked loop with an oscillator locked to the local OMFS. The output error signal from the comparator is used to adjust the frequency of the carrier used in the receiving demodulator. In this way, the receiving carrier "tracks" the frequency offset of the incoming signal and the MMG signal is recovered with no frequency offset.

### *4.4 Equalization*

#### *4.4.1 Static*

The objective for amplitude deviation from flatness over the 30-MHz channel for each switch section is ±0.5 dB. To help achieve this objective on a radio hop, each TR bay receiver includes a basic bay equalizer, which compensates for known systematic amplitude misalignments of the microwave and IF components. Mop-up equalizers are available in ±0.5 dB slope and parabolic shapes to enable each switch section to meet its ±0.5 dB objective. The amount of mop-up equalization required is specified automatically from switch section

amplitude response measurements made by the TSS-R System. Places are available in the 500A protection switch and in each radio receiver to mount the needed mop-up equalizers.

#### *4.4.2 Dynamic*

The objective for time-variant amplitude variations is that they be less than ±2 dB for 99.9 percent of the time. The primary source of transient amplitude variations is selective fading of the microwave signal due to unusual atmospheric conditions. An extensive measurement and analysis program was started in 1970 to determine the effects of selective fading on a 6-GHz microwave radio channel.[5] As a result of computer simulation studies made with the collected fading data, the following three strategies were recommended to minimize selective fading effects:[6]

1. Use space-diversity reception for each hop that has significant selective fading.
2. Use a simple automatic gain control (AGC) amplifier in each receiver.
3. Use a dynamic equalizer that will provide slope and parabolic shape correction at the end of each switch section.

The use of space diversity is recommended to reduce the amount of time the microwave signal is in a deep fade where the largest amplitude variations occur. This is discussed further in Section 4.7. The use of an AGC amplifier in each receiver to compensate for the flat gain portion of microwave signal variations is standard in radio systems. In the AR6A System, the AGC amplifier gain is controlled by the voltage average of the three received radio pilots. To compensate for the shape portion of the amplitude variations, a dynamic equalizer has been designed[7] and is used in each main station receiver. The dynamic equalizer contains one bump network located at each end of the IF band. The gains of these networks are continuously controlled by the voltage difference between the edge radio pilots and the center radio pilot of the received channel. Figure 6 shows a typical fade shape across the IF band before and after equalization.

### *4.5 Interference*

#### *4.5.1 Impairments*

An important aspect of any new radio system design is the consideration of the interference environment. The AR6A System has to coexist with the existing FM radio systems, other AR6A routes, and self-interference. One effect of interference is to add to the background noise of the channel; this is compensated for by the inclusion of 32.3 dBrnc0 for interference in the system noise allocation.

Other effects are crosstalk, babble, and tone interference, which can

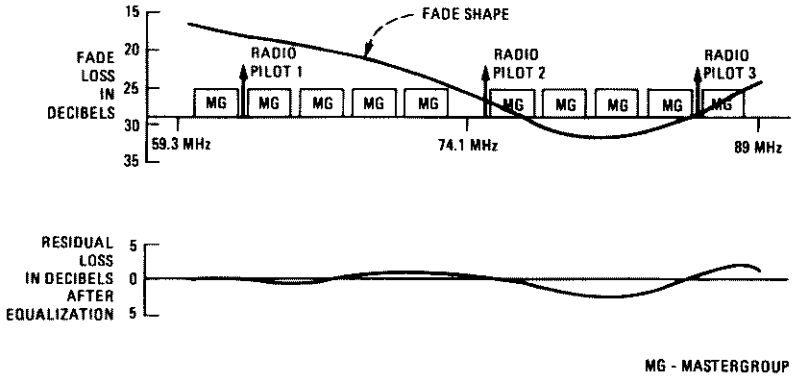



Fig. 6—Dynamic equalization of a fade.

occur when two AR6A channels carrying different traffic interfere with each other.

If both the interfering signal and the desired signal are voice signals and are exactly aligned in frequency, the effect is crosstalk. If the two signals are increasingly offset in frequency, the interference becomes less intelligible until, at ±500 Hz, the interference is judged to be unintelligible crosstalk and referred to as babble. Should the interfering signal be an idle circuit tone or a data signal, the effect will be tone interference regardless of any frequency offset. This type of coupling between two AR6A channels can occur at junction stations or between two routes not sharing a station but in the same geographical area. These effects must be evaluated in terms of the crosstalk, babble, and tone interference objectives. In FM systems, the poorer RF frequency stability and the effects of carrier spreading render these impairments negligible.

Another interference effect is talker echo that results from coupling—primarily antenna coupling—between the east- and westbound channels of the same route. For administrative purposes, the east- and westbound halves of the 4-kHz voice circuits are in corresponding locations in their respective channels. Thus, if these two channels are within 500 Hz of each other, the result is talker echo. Beyond 500 Hz, the effect will be a babbled talker echo, which is judged to be almost as annoying. The magnitude of the talker echo may be described in terms of the effective return loss of the system.

#### *4.5.2 Cochannel interference*

To avoid interference from cochannel FM carriers, the center of the AR6A channel is left vacant. The noise effect of the FM sidebands is controlled by properly selecting the channel center gap and frogging

Table I—Impairment calculation summary

| Impairment | Objective | Estimated Performance (Percent) |
|---|---|---|
| Crosstalk | ≤ 1 percent | 0.12 |
| Babble/Noise | ≥ −10 dB no more than 1 percent of time | 0.32 |
| Tone/Noise | ≥ −10 dB no more than 1 percent of time | 0.55 |

the mastergroups. The noise effect of another cochannel AR6A System is controlled by requiring at least a 64-dB signal-to-interference ratio.

Studies were conducted to evaluate the ability of the AR6A System to meet the crosstalk, babble, and tone objectives. These impairments are largely controlled by the antenna system discrimination against interfering signals and route layout. Most of AR6A installation would be on existing routes, which means antennas and routes are already fixed. A design choice that was open was the frequency stability of the radio line. As noted in Section 4.2, the microwave carrier supplies required stabilization to keep the radio pilots within the passband of their pick-off filters. Several stabilization plans with different degrees of stabilization were analyzed to determine their effect on crosstalk, babble, and tone interference. These included a plan that introduced a series of fixed frequency offsets on different radio channels to reduce crosstalk. The results showed that a highly stabilized plan was the best for balancing the impairments. A summary of these impairments for the 4000-mile system model is shown in Table I.

One remaining cochannel interference that required analysis was the same route interference that gives rise to talker echo. The requirement for a 4000-mile system is an effective return loss of at least 30 dB. Calculations estimate that the return loss of the radio line is 41 dB.

#### *4.5.3 Adjacent-channel and tertiary inference*

The AR6A System has no significant signal-related energy outside of its channel bandwidth. Thus, adjacent-channel noise effects are negligible for adjacent AR6A channels. However, mutual problems occur when an AR6A and an FM channel are adjacent. If the FM channel is an RF squelch-equipped TH-3 System, this feature must be removed because the initiator noise slot is located inside the AR6A channel in a mastergroup position. Without removal the initiator would be continuously operated. Another problem is generated due to the wide bandpass of the IF filters needed in the FM systems to reduce their own channel noise. These filters couple into the FM receiver a portion of the adjacent AR6A signal, which is limited primarily by the cross-polarization discrimination of the antenna system. The noise

caused by this interference restricts the number of exposures to 10 or 15. In addition, the tertiary interference effect of a portion of the AR6A signal coupled into the adjacent FM channel is also undesirable. This AR6A signal when coupled back into the AR6A channel will produce modulation on the edge radio pilot and an amplitude shape deviation, depending upon the phase relationship between the signal and the interference. Estimates of these effects indicate they should be avoided if possible, but if they cannot be avoided during route conversion from FM to AR6A, then the AR6A channel adjacent to an FM channel must be either the protection channel or a working channel with the five adjacent mastergroups vacant.

### *4.6 Linearity*

#### *4.6.1 General*

Low intermodulation distortion (IM) properly optimized relative to other noise contributors is a basic consideration in the design of a transmission system. It is of particular interest in the AR6A System since the achievement of satisfactory linearity in the transmitting microwave amplifier is a key factor in making the SSBAM Microwave System possible. Transmitting amplifiers with power-handling capacities comparable to those commonly used on current FM systems fall short of meeting SSBAM linearity requirements by some 20 to 30 dB. Achieving stable distortion reductions of this magnitude was thus a crucial design task. Furthermore, although transmitter nonlinearity is dominant, care must be exercised in the design of all other in-line elements to ensure that their cumulative distortion contribution is not excessive.

Since at microwave frequencies and even at intermediate frequencies we are dealing with narrowband channels, the only significant distortion contributors are the odd-order, nonlinear terms. In AR6A, third-order distortion is dominant, fifth order is on the fringes of importance, while seventh and higher odd-order terms are negligible.

A related and crucial design consideration is the question of how the distortions from tandem nonlinear networks combine within and between repeaters. This is referred to as the Law-Of-Addition (LOA) question. Differences between the extreme assumptions of coherent and noncoherent addition strongly influence a variety of design choices. For example, the LOA influences the optimum system drive level, it affects how frequently mastergroups must be relocated or frequency frogged, and it affects equalization strategies. For these reasons, studies and estimates of the LOA received early attention, while subsequent verification of the assumptions was an important element in the field evaluation.

#### 4.6.2 Distortion estimates

To make system distortion estimates, first the distortion contribution must be estimated from individual system elements. This, in turn, must be based on some suitable measured parameter(s) indicative of the unit's nonlinearity. For AR6A computations, a practical approach starts from a power series description of the unit's input/output relationship. Functional ratios of the power series coefficients can then be related to measurable modulation coefficients and vice versa.

To estimate distortion, the broadband message load can be simulated by zero-mean Gaussian noise with appropriate spectral shaping. Multiple convolutions weighted according to power series coefficients may then be used to determine expected IM distortion. In AR6A, since the signal spectrum is flat, the IM noise computation can be greatly simplified by using the work of Y. L. Kuo.[8] Distortion-to-signal ratios are evaluated in terms of power series coefficients, which are then related to measurable modulation coefficients. The final result is expressed in the following relationships, which assume that the odd-order power series has no significant terms beyond the fifth

$$N_3 = 4.23 + 2P_{\text{out}} + M_{\alpha+\beta-\gamma}$$

$$N_{35} = 4.77 + 3P_{\text{out}} + \frac{1}{2}\,(M_{\alpha+\beta-\gamma} + M_{2\alpha-2\beta+\gamma})$$

$$N_5 = 8.32 + 4P_{\text{out}} + M_{2\alpha-2\beta+\gamma}, \tag{1}$$

where

$N_3$ = ratio of third-order distortion power in a narrowband $\Delta W$ to signal power in the same bandwidth. The ratio is expressed in decibels and the band $\Delta W$ is located at the center of the simulated, spectrally flat, message load,

$N_5$ = the corresponding fifth-order power ratio,

$N_{35}$ = the corresponding fourth-order power ratio being an interaction term between the third- and fifth-order distortion mechanisms,

$M_{\alpha+\beta-\gamma}$ = a third-order modulation coefficient expressed in decibels,

$M_{2\alpha-2\beta+\gamma}$ = a fifth-order modulation coefficient expressed in decibels, and

$P_{\text{out}}$ = total simulated message-load power in dBm at the device output.

The noise-to-signal power ratios can be converted to dBrnc0s as follows:

$$\textit{noise}\ (\text{dBrnc0}) = N + P_1 + 86.8, \tag{2}$$

where $N$ is the noise-to-signal power ratio in dB and $P_1$ is the average signal power per 4-kHz message circuit at 0 transmission level (TL). These relationships involve a number of simplifying assumptions and should be used with caution. For example, the $M$ values are assumed to be constants independent of drive level and frequency; addition of $M_{\alpha+\beta-\gamma}$ and $M_{2\alpha-2\beta+\gamma}$ in the expression for $N_{35}$ is subject to some uncertainty because of the absence of product phase information in the definition of $M$; and, finally, the expressions apply at the band center of an idealized simulation of the message load. In AR6A applications, cross-checks have been made against pseudorandom noise load measurements[9] with satisfactory agreement if nominal drive levels are not exceeded by more than a few decibels.

With typical values for an output traveling-wave-tube amplifier of $M_{\alpha+\beta-\gamma} = -90$ dB, $P_{\text{out}} = +25$ dBm, and $P_1 = -19.6$ dBm0, the estimated IM noise is 23 dBrnc0 per repeater. Taking account of product addition laws, thermal noise contributions, etc., led to the earlier assertion that distortion reductions of at least 20 dB are needed to make the SSBAM System viable.

#### *4.6.3 Law of addition*

For a string of $n$ identical nonlinear devices, the law of addition ($K$) for intermodulation noise (IM) is defined by

$$W_n = K \log n + W \text{ dBrnc0}, \tag{3}$$

where

$W_n$ = intermodulation noise of the string,

$W$ = intermodulation noise of one device.

Odd-order distortion from some elements of a repeater are going to add systematically corresponding to $K \approx 20$. Other sources of repeater IM are going to accumulate in a random fashion corresponding to $K \approx 10$. The overall law of addition is going to be somewhere between those extremes and will depend on the relative magnitude of the two types of contributors. Clearly, the net value of $K$ is going to be an extremely important factor in estimating total system noise on a long route.

In the case of IM contributed by repeater elements subject to predistortion correction, the residual distortion can be expected to have a random phase orientation. However, some departures from randomness will be experienced if there are systematic degradations in predistorter improvement due to frequency response or aging effects.

Nonlinear elements not subject to predistorter compensation, such as IF amplifiers, will generate IM that adds systematically. Again,

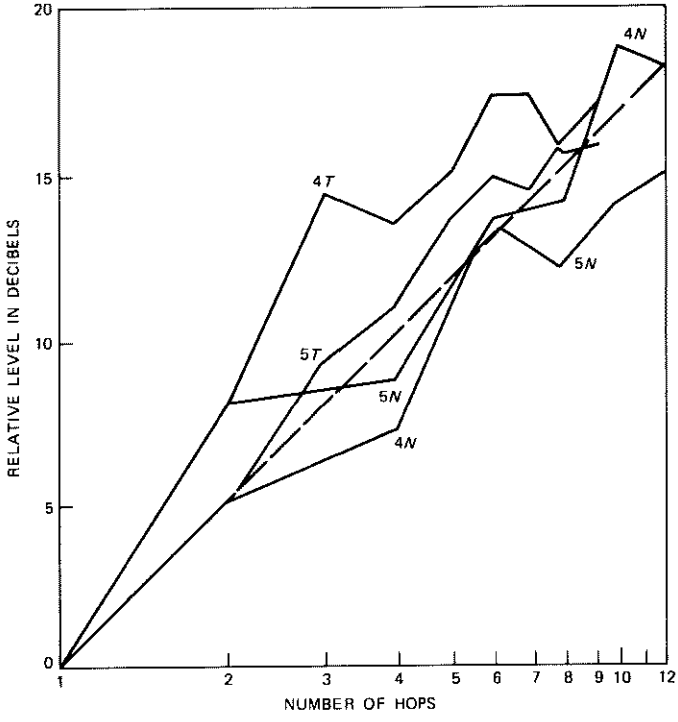



Fig. 7—Intermodulation noise law of addition.

however, there are modifying circumstances such as repeater delay distortion that in this case can reduce the rate of accumulation. Computations involving multiple convolutions of the signal indicate that on AR6A, delay distortion reduces the law of addition from 20 to approximately 19 near band center with larger reductions near the band edges.

For design purposes, law-of-addition estimates were based in broad terms on the power addition of transmitter IM, the voltage addition of receiver IM, and power addition of the combination. Depending on detailed assumptions, this led to estimates for $K$ in the range 16 to 18.

Measurements using pseudorandom noise loading[9] for 12 pairs of repeaters in the initial field installation yielded an average value $K = 16.9$. These same repeaters were then connected in loops of 2 through 12 hops, and the pseudorandom noise load results for radio channels 4 and 5 ($4N$, $5N$) shown in Fig. 7 were obtained. The dotted line in the figure refers to the expected law of addition ($K = 16.9$). Measurements using a three-tone test ($4T$, $5T$) are also shown in the figure. Reasonable agreement with the expected law of addition is observed.

### 4.7 Application of frequency- and space-diversity protection

#### 4.7.1 Microwave propagation of the AR6A signal

In the early development of AR6A, it was recognized that it would be necessary to characterize the effects of selective fading on the broadband SSBAM signal. Propagation studies were conducted in 1971 on a 26.4-mile path from Palmetto to Atlanta, Georgia. The studies quantified the fading effects on a statistical basis and determined many of the basic system design features, including the choice of the three radio pilots to control system gain, the type and amount of dynamic equalization, and the algorithms for space- and frequency-diversity switching.[10]

In microwave radio system design, multipath (Rayleigh) fading is usually characterized by the equation[11]

$$T(L) = rT_0L^2,\ L < 0.1, \tag{4}$$

where

$L$ = the ratio of received carrier amplitude to nominal or free-space received carrier amplitude,
$T_0$ = time base or total time of the study,
$r$ = fade occurrence factor representing the fraction of time that selective fading occurs, and
$T(L)$ = time in seconds during the period $T_0$ that the received carrier amplitude ratio is below the level $L$.

This relation is descriptive of any single frequency in the AR6A spectrum. The propagation studies expanded this signal characterization to four frequencies. Statistics representative of the fading of any independent frequency in the spectrum were generated with conditioning on the state of the three radio pilots. The equation modeling the fading becomes

$$T(L) = rT_0L^2(U(L) + P(L)), \tag{5}$$

where

$U(L)$ = the conditional probability that the radio pilots are less faded than the ratio $L$ given that the frequency of interest is faded at least to the ratio $L$, and
$P(L) = 1 - U(L)$.

The probability $U(L)$, shown as the curve $L_F = L$ in Fig. 8, is less than 0.1 for fades less than about 36 dB ($L = 0.016$) and increases monotonically with fade depth to about 0.5 for fades greater than 48 dB ($L = 0.004$). Since the radio pilots are used to control the system gain and switching functions, this characterization allows various equalization and switching plans to be evaluated.

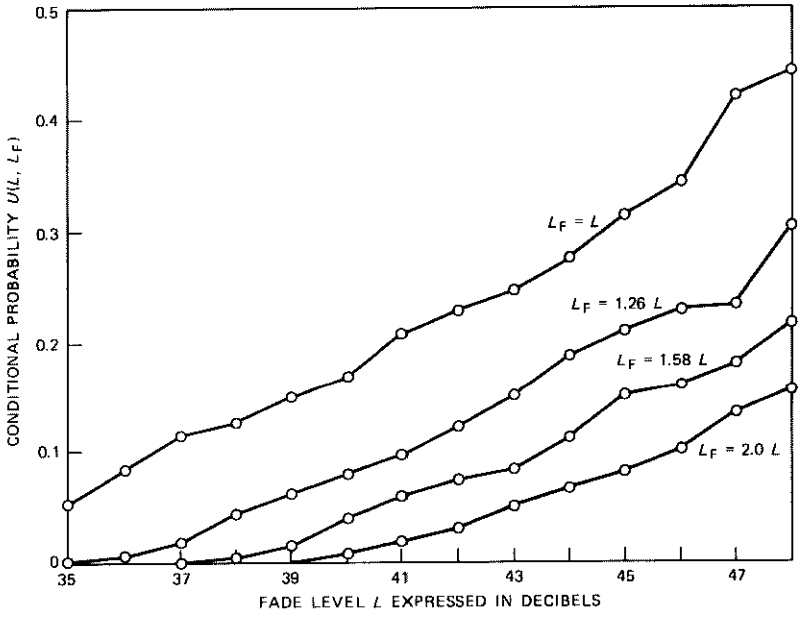



Fig. 8—The conditional probability $U(L, L_F)$.

In addition to the four frequency fading characterization, the 1971 propagation data were also used to characterize the amplitude response of the signal during fading. The data were used as the input to a computer program that simulated the response of the dynamic equalizer. The results of this study are shown in Fig. 9. This plot indicates the expected average severity of gain deviations during fading. The averaging process includes an average over all voice-circuit locations in the signal spectrum. The plot is similar to a cumulative distribution with the maximum fade level of the radio pilots as the independent variable. The 2-dB gain deviation limit corresponds to the dynamic equalization objective. As indicated in the figure, fades with radio pilot attenuation less than 30 dB contribute essentially no gain deviation time. For fades with radio pilot attenuation exceeding 40 dB, comparison of Fig. 9 with the Rayleigh model of eq. (4) shows that almost all of the fade time has gain deviations exceeding ±2 dB. These results led to the recognition that in order to meet the system 2-dB gain deviation objective, space-diversity switching at radio pilot fade levels in the range of 30 to 40 dB would be required.

#### *4.7.2 Frequency- and space-diversity switching during multipath fading*

With the recognition that both frequency- and space-diversity switching would be required for AR6A, a fading model including the

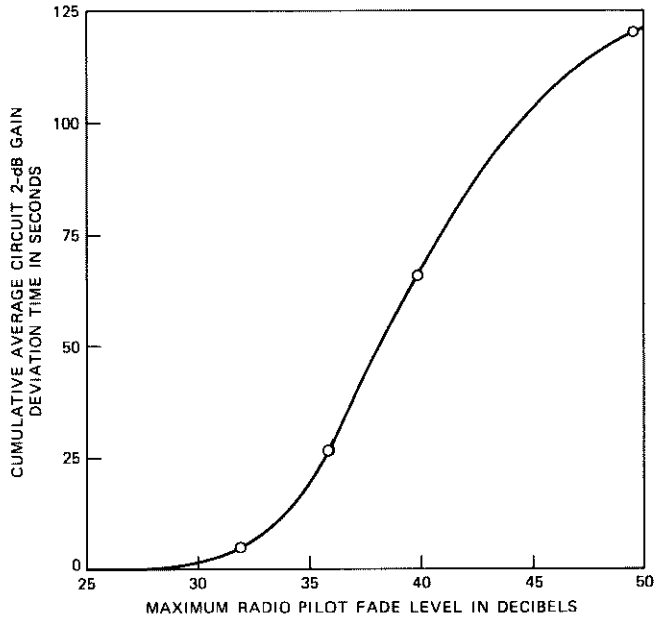



Fig. 9—Average gain deviation time of an AR6A circuit for 600 seconds of multipath fading at the 30-dB level.

effects of both switching systems was developed.[6] The resulting equation representing the fading performance of a hop equipped with both types of switching is

$$T_p(L) = \frac{rT_0L^2}{I_s(L_s)}\left[U(L, L_F) + \frac{P(L, L_F)}{I_F(L_F)}\right], \; L \leqslant L_F < L_S < 0.1, \quad (6)$$

where

$L_F$ = the pilot fade level corresponding to the frequency-diversity switch threshold,
$L_S$ = the pilot fade level corresponding to the space-diversity switch threshold,
$I_F(L_F)$ = the improvement factor resulting from frequency-diversity switching, typically $I_F > 1$,
$I_S(L_S)$ = the improvement factor resulting from space-diversity switching, typically $I_S \gg 1$,
$U(L, L_F)$ = the conditional probability that the three radio pilots are less faded than $L_F$ given that the frequency of interest is faded to level $L$,
$P(L, L_F)$ = $1 - U(L, L_F)$, and
$T_p(L)$ = time that the protected signal is faded below level $L$.

The quantities $I_F$ and $I_S$ vary as $L^2$ but differ from the conventional,[10] single-frequency improvement factors. Typical values in the current system design are $I_F = 15$ and $I_S = 100$. The conditional probability function $U(L, L_F)$ is shown in Fig. 8 with the relationship between $L$ and $L_F$ as a parameter. The results in the figure are calculated from the 1971 data without smoothing. Since $I_F(L_F) > 1$, eq. (6) shows that higher values of $P(L, L_F)$ are desirable. The increase in $P(L, L_F)$ corresponds to an increase in the fade time that is protected by frequency-diversity switching and a decrease in outage time.

The 2-dB gain deviation time for a path protected by frequency- and space-diversity switching is modeled by the equation[6]

$$T_2(L) = \frac{rT_0}{0.6}\left[D(L_s) + \frac{D(L_F) - D(L_s)}{I_s(L_s)} + \frac{D(L) - D(L_F)}{I_s(L_s)I_F(L_F)}\right], \quad L < L_F < L_s, \tag{7}$$

where the function $D$ is the 2-dB gain deviation time given in Fig. 9. Equations (6) and (7) formed the basis for evaluating various space- and frequency-diversity switching algorithms.

#### *4.7.3 Reliability and dynamic equalization requirements*

During the development process, the performance of the radio system with respect to the reliability and dynamic equalization objectives was evaluated for several space- and frequency-diversity switching algorithms. In particular, various combinations of the switching levels $L_S$ and $L_F$ were studied. System performance for both voice and voiceband data was included. It was found that lower values of $L_S$ and $L_F$ improved the reliability performance but degraded the dynamic equalization. The current system design represents an approximate optimum in the trade-off between these requirements.

The space-diversity switching level was chosen to correspond to a 36-dB fade on a 27.1-mile path. For longer or shorter paths, the switching level varies inversely with the received signal level. The frequency-diversity switching level was chosen to be 53 dBrnc0 corresponding to a 44-dB fade on a 27.1-mile path. With this switching arrangement, approximately one-half of the hops in an average fading area are expected to require space diversity.

#### *4.7.4 Computerized implementation*

An interactive computer program was developed to facilitate the application of frequency- and space-diversity protection to new AR6A switch sections.[6] The evaluation of a proposed section's performance with respect to the outage and dynamic equalization objectives is

accomplished by the program after input data are supplied by the program user. The data are used to characterize the expected fading on the various hops in the section. For each hop, the input data include a climatic factor; the mean annual temperature; the hop length, roughness, and fade margin; and a description of any proposed space-diversity antenna.[10] The program first evaluates the performance of the section without space diversity. The estimated performance is compared to the objectives after prorating them to the length in miles of the section. If the objectives are met, the program stops and the route can be installed without space diversity. If they are not met, the program reevaluates the performance with space diversity applied initially on the worst fading hop. The program continues in this manner until the objectives are met. The route would then be installed with space diversity on the specified hops.

### *4.8 Mastergroup frogging*

Noise, reliability, and dynamic equalization in AR6A are functions of voice-circuit location in the IF passband. Circuits located near the IF band center receive relatively high levels of interference noise from FM-FDM and FM-TV systems. Reliability and dynamic equalization are best for circuits located near one of the radio pilots. Because of these variations, more uniform performance is attained on long trunks by varying their location within the IF spectrum. The variation is accomplished by interchanging or frogging mastergroups at specified distances. The maximum allowable frogging distance chosen for field applications is 800 miles. This value provides acceptable system performance and reduces the amount of multiplex equipment required solely for frogging. The effects of frogging distance on system performance are described in the following sections.

#### *4.8.1 Noise performance*

The system noise components that are influenced by frogging include: intermodulation noise, radio pilot modulation noise, and worst-circuit FM RF interference noise. The levels of these components all increase with frogging distance and are listed in Table II for distances of 400, 800, and 1600 miles. The intermodulation noise increase with

Table II—Noise performance with various frogging rules

| | Noise in dBrnc0 | | |
|---|---|---|---|
| Noise Type | 400-Mile Rule | 800-Mile Rule | 1600-Mile Rule |
| Intermodulation | 31.7 | 33.3 | 34.0 |
| Radio pilot modulation | 16.1 | 21.1 | 26.1 |
| Worst-circuit FM RF interference | 27.4 | 30.4 | 31.9 |

frogging distance is due to the radio receiver. The receiver intermodulation products produced on hops within a frogging section are nearly coherent. A small reduction, estimated to be 2.6 dB for an 800-mile frogging section, results from the phase characteristic of the IF filter. The receivers' products in different frogging sections are incoherent because of the changed signal. The radio pilot modulation noise also increases with frogging distance as a result of hop-to-hop coherence.

Although the average noise resulting from the FM RF intereference does not change with frogging distance, the distribution of the noise does. Circuits located near the center radio pilot experience maximum interference. When the frogging distance is increased beyond 400 miles, some circuits receive significantly more noise than the average. The resulting worst-circuit noise is shown in Table II.

#### *4.8.2 Reliability and dynamic equalization performance*

Since the noise initiator slots for the 500A Protection Switching System are located near the radio pilots, voice circuits located near the pilots receive relatively good selective fading protection. For these circuits unprotected fade time is small. For circuits located between the radio pilots, some selective fades can cause outage without producing enough noise in the slots to initiate a switch. As a result, the outage performance of the system varies with IF frequency. With a 400-mile frogging rule, all mastergroups transmitted 4000 miles occupy each mastergroup location in the signal spectrum for approximately 400 miles. When the frogging distance is extended beyond this interval, the performance is no longer uniform. With the 800-mile rule, calculations indicate that the outage of the worst circuits will exceed the average circuit by 44 percent. To ensure that these circuits meet the overall multipath fading objective, an algorithm was added to the computer program for the application of space diversity. For switch sections located in frogging sections between 400 and 800 miles in length, worst-circuit performance is calculated and compared to the objective. Additional space-diversity protection is then required for the switch section to meet the system requirement. This procedure allows the engineers responsible for a route to trade off the cost of additional multiplex required for a shorter frogging section with the cost of additional space diversity required for a longer one.

For frogging distances greater than 400 miles, the dynamic equalization performance of the system parallels the reliability performance. Worst-circuit gain deviation time with the 800-mile frogging rule is approximately 48 percent greater than the average circuit time. This factor is also included in the computer program for the application of space diversity, and the worst circuits are required to meet the system objective.

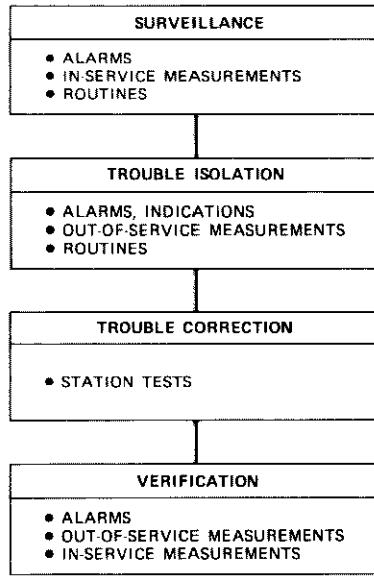



Fig. 10—AR6A generalized maintenance plan.

### *4.9 Maintenance*

#### *4.9.1 General*

Maintenance was discussed in general terms in Section 1.6 with particular mention of the importance of adequate transmission surveillance capability. In AR6A, the new surveillance functions are combined with traditional remote alarm and reporting features to form an overall maintenance plan that is summarized in very general terms in Fig. 10.

The alarm reporting system is a full-time monitoring facility using either C1 or E-type alarm remotes under the centralized jurisdiction of a SCOTS or TASC central minicomputer. In general, remote alarm reporting is limited to isolating trouble to a particular station and direction of transmission, not the particular unit in trouble. Locally, however, more detailed visual indications and test points simplify the trouble isolation process. This approach is based on the premise that prior knowledge of the detailed trouble condition is unnecessary if it is assumed that, on a station visit, the technician is equipped to deal with any on-site problem or any additional problems that become known enroute.

Except for a few legally required routine tests, maintenance is performed on an as-needed basis. When a trouble has been identified and isolated, a technician is dispatched to the trouble site with a defined set of spare parts and station test equipment. The principal station test sets are:

1. Transmitter-receiver transmission test set
2. Scanning intermodulation test set
3. Portable rubidium frequency standard
4. Pilot selection test set
5. TWT power supply test load.

#### *4.9.2 Transmission Surveillance System*

The AR6A Transmission Surveillance System—Radio (TSS-R) consists of programmable equipment for access and testing located at radio switching stations. It can perform a variety of transmission measurements under the direction of a central minicomputer. Communication between the central computer and remote measuring sites is via *DATAPHONE** data communications service with its attendant flexibility and expandability. Unlike the alarm reporting system, TSS-R measures system performance on a scheduled or as-needed basis, providing snapshot rather than continuous information.

Figure 11 is a block diagram of TSS-R. The central minicomputer has a peripheral disk storage capacity of approximately 20 Mb allocated to program, database, and data storage functions. Automatic call-out units provide DDD access to the measuring remotes as well as to maintenance and other centers programmed to receive reports. It is expected that five central computers will be capable of covering all AR6A deployment within the continental United States.

At switching stations, test access, measuring equipment, and data interchange with the central computer are under the control of a microprocessor with its associated firmware. Physically, these functions are grouped in the TSS-R distribution bay as indicated in Fig. 11. All measurements are performed by a Selective Transmission Measuring Set (STMS) with a frequency range of 10 kHz to 160 MHz. The set is controlled by the distribution bay microprocessor via an IEEE control bus. Programmable functions include level and frequency measurements, bandwidth selection, and the selection of a phase-jitter measuring mode.

Switched access for measurement purposes is provided at transmitting and receiving mastergroup combiners and at transmitting and receiving 500A switches. Measurements at mastergroup combiners can be made on a bridged, in-service basis only; measurements at the 500A switches can be made either on an in-service or out-of-service basis.

In-service measurements can be made conventionally using bridging hybrid transformers in conjunction with director switches to route the selected test point to the STMS. Signals available for measurements

* Registered service mark of AT&T.

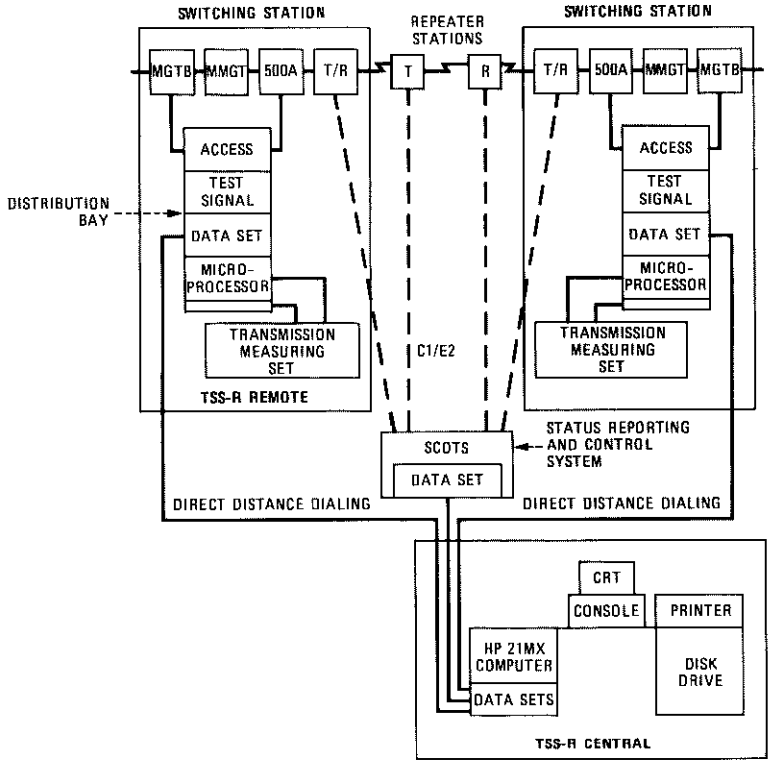



Fig. 11—Transmission Surveillance System.

are elements of the normal system load such as mastergroup pilots, system pilots, and intermastergroup noise. Since in-service measurements are made without affecting the main transmission path, they are made frequently and routinely at intervals of approximately one week.

On the other hand, out-of-service measurements require that service be temporarily transferred to the protection channel. This frees the selected working channel for end-to-end switch section measurements. To measure the channel, access at the head end of the switch section is provided by the 500A maintenance switch as indicated in Fig. 12. Since operation of a maintenance switch opens the channel and removes normal pilots, a substitute set of pilots must be supplied via the maintenance measuring path (see Fig. 12). At the receiving end of the switch section, the channel under test is routed to the STMS for making the selected out-of-service measurements.

Since the operation of a maintenance switch can be done under program control, safeguards must be provided against untimely oper-

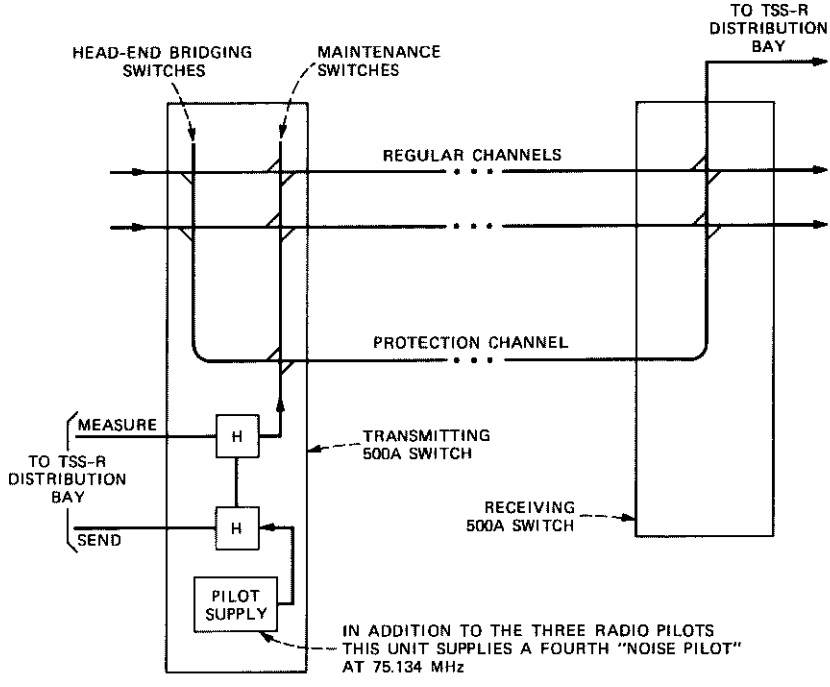



Fig. 12—Simplified diagram of switch section out-of-service test access arrangements.

ation of the switch by TSS-R. This is achieved by making the 500A control circuits the ultimate arbiter of all switching decisions.

Some out-of-service measurements require only that the head-end of the channel under test be terminated. Other tests require the insertion and measurement of a test signal at the head end of the channel as well as its measurement at the receive end. Provisions for inserting test signals, combining them with pilots, and providing head-end measuring access are shown in Fig. 12. For example, to measure out-of-service amplitude response, a comb signal with a tone spacing of 130 kHz that encompasses the entire IF band is available from the distribution bay. This signal, when needed, is sent to the test signal input port (Fig. 12) and measured at each end of the channel under test. The measured data are returned to the TSS-R central minicomputer where a comparison of transmitting and receiving levels provides the desired amplitude response characteristic for the channel.

Special mention is made of the following out-of-service switch-section measuring capabilities:

1. Spurious modulation—This involves the insertion of a +5 dBm0 test tone (83.168 MHz) at the head end of the switch section. At the receiving end, TSS-R looks for spurious modulation sidebands on the

test tone. These might be due to modulation of the test tone by spurious switching tones from a bad power supply, spurious modulation by a faulty microwave local oscillator, or spurious modulation by some other unwanted mechanism.

2. Linearity—This measurement is of special interest for verifying continued satisfactory adjustment of transmitter predistorters. Two high-level (+17 dBm0) test tones located ±44 kHz on either side of the 74.13-MHz channel center frequency are inserted at the head end of the switch section. Third-order nonlinearities produce IM products located 88 kHz on either side of the radio-line pilots. These products are measured at the receiving end of the switch section as an indication of switch section linearity. Positioning the +17 dBm0 full-load test tones close to band center minimizes the impact of cochannel interference on other systems. In other words, the tones are no worse from an interference standpoint than cochannel FM carriers.

3. Transmitter gain check—The +17 dBm0 test tones mentioned in (2) can also be used as a coarse check on transmitter gain. When turned on at the head end of a switch section, power monitors on the output of each succeeding transmitter provide a remote indication if the power is within a ±3 dB window of nominal.

A complete listing of all in-service and out-of-service tests is given below.

1. In-service tests—switch section
   (a) Pilots—level and frequency
   (b) Intermastergroup noise
   (c) Pilot slot noise—before pilot insertion
2. Out-of-service tests—switch section and hop by hop
   (a) Idle noise
   (b) Noise and tone scan
   (c) Linearity
   (d) Amplitude response
   (e) Interference
   (f) Cross-polarization discrimination
   (g) Transmitter gain
   (h) Space-diversity switch point
3. Other tests
   (a) Head-end pilot supply on 500A
   (b) Phase jitter—in-service and out-of-service
   (c) Diagnostic tests

#### *4.9.3 Stress-test and trouble-isolation features*

Distinct from, but enhanced by, TSS-R are a number of supplementary features for stress testing and trouble isolation. Central to implementing these features is the capability to remotely operate any

selected resupply switch at intermediate repeaters via the alarm reporting and command system. When a resupply switch is operated, a locally generated set of pilots is inserted into the channel under test. By measuring the channel with TSS-R at the receiving end of the switch section, it is possible to characterize systematically shortened portions of the switch section to isolate trouble.

To ensure that a resupply switch can only be operated remotely on an out-of-service channel, the switch command is enabled by the presence of the noise pilot on the out-of-service channel (see Fig. 12). Since the noise pilot can only be present subsequent to the operation of a head-end maintenance switch and the prior transfer of service to a protection channel, the requisite protection is provided.

An important additional function of the noise pilot is associated with the switching system preemption feature. If TSS-R is measuring a foreshortened portion of a channel in a switch section, rapid restoral of the intermediate resupply switch is essential if the channel under test has to be preempted for service. Since a first step in the preemption process is to pull down the head-end maintenance switch and thereby remove the noise pilot provided by the head-end resupply, all downstream resupply switches are automatically disabled and returned to normal. This avoids the slow restoral procedure that would result if resupply switches had to be reset via the alarm reporting and command system.

Trouble isolation capabilities associated with the remote control of resupply switches are further enhanced by providing means to remotely modify the resupply signal. For example, the two high-level test tones used for switch section linearity measurements can also be turned on at intermediate resupply points. This provides the capability to isolate hops with high nonlinearity by looking at systematically foreshortened portions of the switch section. The isolation of linearity problems to a particular hop is further enhanced by turning on the aforementioned linearity test signal at a selected intermediate station; but instead of applying the radio pilots at their normal −10 dBm0 level, they are applied at 0 dBm0. This enhances the third-order nonlinearity product contributed by the first hop of the foreshortened switch section by 10 dB. At succeeding receivers on the switch section, AGC restores the radio pilots to −10 dBm0 and correspondingly reduces the high-level test tones by 10 dB to +7 dBm0. Thus, the third-order products contributed by succeeding hops are reduced by 20 dB, thereby making the first hop contribution dominant.

Another stress test feature is provided by remotely commanding the resupply pilots to be inserted 32 dB below their normal level of −10 dBm0. When this is done, AGC on the first receiver beyond the resupply point increases its gain by 32 dB to restore the pilots to their

normal −10 dBm0 level at the receiver output. This enhances the front-end idle noise contribution of that particular receiver by a corresponding amount and will, therefore, be dominant when TSS-R measures noise at the end of the switch section. In effect, therefore, this measurement gives an indication of the receiver noise figure on the artificially faded hop. There is a corresponding 32-dB enhancement of incident interference on the faded hop providing a means for identifying the interference level and point of entry. This same technique may also be used to measure the level of the nearest radio pilot in the adjacent AR6A channel to obtain a measure of cross-polarization discrimination on a hop-by-hop basis.

## V. TESTING

### *5.1 Laboratory testing*

A test facility was established at Merrimack Valley to perform initial testing of AR6A bays. This test facility served the needs of developing shop and field testing information, initial testing for radio bay components, and pursuing problems observed in the later planned field trial program. Two sets of frequency-compatible bays were installed along with a support bay. The bays were configured so that each set had a main-station-configured bay and a repeater-station-configured bay. This arrangement allowed for the greatest flexibility for an ongoing testing program.

Figure 13 depicts the laboratory bay arrangement. Bays 1 and 2 were configured as main station bays. The waveguide runs connecting repeater bays 3 and 4 contain RF preamplifiers and the loss of each run was built out to simulate a nominal hop length of 27.1 miles (61.7 dB of section loss). Two MCSS supplies were available in the support

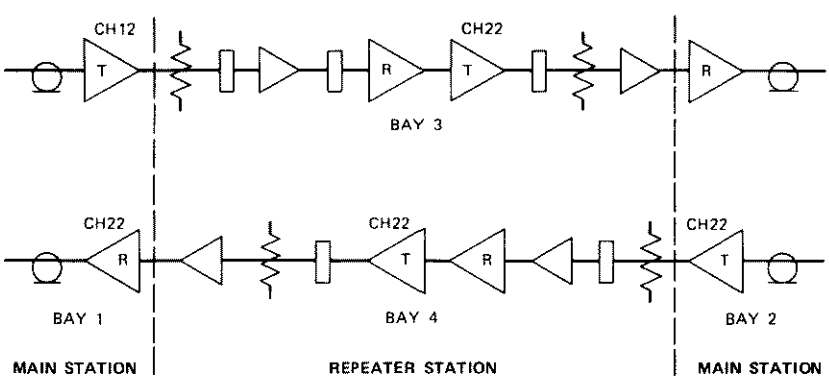



Fig. 13—Laboratory bay arrangement.

bay to allow for all bays to be locked to the same unit or for separate MCSS signals for bays 1 and 2 as testing required.

Initial testing concentrated on tests and measurements to ensure that an assembled bay met system requirements. Extensive tests were performed to isolate and reduce noise pickup and cross-coupling of signals in the bay wiring. These and results from the later field trial showed that a frame filter was necessary to filter the power from the −24 volt plant if noise objectives were to be met.

Modular design of the radio bay implied that a set of pretested modules could be inserted into a basic framework and a working bay obtained with a minimum of alignment. Initial testing was directed toward this goal. It was necessary to specify a series of tests for each module to ensure that when a set of modules was assembled into a bay, the bay performed satisfactorily. Test procedures for the overall bay were then developed for bay alignment. Once these procedures were deemed sufficient, they were incorporated into Bell System Practices (BSPs), which were then debugged using the laboratory bay facility.

### *5.2 String tests*

Since all components of the AR6A System were designed and available at Merrimack Valley, a tandem test arrangement was achieved by utilizing trunking between laboratories to construct a complete system comprised of MGTBs, MMGT-R multiplexing equipment, 500A Switching System, and the radio bays. This permitted injecting a signal into the system at basic mastergroup level and testing for switching transients with tandem units. A fade could be introduced producing a space-diversity switch, and as the fade was increased, correct switching activity for the 500A switch and the pilot resupply function in a radio bay could be observed and tested.

Noise load tests were performed on the tandem system, first on the MMGT-R connected back to back, then with the loopback at the 500A switch, and finally with a radio bay. This allowed testing of up to a four-hop switch section.

Phase jitter was also investigated on the system. Figure 14 depicts the test setup for a typical measurement. Pictures A and B in the figure show examples of the demodulated phase-jitter voltage display on an oscilloscope and a spectrum analyzer, respectively. Peak-to-peak switch-section jitter, measured in this fashion, is typically 1.2 degrees.

This laboratory facility was used constantly over the next two years as the minitrial was conducted in western Massachusetts and the field trial was conducted in Missouri. Problems seen in the field were

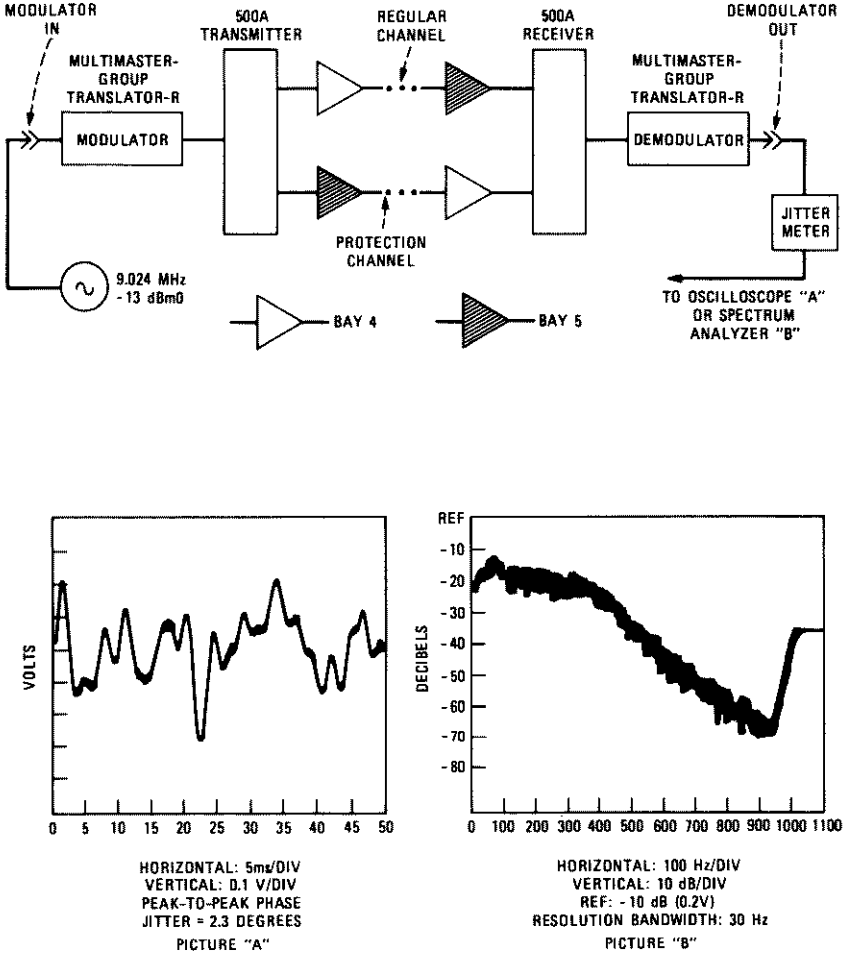



Fig. 14—Phase-jitter measurements.

pursued on the laboratory setup, thus eliminating a great deal of travel time and allowing testing, which led to a rapid resolution of problems.

For example, during the minitrial in western Massachusetts, a fade deep enough to produce either a space-diversity switch or pilot resupply switch caused multiple switching activity. The laboratory setup showed that although the average signal level was insufficient to call for the switch, switching was being initiated by noise spikes due to a peak detector with minimum filtering. Modifications were made to the detector circuitry such that switching was done on an average value over a few milliseconds rather than on an instantaneous peak value, thus eliminating the great number of switches that were occurring.

### 5.3 Ashburnham-Wendell field trial

#### 5.3.1 Configuration

A field trial consisting of a one-hop loopback of AR6A prototype radio bays was conducted during the period of October 1977 to July 1978 between Northeastern Area Long Lines radio stations in Ashburnham and Wendell, Massachusetts. The Ashburnham station was equipped as a main station and the Wendell station as a repeater site. The hop length was 29.3 miles, and BTL model 656A microwave preamplifiers were used at both receivers. The Ashburnham radio bay transmitted on radio channel 22 (6226.89 MHz) with vertical polarization and the Wendell bay transmitted on channel 12 (5974.85 MHz) with horizontal polarization, which made both bays standard in their channel arrangements. Both stations had prototype support bays that contained the pilot resupply and microwave carrier synchronization supply (MCSS) equipment. In addition to the radio equipment, each station was equipped with a PDP-8 computer which served as a Data Acquisition System (DAS) for complete monitoring and recording of all bay alarm and indication activity. A simplified diagram of the trial equipment is shown in Fig. 15.

#### 5.3.2 Installation and alignment

As a test of the modular design concept Western Electric installers installed the basic bay framework and wiring. The actual assembly of the TR modules in the bay framework was performed by AT&T Long Lines personnel. The installation process was photographed and these photographs along with the documentation from Bell Laboratories were used to generate the installation Bell System Practices.

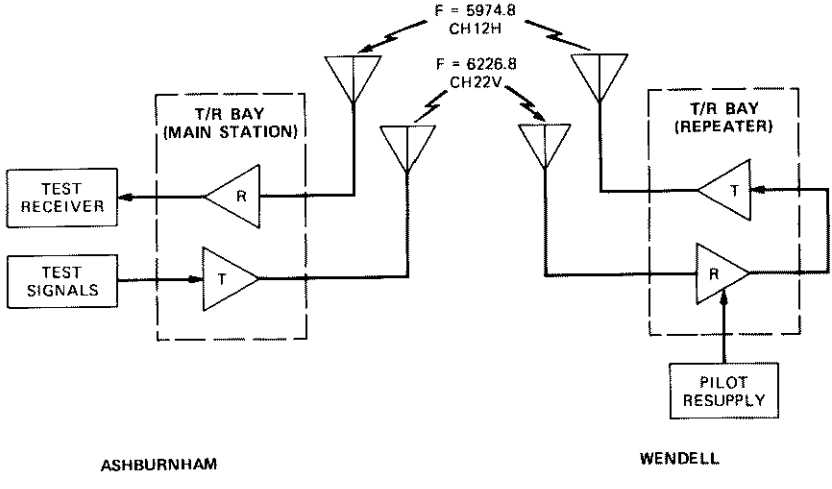



Fig. 15—Ashburnham-Wendell field trial equipment configuration.

#### *5.3.3 System performance*

Prior to shipping, the AR6A radio equipment underwent extensive laboratory measurements to fully characterize it. The measurements included thermal and intermodulation (IM) noise, and amplitude response. Initial system measurements made on the Ashburnham-Wendell installation showed good agreement with the laboratory results. Received signal levels agreed with calculated values indicating nominal section loss in the two directions of transmission. These results were further confirmed by thermal noise measurements. Long-term measurements were conducted on received signal strength, IM performance, and thermal noise. The results of these measurements indicated performance adequate or better than that required to meet system requirements. In particular, the long-term behavior of the IM noise was encouraging, indicating that the predistorter realignment intervals would be acceptably long. Examination of the received signal strength and noise slot data during fading activity indicated that the AGC amplifier and dynamic equalizer responded adequately and that space-diversity switch requests were generated at the planned level of 36 dB of fading.

### *5.4 Factory testing*

Historically, radio bays have been installed and tested on site by Western Electric installers. Fully tested and operating bays are then turned over to the operating company. The AR6A modularity concept changed this procedure to one where the customer receives factory tested modules and assembles them to produce a working bay. To ensure that the customer could bring his equipment on line in a timely manner, two procedures were initiated. First, a level of spares was specified that should ensure inhand replacement units to offset initial failures and units damaged in shipment. Secondly, a factory testing program was initiated requiring that all modules be assembled in a test bay and functionally tested before shipment. Both of these procedures are now standard practice.

#### *5.4.1 Bay alignment*

The set of modules comprising a radio bay are installed into test bays and then the bay is aligned. Any units not functioning correctly are replaced and trouble sheets on the failed units are filed. The data collected in this procedure are then used to identify problem units and the type of problem often helped identify the solution.

#### *5.4.2 Back-to-back testing*

To ensure that a quality product was shipped from the factory, the eight performance tests listed below were performed on each aligned bay:

1. Amplitude response
2. Phase jitter—TR pair
3. Phase jitter—microwave generator
4. Spurious tones
5. Spurious modulation
6. Thermal noise
7. Intermodulation noise
8. Pilot resupply operation.

When factory production was first begun, the bays were produced in frequency-compatible pairs. After bay alignment was completed, the pair was connected back to back (at RF) and the performance tests conducted. As production increased, test bays containing a frequency shifter were constructed so that the performance test could be performed on individual bays. After 126 bays had been tested it was determined that other bay tests were sufficient, so that back-to-back tests were discontinued.

### *5.5 Missouri trial*

During the period from July 1979 through June 1980, an initial evaluation of AR6A was conducted on a six-hop trial switch section between Hillsboro and Windsor, Missouri. This was part of a new 6-GHz route being established between Hillsboro, Missouri, and La Cygne, Kansas, for the first service route of AR6A (see Fig. 16). To allow field evaluation of upper and lower sideband transmitter modulators, radio channel 4 employing the lower sideband and channel 5 employing the upper sideband were installed. This arrangement also allowed for the measurement of adjacent-channel interference, an important factor in overall noise performance.

#### *5.5.1 Installation*

The 6-hop trial route was equipped as a two-way, 1 × 1 frequency-diversity protection system. Hillsboro and Windsor were equipped as terminal main stations and the five stations, Richwood, Rosati, Brinktown, Barnett, and Cole Camp were the connecting repeater stations. Installed at each of the repeater stations were four repeater bays and a support bay. Hillsboro, which is a terminal main station, was equipped with standard radio bays configured for main station application. In addition to the radio bays, a 500A protection switch, MMGT-R, 500B protection switch, and MGTB multiplex were also installed.

Windsor is a repeater station on the final expansion of the route to La Cygne, but for this trial was to serve as a terminal main station for AR6A. This required that two transmitter-only bays and two receiver-only bays (standard options) be installed at Windsor for normal

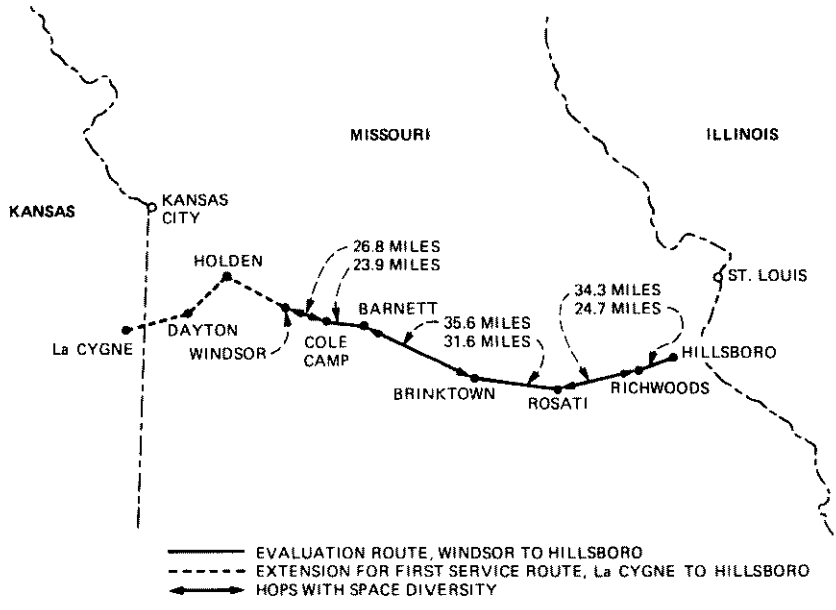



Fig. 16—First service route of the AR6A System.

connection to the indoor waveguide. And, of course, a 500A protection switch, MMGT-R, 500B protection switch, and MGTB multiplex were also installed. To allow signals transmitted from Hillsboro to be looped back to Hillsboro, a set of mastergroup connectors also were installed at Windsor.

Alternate hops on this route were equipped with the space-diversity option. This required that a space-diversity antenna be installed at all stations but Hillsboro.

A Bell Laboratories prototype TSS-R remote distribution bay was installed at both Hillsboro and Windsor. This allowed field evaluation of the distribution bays and their interaction with the TSS-R central minicomputers located at Bell Laboratories.

In addition to the radio equipment, each station was supplied with a PDP-8 computer that served as a Data Acquisition System for complete monitoring and recording of all bay alarm and indication activity. This system was connected to a PDP-8 computer at Bell Laboratories, Merrimack Valley, to allow for continual monitoring of the trial.

### *5.5.2 Antenna tests*

The trial route, being part of a new 6-GHz route, had newly installed antennas between Hillsboro and Richwood and newly installed space-diversity antennas. A series of antenna measurements were made to

characterize the transmission environment for AR6A radio. The measured parameters included section loss (SL), amplitude response, cross-polarization discrimination (XPD), and front-to-back (FB) ratio. In addition, checks for RF interference were made at each receiving antenna. The test results indicated generally good antenna performance; however, one antenna at Cole Camp did have to be reoriented slightly to meet AR6A System requirements. Junction interference from a TH-3 crossing route at Rosati was severe and had the potential to cause an improper sequence of operation of the space-diversity and 500A switching systems. This situation was corrected before the AR6A hop and switch section tests began.

#### *5.5.3 Hop tests*

The installation and alignment of the TR bays were performed by AT&T Long Lines technicians and served as a field trial for the initial issue of the BSPs. Radio bay hop tests were then performed jointly by AT&T Long Lines communication technicians and Bell Laboratories personnel. These tests afforded an opportunity to update and refine the hop turn-up BSP. These tests check such parameters as path loss space-diversity switch point, and pilot resupply initiate switch point.

#### *5.5.4 Switch section tests*

Switch section tests were performed in three parts. First, those tests necessary to condition a switch section for service were performed. Second, a series of tests were performed to measure the channel performance, and, third, testing was done with TSS-R to verify its performance. Initial switch section testing normally is performed by TSS-R, as discussed in Section 4.9.2. However, at the start of this field trial, TSS-R was not in operation. The same measurements were performed manually as would have been performed by an operational TSS-R System to condition the switch section for service. Performance of the test in this manner served two purposes. First, a database was generated for comparison with initial TSS-R measurements and, second, the latter measured performance of the switch section would determine if the set of TSS-R tests was adequate to condition a channel for service.

The amplitude response of each of the four channels of the switch section from 500A transmitting switch to 500A receiving switch were measured. The results of these tests were then used to compute the number and type of 989-type mop-up equalizers required to equalize the channel to a 0.5 dB flatness. Figure 17 shows a typical radio channel response after amplitude equalization. These tests showed that ordering information for mop-up equalizers did not supply a proper selection of equalizers. New ordering information for the mop-

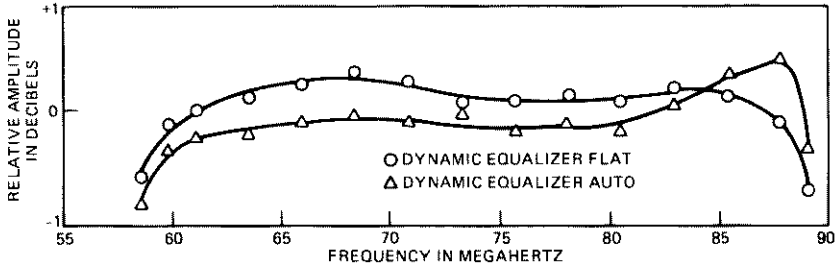



Fig. 17—Typical radio channel response after amplitude equalization.

up equalizers was generated based upon the results of these tests and analysis of factory test data from the initial bays.

The rest of the out-of-service switch section tests listed in Section 4.9.2 were performed on the four channels. The measured thermal noise was slightly higher than desired. This result was not totally unexpected since some of the initial TWTs had a higher noise figure plus gain sum than specified. This sum (noise figure plus gain) is now used as one of the screening tests for tubes used in systems now operating.

A test of system linearity was of significant importance. Hop and switch section linearity were specified based upon the system model. During the trial, it was found that even though every transmitter met its linearity requirement, some hop and switch section limits were difficult to meet. It was determined that these limits were too stringent because of such variables as hop length, systematic level error, and intermodulation noise contribution from the end hop containing a 500A switch. As a result, new limits for hop and switch section linearity were developed that were more realistic, yet ensured system integrity.

The noise and tone scan performed on each channel showed a few undesired tones in the band, which were identified as originating in the multiplex. These tones are harmonics of carriers and, with production-type filters, are eliminated. With the out-of-service tests completed, noise loading tests began. During this second phase of testing, three types of noise load tests were made. The first type was performed using a test signal generated at basic mastergroup level. Ten basic mastergroups of noise were generated and loaded onto the system at the MGTB. The second type of noise loading was performed by bridging on a live signal obtained at the mastergroup distribution frame. This allowed a comparison of random noise load results with a live signal result. The third type of noise loading used a pseudorandom noise source.[9] These tests were performed to determine (1) the level and type of nonlinearities in the system, and (2) the law of addition, i.e., how these nonlinearities add on a hop-to-hop basis.

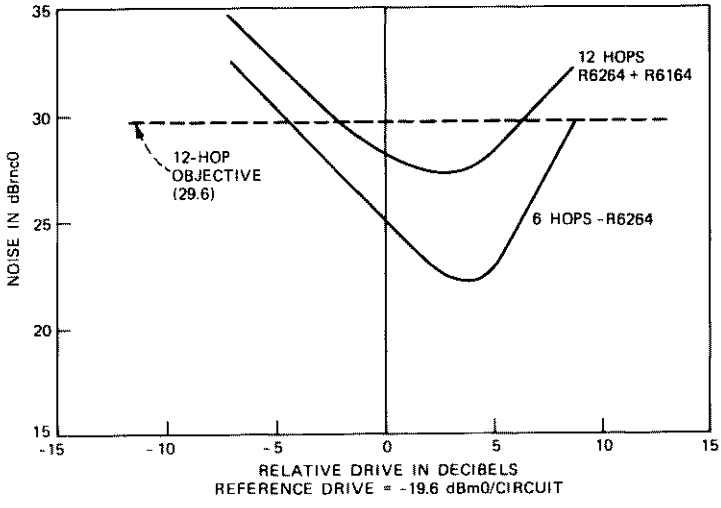



Fig. 18—Noise load tests made during the AR6A field evaluation.

Random noise load measurements were made first on each of the six-hop channels using a random noise load. This measurement was made by looking in a noise slot in each of the ten mastergroups. Figure 18 depicts the results of one such measurement. Measurements were then repeated for a 12-hop switch section by looping back at Windsor first at the 500A, and then at the MGTB employing mastergroup connectors between the transmit and receive MGTB. The results of 6- and 12-hop switch section measurements are shown in Fig. 18. These types of measurements were made early during the field trial, again after several months of operation in which no particular effort was made to keep the channel in peak operating condition. They were then repeated near the end of the trial after all transmitters were realigned to ensure their best linearity performance. A hands-off period of two months was then allowed, and the noise loading tests were repeated to obtain aging information.

Noise loading was also performed using live signal load before and after the hands-off period. The tests employing a live load used nine mastergroups bridged from the mastergroup distributing frame and one mastergroup of random noise containing the measurement slot. This mastergroup could be placed on any of the ten mastergroup positions in the channel. The results showed slightly better performance than the random signal, since the average load with a live signal is less than a noise load due to varying talker activity and volume.

Pseudorandom noise loading allowed measurement of IM distortion produced in the channel. These measurements were performed on the radio portion of the switch section only, that is, from transmitter "in"

to receiver "out" of the switch section. The test signal was applied at the transmitter "in" on the radio bay at Hillsboro and recovered from the receiving "out" port of the same bay. To complete the circuit, a patch was made at Windsor to loop back the signals between receiver and transmitter. The receiver of the test set measures the total IM signal (generated by the 12 radio bays) that falls into a narrow (3-kHz) band. These measurements were made for the expected range of system signal powers to ensure that the IM products were within limits for all operating conditions.

As we discussed in Section 4.6.2, the IM products and their law of addition when a large number of repeaters are connected in tandem is of critical importance. A scheme was derived whereby the law of addition could be determined from a series of pseudorandom noise measurements. To perform these measurements, it was necessary to have personnel present at three stations to perform the patching so that each repeater could be measured individually and then in combination. The results of the measurements were discussed in Section 4.6.2 with the results shown in Fig. 7.

A period of the trial was devoted to TSS-R testing. During this phase of testing, each of the TSS-R measurements was performed remotely from the Merrimack Valley central computer and then repeated manually on site. This was to ensure that both the proper measurement was being made, that it was being made on the intended unit and, finally, that the results were correct. Maintenance of the AR6A System relies upon a combination of SCOTS alarm and radio indications and TSS-R in-service and out-of-service measurements. When a problem occurred on a channel, the indications from SCOTS in conjunction with TSS-R measurements were used to identify the problem and its location, so that maintenance personnel could be dispatched to that location with the necessary equipment and spare parts to effect a repair. As a result of this brief evaluation period, the maintenance scheme was judged viable, but further documentation was necessary for the users to take full advantage of the powerful tools put at their disposal.

During the course of the field trial, several minor problems were uncovered in the functional operation of both radio bays and associated 500A Switching System and the multiplex. Identification of these problems allowed for the necessary circuit changes to be implemented into the product prior to full production.

## VI. HOT STANDBY

### *6.1 System configuration*

#### *6.1.1 Repeaters*

As we mentioned in Section 1.5, the AR6A hot-standby configura-

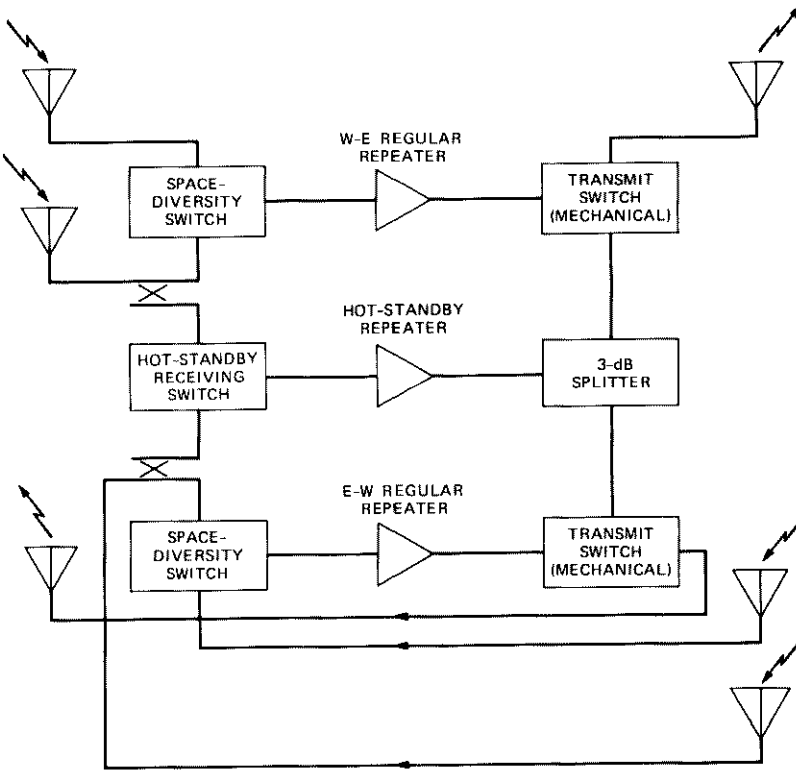



Fig. 19—Hot-standby repeater configuration with space diversity.

tion uses a single hot-standby repeater to protect both directions of transmission. This is illustrated in Fig. 19, which shows a repeater configuration using space diversity.

The primary monitoring function determining whether a TR bay is in satisfactory condition is provided by a broadband power monitor at the transmitter output. Under some trouble conditions associated with high receiver gain, it would be possible for broadband receiver noise to look like a message load, thereby deceiving the power monitor. To avoid this difficulty, high receiver gain operates the pilot resupply switch at the receiver output in the same way as it does in frequency-diversity applications. However, since resupply pilots are not needed for hot standby, the normal pilot input port can be terminated. Thus, when the resupply switch is operated, the transmitter input is automatically terminated, providing the equivalent of a squelch.

Figure 19 shows that in the hot-standby mode the hot-standby receiver does not have space diversity. This eliminates the necessity and complication of transferring the space-diversity control function

between regular and hot-standby bays. The simplification is justified on the grounds that the estimated simultaneity of equipment failure and fading is small enough to be neglected.

The microprocessor-based control unit, in addition to monitoring transmitter power output and other status indicators, arranges the sequence of switch operations during the transfer to or from hot standby. In addition, it provides local and remote status information and supervises local and remote commands with predetermined override priorities. Important considerations relating to the reliability of the 1 × 2 hot-standby configuration are discussed in Section 6.2.

### *6.1.2 Main stations*

Main stations do not lend themselves to a 1 × 2 hot-standby protection arrangement since in general there is only one TR unit per channel at these locations. As a result, main station protection is provided on a one-for-one basis for both transmitters and receivers. A block diagram of the protection arrangement that is integrated with the 501A Protection Switching System is shown in Fig. 20. In essence, the transmitting portion of the 501A System double feeds two transmitters whose outputs are selected by a transmitting switch supervised by a control unit similar to the one used at hot-standby repeaters. On the receive side, receiver selection is performed by the 501A switch on

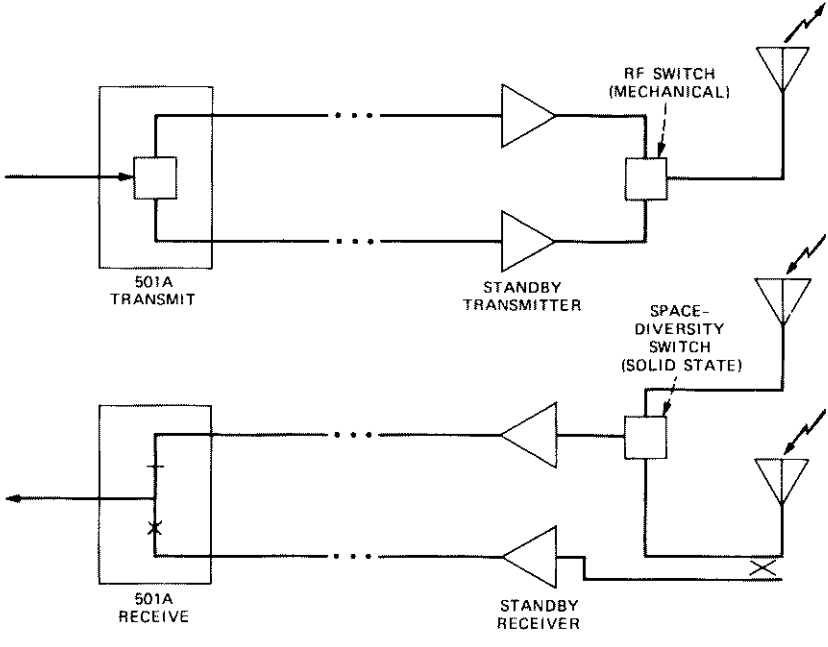



Fig. 20—Hot-standby main station configuration with space diversity.

the basis of noise and pilot monitor information identical to that employed on the 500A System.

Interface levels between the 501A System, TR equipment, and multiplex equipment are identical to those employed in the frequency-diversity 500A configuration. This simplifies the conversion from hot standby to frequency diversity when justified by growth in the route cross section.

### *6.2 Reliability considerations*

The main station 1 × 1 hot-standby configuration follows conventional design practice and will not be discussed in detail. For these systems, the use of high-reliability components and subsystems provide adequate overall equipment reliability.

The outage probability of the repeater station configuration shown in Fig. 19 can be expressed as

$$P_0 = P_B^3 + 3P_B^2(1 - P_B) + 2P_S P_B(1 - P_B)^2 + 3P_S' S_B(1 - P_B)^2 + P_S'(1 - P_B)^3, \quad (8)$$

where

$P_0$ = two-way equipment outage probability,
$P_B$ = outage probability of a radio bay,
$P_S$ = probability of switching system failures that do not cause immediate service outage, and
$P_S'$ = probability of switching system failures that cause immediate service outage.

An example of a failure included in $P_S$ is a loss of power to the switching system when it is in the normal state with both regular radio bays operable. The system equipment outage objective results in the requirement that $P_0 \leq 3.21 \times 10^{-7}$. Preliminary calculations showed that, for expected equipment failure rates and a 2.25-hour mean repair time,[12] the sum of the first four terms of eq. (8) would be less than the required $P_0$ with reasonable margin. Preliminary calculations for $P'$s, however, showed that the last term would be roughly equal to the requirement and special design features would be needed. The probability $P'$s was reduced by requiring two independent circuits to simultaneously cause the electromechanical switch shown in Fig. 21 to connect the hot-standby bay to the antenna. Failures in either circuit then cannot cause a loss of service if the opposite direction of transmission is connected to the hot-standby bay input. With this design feature, the equipment outage objective is met for the calculated equipment failure rates.

For hops having space diversity, it was found that in order to meet equipment reliability objectives the space-diversity switch would have

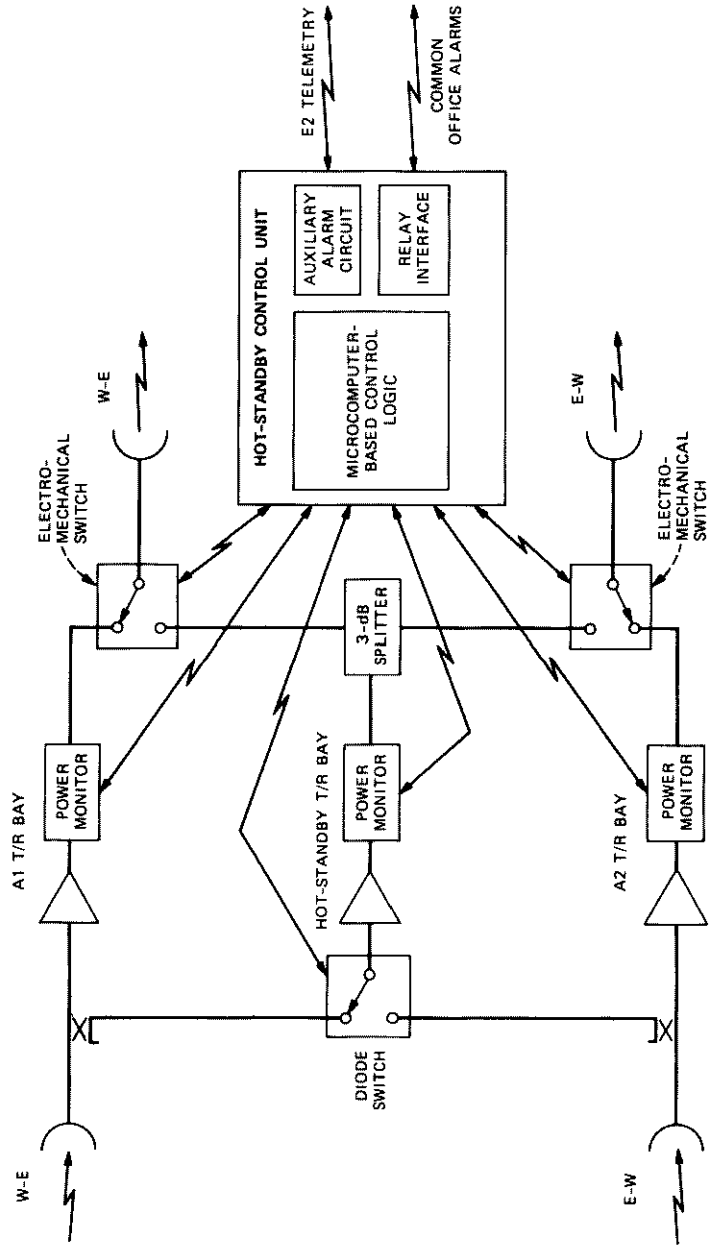



Fig. 21—Block diagram of the 1 × 2 Hot-Standby Protection System.

to be included within the transmission path protected by the standby bay. This leaves the hop without space-diversity protection when the regular radio bay fails but the calculated increase in fading outage is less than 5 percent. The overall reliability of this configuration is better than a design that leaves the switch outside the protected path.

### *6.3 Switching and control features*

A block diagram of the hot-standby switching and control system configuration for repeater stations is shown in Fig. 21. A microcomputer-based control unit is used to control all switching activity and provide equipment checking for alarm purposes. The auxiliary alarm circuit monitors dc power and provides information to the common office alarms. The relay interface conditions the alarm signals for the C1 or E2 telemetry and common office alarm systems.

The hot-standby control unit has three modes of operation: automatic, local, and remote manual. Under normal conditions in the automatic mode, the transmitter electromechanical switches connect the regular (A1 and A2) TR bays to the transmitting antennas. Radio bay failures are detected by the power monitors. A failure of either bay A1 or A2 will cause the system to replace it with the hot-standby bay. For transmitter failures, the loss of signal during switching will normally be less than 15 ms. For receiver failures, the loss will be approximately 50 ms. The standby bay will remain in service until either the regular bay returns to normal or a switch command in the local mode is issued. To revert automatically from the standby bay, the regular bay must be operating correctly for 2 seconds. If another bay should fail when the standby bay is in service, then the appropriate service failure alarm will be set. To prevent unnecessary switching due to failures at a previous repeater, a simultaneous failure of a regular bay and the hot-standby bay is interpreted as an input signal failure and no switching action is taken.

The local mode of operation is provided for maintenance and trouble isolation. The local actions include manual switching of the hot-standby bay to replace either regular bay and a lockout of the standby bay. When the local mode is entered, the pushbutton actions take precedence over switch requests indicated by the power monitors.

If the system is in the automatic mode and the standby bay is idle, the remote manual mode can be enabled by the C1 or E2 telemetry. The remote switching and lockout commands are identical to the local ones. If a service-affecting failure occurs while the system is in the remote mode, a reversion to the automatic mode takes place and the service is protected.

## VII. CONCLUSION

The economic attractiveness and efficient spectrum use of the AR6A

System has led to its rapid deployment throughout the United States. The relatively trouble-free introductory experience with this new technology system is a tribute to the many people who contributed to its development. As the authors of this overview article reporting on the work of many colleagues, we wish to acknowledge the contributions of all who participated in the project.

## REFERENCES


1. R. I. Felsberg and M. E. Sands, "The AR6A Single-Sideband Microwave Radio System: Special Test Equipment," B.S.T.J., this issue.
2. R. P. Choyce and G. J. O'Connor, unpublished work.
3. W. C. Ahern, F. P. Duffy, and J. A. Maher, "Speech Signal Power in the Switched Message Network," B.S.T.J., *57*, No. 7, Part 2 (September 1978), pp. 2695–726.
4. Bell System Technical Reference, "Transmission Parameters Affecting Voiceband Data Transmission—Measuring Techniques," May 1975, AT&T Publication 41009.
5. G. M. Babler, "A Study of Frequency Selective Fading for a Microwave Line-of-Sight Narrowband Radio Channel," B.S.T.J., *51*, No. 3 (March 1972), pp. 731–57.
6. G. M. Babler, unpublished work.
7. N. O. Burgess, R. C. Maclean, G. J. Mandeville, D. I. Mclean, M. E. Sands, and R. P. Snicer, "The AR6A Single-Sideband Microwave Radio System: Equalization for Multipath Fading," B.S.T.J., this issue.
8. Yen-Long Kuo, IEEE Trans. Instrumentation and Measurement, *IM-22*, No. 3 (September 1973), pp. 246–9.
9. R. H. Moseley, S. B. Pirkau, and R. B. Swerdlow, "Pseudorandom Noise Loading for Noise Power Ratio and Law of Addition," B.S.T.J., *56*, No. 4 (April 1977), pp. 511–34.
10. A. Vigants, "Space Diversity Engineering," B.S.T.J., *54*, No. 1 (January 1975), pp. 103–42.
11. W. T. Barnett, "Multipath Propagation at 4, 6, and 11 GHz," B.S.T.J., *51*, No. 2 (February 1972), pp. 321–61.
12. W. Y-S Chen, "Estimated Outage in Long-Haul Radio Relay Systems," B.S.T.J., *50*, No. 4 (April 1971), pp. 1455–85.


## AUTHORS


**James Gammie,** B.S. (EE), 1944, University of Aberdeen, Scotland; B.S. (Mathematics), 1951, Birkbeck College, University of London; Bell Laboratories, 1952—. Mr. Gammie's early assignments at Bell Laboratories included circuit design work on the L3 Coaxial Cable System and the TJ, TL, and TM series of short-haul microwave radio systems. More recently, he had responsibility for the development of 3- and 4-type solid-state FM terminals. Since 1975 he has headed the group responsible for the system design aspects of the AR6A Single Sideband Microwave Radio System.

**James P. Moffatt,** B.S.E.E., 1963, Princeton University; M.E.E., 1965, Ph.D., 1968 (Electrical Engineering), North Carolina State University at Raleigh; U. S. Army 1968–1970; Bell Laboratories, 1970—. Mr. Moffatt has worked in various areas contributing to the development of microwave radio systems. He is currently involved in the development of long-haul digital radio systems. Member, Phi Kappa Phi, IEEE.

**Richard H. Moseley,** Associate Degree (EE), 1953, Connecticut State Technical Institute; B.S. (EE), 1965, M.S. (EE), 1969, Newark College of Engi-

neering; Bell Laboratories, 1953—. Mr. Moseley worked on various defense systems until joining the Single Sideband Microwave Radio Department in 1970. He is currently engaged in digital radio systems design.

**Willard A. Robinson,** B.S.E.E., 1962, New Jersey Institute of Technology; M.E.E., 1964, New York University; Bell Laboratories, 1954—. Until 1970 Mr. Robinson worked on missile guidance systems and research for antiballistic missile systems. Since then he has been engaged in the development of the AR6A System, particularly in the areas of system studies, testing, and evaluation. His current assignment is system engineering for new features for the AR6A System. He holds a patent (co-inventor) on a radar range tracking unit used in missile guidance systems. In July 1982, Mr. Robinson received a Bell Laboratories Distinguished Technical Staff Award. Member, Eta Kappa Nu, IEEE.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Radio-Line Physical Design

By S. A. HARVEY* and P. D. PATEL*

(Manuscript received February 7, 1983)

The physical design of the AR6A radio-line equipment is the product of combining the system operating requirements with the technology available to meet those requirements economically. As part of the description of the radio-line equipment, we discuss the practical considerations for partitioning the circuit block diagrams into realizable units. Modularity is a new feature incorporated into the design of the transmitter and receiver radio bay. We also describe a novel thermal design detail used with discrete transistors in the IF circuits to ensure their reliable operation.

## I. INTRODUCTION

AR6A† radio is a new high-capacity, long-haul microwave radio relay system operating in the 6-GHz common carrier band. The use of single-sideband modulation for this high-capacity system was made possible only through the achievement of a high degree of system linearity, which was obtained by designing each circuit within the AR6A System to meet a linearity performance objective. Closely coupled with the system and circuits development of AR6A was the physical design of the radio-line equipment. The Transmitter-Receiver

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

(TR)* bay, the TR support bay, and their constituent modules are the principal elements of the AR6A radio-line equipment.

The configuration of this equipment is the culmination of the physical design effort applied to the AR6A System requirements. Physical design is a product creation process that takes a design concept through an iterative development stage and then into manufacture. The driving force behind any new product development, such as AR6A, is the potential market for that product. The AR6A market was identified as the need for a high-capacity, long-haul 6-GHz microwave radio system to overbuild the existing operating companies' long-haul 4-GHz TD-2 and TD-3 radio routes. The principal operating environment for the AR6A radio-line equipment would be within the existing radio station buildings of the TD radio network.

The physical design process began when the AR6A transmitter-receiver was transformed from a systems concept into a functional block diagram. Partitioning the system block diagram into realizable circuit modules was the first step toward the physical design concept. Consideration for the existing buildings and the goals of the operation and maintenance plans was used to determine several fundamental physical design objectives. These objectives were: (1) to assemble a single AR6A transmitter and receiver unit into one radio bay framework, (2) to assemble the common radio-line equipment into one support bay unit, (3) to design for modular field assembly of the TR bay, and (4) to design the radio-line equipment for a nonair-conditioned building environment.

The physical design effort can be divided into two categories: (1) an overall equipment design level, and (2) an individual module design level. The overall equipment design effort is concerned with putting together the many individual circuit modules into one functioning unit, while at the individual design level we concentrated on providing the required circuit functions within the physical bounds determined by the overall equipment design concept. These two categories of design must go on together during development as neither can proceed very far without some knowledge of the other.

The overall radio-line equipment concept was the outcome of an iterative process of configuring each circuit module and then arranging the modules spatially within the framework so that all the physical design objectives were met simultaneously. The particular configuration of each circuit module was influenced by three principal factors: (1) the technology available to implement the required circuit function, (2) the thermal design requirements of the module, and (3) the spatial

* Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

relationship of the circuit module with respect to the overall equipment concept. The final physical form of each module emerged from the integration of its individual design objectives into the overall equipment concept.

The challenge of this physical design process was to produce AR6A radio-line equipment designs which met both the system performance requirements, and were economically attractive to purchase, to operate, and to maintain.

## II. RADIO-LINE EQUIPMENT DESCRIPTION

Figure 1 is a simplified block diagram showing the AR6A transmitter, receiver, and the common radio-line functions. The physical design realizations of this functional diagram are the AR6A TR bay and the AR6A TR support bay.

### *2.1 Transmitter-receiver bay*

The AR6A TR bay, as shown in Fig. 2, is assembled onto a 19-inch panel framework with overall dimensions of 22-3/8 inches wide by 15-1/2 inches deep by 9 feet tall. All the TR bay components are mounted below the 7-foot level in accordance with human factors design considerations. The TR bay is composed of three sections: (1) an upper section consisting of an Intermediate Frequency (IF) shelf assembly and plug-in units, (2) a middle Radio-Frequency (RF) section made up of waveguide-type components, and (3) a lower housing for power supplies and the microwave generator. The predominant color is light gray, which is in harmony with existing radio station equipment. The rest of the color scheme is black, red, and dark blue-gray blended with natural and applied finishes on aluminum, brass, copper, and steel.

The six common alarm and control interface circuits are on printed wiring boards complete with faceplates and are located on the top left level of the IF shelf assembly. A dc-to-dc power converter, which is the common low-voltage power supply for the IF shelf, is located at the upper right-hand side. The majority of the transmitter and receiver IF circuit functions are housed in 12-inch by 12-inch by 2-inch plug-in modules. These IF modules are arranged together for ease of interconnection and operational access into the IF shelf. The transmission path connections are made coaxially across the front of the plug-in modules. A display panel, located immediately below the IF shelf, provides a surface for visual information and dresses the interface between the IF and RF sections.

The receive RF signal path from the indoor waveguide run or from an adjacent AR6A TR bay comes into the channel-separating filter at the 5-1/4 foot level and then down the center of the RF section. With the space-diversity option there are two receive signal paths and two

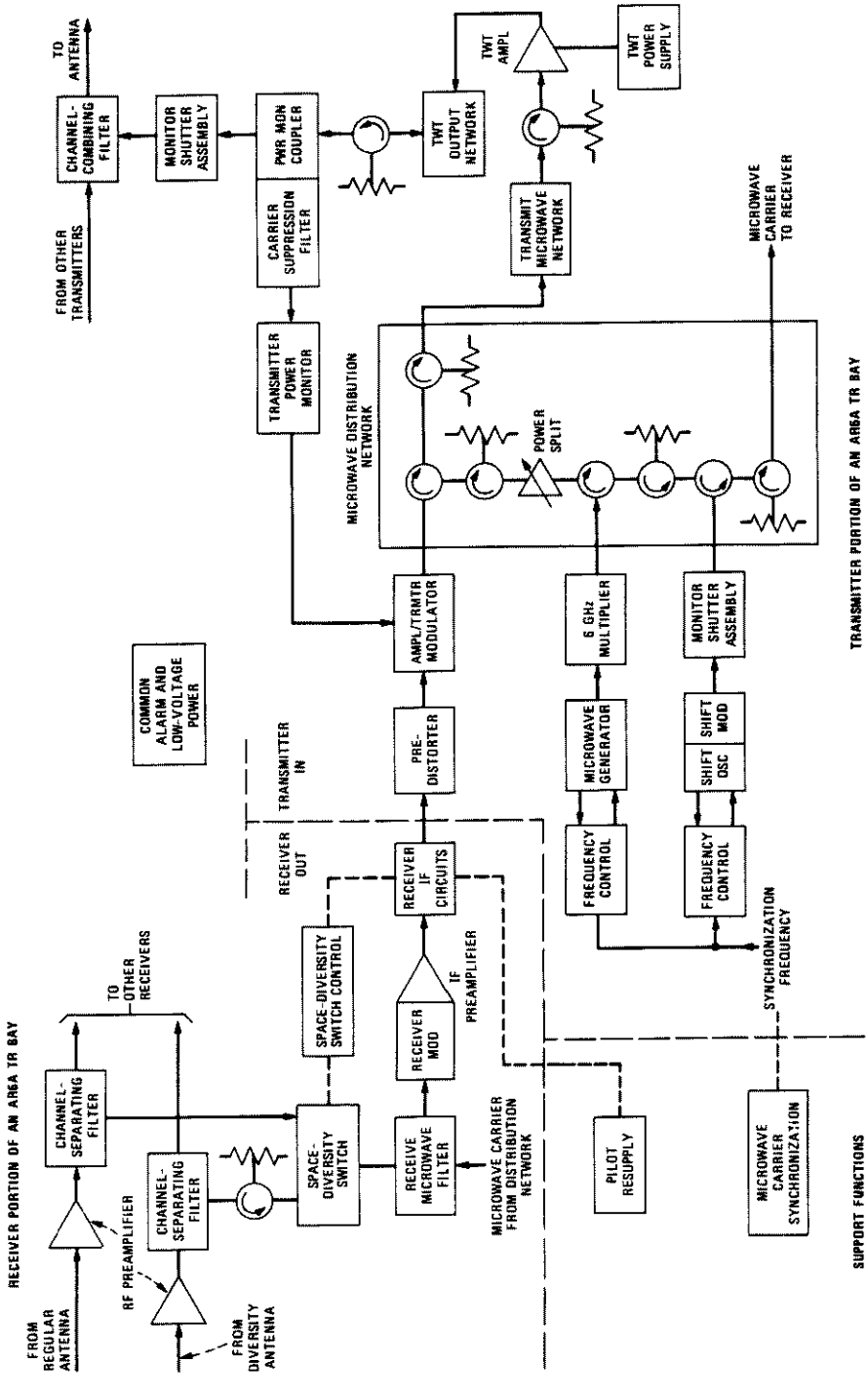



Fig. 1—Block diagram of the AR6A radio line.

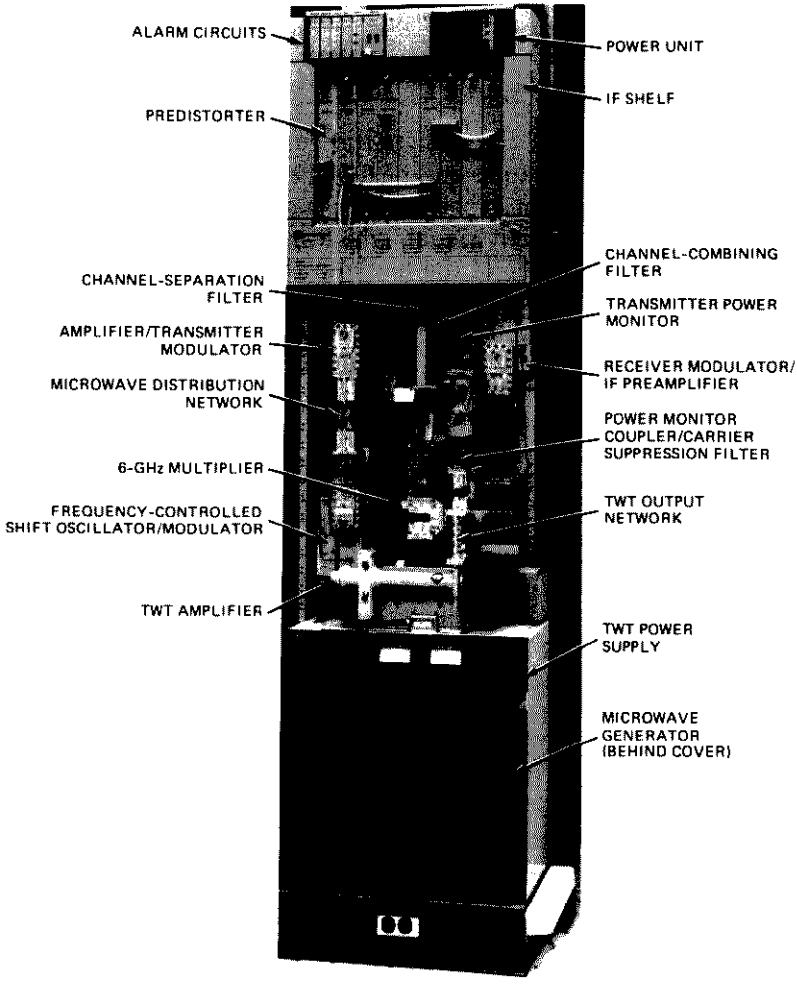



Fig. 2—AR6A transmitter-receiver bay.

channel-separating filters. The upper portions of the RF channel filters are hidden from view behind the removable display panel. The space-diversity switch allows either the RF signal from the main or the diversity antenna to be connected into the receiver. Control of the switch is provided through the switch control unit under the direction of the receiver IF section. The receive signal output from the space-diversity switch is fed through flexible waveguide up to the right, through a waveguide RF filter, and into the receiver modulator. Here the RF signal is down converted to a 59- to 89-MHz IF signal and

then amplified in the IF preamplifier unit. This IF signal is cabled up the right to the IF shelf modules where it is passed through each IF module from right to left. The IF modules include filters, amplifiers, and equalizers.

The first element of the transmitter function is the predistorter, which is the leftmost unit on the IF shelf. In a repeater TR bay configuration the output of the receiver is connected directly into the transmitter. The transmit signal, which is still in the IF format, is cabled down the left side of the bay into the IF driver amplifier and transmitter modulator. After the up-conversion process, the resulting signal is transmitted through an RF waveguide filter to the Traveling-Wave-Tube (TWT) amplifier. A transition-to-coaxial cable is used to provide the input to the TWT. The amplified RF signal continues up through a reduced-height waveguide filter and into a waveguide directional coupler. A small sample of the output RF signal is passed through a waveguide filter and then coaxially fed into a power monitor. The RF signal path continues up through a monitor port and into a fixed waveguide-to-coaxial-to-waveguide network. This network makes a compact RF connection to the channel-combining filter. The transmit RF signal path continues out from the combining filter into an adjacent AR6A TR bay or into the indoor waveguide run.

The TWT high-voltage power supply is located in the lower housing directly below the traveling-wave-tube amplifier. The outer shell of the TWT power supply fills the available space for safety and operational considerations. The distinctive shape and color forms a visual transition between the RF section and the lower third of the TR bay. The microwave generator and a −19V dc power regulator unit are mounted behind the cover in the lower housing. The microwave generator is an oven-controlled crystal oscillator with multiplier stages to achieve a 1-GHz output. An external reference frequency is used to stabilize the output frequency. The 1-GHz output from the microwave generator is fed coaxially up to the RF section where a microstrip RF multiplier unit provides the local oscillator signal. This local oscillator signal is supplied via coaxial cable to a distribution network where it is split into two signal paths. One signal path goes directly to the transmitter modulator and the other signal goes to a frequency shifter. The shift oscillator/modulator shifts the transmitter local oscillator signal frequency by 252 MHz to provide the receiver with its local oscillator signal. The shift oscillator is also stabilized by an external reference through a frequency control unit.

### *2.2 TR support bay*

The common functions that are required to operate and maintain the AR6A radio-line transmission path have been configured into a

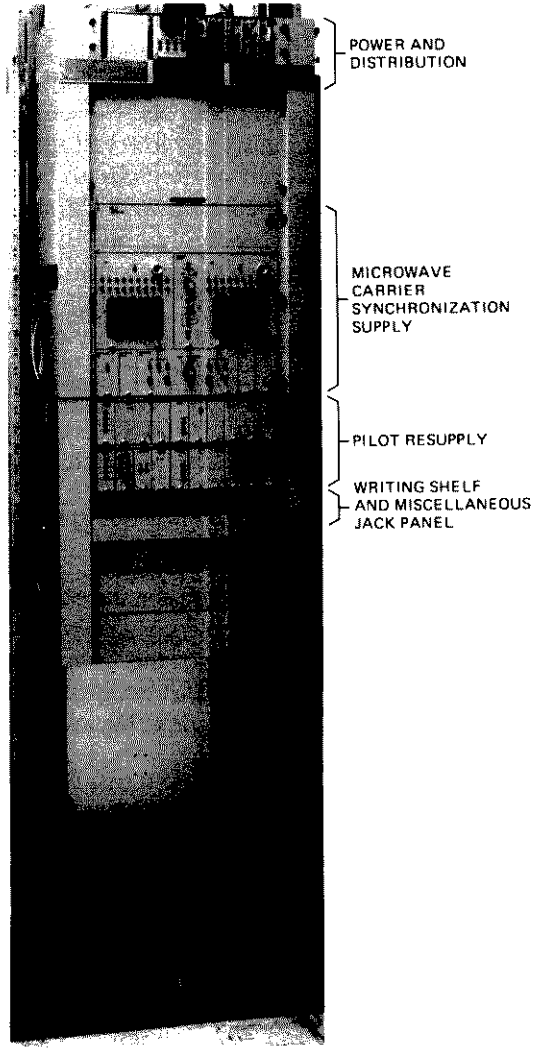



Fig. 3—AR6A transmitter-receiver support bay.

TR support bay unit, which is shown in Fig. 3. These common functions are: (1) the station reference oscillator and its control circuits for synchronization of the local oscillator frequencies within the TR bays, (2) the radio-line pilot resupply oscillators, (3) the additional oscillators and control circuits for out-of-service testing of the TR bays, and (4) the fusing and alarm circuits for the common waveguide RF preamplifiers.

The TR support bay uses the same size framework as does the TR

bay so that it can be placed compatibly in the same equipment lineup as the TR bay. Its color scheme is light gray and black in keeping with the TR bay. As with TR bay, all the TR support bay components are mounted below the 7-foot level. The upper shelf assembly provides the mounting for the common TR support bay low-voltage power converters and for the fuse and alarm circuit printed circuit boards associated with the RF preamplifiers. Both the Microwave Carrier Synchronization Supply (MCSS) and the pilot resupply units, configured as shelf assemblies with plug-in modules, are positioned in the upper half of the bay framework for easy access during operation. The panel above the MCSS is reserved for the optional hot standby control unit, while the panels below the jack and access panel provide storage space for the pilot resupply distribution cables.

Additional descriptive information for the TR bay and TR support bay modules is detailed in companion articles in this issue.

## III. THE MODULAR APPROACH

An important consideration in any equipment design concept is how the individual parts are put together to form the whole. This section reviews the development of the AR6A radio-line equipment design concept as a modular approach. The AR6A radio-line equipment just described is the result of physical design planning and decision making. The planning and decision process began with the conversion of the system blocks into realizable circuits. Further refinement of the system block diagram produced the evolutional equipment partitioning into the circuit modules.

### *3.1 Modular design concept*

Recall that the physical design objectives included the design of the TR bay for modular field assembly. Station engineering and installation costs may be reduced by initially installing a complete lineup of unequipped, factory-tested, wired bay frameworks in a station, as opposed to engineering each radio-channel addition separately. Channel growth can then be accomplished by installing factory-tested TR modules, as needed, to fill out the radio route. This capability for TR bay modular field assembly provided an added challenge in the development of the radio-line equipment concept.

The initial equipment-concept decision was the selection of the 19-inch panel, duct-type framework for the radio-line equipment. We determined that the floor space available in existing radio station buildings would be sufficient for a full array of frameworks of this size. A smaller framework size would not, therefore, "save" floor space. On the contrary, a more compact framework size might create addi-

tional difficulties in engineering for the thermal design, for maintenance access, and for future changes.

A decision on the framework size could not be made without some confidence that its volume would be appropriate for the transmitter and receiver circuit modules. The size and shape of each circuit module was initially approximated based on the proposed technology for each circuit module. Alternate design options were evaluated within the criterion of meeting the overall AR6A System objectives economically. Basic decisions such as these were part of an iterative decision-making process where the impact of a decision on the adjacent levels of the physical design process was fed back to the decision makers.

Various equipment configurations of the TR bay were evaluated as possible design alternatives. The significant factors that influenced the spatial organization of the circuit modules into the final radio-line equipment concept were: (1) the modular assembly objective, (2) the human factors considerations for operation and maintenance, (3) the integration of the overall thermal design concept into the overall equipment concept, (4) the inherent size and shape of some of the circuit modules, and (5) the performance-related objective of minimizing the lengths of the transmission path circuit interconnections.

Modular field assembly was successfully accomplished by structuring the TR bay as the combination of (1) a wired basic bay, (2) a wired IF shelf assembly, and (3) an assemblage of modular RF components. The basic bay is the 19-inch panel, duct-type framework with a minimal amount of factory-installed wiring and a few mechanical piece parts to accept the mounting of the TR bay modules. The installed wiring provides an interface to the radio station dc power and alarm systems and provides the nontransmission path interconnections between the TR bay modules. Some of the mechanical piece parts are factory located with fixtures to assure the proper alignment of the RF components when they are assembled into the basic bay. This bare-bones design approach allows the basic bay to be installed in a radio station with a minimum of additional capital investment.

In the modular design concept, the IF shelf is a wired framework for plug-in modules that has a connectorized interface to the basic bay. Installation of the IF shelf and its plug-ins into the basic bay is a straightforward procedure. In contrast, the RF components did not lend themselves to a traditional modular assembly concept. However, by using the modularity guideline of ease of assembly and by carefully engineering the interface with the basic bay, an innovative modular design was achieved. Each of the RF circuit modules was connectorized, and their mounting arrangements were made easily accessible from the front of the TR bay. The frequency-sensitive waveguide components were assembled together on a supporting framework to

minimize the number of waveguide connections that would need to be made in the field. This main RF module is positioned into the basic bay on locating pins for accurate alignment of the channel filters. As an additional dividend from the modular design approach, both the factory assembly of complete TR bays and the field replacement of TR units have been made uncomplicated. The TR support bay has followed the modular design example by using wired shelf assemblies and plug-in modules. There is no advantage in field assembly, however, as the support bay must be installed completely for use with the initial group of TR bays in a station.

### *3.2 IF circuits subsystem*

The IF shelf assembly is a subsystem module within the TR bay as a whole, and, as such, it was treated as a separate entity. A close-up view of the IF shelf assembly is shown in Fig. 4. The subsystem consists of the receiver IF circuits (represented as a single block in Fig. 1) and the closely associated predistortion, alarm and interface, and low-voltage power circuits. These modules are mounted in a two-level shelf assembly that interconnects the modules via backplane wiring. The mechanical design of the shelf assembly is a combination

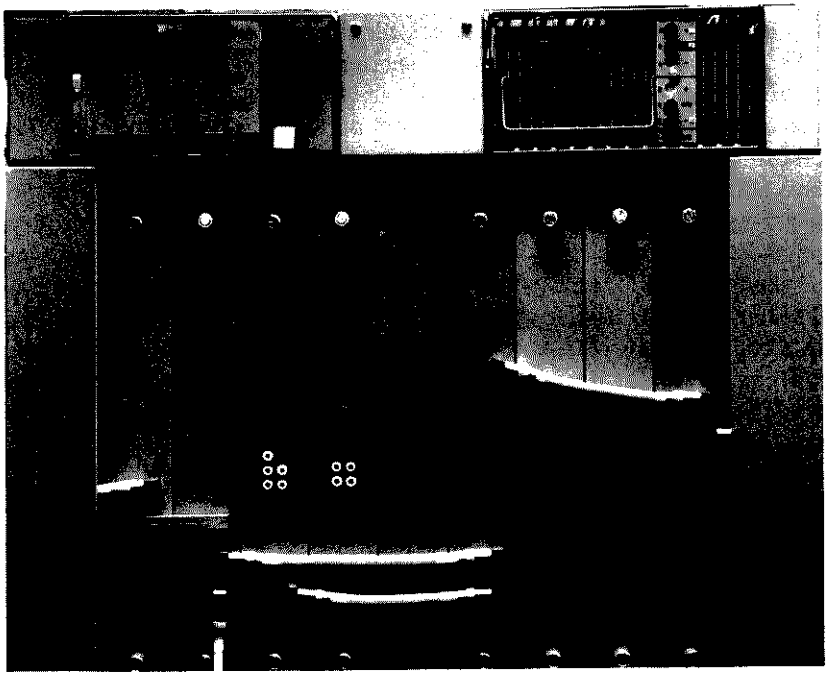


Fig. 4— IF shelf assembly.

of sheet metal fabrication and assembly techniques. Access to the bay framework mounting and the bay cabling is provided by hinged side panels.

A manufacturability cost study showed that for the IF circuits a few large integrated units would be more economical to produce than many smaller units. An electrical and physical partitioning of the receiver IF circuits was achieved based on this study. The block diagram of the receiver IF circuits is shown in Fig. 5. Each of the dashed blocks represents an identifiable circuit function. These circuit functions have been packaged as individual modules: filter/equalizer, dynamic equalizer, Automatic Gain Control (AGC) amplifier, receiver control, pilot detector, and resupply switch. A spare plug-in space is available for future needs. Further description of these modules is found in Ref. 1.

## IV. TECHNOLOGY SELECTION

The single most important physical design requirement is that the manufactured product be capable of meeting the AR6A System performance requirements. Therefore, a suitable design technology must be selected for each circuit module that will allow us to meet the individual performance requirements. The choice of design technologies affects the basic partitioning of the circuit functions into physical modules and the total volume needed to realize an equipment design concept. The evaluation of the design technology to implement any circuit function was based on an engineering estimate of: (1) the technical risk factor to achieve the performance requirements, (2) the anticipated cost to manufacture, and (3) the expected reliability.

### *4.1 Circuit implementation*

Circuit linearity is one of the most important of the AR6A performance requirements. The traveling-wave tube was selected for the critical RF amplifier function with the assurance that its inherent nonlinear, third-order characteristic could be controlled and then compensated by predistortion. The performance requirements for the three modulators (receive, shift, and transmit) could be met using modified versions of waveguide structure designs previously used in TH-3 radio.[2] Existing waveguide-based designs were selected as the technology for most of the remaining RF circuit functions. These selections were based on the technical risk confidence factor, performance, and on economic compatibility with the RF section equipment concept. For example, waveguide filters have less loss than dielectric resonator-type filters.

Key elements in the IF circuitry were developed in printed circuit board technology using discrete components. Computer-aided design

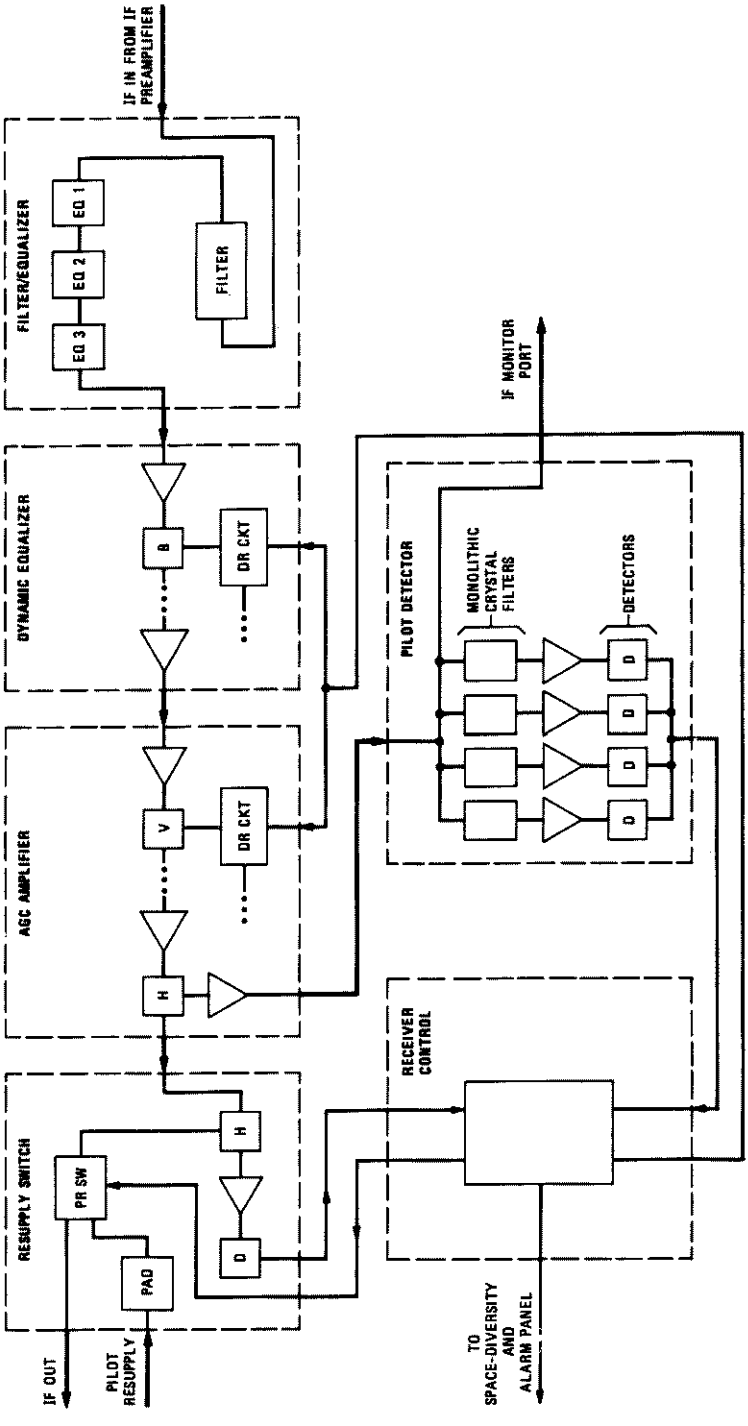



Fig. 5—Receiver IF circuits.

facilities were used to document the critical layout geometries which are characteristic of IF circuits. The performance requirements in the IF circuits were met with amplifier designs based on a hybrid-transformer feedback-transistor stage biased for linear operation. Magnetic components and crystals needed in the IF frequency range were most compatible with the use of printed circuit board technology.

### *4.2 Mechanical implementation*

A technology selection of a different kind was applied to the design of the mechanical piece parts that make up the housings and structures of the AR6A radio-line equipment. The cost of these parts and the labor to assemble them together make up a significant fraction of the total equipment cost. The economics of providing and assembling the mechanical parts is a function of the number of units to be produced. Typically for the radio-line product lines the number of units is not very large, and we use existing manufacturing facilities and tooling wherever possible. Aluminum die castings are used for the microwave assemblies to achieve the required close tolerance forms at relatively low cost due to the reduced need for secondary machining. Aluminum extrusions were utilized as linearly shaped raw material for fabrication into the IF circuit housings. These extrusions can be combined to form electrically isolated compartments within the same plug-in housing. Dip brazing techniques were used to seal those compartment sections that were particularly sensitive to Electromagnetic Interference/Radio Frequency Interference (EMI/RFI) leakage. Close tolerance stamping and forming of sheet metal provides an economical method of creating structural members and parts for the shelf assemblies.

## V. THERMAL DESIGN

The critical importance of thermal design to the operation of electronic circuits should not be underestimated. The operating temperature of the components in a circuit affects the performance of the circuit as well as its reliability. The primary objective of thermal design is to maintain the individual circuit components within the temperature limits necessary to achieve circuit performance and reliability objectives. The component temperatures will be determined by: (1) the amount of heat generated by the components, (2) the thermal impedance (°C/watt) between the components and the local ambient, and (3) the local ambient temperature. The thermal environment for the AR6A radio-line equipment is the moderately controlled interior of the radio station. The ambient temperature in a typical, nonair-conditioned, but heated, radio station may range from +4°C (40°F) to +49°C (120°F), depending on the region of the country and the season of the year. It is the upper end of this temperature range that concerns

us the most, although some unusual problems may occur at low temperatures.

### *5.1 Thermal management*

Thermal management is a design coordination method that economically achieves the thermal design objectives of the individual circuit modules within the design intent of the overall equipment concept. The thermal aspects of the AR6A radio-line equipment concept were a result of design considerations taken from two perspectives. A total assembly view arranges the circuit modules within the equipment framework to minimize the surface temperature rise of the modules above the station ambient. As part of the initial allocation of physical space within the AR6A TR bay and TR support bay, the thermal design requirements of the individual modules were important decision parameters. The individual module view considers the circuit components as heat sources and devises low thermal impedance paths to the local ambient surrounding each module. In both views physical separation of major heat sources was used to reduce the effects of localized heating on adjacent parts. Potential hot spots were identified by estimating the heat load per volume from the circuit and component information. Natural convection techniques dominate the AR6A radio-line thermal management plan in the total assembly view. In the individual module view, conduction and radiation are combined with natural convection to meet the thermal design objectives. These passive techniques are preferred over aided thermal techniques, such as fans, because of their inherent reliability and no additional energy requirement. Another aspect of the thermal management plan was the design influence directed toward energy efficiency. By reducing the power requirements for the individual circuit designs, we reduced the thermal design heat load as well.

### *5.2 Thermal design concept*

Due in large measure to the thermal management influence, the volume of the AR6A radio-line equipment framework is large enough to rely on natural convection for cooling the component modules. The thermal management plan for the TR bay is to induce air from the station to enter and flow around or through the modules. Conduction paths between the modules and the mounting bracketry aid in effectively increasing the surface area for the ultimate convection heat transfer to the station ambient. The open expanse of the middle RF section allows air from the station to circulate around the waveguide components. Some of this air is drawn up and through the IF shelf, while the rest of the air mixes back into the station ambient. The purposeful up-front position of the TWT provides for an almost direct

free-convection transfer of heat to the station ambient. Equipment mounted in the lower housing is cooled by the air that enters through openings in the lower cover, circulates, and then exits up into the rear of the RF section. For the relatively lightly loaded TR support bay, the plan was to design for natural convection from the module faceplates as the principal heat transfer mechanism. Some airflow is induced into and through the shelf assemblies for added heat dissipation.

### *5.3 Thermal design of modules*

The thermal design of each circuit module must be compatible with its other functional requirements. An ideal application of natural convection cooling would be to have the individual circuit components located within unimpeded natural airflow paths. However, because of EMI/RFI requirements, many of the AR6A circuit modules are constructed with completely closed housings. In these modules the power densities are low enough that we can use passive internal heat transfer mechanisms to provide adequate thermal paths. Special attention was paid to the localized hot spots to assure low thermal impedance paths to the outer housing. With a large surface area relative to the thermal load, the heat from these modules can be transferred to the local ambient with a small temperature gradient.

### *5.4 Special thermal considerations*

The largest single heat source in the TR bay is the traveling-wave tube used as the transmitter RF amplifier. Most of the energy in the electron beam is converted into heat at the collector end of the TWT. A critical operating requirement for the TWT is to maintain the temperature of the glass-to-metal seal at the collector below 150°C. This requirement was successfully met by a carefully designed conduction path from the collector to an external-finned heat sink. The thermal impedance between the TWT heat sink and the local ambient was minimized by choosing the optimum fin spacing permitted within the physical space allocated to the TWT.

When a thermal design requirement specifies that a component is to be operated within a narrow temperature range, it may be necessary to control the component's temperature above the ambient temperature by the addition of heat. Several of the frequency-controlling crystal oscillators in the AR6A radio-line equipment are temperature stabilized with thermally insulated oven units. The insulation provides a relatively high thermal impedance path to reduce the heating energy requirements. The control circuit of the oven is set to add no heat at the maximum expected ambient temperature and the full output at the minimum expected ambient temperature.

Two of the closed housing IF modules, the dynamic equalizer and the AGC amplifier, required additional thermal design treatment. Both these modules were provided with perforations on the top and bottom of their frames to allow air to circulate through them. The size of these perforations was selected to use the cutoff waveguide effect to minimize the EMI/RFI leakage. The AGC amplifier dissipates approximately 30 watts of power, mostly from 15 discrete transistors. A vertically finned heat sink has been integrated into the design of the frame for this module. A low thermal impedance path between the discrete transitors and this heat sink is described in the next section.

### *5.5 Heat-dissipating device*

One of the most important circuit designs used in the AR6A System is an IF amplifier stage with very stringent amplitude linearity requirements. To achieve the required linearity, each transistor is biased with a relatively large collector current. Two watts of transister bias power that is converted to heat must be dissipated while the junction temperature is maintained below 125°C. The internal thermal impedance between the junction and the external package of the transistor is 30°C/watt. A patented conduction path package[3] has been devised for this application. Figure 6 shows a cross-section of the heat-dissipating assembly. The purpose of this assembly is not only to provide a low thermal impedance between transistor header and external heat sink but also to provide a means to connect several devices mounted on a printed circuit board to a common heat sink.

The assembly consists of a cylindrical tellurium-copper body, a phenol fiber insulator, a curved stainless steel washer, and a brass ferrule. The body consists of two annular seating surfaces, or "shelves," on the inside surface. The lower shelf provides contact with the transistor, while the upper one provides a stop for the ferrule. Two diametrically opposite pins on the bottom surface of the body engage holes in the printed circuit board to provide rotational locking. The ferrule compresses the spring washer, which exerts a controlled axial force through the insulator to the transistor and, thus, provides a uniform contact pressure between the transistor header and the copper body. A selective indium plating on the inner surface of the body forms a soft interface to improve thermal contact resistance. The insulator provides electrical insulation between the transistor cover and the copper body. The ferrule has a tapped hole for assembly to the common heat sink. It seats on the upper shelf of the body when assembled. An annular groove in the top of the ferrule provides a cylindrical surface for grasping the ferrule during assembly. The outer perimeter of the top of the ferrule is chamfered so that the body may be crimped over for locking. An internal groove in the body simplifies crimping. The

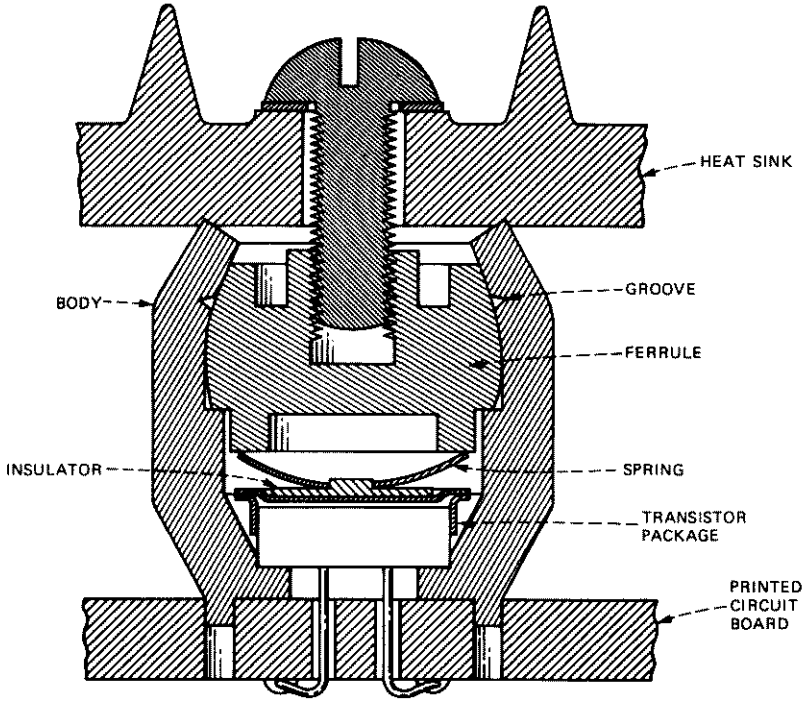



Fig. 6—Heat-dissipating assembly.

tapped hole in the ferrule is "blind" to ensure that the inadvertent use of a long screw does not damage the transistor and to keep metallic debris out of the assembly. When assembled, the ferrule is recessed slightly below the top of the body to provide consistent annular thermal contact between the body and the heat sink.

The material chosen for the body is copper alloy 145. It has high thermal conductivity and good machinability. The free-machining brass of the ferrule is excellent for screw-machine production. Most of the components are designed for screw machining for low-cost production. An assembly fixture presses the ferrule into the body and crimps the body over in a single operation. It also aligns the three transistor leads and two antirotation pins during assembly.

## VI. ADDITIONAL DESIGN CONSIDERATIONS

Every product line has a set of standard physical design criteria that all products are expected to meet. Included among these criteria, which have also been applied to the design of the radio-line equipment, are minimum requirements for structural integrity, shock and vibration, and materials selection for nonflammability and corrosion protection.

Beyond these criteria are the special physical design considerations for electromagnetic interference, human factors, and appearance.

### *6.1 Electromagnetic interference/radio frequency interference*

A continued awareness of the potential electromagnetic and radio frequency interference (EMI/RFI) problems was maintained throughout the physical design process. To prevent unwanted signals from propagating into the radio-line circuits, careful attention was placed on the application of interference suppression techniques. Closed conductive housings were used for the sensitive modules and low-pass filters were applied to their dc power and control leads. Grounding locations, and the integrity of those grounds, were particularly reviewed. Wiring and cabling paths were planned to minimize the possible radiated pickup of other signals. These requirements were purposefully blended into the physical fabrication objectives of the modules.

### *6.2 Human factors*

Human factors is a physical design consideration that includes all the interfaces a product may have with people. Of these interfaces, the most important one is safety and personnel protection. Both defined objective safety criteria and subjective value judgments were applied to design details ranging from interlocked high-voltage compartments to the removal of sharp edges and corners from exposed surfaces. Another human factors concern was the operation and maintenance aspect of the radio-line equipment. Panel and faceplate displays should be easily seen and understood by operating personnel, and normal access to the equipment should be unobstructed. On the TR and TR support bays all the important display features have been assigned to favorable eye-zone regions. The AR6A radio-line equipment has been designed for access completely from the front of the bays. Test equipment access points have been brought to front-facing surfaces, and the circuit modules are easily replaced as a part of the modular assembly concept.

### *6.3 Equipment appearance*

Appearance or visual appeal was an important design consideration in the development of the AR6A radio-line equipment. While appearance may not be the final factor in judging any equipment design, first impressions are often lasting impressions. A harmonious design is easier to work with and is given more care by operating personnel. Good design appearance in the radio-line equipment was achieved by incorporating balance into the size, the shape, and the placement of the circuit modules. The same planned attention to detail was applied

to the design of the circuit modules to provide functional visual appeal. Color has been used with the radio-line equipment where appropriate to enhance the visual effect. Aesthetically pleasing designs need not be at odds with any other design requirements. Rather, the ultimate objective in physical design is to create simple, but truly elegant, designs.

## VII. STATION ARRANGEMENTS

The most economical application of AR6A is to overbuild established TD microwave radio routes. Existing building floor space and facilities, as well as the microwave antennas and towers, can be further utilized at minimal incremental cost. The physical size of the AR6A TR bays was determined with this specific overbuild objective in mind.

### *7.1 Repeater station—frequency diversity*

AR6A uses the same channelization plan as does the TH-3 Microwave Radio System.[4] At a repeater station the two RF signal polarizations in each direction of transmission result in four lineups of four TR bays each. In addition, one TR support bay is required at each station with up to a maximum of 16 TR bays per route. The TR support bay is constrained to be located within 40 cabling feet of any TR bay so that the pilot resupply tone signals may be provided to each TR bay at the same power level.

A typical AR6A repeater station floor plan layout is shown in Fig. 7. Since the radio-line equipment has been designed for front access, the radio bay lineups may be placed against a wall or back to back with another radio bay lineup. The empty bay position opposite the

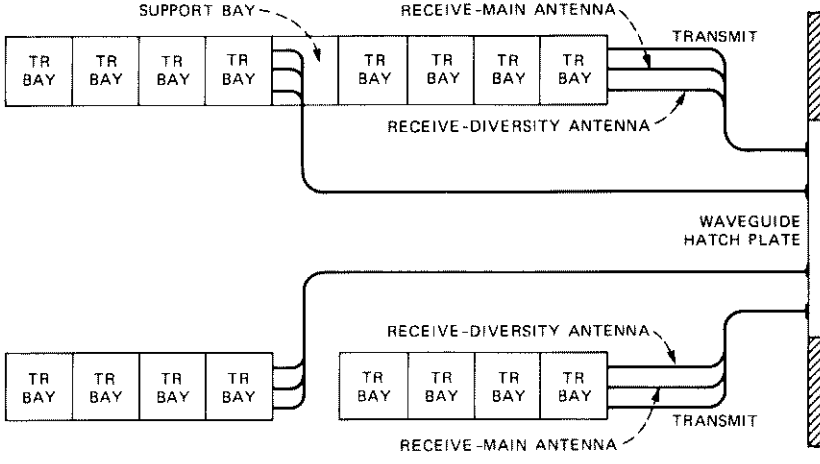



Fig. 7—Floor plan layout for a typical AR6A repeater station.

TR support bay is maintained to provide E-W and W-E channel symmetry across the aisle for operating clarity.

The indoor waveguide arrangements are similar to those used in TH-3. When space diversity is applied there will be three indoor waveguide runs (transmit, main receive, and diversity receive) connected to each bay lineup. To minimize the frequency shift of waveguide filter cavities, dry air is supplied from the station dehydrator using the indoor waveguide for distribution to the TR bays. The indoor waveguide is routed above the bay frameworks and brought down to the 5-1/4 foot level of the channel filters at the end of each TR bay lineup. The common channel RF preamplifiers are mounted in the receive waveguide runs at the top of the first bay in the lineup. Either right-hand or left-hand feed direction is permitted due to the mounting of the channel filters on the TR bay center line. Space has been allocated in the upper section of the TR support bay to allow the indoor waveguide to pass behind the upper side covers. RF connections between the channel filters of adjacent radio bays are made using 4-inch sections of flexible waveguide. Alignment of these connections is made possible by having the radio bay frameworks bolted together through factory-fixtured brackets on the basic bays. Low-pass power filters, which reduce any 60-Hz and harmonic tones from the -24 volts dc power feeds, are located in each bay at the 8-foot level. A photograph showing the AR6A radio-line equipment installed in the Hillsboro, Missouri station is shown in Fig. 8.

### *7.2 Repeater station—hot standby*

When the forecasted channel growth does not permit a frequency-diversity route plan, a hot-standby arrangement is required. Figure 9 shows a hot-standby floor plan for a typical repeater station with one working channel in each direction of transmission. A single standby TR bay protects the two opposite direction regular channels from equipment outage. The receive RF signal paths from the main antenna are split in couplers to provide the inputs from two transmission directions to the standby bay. The microprocessor-based hot-standby control unit, mounted in the TR support bay, controls the selection to an RF switch in the standby bay. The transmit output from the standby bay is split in a coupler above the standby bay and the same transmit signal is sent to another RF switch in each of the regular TR bays. This switch, also under control of the microprocessor controller, determines whether transmit signal from the regular or the standby bay goes out to the antenna. Space diversity may be applied to a hot-standby arrangement but only on the regular TR bays. An additional working channel would share the common receive RF paths but would require a duplication of the other equipment required for the first

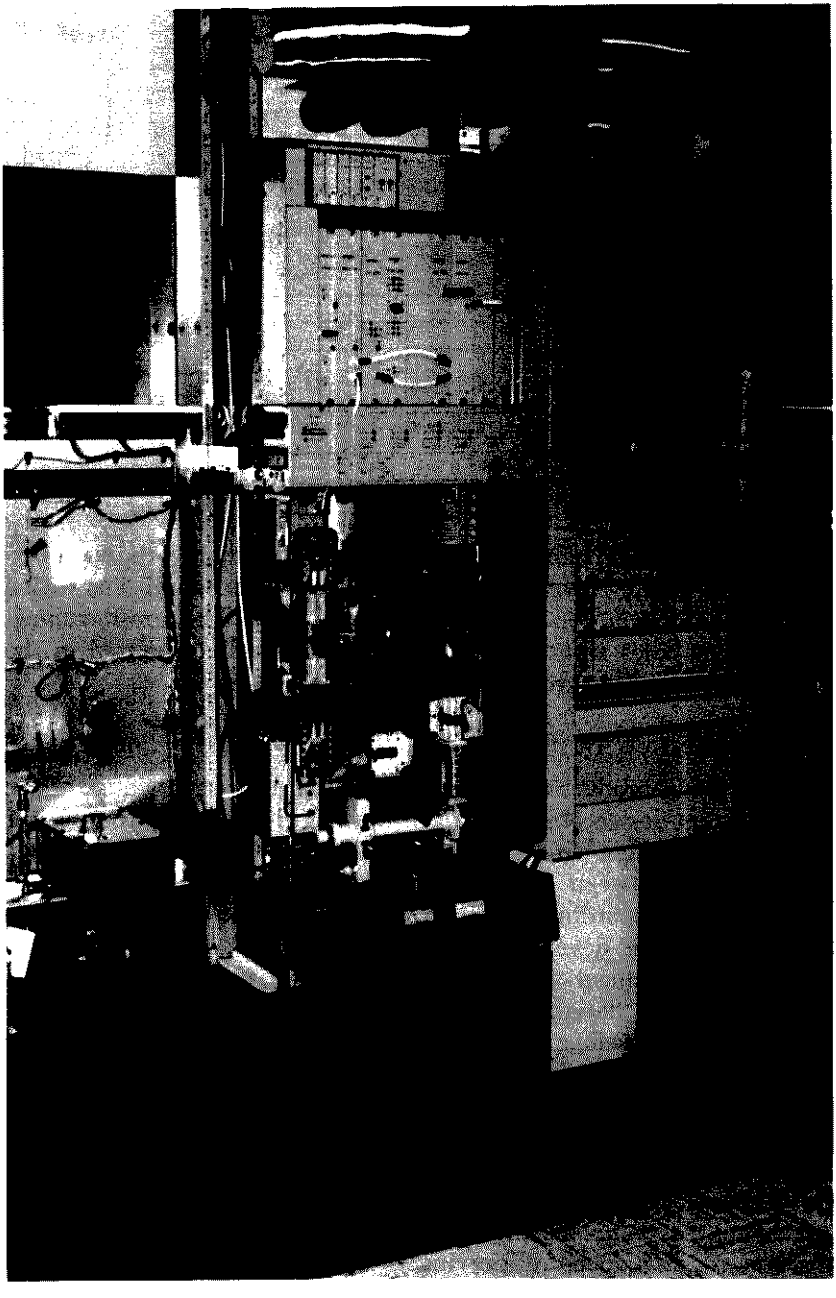


Fig. 8—AR6A radio-line equipment installed at Hillsboro, MO.

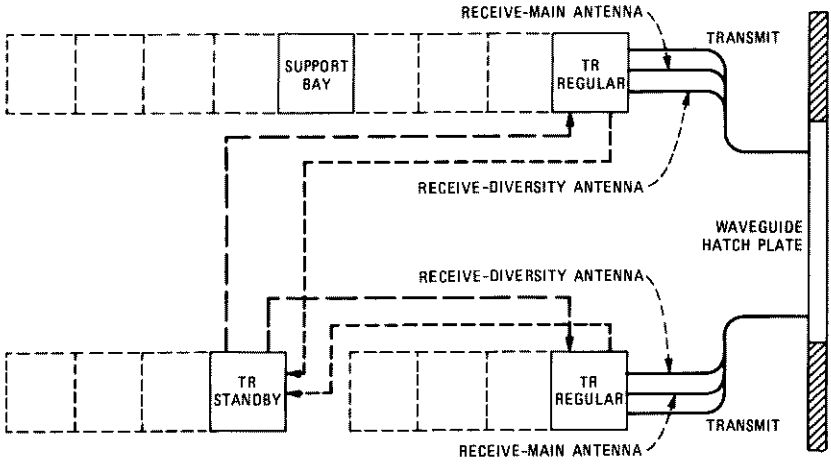



Fig. 9—Hot-standby floor plan layout for a typical AR6A repeater station.

working channel. When there is sufficient channel growth, a conversion to a frequency-diversity arrangement can be accomplished with a minimum of disruption to the working channels. The nonfrequency-sensitive modules from the standby TR bays may be reused in the field assembly of the growth channel TR bays.

## VIII. FINISHING TOUCHES

A successful product design is the result of a careful balance of the many system design objectives. Maintaining that balance throughout the physical design process called for increased emphasis on design coordination between the many engineers involved in this major project. Design coordination has become an area of increasing importance in product developments to evaluate design alternatives, to reduce the technical risks, and to keep the project on schedule. Throughout the design cycle of AR6A many formal and informal design review sessions were held to monitor design status and to maintain continuity with the overall design concept. For convenience in keeping all the design engineers informed, the documentation of the equipment design concept was maintained in the format of size-and-feature outline control drawings.

The physical design of the AR6A radio-line equipment will continue to change as technologies improve and new system concepts and circuits are introduced. Improving the product has always been part of the physical design process. In the design of this equipment we have tried to anticipate these future changes so that the physical design modifications may be accomplished gracefully.

It would be unusual in our modern world of design if a complex

product, such as AR6A, were created by a few individuals. In addition to the many individuals who were involved in the project, we would like to acknowledge in particular the contributions made by R. E. Caron, G. B. Gregoire, T. Kuliopulos, and L. F. Travis.

## REFERENCES

1. N. D. Burgess, R. C. McLean, G. J. Mandeville, D. I. McLean, M. E. Sands, and R. P. Snicer, "The AR6A Single-Sideband Microwave Radio System: Equalization for Multipath Fading," B.S.T.J., this issue.
2. O. Giust, "Modulators," B.S.T.J., *50,* No. 7 (September 1971), pp. 2155–73.
3. P. D. Patel, "Heat Dissipating Assembly for Semiconductor Devices," U.S. Patent 4,167,031, issued September 4, 1979.
4. R. M. Jansen and R. C. Prime, "System Considerations," B.S.T.J., *50,* No. 7 (September 1971), pp. 2085–2115.

## AUTHORS

**Stephen A. Harvey,** B.S.M.E., 1964, Worcester Polytechnic Institute; M.S.E.M., 1966, New York University; M.S. (E.E.), 1980, Northeastern University; Bell Laboratories, 1964—. Mr. Harvey has worked on a variety of physical design assignments in the microwave radio area. He is currently involved in the systems design of digital microwave radio. Member, Tau Beta Pi.

**Parbhubhai D. Patel,** B.E. (Mechanical Engineering), 1962, M. S. University, at Baroda, India; M.S. (M.E.), 1963, and Engr. Sc. D. (M.E.), 1968, Columbia University; Bell Laboratories, 1969—. Mr. Patel began his career at Bell Laboratories in the Transmission Technology Laboratory. Since 1979 he has worked in the Lightguide Cable design and development area. Member, A.S.M.E.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Radio Transmitter-Receiver Units

By R. C. HEIDT,* E. F. COOK,* R. P. HECKEN,* R. W. JUDKINS,* J. M. KIKER, Jr.,* F. J. PROVENZANO, Jr.,* and H. C. WANG*

(Manuscript received February 7, 1983)

This paper describes the essential features and the performance of the AR6A microwave transmitter and receiver. Subsystems that are major contributors to repeater noise are explained in detail. Design objectives for each of these units are given and compared to typical results obtained from measurements. Transmitter linearity was improved substantially through the use of predistortion. The thermal noise contribution of the receiver was reduced by use of a new low-noise microwave preamplifier.

## I. INTRODUCTION

The two basic portions of the AR6A† radio bay are the transmitter and receiver units. These units have a natural interface at the Intermediate Frequency (IF)‡ and are described in the next two sections.

In general, the receiver establishes the required noise figure, and provides the required frequency selectivity, automatic gain control,

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

‡ Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

and both dynamic and static equalization. The transmitter supplies microwave gain with low intermodulation distortion at the required output power.

The alarm and control system spans the entire bay. This system monitors the operation of the radio bay and provides remote switching capability to exercise important repeater functions to facilitate maintenance testing.

## II. THE AR6A RECEIVER

### *2.1 General*

Figure 1 is a block diagram of the AR6A receiver equipped for the space-diversity application. The multichannel Radio Frequency (RF) signals from both the main antenna and diversity antenna (when implemented) are amplified by the optional waveguide microwave preamplifier in each waveguide run. The desired channel is then selected by channel-separation filters. The outputs of the separation filters are input to the space-diversity switch at a nominal full-load power level of −28.6 dBm. The output of the diversity switch is combined with the receive microwave carrier (local oscillator) signal in the receive microwave filter and applied to the Schottky diode single-ended down converter. Integral to the down converter is an IF preamplifier that boosts the signal level to about −4 dBm. In the case when the receiver is not equipped for space diversity, the waveguide from the diversity antenna and the associated channel-separation filter are omitted and the space-diversity switch is replaced by an isolator and a shutter monitor assembly.

The remaining elements of the receiver are located in the IF shelf. The filter portion of the filter/equalizer[1] provides the basic selectivity for the repeater, while the equalizer is used to compensate for static residual amplitude shape due to the various repeater components. At a repeater station the output of the filter/equalizer is applied to the Automatic Gain Control (AGC) amplifier, which provides flat gain with a dynamic range of 61 dB to compensate for fading. Correction is made for both up fades and down fades. The AGC is controlled by the receiver control unit, which maintains the average of the three detected radio pilot voltages constant. The radio pilots are fed to the pilot detector unit through an auxiliary output port of the AGC amplifier. The pilot detector unit filters and detects each of the radio pilots. The output level of each of the detectors is provided as the inputs to the receiver control unit. An IF monitoring point for the radio channel is accessible at the front panel of the pilot detector unit.

At main stations, a dynamic equalizer is used in conjunction with the AGC amplifier to provide up to 16 dB of linear slope or 8-dB

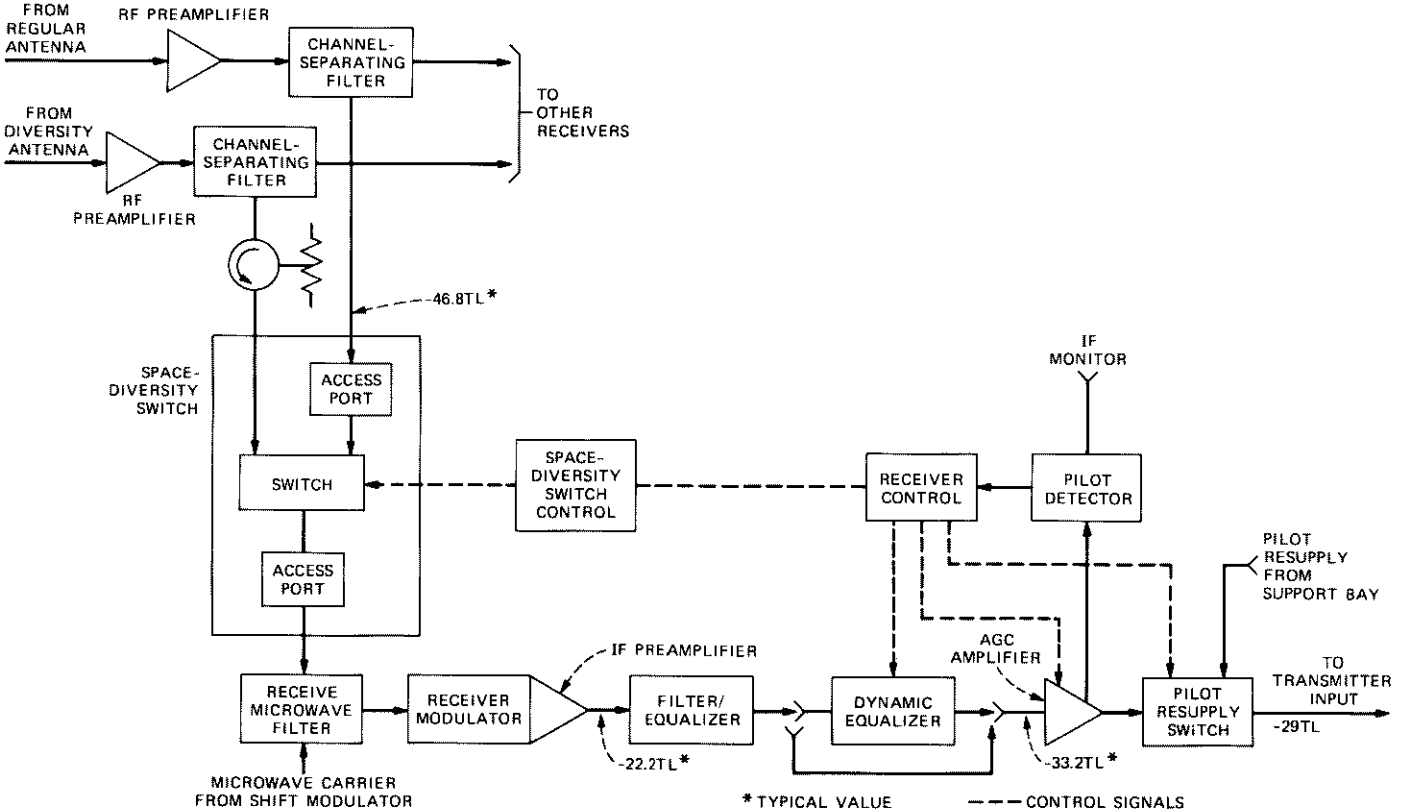



Fig. 1—Receiver portion of an AR6A TR bay.

parabolic amplitude shape to compensate for dispersive fades. The operation of the dynamic equalizer is also governed by the receiver control unit by monitoring the relative differences in level of the three radio pilots.

The output of the AGC amplifier is passed through the pilot resupply switch to the receiver output. Signal loss or signal overpower will cause the receiver control unit to actuate the pilot resupply switch. This action terminates the AGC amplifier output and connects another set of radio pilots from the AR6A support bay to the receiver output. The resupply pilots prevent following repeaters from going to full gain on loss of the channel load.

The receiver control unit also signals the switch request to the space-diversity switch control unit. This request is given when any radio has faded 36 dB or more below the normal received level on a nominal 27.1-mile hop length. This information is derived from the pilot detector signals and the control voltage outputs.

The equalization strategy and operation of the AGC amplifier, dynamic equalizer, receiver control unit, and pilot detector unit are described elsewhere.[2] Further details of the space-diversity switch and control circuit, the receiver modulator/IF preamplifier, pilot resupply switch, and the microwave preamplifier are presented in the following sections.

### *2.2 Circuit description*

#### *2.2.1 Low-noise microwave preamplifier*

An optional low-noise microwave preamplifier (656A) is used in the common waveguide run feeding TH-3, as well as AR6A, receivers on a given polarization. The signal on a TH-3 channel is 16 dB higher in power than the full-load AR6A signal; therefore, the amplifier must exhibit excellent linearity to prevent intermodulation "crosstalk" between radio channels. The 656A amplifier intermodulation distortion performance is good enough to allow this common waveguide amplifier to be used with a mix of TH-3 and AR6A on a given route.

The single-stage amplifier using a gallium arsenide (GaAs) field effect transistor (FET) (103B) was designed to operate over the band of 5.925 to 6.425 GHz. The average gain ranges from 8 to 10 dB, with a ±0.5 dB gain shape allowed over the frequency band. A maximum loss requirement of 10 dB is imposed on the amplifier when the transistor is unpowered. The maximum noise figure is 3 dB with typical values measured around 2.7 dB. Typically, the third-order intermodulation coefficient, $M_{A+B-C} = -45$ dB.

Mechanical construction of this amplifier is very similar to the 4-GHz amplifier previously designed for use in the TD Radio System.[3] The GaAs FET is mounted in a microstrip circuit module. Three

circulators in air strip-line (Fig. 2) provide good return loss at the input and output waveguide ports as well as an effective passive bypass circuit in case the amplifier module fails. The amplifier has a return loss requirement of 30 dB minimum at both waveguide ports.

The dc operating point for the GaAs FET is a compromise between noise figure and linearity. A built-in regulator sets the gate voltage so that $I_D = 15$ mA at $V_{DS} = 4.8$ volts. The amplifier operates from a −24 volt supply drawing 60 mA. In case of transistor or power supply failure, a contact closure is provided for an alarm indication. At the same time, a visible green Light-Emitting Diode (LED) will be turned off.

#### *2.2.2 Receiver modulator/IF preamplifier*

The modulator mixes the received RF signal with the locally generated microwave carrier (LO), which is at a frequency 74.1 MHz above or below the center frequency of the received signal. The modulator, a direct use of an existing TH-3 FM radio design,[4] consists of a stepped waveguide transformer, a waffle-iron low-pass filter, and a diode-mount waveguide section. The basic arrangement of the circuit is shown in Fig. 3. The IF signal is fed through a coaxial low-pass filter to the IF preamplifier input. To obtain improved third-order intermodulation performance for AR6A applications, an LO level of +10 dBm is used instead of +6 dBm as for TH-3. An $M_{A+B-C}$ of about −21 dB is obtained for this LO drive level and optimized diode dc biasing.

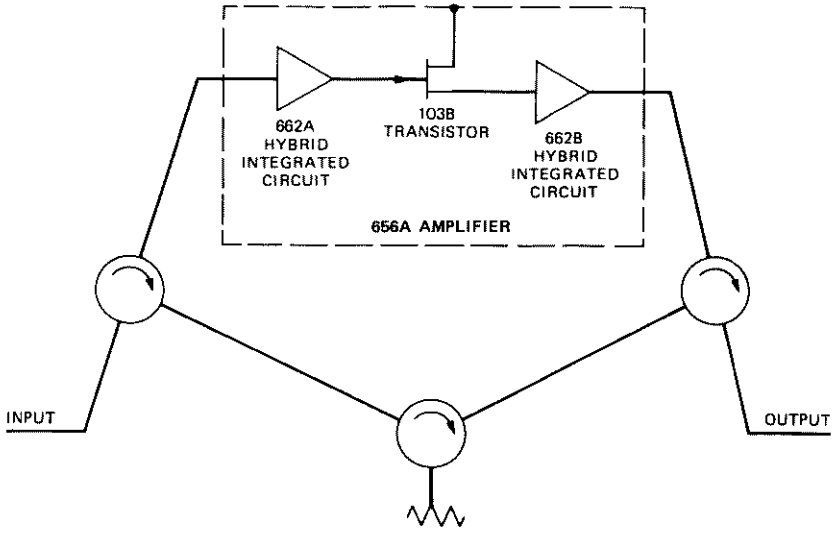



Fig. 2—RF preamplifier.

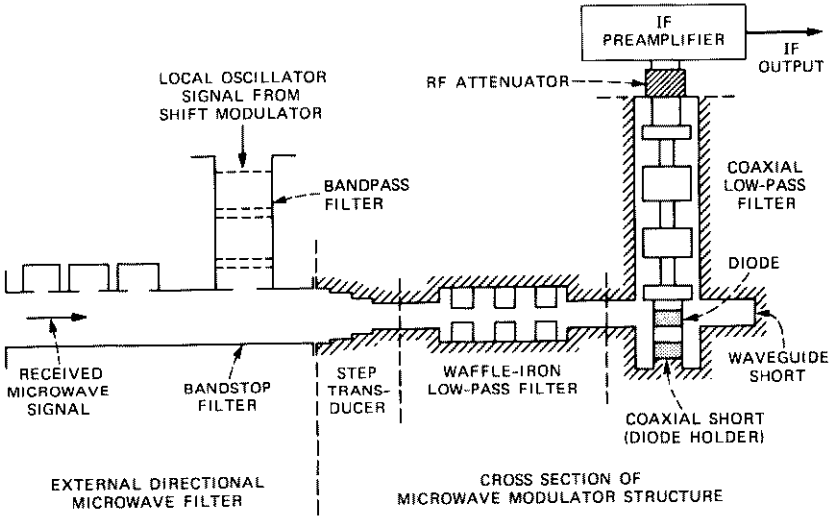



Fig. 3—Basic arrangement of receiver modulator and IF preamplifier assembly.

The 3-stage IF preamplifier has an overall gain of approximately 32 dB. A gain control allows a gain variation of ±3 dB. A slope control is used to achieve flat gain across the band, and two return loss controls are used to provide 30 dB or greater output return loss. The first stage is a low-noise common emitter configuration followed by two hybrid-feedback stages in the output. This configuration provides low-distortion and high-output return loss. The first-stage transistor is designed for low thermal noise performance, but a trade-off was made between best thermal noise and low intermodulation noise. With a dc bias current of about 25 mA and $V_{CE}$ of 3 volts, a gain of 17 dB is realized in this stage with a noise figure of 2.7 dB and an intermodulation coefficient $M_{A+B-C}$ of −45 dB. The following two stages are biased for high-linearity performance. With both transistors biased at an $I_c$ of 120 mA and $V_{CE}$ of 13 volts, the noise figure of the two stages combined is 8 dB with $M_{A+B-C}$ of −80 dB. The complete receiver modulator/preamplifier provides a nominal 28 dB of RF-to-IF gain at a full-load output level of −4 dBm with a noise figure of 8 dB and an overall $M_{A+B-C}$ of −75 dB.

### *2.2.3 Pilot resupply switch*

Figure 4 is a simplified block diagram of the pilot resupply switch circuit. The unit contains a 10-dB directional coupler, a wideband power detector, and an IF switch. The −10 dB port of the coupler is connected through an IF isolation amplifier and power detector. The signal path through the coupler is connected to the receiver-out port

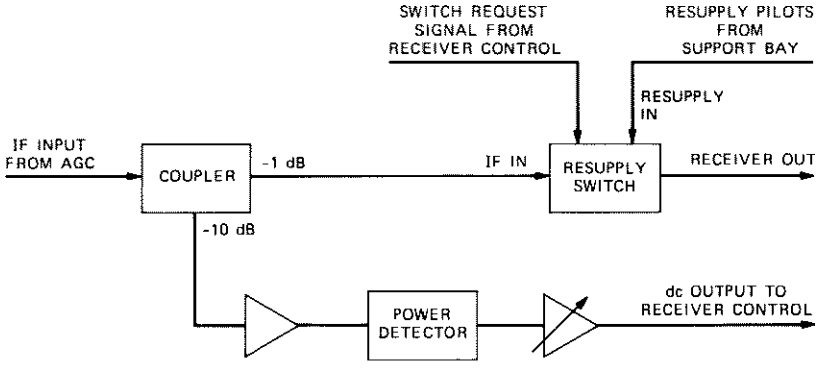



Fig. 4—AR6A resupply switch circuit.

through the normally closed contacts of the switch. The insertion loss through the pilot resupply switch unit is 1 dB. The local set of resupply pilots from the AR6A support bay is applied to the normally open contacts. In this condition, there is 90 dB or greater isolation between the resupply input port and the receiver output port.

The dc output from the power detector is buffered by a variable gain operational amplifier and connected to the receiver control unit. The gain of the amplifier is adjusted to achieve a prescribed output level for a corresponding reference IF power input. This circuit also provides an overpower monitoring feature. If the output exceeds a given threshold for a period of about 80 ms, the receiver control unit will initiate a pilot resupply switch request. The threshold corresponds to an overpower condition of about 12 dB. The receiver control will also initiate a switch request for an underpower condition and in response to a local or remote switch request.

Operation of the resupply switch is controlled by the receiver control unit. When the switch request is made, the IF input from the AGC amplifier is disconnected from the receiver output port and terminated internally in 75 ohms. At the same time the internal termination is disconnected from the local set of resupply pilots, which are then connected through to the receiver output port.

#### *2.2.4 Space-diversity switch and control*

The 436A switch, shown in Fig. 5, provides switching between two input ports and a single waveguide output port. One input is WR159 waveguide and is connected to the channel-separation filter that is part of the common waveguide lineup connecting to the main (or preferred) antenna. The other input, an SMA-coaxial port, is connected to the channel-separation filter that connects to the diversity

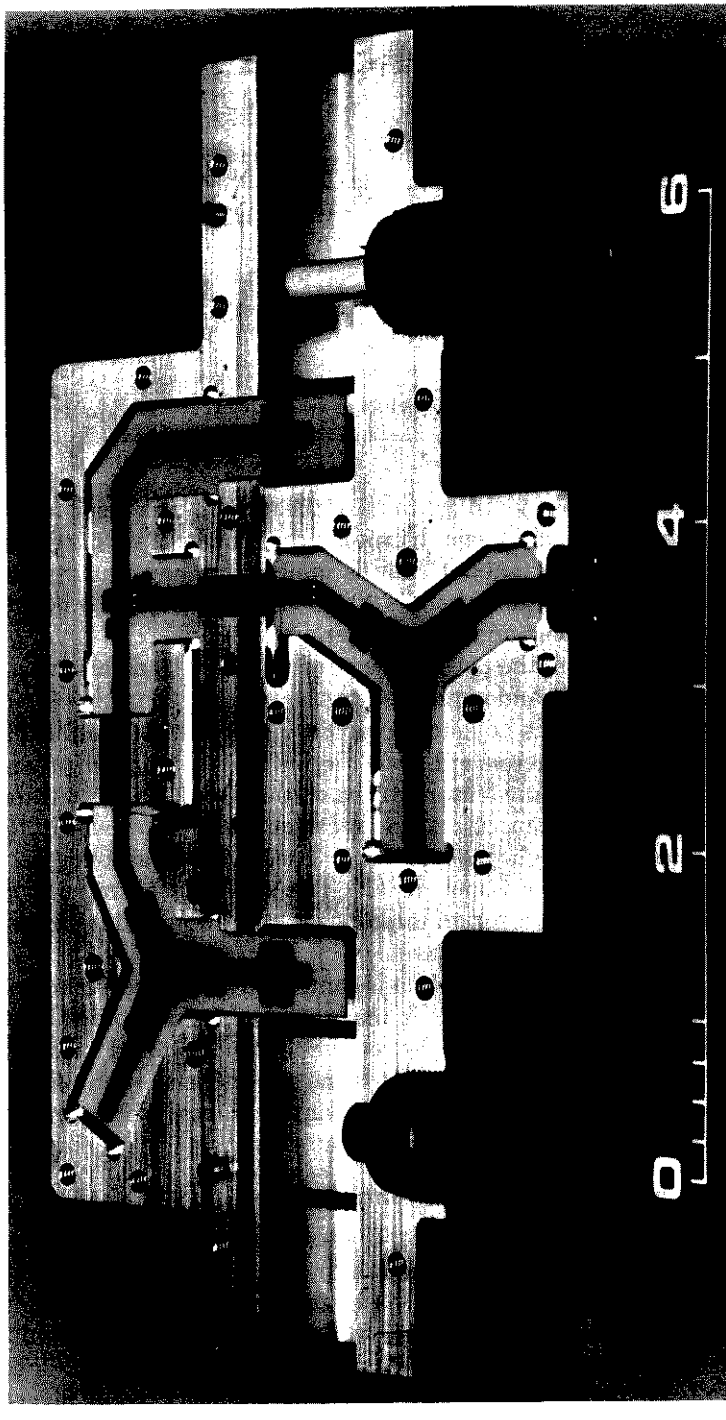



Fig. 5—Space-diversity switch.

antenna. Isolators are built into each port. They provide a load to the channel-separation filters when either path of th is in its open state. The return loss of the input waveguide required to exceed 30 dB; the output waveguide port, 20 dB; i input coaxial port 25 dB. The insertion loss is required to be les ...an 1.75 dB. An air leakage path is provided between input and output waveguide ports to provide a path for air when used in a pressurized waveguide system. The switch is designed to fit into the space normally occupied by the 16B shutter monitor and 23A isolator in nonspace-diversity applications. Therefore, the switch is equipped with test access ports and shorting plates to perform the functions of a shutter monitor.

Diode modules in each of the input paths of the switch incorporate three p-i-n diodes. In the preferred signal path the diode module is zero biased to its low-loss (closed) state. In the nonpreferred signal path the diode module is forward biased to its high-loss (open) state. Isolation between input and output ports in the open state exceeds 70 dB.

The space-diversity switch control unit contains the control circuitry for the 436A switch. The circuit can be separated into an analog portion that delivers the driving current function for each p-i-n diode module and a digital portion that includes the fault and alarm circuitry, the reversion circuitry, the main switching circuitry, and the test circuitry.

The switch control panel contains three red LEDs for indicating fault conditions, two green LEDs for indicating which antenna is connected to the receiver, a momentary pushbutton switch for resetting fault conditions or manual antenna selection, and a mode select switch that allows manual or automatic operation.

Although there are no field adjustments, it is necessary to position two miniplugs for selecting various options. One of these user-selectable options allows either antenna to be designated as the preferred antenna and the other enables the automatic reversion option. Other miniplugs are for factory test purposes only.

In operation the main switching circuitry responds to a digital switch request signal generated in the receiver control unit. This switch request will be honored only if the control unit is in the automatic mode and has not switched in the previous 10 seconds. Two driving functions are employed to control the separate diode modules in the space-diversity switch. These driving functions are identical but 180 degrees out of phase and are overlapped to provide a make-before-break type of transition. The maximum rate of switching is limited to one transition every 10 seconds.

If the automatic reversion option was chosen and the receiver has

been connected to the nonpreferred antenna for 30 minutes, the reversion circuitry will cause a switch back to the preferred antenna. If a switch request is received during the next 3 minutes, indicating that signal fading is still excessive on the preferred antenna, the control circuit will switch the receiver input back to the nonpreferred antenna and reinitiate the 30-minute timer. If a switch request had not been received during the 3-minute interval, the receiver would remain connected to the preferred antenna.

Any one of six fault conditions will result in a space-diversity switch alarm. Seizure faults are generated whenever either antenna is selected manually or whenever either test function is selected. A reversion fault occurs after 14 consecutive reversion attempts fail. The 15th attempt will cause a 7.5-hour timer to time out thereby inhibiting the operation of the reversion circuitry while allowing the rest of the control circuitry to function normally. A reversion fault is reset by depressing the EXCHANGE ANTENNA/RESET switch. A switch request fault is produced whenever a continual switch request is present for more than 85 minutes. The antenna in use is then seized and further operation of the control is blocked to prevent indefinite switching at a 10-second rate. This fault is reset when a low-to-high logic-level transition is received on the switch request input or the EXCHANGE ANTENNA/RESET switch is depressed.

All control unit input and output levels are Transistor-Transistor Logic (TTL) except the two analog signals driving the space-diversity switch. The ALARM output provides a signal to the alarm and control interface circuit. The output labeled SDS ACTIVITY provides a means of monitoring the space-diversity switch performance.

## III. THE AR6A TRANSMITTER

### *3.1 General*

Figure 6 shows the signal-carrying components of the transmitter, together with other ancillary units. The input signal to the transmitter is the 10-mastergroup IF spectrum and includes all pilot tones. This spectrum at a full-load power level of −10.8 dBm comes from either the 500A switch or the output of the preceding receiver and is first applied to the predistorter.[5] The predistorter adds a specific amount of third-order intermodulation distortion with the proper phase relative to the main signal to reduce the intermodulation distortion generated by the remainder of the transmitter. After the signal is predistorted, it enters the amplifier/transmitter modulator, which consists of an IF buffer amplifier, slope equalizer, and a single-ended up converter. The double-sideband RF signal out of the modulator, at a nominal full-load level of −16.7 dBm per sideband, is directed

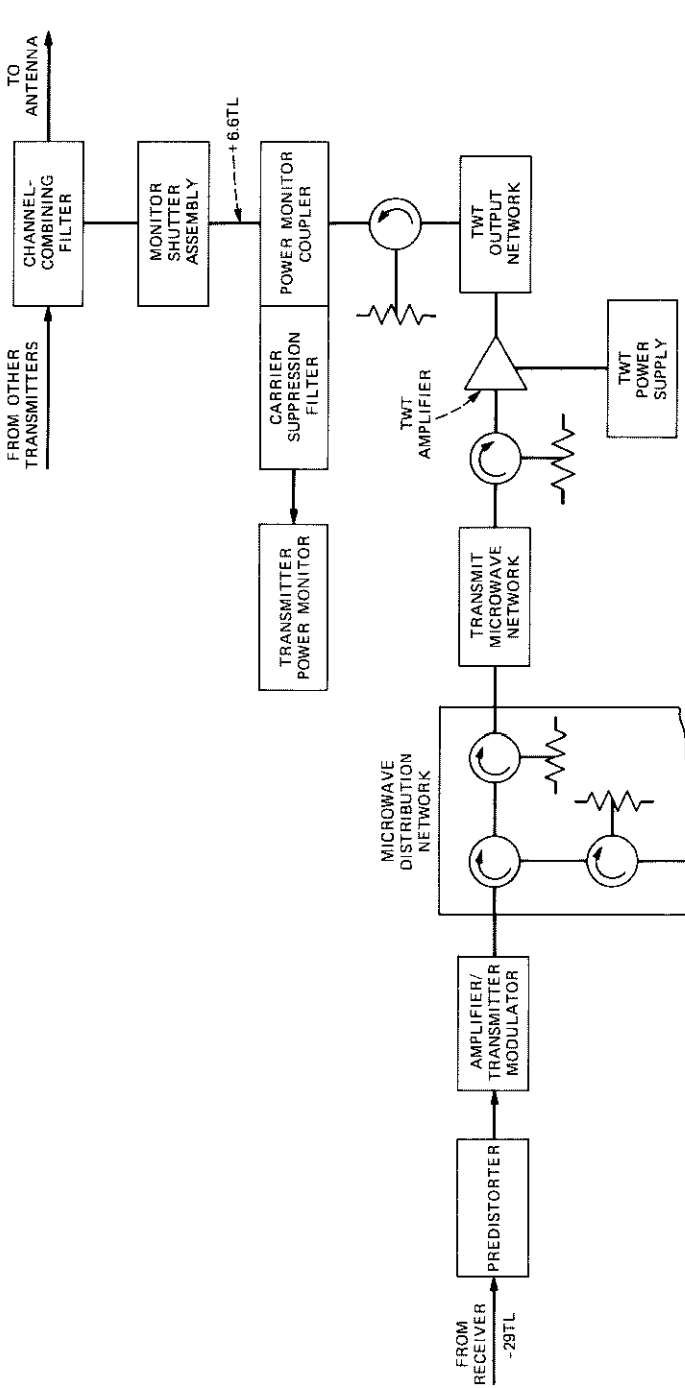



Fig. 6—Transmitter portion of an AR6A TR bay.

through the microwave distribution network to the transmit microwave network (1450 type), which is a bandpass filter with a 1-section delay equalizer. The bandpass filter selects the appropriate sideband and attenuates the carrier and unwanted sideband. More details are given in the next section of this chapter. From the transmit microwave network the signal passes through an isolator to the input of the TWT amplifier.[6] The traveling-wave tube has a typical gain of 44 dB, with gain plus noise figure ≤69.5 dB, and a third-order intermodulation coefficient $M_{A+B-C} \leq -90.5$ dB. A coaxial-to-waveguide transducer is integrated with the tube mount at the input. At the output of the TWT the signal is stripped of its second harmonic content in the TWT output network, which is the combination of a waffle-iron filter, stepped waveguide transformer, and an isolator. The signal is directed through the power monitor/coupler and shutter monitor assembly where the full-load RF power is approximately +25 dBm. The channel-combining filter combines the signal with the spectra of other radio channels.

### *3.2 Circuit descriptions*

#### *3.2.1 The IF predistorter*

Figure 7 is a block diagram of the predistorter used in AR6A. After amplification to a level of approximately +1 dBm, the spectrum enters the phase resolver circuit.[1] The phase resolver splits the input signal into two parts with the phase angle between the two output ports being adjustable. The signal passing through the delay line is the main signal and constitutes the reference. The phase-shifted signal is applied to the other path, which is referred to as the cuber path. The delay line in the main signal path is used to match the delay of the other path to within ±0.1 ns.

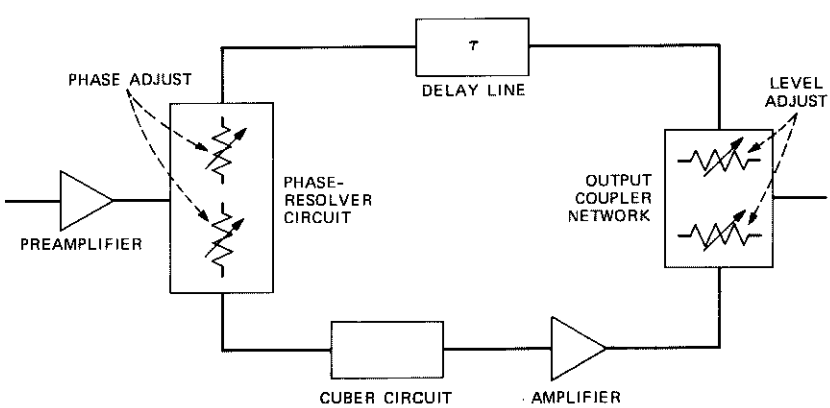



Fig. 7—Block diagram of a predistorter.

The cuber circuit[7] generates the desired complementary distortion products. The cuber circuit consists primarily of two pairs of back diodes connected antiparallel and in opposite branches of a bridge circuit. In this bridge arrangement fundamental frequency signals are canceled by at least 60 dB. The antiparallel pair of diodes generates predominantly odd-order intermodulation products. They are carefully selected for best conversion efficiency and performance that is not level-dependent. Other design considerations and requirements are described in a companion article in this issue.[5] The distortion amplifier brings these complementary distortion products to the desired level and feeds them directly into the output coupler, where the complementary distortions products are combined with the main signal.

The output coupler circuit contains two variable attenuators allowing the independent level adjustment of the main signal and distortion signal prior to combining them. At the output of the predistorter the main path nominal signal level is approximately −14 dBm and the complementary distortion products are at a level of about −71 dBm.

#### *3.2.2 Amplifier/transmitter modulator*

The amplifier/transmitter modulator, shown in Fig. 8, is used to translate a 59- to 89-MHz IF signal to the 6-GHz communications band with minimum distortion. The amplifier serves as a buffer between the IF input port and the Schottky diode of the transmitter modulator to minimize variations in the input return loss. Slope and gain compensation of the overall transmission characteristic is accomplished by adjustments located on the amplifier. The amplifier is also equipped with a p-i-n diode attenuator that protects the transmitter output networks by limiting the maximum input power to the transmitter.

The transmitter modulator combines the IF input signal from the amplifier with the microwave carrier. This combining or mixing action results in the creation of a lower sideband (LSB) and a upper sideband (USB) referenced to the microwave carrier.

The amplifier consists of two transistor stages. Ahead of these stages is a transmitter gain adjustment, which sets the transmitter output power by attenuating the IF input signal level. This adjustment is capable of varying the IF signal level over a 20-dB range although, typically, an adjustment less than ±3.9 dB is required.

The first amplifier stage is a common base configuration designed to exhibit very low third-order intermodulation distortion. A slope equalizer network follows the first amplifier stage. When the slope adjustment is rotated clockwise, the network forms a low-pass filter with a negative slope transmission characteristic. When rotated counterclockwise the network becomes a high-pass filter with a positive

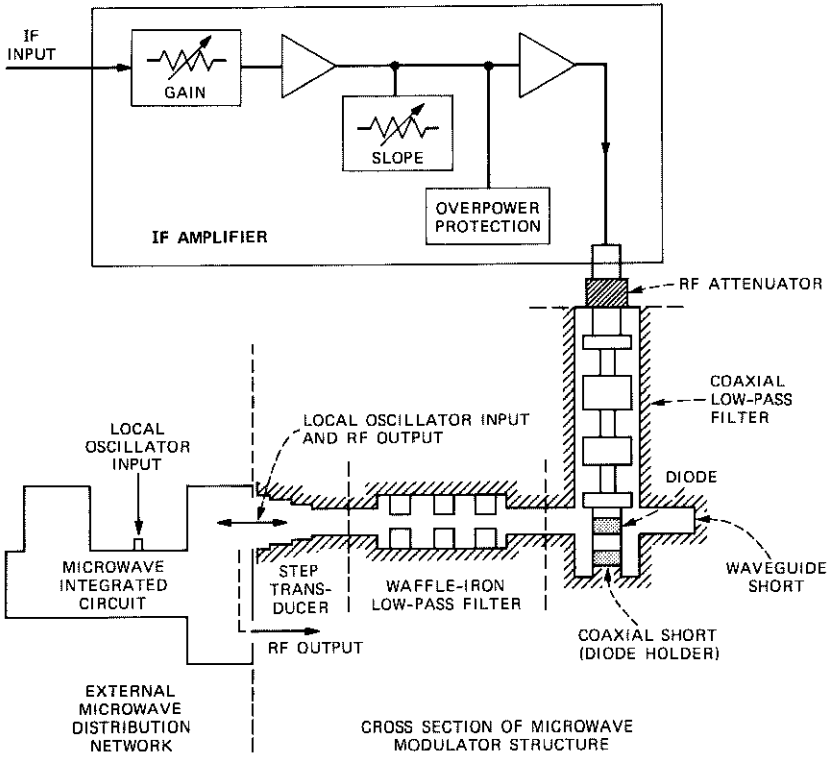



Fig. 8—Basic arrangement of an amplifier/transmitter modulator.

slope transmission characteristic. The slope equalizer network adjustment is capable of varying the IF channel slope over a range greater than ±1 dB although, typically, an adjustment less than ±0.2 dB is required.

The second amplifier stage is a common emitter configuration. It provides approximately 5 dB of additional gain and buffers the transmitter modulator Schottky diode from the slope equalizer network.

The p-i-n diode attenuator network is controlled by a third transistor. Under normal operating conditions, no dc current flows through the p-i-n diode and, therefore, its ac impedance is extremely high. However, should the transmitter output power increase to the point where limiting is required, the transistor will drive dc current through the p-i-n diode and cause its ac impedance to become very low. This loads down the collector of the first-stage transistor thereby attenuating the IF signal by a nominal 10 dB.

The 59- to 89-MHz IF signal from the amplifier and the dc bias for the Schottky diode are fed through an RF attenuator and low-pass

coaxial filter to the diode in the transmitter modulator. The coaxial filter suppresses 6-GHz fundamental frequencies and their second and third harmonics while the attenuator provides additional loss at frequencies above 1 GHz to prevent this energy from feeding back to the amplifier. The combined insertion loss of the coaxial filter and attenuator is approximately 3 dB at 800 MHz and 50 dB at 6 GHz.

The microwave carrier frequency is applied to the Schottky diode through a waveguide input port, a stepped waveguide transformer and a waffle-iron filter. The Schottky diode is physically mounted between a coaxial short, which is also the diode holder, and one end of the coaxial low-pass filter. The distance that the coaxial short extends into the diode cavity and the position of the waveguide short are designed to achieve the desired conversion efficiency and eliminate second-harmonic waveguide resonances, which could adversely affect the linearity of the transmitter modulator circuit.

The stepped waveguide transformer provides a smooth transition of the RF transmission characteristic from the reduced height waveguide, used in the design of the waffle-iron filter and diode mount, to full-height WR159 waveguide. The waffle-iron filter prevents the second-harmonic frequencies, which are generated in the transmitter modulator, from appearing at the waveguide port. The waffle-iron filter has an insertion loss of approximately 3 dB at 8.2 GHz and 40 dB at 12 GHz.

The IF signal and the microwave carrier are mixed in the Schottky diode to obtain the RF sideband frequencies. The sideband frequencies appear at the waveguide port and are then filtered to select the desired sideband. The single-tone output power level of the transmitter modulator is nominally −16.7 dBm when a single-tone IF input power level of −13.8 dBm is applied to the amplifier and a microwave carrier power of +16 dBm is applied to the transmitter modulator. A supply voltage of −15 volts dc at approximately 155 mA is required to operate the amplifier/transmitter modulator.

#### *3.2.3 Traveling-wave-tube amplifier*

The traveling-wave-tube amplifier supplies the necessary microwave gain. The amplifier has unique linearity requirements: it must provide low third-order intermodulation levels ($M_{A+B-C} < -90.5$ dB) and must have a constant gain and third-order intermodulation coefficient over a wide range of output power (from 23 to 35 dBm).

The amplifier, which is convection cooled, consists of a TWT envelope, focusing mount, and power supply. The input to the amplifier is through an SMA-coaxial connector. The output is reduced height (0.1-inch) WR159 waveguide. Focusing adjustments on the mount minimize helix current.

The power supply provides all voltages required by the TWT. The -24 volt battery, fed through a frame filter to reduce low-frequency (60-Hz harmonics) ripple, is used to operate the high-voltage TWT power supply. Front-panel meters measure helix and collector (or beam) currents. A helix voltage adjustment on the front panel is used to set the TWT to its proper operating point.

## IV. MICROWAVE NETWORKS

Various microwave components used in the TR bays were designed some time ago for applications in previous 6-GHz microwave radio systems. Therefore, we are limiting our discussion to new components designed for AR6A.

The 1379-type filter is used as both a channel-separating and channel-combining filter. This filter is mechanically identical to the 1340-type channel-combining filter of the TH-3 System.[8] It is a six-cavity, pseudodirectional filter in WR159 waveguide with a 3-dB bandwidth of 54.15 MHz. To reduce the passband insertion loss slope due to the inductive nature of discontinuities used in the design, the filter is tuned 1 MHz lower than the channel frequency. The insertion loss at the channel frequency should be less than 0.75 dB. In the stopband the loss follows well with the calculated loss of a six-cavity, maximally flat, waveguide bandpass filter.

The receiver modulator filter (1374 type) is a three-cavity directional filter in WR159 waveguide that couples the microwave carrier (LO) and the signal into the receiver modulator. The insertion loss at the LO frequency is no more than 1 dB and the 3-dB bandwidth is 17 MHz. Selectivity at frequencies ±25 MHz away from the center of the bandpass is more than 30 dB. The return loss for both signal and LO ports is required to be at least 30 dB.

The transmitter microwave network (1450 type) is used to block the unwanted sideband at the output of the transmitter modulator. It consists of a seven-cavity, maximally flat, bandpass filter (1373 type) in WR159 waveguide and a microwave delay equalizer.[9] The 1373 filter has a 3-dB bandwidth of 70 MHz. The return loss over the 34-MHz passband is required to be better than 30 dB. The insertion loss at midband is less than 0.65 dB. The delay equalizer adds no more than 0.35 dB to the filter passband loss. The residual delay distortion of the network should be less than ±0.2 ns over a 30-MHz passband.

The TWT output network (1451 type) serves the two functions of transducing the reduced height waveguide at the TWT output port to the full-height WR159 waveguide and of rejecting harmonics from the TWT. It consists of a three-section waffle-iron low-pass filter and a three-section quarter-wavelength step impedance transducer. The net-

work has a 30-dB return loss requirement over the 6-GHz band. The insertion loss at the second-harmonic frequency is about 40 dB.

The 1452 network is a power-monitoring coupler in the transmitter. The main signal path consists of a 26.5-dB cross-guide coupler and a 90-degree step twist. The coupled arm has a two-cavity, band-reject filter to block the LO frequency and a waveguide-to-SMA coaxial transducer in the output port. A 30-dB return loss requirement over the entire frequency band is imposed on the main signal path. The coupled arm requires a 20-dB return loss over a 30-MHz band centered at the channel frequency. About 55-dB rejection is provided at the LO frequency by the band-reject filter.

To reduce multiple reflections in the RF signal path of the radio bay, a long section of bent semirigid coaxial cable in the transmitter was tuned and tested together with its two transducers to WR159 waveguide, as a network (1453 type). It requires 30-dB return loss over the entire frequency band. This network is particularly sensitive because of its connection directly to the 1379-type filter. If a mismatch occurs in the 1453 network, severe ripple may occur in the channel.

Three shutter monitors are also required in the bay: the 16A, used in the transmitter, consists of three sliding shorts and two probes; the 17A shutter-monitor, used in the shift modulator circuit, consists of one sliding short and one probe; the 16B, used in the receiver when the space-diversity switch is not used, is similar to the 16A. The shutter-monitor is a waveguide component that has an access port in the broad wall in which a coaxial probe can be inserted. Slides may be inserted into slots cut into the narrow wall, thus providing a short across the waveguide. The resulting assembly is similar to a coaxial-to-waveguide transducer. With the probe and slides removed, the assembly becomes a simple waveguide section. A minimum return loss of 23 dB can be expected from the coaxial probe, and about 35-dB insertion loss per inserted sliding short can be achieved.

## V. ALARM AND CONTROL INTERFACE CIRCUITS

The alarm and control interface circuits consist of six units that provide the alarms, indications, and commands needed for one TR bay. These circuits provide a means for external identification of the particular failure, visual indications of trouble conditions on the TR bay display panel, remote indications of trouble conditions, maintaining visual alarms when audible alarms are silenced or disabled, and converting a command from C- or E-type telemetry equipment to a TTL level for interfacing with the TR bay.

The command unit translates command pulses (generated by a contact closure) from a C- or E-type telemetry system to a latched

logic-level change for the remote operation of a function in the TR bay. The command circuits in the bays use a multiplex scheme requiring two command pulses from the telemetry system for the operation of any one function. One command pulse from the telemetry equipment for each bay is used for addressing that bay. This is followed by a second command associated with a particular function. This second command is fed to all other radio bays in both directions of transmission but is accepted only by the addressed bay. It is necessary to readdress the bay for each subsequent operation of a function. The five operations that can be remotely performed by command are as follows: RESUPPLY INITIATE, FLAT EQUALIZER SHAPE, NOMINAL RECEIVER GAIN, GAIN MEASUREMENT REQUEST, and BAY RESET.

Conditions causing alarms and indications are listed in Table I. Inputs to the receiver alarm unit are from the receiver or common equipment only; inputs to the transmitter alarm unit are from the transmitter or common equipment only; and inputs to the common alarm unit are from the equipment shared by both the transmitter and the receiver.

The alarm fuse unit provides two fused sources of -24 volts. One source is used for powering all of the alarm and control interface circuit relays. The other is an additional source of power for the power alarm relays.

The alarm relay unit translates TR bay alarms into contact closures for the operation of external audible, visual, and remote alarms. The

Table I—Conditions causing alarms and indications

| Condition | Receiver Alarm | Transmitter Alarm | Remote Indication | Display Panel Indication |
|---|---|---|---|---|
| Space-diversity switch alarm | X | | | X |
| Receiver control alarm | X | | | |
| Resupply initiate alarm (delayed) | X | | | |
| Equalizer at maximum range (delayed) | X | | | |
| Shift oscillator loss of lock | X | | | X |
| Radio bay enabled | | | | X |
| Predistorter off | | X | | |
| Low or high transmitter power | | X | | X |
| Low dc power | X | X | | |
| Microwave generator low power | X | X | | X |
| Microwave generator loss of lock | X | X | | X |
| Transmitter gain normal | | | X | X |
| TWT beam fault | | | X | |
| Space diversity initiated | | | X | |
| Shift oscillator or microwave generator memory fault | | | X | X |
| Resupply initiated by command | X | | X | X |
| Receiver gain nominal by command | X | | X | |
| Equalizer flat shape by command | X | | X | |

unit also has an alarm cutoff function for silencing the audible alarms. LEDs on the faceplate provide a visual indication for a bay alarm or a power alarm.

## REFERENCES


1. R. L. Adams et al. "The AR6A Single-Sideband Microwave Radio System: System Networks," B.S.T.J., this issue.
2. N. O. Burgess et al. "The AR6A Single-Sideband Microwave Radio System: Equalization for Multipath Fading," B.S.T.J., this issue.
3. R. H. Knerr and C. B. Swan, "A Low-Noise Gallium Arsenide Field Effect Transistor Amplifier for 4-GHz Radio," B.S.T.J., *57*, No. 3 (March 1978), pp. 479–90.
4. O. Giust, "TH3 Microwave Radio System—Modulators," B.S.T.J., *50*, No. 7 (September 1971), pp. 2155–73.
5. R. P. Hecken, R. C. Heidt, and D. E. Sanford, "The AR6A Single-Sideband Microwave Radio System: Predistortion for the Traveling-Wave-Tube Amplifier," B.S.T.J., this issue.
6. J. F. Balicki et al. "The AR6A Single-Sideband Microwave Radio System: The Traveling-Wave-Tube Amplifier," B.S.T.J., this issue.
7. R. P. Hecken, U. S. Patent No. 4,157,508, issued June 5, 1979.
8. E. J. Drazy, R. E. Sheehey, and H. C. Wang, "TH-3 Microwave Radio System—Networks," B.S.T.J., *50*, No. 7 (September 1971), pp. 2137–53.
9. T. A. Abele and H. C. Wang, "An Adjustable Narrow Band Microwave Delay Equalizer," IEEE Trans., *MTT-15*, No. 10 (October 1967), pp. 566–74.


## AUTHORS


**Edmund F. Cook,** B.S.M.A., 1965, Fairleigh Dickinson University; Bell Laboratories, 1954—. At Bell Laboratories, Mr. Cook was first engaged in the development of radio telemetry and electronic switching systems. Since 1971 he has participated in the development of the AR6A Microwave Radio System with particular emphasis on modulator and traveling-wave-tube studies.

**Rudolf P. Hecken,** M.S.E.E., 1959, Ph.D., 1964, Technische Hochshule Aachen, West Germany; Institute for High Frequency Technology, Aachen 1959–1964; CERN, Geneva, Switzerland, 1964–1968, Bell Laboratories, 1968—. Mr. Hecken has been involved in the design of networks for the L5 coaxial system, FM microwave radio systems, and the AR6A Radio System. Since 1982 he has been Director of the Radio Systems and Terminals Laboratory.

**Richard C. Heidt,** A.S.E.E., Arlington State Jr. College, 1956; B.S.E.E., Southern Methodist University, 1959; M.E.E., New York University, 1963; Bell Laboratories, 1961—. Mr. Heidt worked first in microwave transmission circuits and antennas. Since 1969 he has been involved in the design of transmitters for the single-sideband microwave radio system. He is currently Supervisor of a group developing digital microwave radio systems. Member, IEEE, Eta Kappa Nu, Sigma Tau, Kappa Mu Epsilon.

**Robert W. Judkins,** Bell Laboratories, 1954—. Mr. Judkins' early work was directed towards design and development of microwave circuits, such as up converters, down converters, frequency multipliers, and waveguide networks for use in TD-3, TH-3, and TN-1 radio systems. Since 1975 he has worked on the design and development of solid state microwave switches, preamplifiers, and power amplifiers.

**J. Morgan Kiker, Jr.,** B.S.E.E., 1962, University of California at Berkeley; M.S.E.E., 1964, New York University; Bell Laboratories, 1962—. Mr. Kiker began work on high-resolution radar systems. Since 1971 he has supervised work on ultralinear IF amplifiers and low-noise carrier and oscillator supplies for the AR6A Radio System. He is now working on digital radio systems. Member, Tau Beta Pi, Eta Kappa Nu.

**Frank J. Provenzano, Jr.,** E.T., RCA Institutes, 1969; Merrimack College, 1982—; Bell Laboratories, 1969—. Mr. Provenzano has worked on various aspects of the single-sideband microwave radio system since starting at Bell Laboratories. He currently is working on alarm systems for digital radio systems.

**Han-Chiu Wang,** B.S.E.E., 1955, Cheng-Kung University, China; M.S.E.E., 1960, University of Notre Dame; Ph.D. 1965, Polytechnic Institute of Brooklyn; Bell Laboratories, 1965—. Mr. Wang has worked on microwave generators, microwave amplifiers, and various microwave and millimeter wave networks. Presently, he is Supervisor of the Microwave Network and Antenna Group supporting development of microwave radio transmission systems.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Terminal Multiplex Equipment

By A. DUBOIS,* D. N. RITCHIE,* and F. M. SMITH*

(Manuscript received February 2, 1983)

A combination of new multiplex facilities provides the 6000-message-circuit loading for AR6A radio. It includes a Mastergroup Translator Type-B (MGTB) arrangement that accepts five U600 mastergroups to translate and combine them into a Multimastergroup (MMG) spectrum. A new multimastergroup translator for radio accepts two such MMG spectra from MGTB to produce the intermediate frequency signal and pilots for the AR6A line. A pair of pilots from the Bell System reference frequency transmission unit may be added to the MMG spectrum to be recovered at the receiving end for office synchronization. An office master frequency supply sends synchronization frequencies accurate to within one part in $10^9$ for the MGTB and Multimastergroup Translator-Radio translating equipment. Description, design features, and performance characteristics are described in detail below.

## I. INTRODUCTION

This section describes the frequency-division multiplex facilities that form and recover the 6000-message-circuit load for the AR6A† Radio System. Starting with the U600 mastergroup (MG)‡ spectrum formed by existing multiplex facilities, terminal equipment recently

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

‡ Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

developed to form the multimastergroup (MMG) spectrum for the L5E Coaxial Cable System is also used to form the MMG spectrum for AR6A. This equipment is the Mastergroup Translator Type B (MGTB).

An assembly of five MGTB mastergroup translators translates for five U600 input signals (0.564 to 3.084 MHz), each having a 2.840-MHz mastergroup pilot. The combined outputs of the five translators form an MMG spectrum to which a 13.920-MHz continuity pilot is added. This MMG spectrum occupies the frequency band of 8.628 to 21.900 MHz.

Two such MMG spectra are used as inputs to the Multimastergroup Translator for Radio (MMGT-R). The transmitting MMGT-R terminal combines each MMG with specific recovery and radio-line pilots and modulates each MMG separately to form the IF spectrum of 59.844 to 88.460 MHz. Figure 1 shows the basic mastergroup, MMG, and IF spectra.

Figure 2 shows the process used in MGTB and MMGT-R terminals to translate and recover the ten U600 basic mastergroups.

An accuracy of one part in $10^9$ is required for some carrier and pilot frequencies to generate, administer, and recover the mastergroup, MMG, and IF spectra. This accuracy is achieved by synchronizing the carrier and pilot generators to the 2.048-MHz Bell System frequency standard. For locations where this signal is not available, two reference pilots generated by the Bell System Reference Transmitting Unit (BSRTU) are added to the first MMG of an AR6A channel. The precise 64-kHz frequency difference between these pilot tones of 11.200 and 11.264 MHz is derived from the nationwide Bell System Reference Frequency Distribution System. The Bell System Reference Tone (BSRT) receiver unit retrieves the 64 kHz from these pilots for synchronization of the Office Master Frequency Supply (OMFS), which serves the MMGT-R and MGTB facilities.

## II. MGTB TRANSLATOR

### *2.1 Introduction*

The block diagram of the mastergroup translator is shown in Fig. 3. Each translator provides modulation and demodulation, serving both directions of transmission. The same basic design is used for all five translators for AR6A. Only the carrier generator frequency and the bandpass filters in the modulator and demodulator are changed to achieve the desired translated spectrum.

### *2.2 Improved spectrum utilization*

Earlier designs of mastergroup multiplex equipment translate and combine up to six mastergroups to form an MMG spectrum with

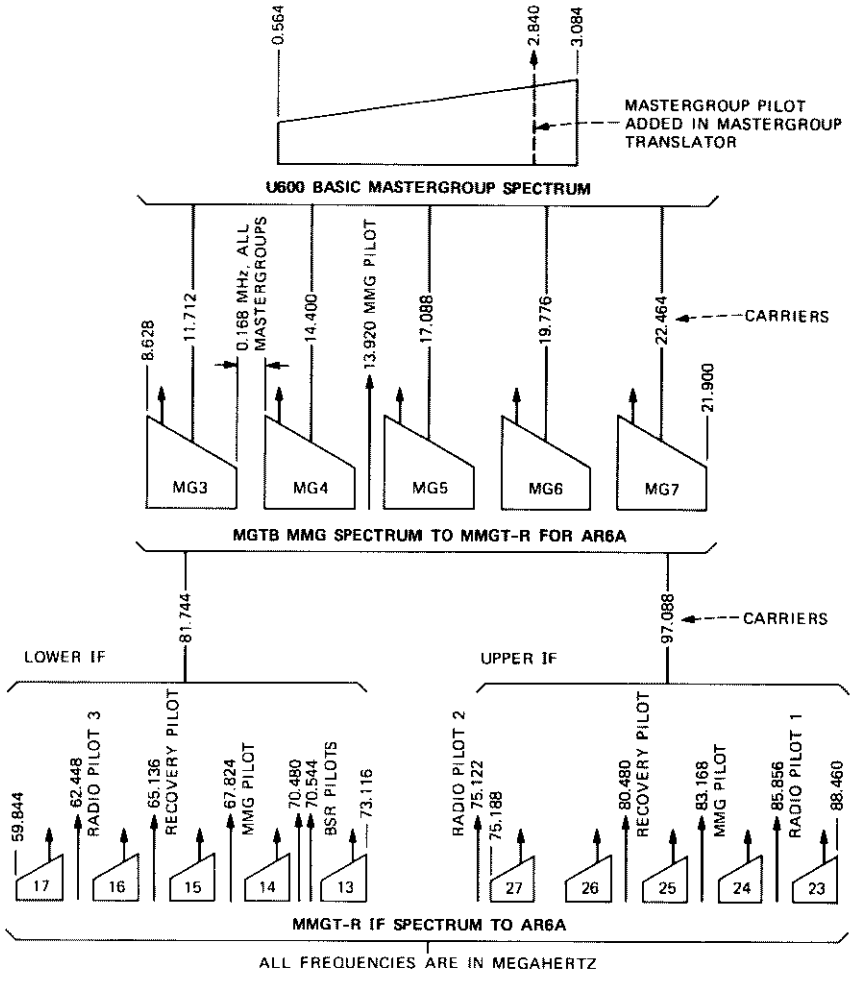



Fig. 1—AR6A terminal equipment spectra.

unequal spacing between the mastergroups. Use of that equipment permitted transmission of 18 mastergroups (10,800 voice circuits) per single coaxial cable or four mastergroups (2400 voice circuits) per FM radio channel.

For the L5E System[1] MGTB equipment translates and combines up to eight mastergroups with a fixed spacing of 168 kHz between mastergroups in the MMG spectrum. By applying three of the MGTB MMG spectra to associated equipment, a total of 22 mastergroups (13,200 voice circuits) are transmitted over a single coaxial cable.

This concept of improved spectrum utilization was carried over into the AR6A development by using five MGTB mastergroups (3 through

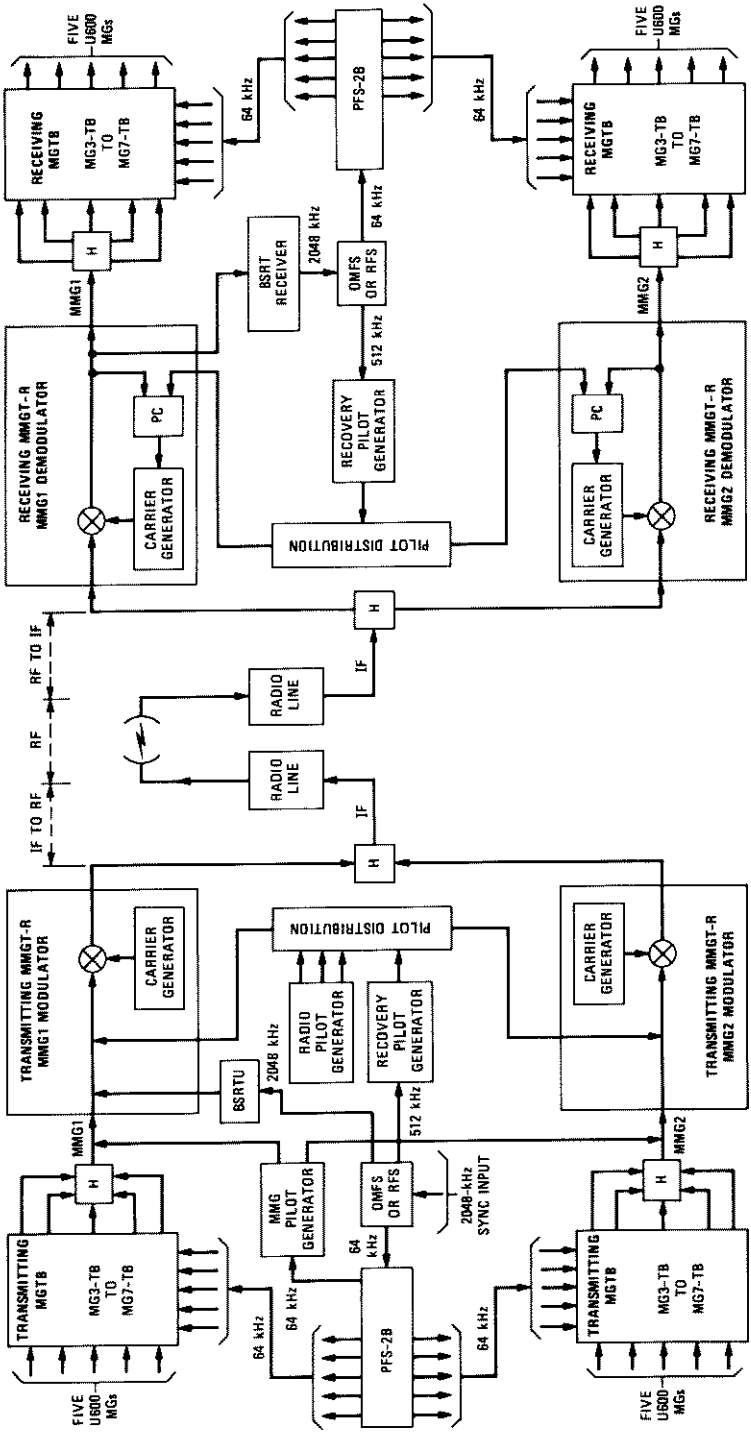



Fig. 2—Block diagram of AR6A terminal equipment.

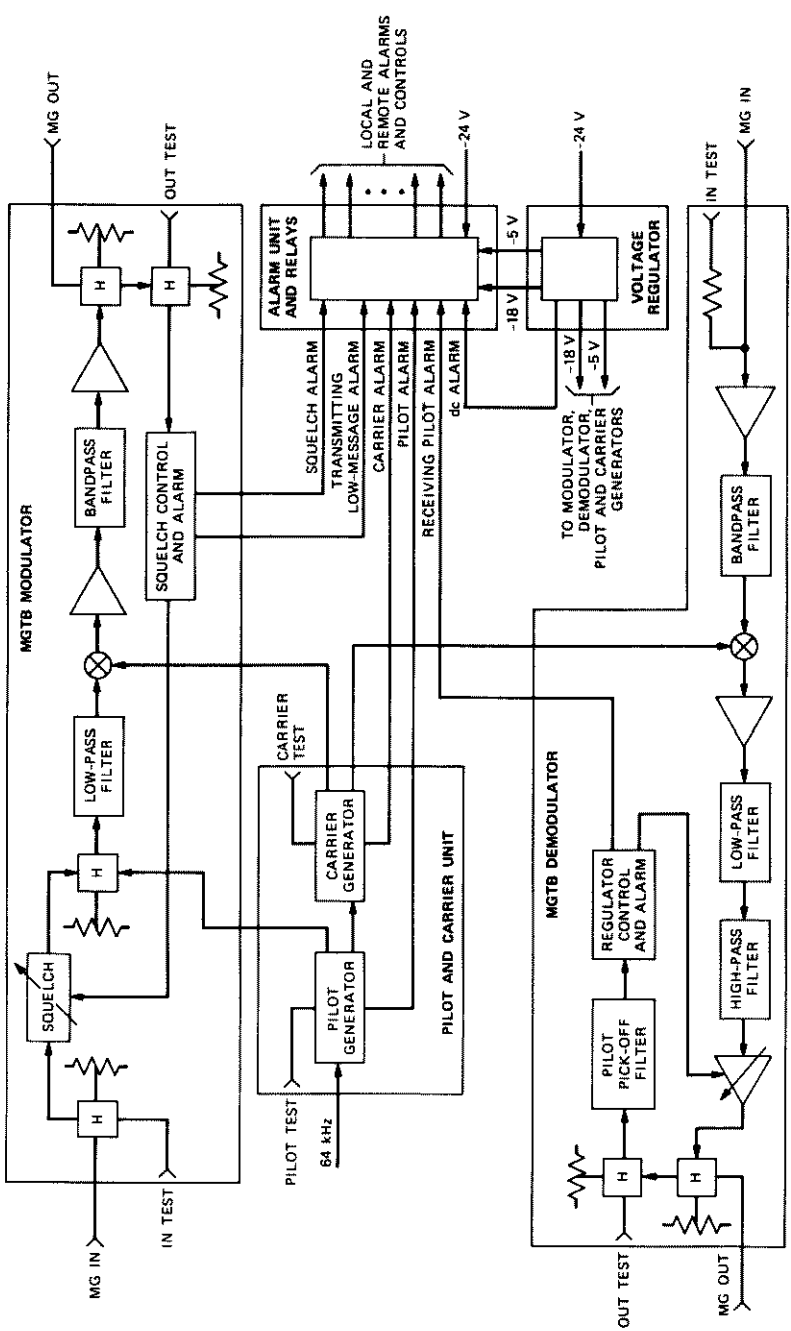



Fig. 3—Block diagram of mastergroup translator shelf.

7) to form the MMG spectrum for input to the MMGT-R. These particular mastergroups were selected to facilitate the realization of filters within the MMGT-R equipment.

### *2.3 Small size and low initial cost*

With MGT designs, the only common equipment in a bay is its battery, fuse, and synchronization distribution panel. Cost of this common equipment is a small portion of the cost of a fully equipped bay. Hence, the incremental growth cost per mastergroup is essentially the cost of the MGT shelf itself. An MGTB bay occupies only half the space needed for earlier equipment designs providing equivalent capability. All circuits for an MGT mastergroup are contained in five small plug-in units per shelf, permitting simplified maintenance or replacement.

### *2.4 Overall design features*

The five plug-in units of an MGTB mastergroup translator are a voltage regulator, alarm unit, pilot and carrier generator, and modulator and demodulator units. These five units contain the circuitry required to translate the U600 mastergroup signal to and from its line spectrum. Test jacks on the plug-in units permit signal monitoring at the inputs and outputs of the modulator and demodulator and at the outputs of the mastergroup pilot and carrier generators.

Transmission and circuit function alarms are displayed locally by indicators on the alarm plug-in and atop the bay. They are also made available to remote surveillance systems. These alarms may be operated in a latched mode, which causes retention of alarm indications until a memory release function is performed.

During a persisting alarm state, the audible alarms are silenced by depressing an Alarm Cutoff (ACO) switch on the alarm unit.

### *2.5 Circuit design features*

#### *2.5.1 Voltage regulator circuit*

A −24 volt office battery input drives two series voltage regulators to provide −18 and −5 volt outputs to power the mastergroup translator shelf circuits. The dc drain on the office battery is 1.3 amperes per MGTB shelf.

#### *2.5.2 Pilot and carrier generator circuit*

A −20 dBm0, 2.840-MHz mastergroup pilot is used to indicate the relative signal level of the mastergroup. This pilot is added in between supergroups of the mastergroup signal in the transmitting circuit before modulation. Thus, it appears in the line spectrum translated by the carrier frequency. This pilot is used for automatic regulation

and alarming. The pilot generator oscillator is phase locked to a stable Primary Frequency Supply (PFS) synchronization source which is locked to an OMFS. Hence, the stability and frequency accuracy of the mastergroup pilot are essentially the same as those of the OMFS supply.

The 64-kHz synchronization signal from the primary frequency supply is divided down to 32 kHz. This 32 kHz is used as a reference signal at one input of a phase comparator. The output of a voltage-controlled crystal oscillator, also divided down to 32 kHz, is fed to the second input of the phase comparator. Phase comparison of these two signals results in a dc control voltage to the oscillator to alter its frequency until the phases of both 32-kHz signals at the comparator inputs are identical. This constitutes the phase-locked loop concept of signal generation wherein the generated signal is stated as being phase locked to a reference signal. A filter at the generator output suppresses harmonics of the pilot frequency.

The MGTB carrier generation is similar to that of the mastergroup pilot generation previously described. The same 64-kHz synchronization signal used for pilot generation is used directly to phase lock the carrier generator to the Bell System reference signal. Two outputs at +10 dBm serve the modulator and demodulator in each MGT shelf. (See Table I for characteristics and performance of MGTB.)

Table I—Characteristics and performance of MGTB

| Characteristic | Performance |
|---|---|
| Type of modulation | SSB-SC-AM |
| Number of VF channels | 600 |
| Basic MG spectrum (U600) | 0.564 to 3.084 MHz |
| MG3B spectrum | 8.628 to 11.148 MHz |
| MG4B spectrum | 11.316 to 13.836 MHz |
| MG5B spectrum | 14.004 to 16.524 MHz |
| MG6B spectrum | 16.692 to 19.212 MHz |
| MG7B spectrum | 19.380 to 21.900 MHz |
| Return loss, all ports across their spectrum | >20 dB |
| Back-to-back frequency response | flat within ±0.4 dB |
| Receive pilot regulation range | ±5 dB |
| Loss of pilot regulation | Receiver reverts to normal manual gain setting when input pilot level deviates >−10 dB from nominal. |
| Noise performance (fully loaded and back to back) | <10 dBrnc0 |
| Phase-jitter performance (back to back) with two independent carrier supplies | <1° peak to peak, Bell weighted |
| MG carrier phase jitter | <0.7° peak to peak, Bell weighted |
| MG carrier supply frequency stability | $1 \times 10^{-9}$ with PFS-2B supply and OMFS |
| dc current drain per MGT shelf | 1.3 ampere, maximum at −24 volts dc |
| dc supply voltage | −20 to −28 volts dc |

## III. MMGT-R TRANSLATOR

### *3.1 Introduction*

The MMGT-R provides frequency translation between MMG spectra at the MGTB translators and the IF spectrum for AR6A radio. Two MMG signals (MMG1 and MMG2) are individually translated and combined to form the lower and upper portions of the AR6A IF spectrum centered at 74.1 MHz. Three radio-line pilots, required for regulation in the AR6A System, are generated in the MMGT-R transmitter.

Because the signal output of the MMGT-R transmitter is single sideband, suppressed carrier, and amplitude modulated (SSB-SC-AM), minimal carrier energy is transmitted with the sideband information. To enable the receiving translator to track and reconstruct the transmitted message accurately, a substitute form of carrier is added to the message in the transmitting translator. This is a highly stable, 16.608-MHz recovery pilot that serves as an input to a phase-locked loop in the receiving translator to control its demodulator carrier frequency. This corrects frequency offset on message due to repeatered transmission.

MMGT-R terminals are protected on a basis of one protection T and R translator pair for up to 14 regular T and R translator pairs.

### *3.2 Overall description*

#### *3.2.1 Regular terminal*

The block diagrams for the regular transmitting and receiving translator portions of the MMGT-R terminal are shown in Figs. 4 and 5, respectively.

In the transmitting direction, MMG1 and MMG2 with 13.920-MHz continuity pilots from MGTB are each translated independently up to adjacent frequency bands of the radio IF spectrum. Their combining produces a composite IF signal of two uncorrelated spectra of 3000 circuits each in the band from 59.8 to 88.5 MHz, centered at 74.1 MHz.

A number of pilots for functions required by MMGT-R and AR6A are generated within the MMGT-R transmitter and are added to the MMG signal at a −10 dBm0 level prior to translation. These include three radio-line pilots at 11.232 and 21.966 MHz in MMG2, and 19.296 MHz in MMG1. Their translated positions in the IF spectrum are shown in Fig. 1. They control automatic gain and dynamic amplitude equalizers in the radio receivers. They also serve as continuity pilots for the 500A Protection Switching System.

A 16.608-MHz recovery pilot is added to both MMG1 and MMG2 spectra prior to translation in the MMGT-R transmitters. This pilot is used in place of the suppressed carrier for demodulator carrier

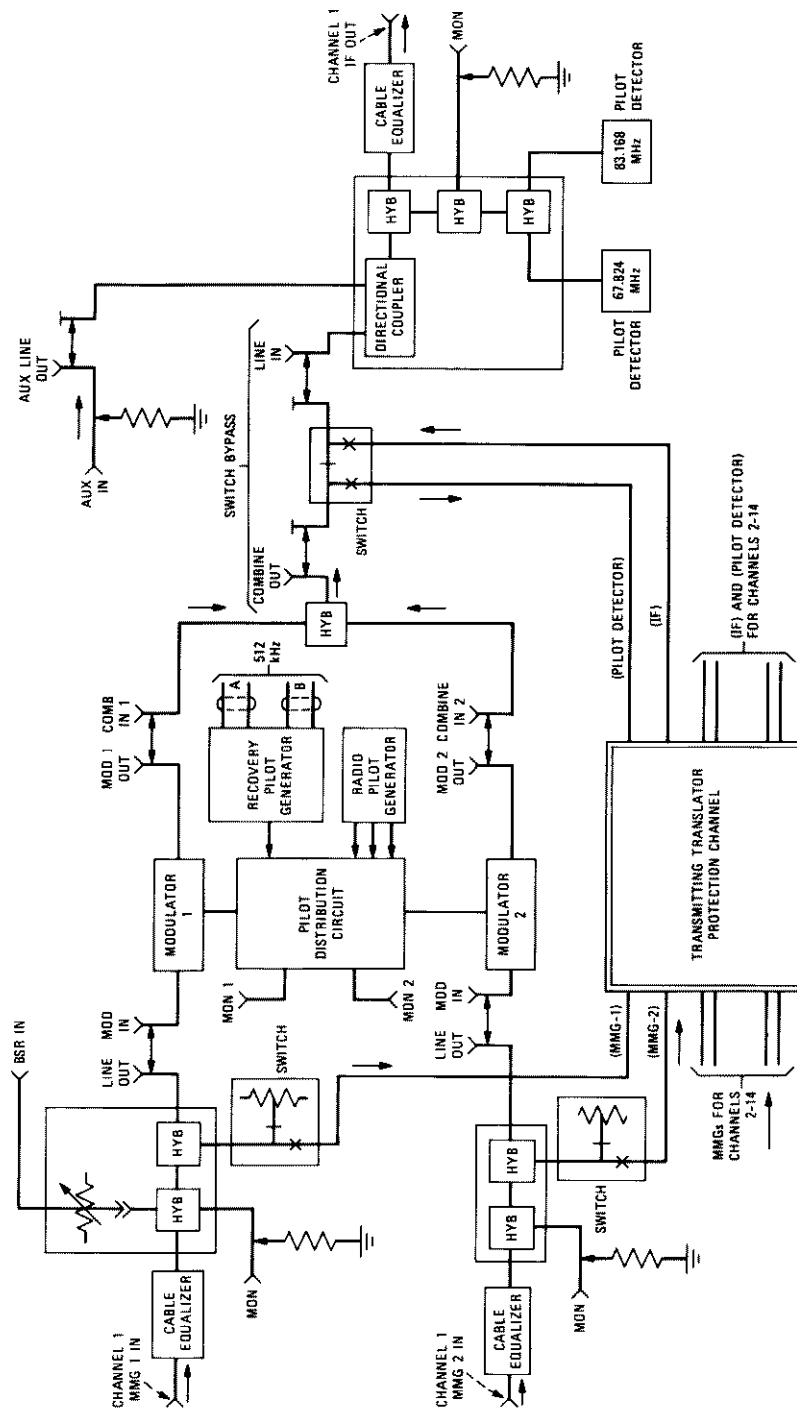



Fig. 4—Simplified block diagram of transmitting MMGT-R translator.

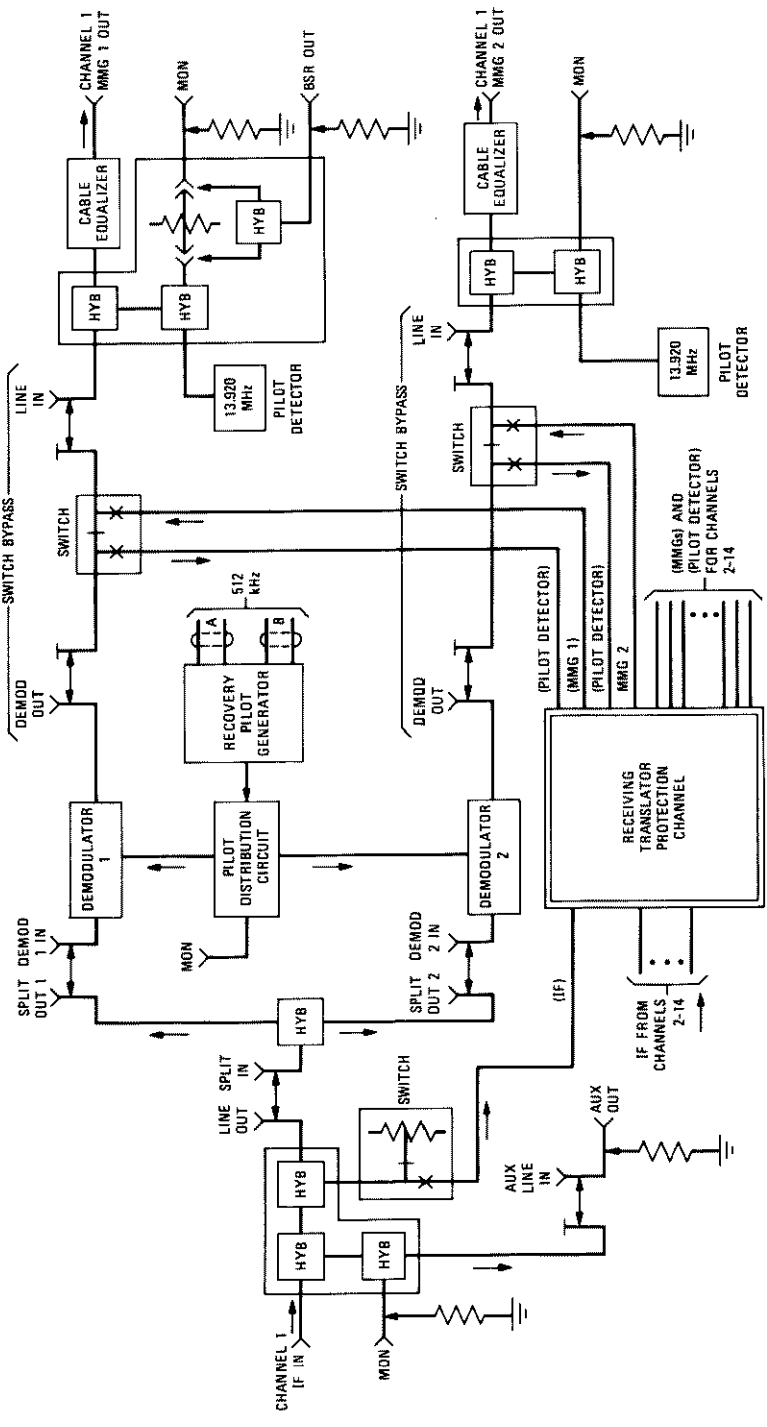



Fig. 5—Simplified block diagram of receiving MMGT-R translator.

frequency control at the receiving station. This function is discussed in greater detail later.

On occasion, it is necessary to use AR6A to transmit the Bell System reference frequency generated at Hillsboro, Missouri, to points not served by other carrier facilities. A BSRTU generator converts a 2.048-MHz reference frequency to two pilot tones separated by 64 kHz. These 11.200- and 11.264-MHz tones are added to MMG1 prior to translation in the MMGT-R transmitter. They are recovered at the receiving MMGT-R where a BSRT receiver converts their difference frequency to the original 2.048-MHz reference frequency. Although the absolute frequencies of the two tones may have shifted because of frequency error attributed to radio repeaters, their precise 64-kHz difference is preserved. Thus, the Bell System reference frequency integrity is preserved.

In the receiving direction, the IF signal from AR6A radio is passed through a splitting hybrid. A pair of upper and lower IF sideband filters separates the signal into its two original sideband components. Each sideband is then translated downward to the multimastergroup spectrum for further demodulation to the basic mastergroup spectrum in MGTB equipment.

### *3.2.2 Protection terminals*

As shown in Figs. 4 and 5, protection transmitting and receiving translators are provided for equipment protection, maintenance, and restoration. The equipment required for this includes switches, pilot detectors, pilot oscillators, and a microprocessor control circuit. The switches, pilot detectors, and oscillators are located in the MMGT-R shelves. The 500B protection switch microprocessor control unit is located in an adjacent bay. Thus, only dc control leads are required between the 500B and MMGT-R bays, and transmission cable lengths are minimized.

Service is routed through the protection translator when a fault is detected in a regular translator or when a manual switch is requested. One protection translator provides protection for up to 14 regular translators. Receiving and transmitting translators are protected independently of each other. This results in two separate 1 × 14 switching arrangements under the common control of a single 500B controller at each MMGT-R location.

Loss of the 13.920-MHz continuity pilot in an MMG is detected by the MMGT-R equipment. When a failure occurs, the 500B controller transfers both MMG1 and MMG2 to protection even though a failure may have been associated with only one MMG. Local and remote alarms and indications are provided by the 500B.

### 3.3 Detailed description

#### 3.3.1 Transmitting translator

Each transmitting translator includes all of the networks and circuits required to translate a pair of MMG signals to the lower and upper IF frequencies (see Fig. 1). The protection transmitting translator differs from the regular one only in the peripheral circuits associated with the input and output functions. Plug-ins MODULATOR 1 and MODULATOR 2 (see Fig. 6) each contain a balanced diode ring modulator, amplifiers, and a free-running, crystal-controlled carrier oscillator. IF bandpass filters select the lower sideband of each modulator.

A 16.608-MHz recovery pilot generator (RCVRY PLT GEN) is phase locked to a local 512-kHz signal derived from the OMFS which is locked to the Bell System reference. A hybrid transformer network distributes the recovery pilots to the MMGT-R modulators.

The radio-line pilots are generated by three free-running crystal oscillators at frequencies of 11.232, 19.296, and 21.966 MHz and are added to the MMG signals by the hybrid transformer network above.

The distribution network maintains at least 85 dB of isolation between the two MMG signal paths of an MMGT-R.

MMG1 and MMG2 input networks provide terminated monitoring access and split the signal to provide inputs to the protection translator. The Bell System Reference (BSR) input option is required only for MMG1.

An output port is available to make terminated measurements of the IF output signal. An auxiliary input port may be used for restoration. Pilot detectors monitor the translated 13.920-MHz continuity pilots to initiate a switch of both MMGs to protection when the level of either or both drops 6 dB or more.

#### 3.3.2 Receiving translator

The receiving translator accepts an IF signal from AR6A via the 500A protection switch. This IF signal is split into two paths by a hybrid, following which the upper and lower IF bands are selected with IF bandpass filters. Each is then demodulated to the MMG

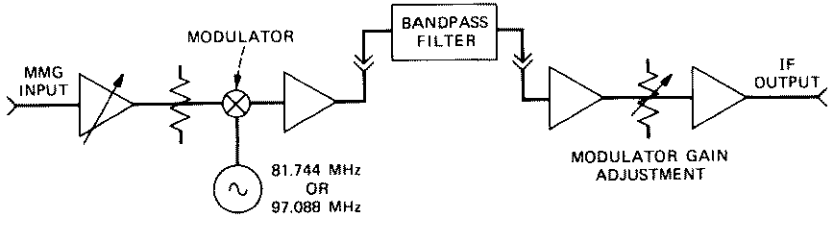



Fig. 6—MMG1 or MMG2 modulator.

spectrum for connection to receiving MGTB equipment. Plug-ins DEMODULATOR 1 and DEMODULATOR 2 (see Fig. 7) use ring-type modulators as in the transmitting direction. The demodulator carriers are generated by voltage-controlled crystal oscillators. Their frequency is controlled by a phase-locked technique to correct the frequency error that accumulates along the radio line. This technique is described below.

A 16.608-MHz recovery pilot is generated in the receiving office translator in a manner identical to that used in the transmitting office translator. It is phase locked to the Bell System reference via the local 512-kHz reference frequency at the receiving site. This 16.608-MHz signal serves as the reference input to the phase-locked loop of each demodulator carrier generator.

From each demodulated MMG in the receiving MMGT-R, the recovered 16.608-MHz pilot, with any frequency offset acquired over the radio line, is selected by a narrowband crystal filter. The phase of this received pilot is compared to that of the locally generated 16.608-MHz signal. Any phase difference between them is converted to a dc control voltage change. This voltage change alters the frequency of the carrier oscillator in a direction to correct frequency offsets of at least ±2500 Hz in the received signal. This technique ensures that the recovered MMG signal is a faithful reproduction of the original transmitted signal.

To enhance the pull-in range of the phase-locked loop, a low-frequency (≈2 Hz) oscillator has been added to the loop filter feedback circuit. This oscillator is activated upon interruption of one or both phase comparator input signals, and when activated, it modulates the

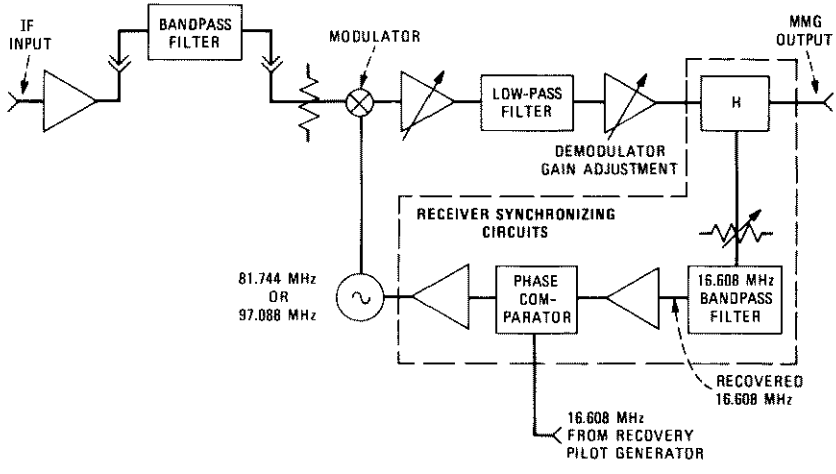



Fig. 7—MMG1 or MMG2 demodulator.

oscillator dc control voltage with sufficient amplitude to sweep the carrier frequency over a range of ±2500 Hz from nominal. This facilitates recapture of a restored signal. In addition, the low-frequency signal is peak detected to generate an out-of-lock alarm and a protection-switch request. Once the loop reacquires lock, the low-frequency oscillator is immediately disabled and has no effect on normal operation.

To accommodate space-diversity applications, the receiver phase-locked loop has been designed such that large phase hits coupled with sudden signal up fades will result in less than 2 ms of loop-phase instability.

The receiving IF network provides terminated monitoring access and splits the signal for protection switching. It also provides an auxiliary output for restoration. Signal monitoring access is provided at both translated MMG outputs. Pilot detectors monitor the recovered 13.290-MHz continuity pilots to initiate protection switching. On a 6 dB or more drop in level of either or both MMG pilots, both MMGT-R translators will switch to protection.

#### *3.3.3 Recovery pilot generator*

The 16.608-MHz recovery pilot generator is shown functionally in Fig. 8. Reproducing the transmitted signal accurately is highly dependent upon synchronizing the pilot generator to accurate 512-kHz reference frequency signals. These are derived from the OMFS, which, in turn, is locked to the Bell System frequency source at Hillsboro, Missouri.

To enhance overall reliability, redundant 512-kHz inputs, designated A and B, are supplied to each recovery pilot generator. This is done over separate 135-ohm balanced cable buses and, where possible, via different cable ducts.

The recovery pilot generator accepts two 512-kHz signals, filters them in narrowband crystal filters, amplifies them, and via a divider, converts each to a 32-kHz square wave. A logic switch, preferential to the A signal, provides the selected 32-kHz square wave to one input of a digital phase comparator. The second input to the phase comparator is derived from a portion of the 16.608-MHz crystal oscillator output, which is converted to a 32-kHz square wave. Phase comparison of these two signals results in a dc control voltage that locks the oscillator precisely at 16,608,000 Hz.

Manual test operation of the 32-kHz logic switch is provided. However, phase hits could occur due to reference frequency signal switching. To prevent this, manual switching in the regular channel recovery pilot generator is inhibited unless service is being carried on the protection channel. Conversely, a manual switch in the protection

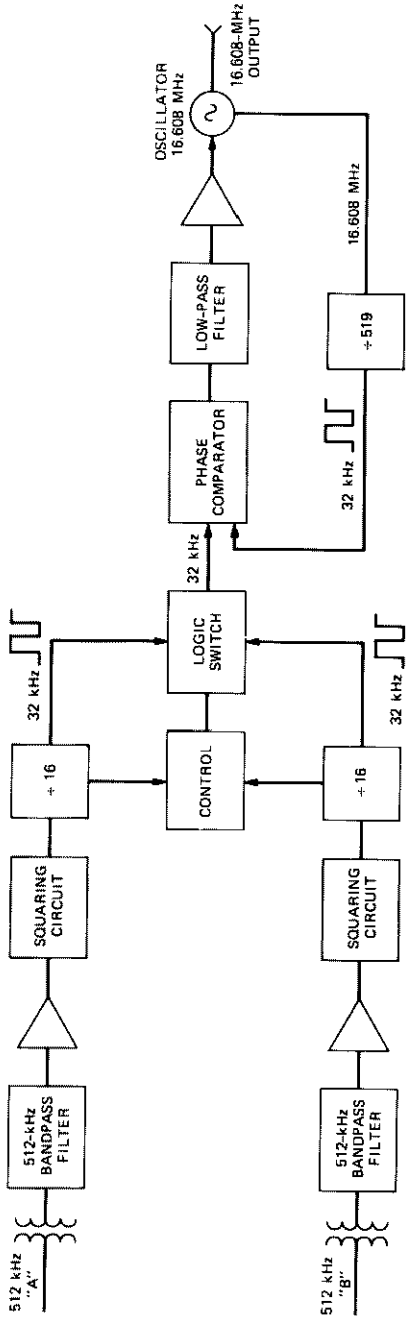



Fig. 8—Functional diagram of a recovery pilot generator.

channel recovery pilot generator can only be made when the protection channel is idle. (See Tables II and III for characteristics and performance of MMGT-R.)

### *3.4 Realization of objectives*

The performance objectives listed represent acceptable levels of performance. Measured data reflect a level of performance better than the values shown. Examples may be seen in Figs. 9 and 10. Figure 9 shows a plot of amplitude versus frequency for a production transmitting and receiving translator pair connected back to back. Figure 10 shows representative noise load data for the same equipment in the same configuration. As the figure indicates, noise performance is thermally limited.

The overall amplitude versus frequency curve in Fig. 9 is a nonadjustable parameter. It is influenced by the response of the individual components (amplifiers, filters, etc.). The only equalization used with

Table II—Characteristics and performance of MMGT-R

| Characteristics | Performance |
|---|---|
| Type of modulation | SSB-SC-AM |
| Number of MMGs | 2 per AR6A channel |
| Number of MGs | 5 per MMG (10 total) |
| MMG spectrum (MG 3-7) | 8.628 through 21.900 MHz |
| IF output spectrum | 59.844 through 88.460 MHz |
| MMG transmit input level | −41.6 dBTL |
| IF transmit output level | −23.9 dBTL |
| IF receive input level | −32.1 dBTL |
| MMG receive output level | −22.4 dBTL |
| Return loss (all ports) | >25 dB |
| dc power—input voltage | −20 to −28 volts dc |
| dc current drain (per shelf) | 2.8 amperes maximum at −24 volts dc |
| Normal power dissipation per 7-foot bay (3T/R pairs) | 400 watts |
| Protection | 500 B − 1 × 14 maximum |

Table III—Performance of MMGT-R

| Performance Objectives | Worst-Case Deviation |
|---|---|
| Gain frequency response (1 − TR pair) | <±0.3 dB peak to peak across any MG |
| Noise performance (equivalent 6000-circuit load, 1 − TR pair) | <15 dBrnc0 |
| Crosstalk | <−100 dB |
| Carrier leak | ≤−70 dBm0 |
| Spurious tones | ≤−70 dBm0 |
| Phase-jitter performance for ten terminal pairs in tandem | ≤1° peak to peak, Bell weighted |
| Frequency stabilities (over 80° ±20°F temperature range and permissible power supply variation) | |
| Transmitter carrier oscillator frequency | ±200 Hz |
| Radio-line pilot oscillator frequency | ±4 parts per million |

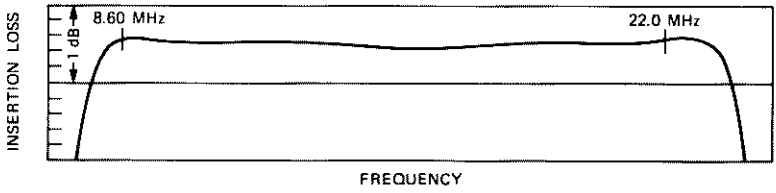



Fig. 9—Typical amplitude vs. frequency response of transmitting/receiving translator.

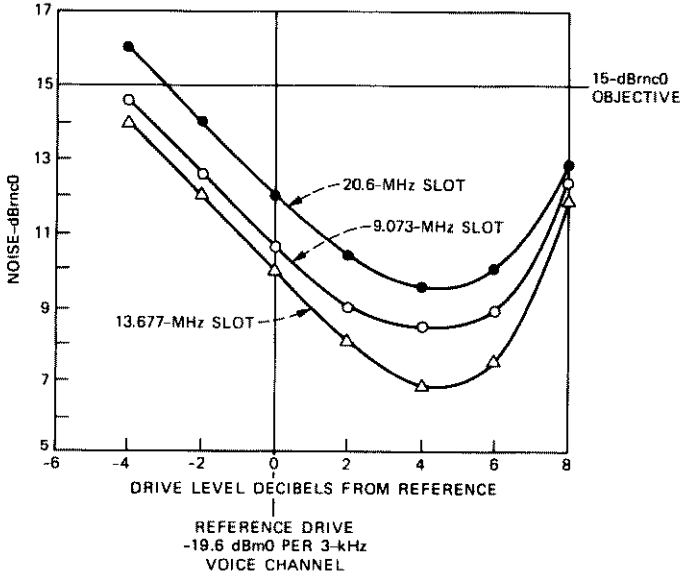



Fig. 10—Typical noise load performance of transmitting/receiving translator.

MMGT-R compensates for amplitude slope due to cable lengths in trunks between the MMGT-R, the MGTB, and the 500A equipment.

Mastergroups assigned to AR6A contain a mixture of analog-voice and voiceband data circuits subject to impairment due to excessive phase jitter. Strict phase-jitter requirements are placed on the transmit and receive carrier generators and on the recovery pilot generator to minimize phase-jitter transfer to data signals.

### *3.5 Physical design considerations*

#### *3.5.1 Goals*

The physical arrangement of the MMGT-R equipment is oriented to achieve three major goals: (1) high reliability; (2) ease of installation, maintenance, and growth; and (3) cost-effectiveness. The first two items influence the third.

### 3.5.2 Reliability

When considering that an MMGT-R shelf carries up to 6000 two-way voice circuits and that a 7-foot bay carries up to 18,000 such circuits, reliability is of utmost concern. With this in mind, an early decision was made to minimize the amount of common equipment and to have each transmitting and receiving translator function independently.

This simplified the protection aspects of the system and minimized the risks to remaining operational parts of the system when equipment failures require repair or maintenance.

Carriers and recovery and radio-line pilots are all generated on a per-shelf basis with plug-in modules that may be replaced easily.

Converters of the dc-to-dc type are provided on a per-shelf basis with separate power for switching and transmission functions. A switching power failure does not impair transmission, and active transmission circuits are fully protected.

By locating switches and detectors in the same shelf as the equipment they protect, transmission cable lengths and interbay wiring and cabling are minimized.

The shelf and plug-in layouts promote rapid and natural heat dissipation. A combination of baffles and perforated cover assemblies permit the free flow of air vertically and from front to rear of each shelf. Each shelf has its own airflow pattern, avoiding high concentration of heat in the upper region of the bay.

### 3.5.3 Ease of installation, maintenance, and growth

The combination of the factors described previously led to a completely self-contained, totally connectorized shelf design. Thus, it requires no installer wiring for initial or subsequent growth installation. An alarm and fuse panel at the top of each bay constitutes the only MMGT-R common equipment. The protection MMGT-R and switch controller are naturally required with the initial installation to permit subsequent growth.

Redundant diode-firmed −24 volt power leads and 512-kHz reference frequency leads are connectorized and provided from the fuse panel to each shelf position in the bay. All active circuitry is of the plug-in type and all passive circuitry uses connectors for ease of removal or replacement. Access ports are provided for components outside protection switching boundaries to readily permit their bypassing for removal or replacement without delaying service restoration. These design concepts achieve a highly reliable and easily maintained circuit and equipment arrangement capable of "graceful growth."

### *3.5.4 Cost-effectiveness*

As we stated earlier, achievement of the first two goals contributes significantly to providing a cost-effective system. By minimizing the amount of common equipment, the costs for new installations or for low cross-section systems with limited growth potential are reduced. Thus, the costs per circuit mile tend to be distributed more evenly. The reduced time required for installation and maintenance also contributes to lower overall operating expenses as well.

## REFERENCE

1. T. J. Aprille, "The Multiplexing of 22 Mastergroups for L5E," Int. Commun. Conf., Chicago, Illinois, June 14, 1977.

## AUTHORS

**Antoine Dubois,** E.E. Assoc., 1968, Merrimack College; Western Electric 1960–1967; Bell Laboratories, 1967—. Mr. Dubois began working at Western Electric Quality Assurance in 1960, and in 1967 joined Bell Laboratories, where he participated in the lineup of line-connecting and mastergroup-multiplexing circuits along the initial L-4 route. Since then, his responsibilities expanded to mastergroup administration. This includes mastergroup-multiplexing and associated circuitry used for multimastergroup transmission in cable and radio systems.

**Donald N. Ritchie,** Certificate (Mechanical Engineering), 1941, Stevens Institute of Technology; Western Electric, 1941–1948; Bell Laboratories, 1948—. While at Bell Laboratories Mr. Ritchie has been involved with design standards, short-haul radio system applications, outside supplier auxiliary channel radio applications, data under voice, multiplex terminal physical design, and pulse-code modulated terminal physical design.

**Frederick M. Smith,** A.E.E., 1960, Wentworth Institute; Bell Laboratories, 1960—. Since starting with Bell Laboratories Mr. Smith has been involved with such projects as short-haul microwave radio and terminal facilities. He is presently responsible for the MMGT-R multiplex terminal used with AR6A single-sideband radio systems.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Frequency Control

By J. M. KIKER, Jr.* and S. B. PIRKAU*

(Manuscript received February 14, 1983)

The AR6A Radio System requires precise control of the local oscillator frequency at repeaters and main stations, so that the IF equalizer pilots will be accurately located with respect to the narrowband crystal pick-off filters. In addition, the accumulated frequency error from each repeater and main station must be limited to a value that can be corrected by the tracking receiver of the multimastergroup terminal. Precise frequency control is maintained along the route by locking local oscillators to a stable and accurate frequency reference provided at each repeater station by the Microwave Carrier Synchronization Supply (MCSS). The MCSS provides a source and a backup for this signal. Extensive monitoring circuitry in the MCSS detects failed conditions. The accuracy of the two oscillators is continuously monitored by making frequency comparisons of their output. These frequency-monitoring circuits are also used to make yearly adjustments to the oscillators using a rubidium frequency standard as reference. This paper describes the MCSS System that has been developed for the AR6A Radio System.

## I. INTRODUCTION

Compared with current 6-GHz FM Radio Systems like TH-1 or TH-3, the new AR6A† Radio System requires much tighter radio-frequency tolerances to keep radio-line pilots within narrowband pick-

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

off filters. These pilots should remain within ±2 kHz of the center frequency of their pick-off filters, which translates to a frequency tolerance of about three parts in $10^7$ compared to the TH-3 tolerance of two parts of $10^5$. This tighter tolerance is beyond the capability of currently available microwave generators used as carrier sources for up conversion and down conversion in the radio Transmit-Receive (TR)* units. The required stability is obtained by synchronizing the microwave generators and shift oscillators to a highly stable frequency synchronization supply located in each radio station. This AR6A Microwave Carrier Synchronization Supply (MCSS) is described in the following sections. The paper then presents an overall block diagram description of the MCSS and discusses in more detail the various subsystem components.

## II. FUNCTIONAL DESCRIPTION OF THE MCSS

The MCSS distributes a highly accurate sine-wave frequency of 308.8735416 kHz (abbreviated as 308.9 kHz) to each TR-bay microwave generator and shift oscillator as illustrated in Fig. 1. All microwave-generator frequencies and the shift-oscillator frequency are harmonics of this reference signal and are locked to it by phase-locked loop circuitry located in the frequency control units. The distribution bus provides 32 isolated outputs that can supply reference signals to a fully loaded route of eight two-way AR6A radio channels (eight channels by two directions by two oscillators per bay).

Figure 2, a block diagram of the MCSS, shows two symmetrical sections (frequency supplies), one of which is used for on-line operation and the other for standby. The major component in each of the two sections is a crystal-controlled, 4.94197666-MHz (hereafter abbreviated to 4.94 MHz) oscillator that has a long-term frequency stability of 1.8 parts in $10^7$ per year. Each oscillator is associated with a divider/switch unit, which performs both a frequency division of 16 and power monitoring functions. Although only one of the two oscillators is used at a time, a continuous frequency comparison between them checks for an excessive frequency difference and generates appropriate alarm signals if the difference exceeds certain values. The interconnection between the two sections is accomplished through the control unit, which offers automatic switching between output signals under trouble conditions or manual switching for maintenance purposes.

Figure 3 is a picture of the MCSS that shows the major component parts described above.

---

* Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

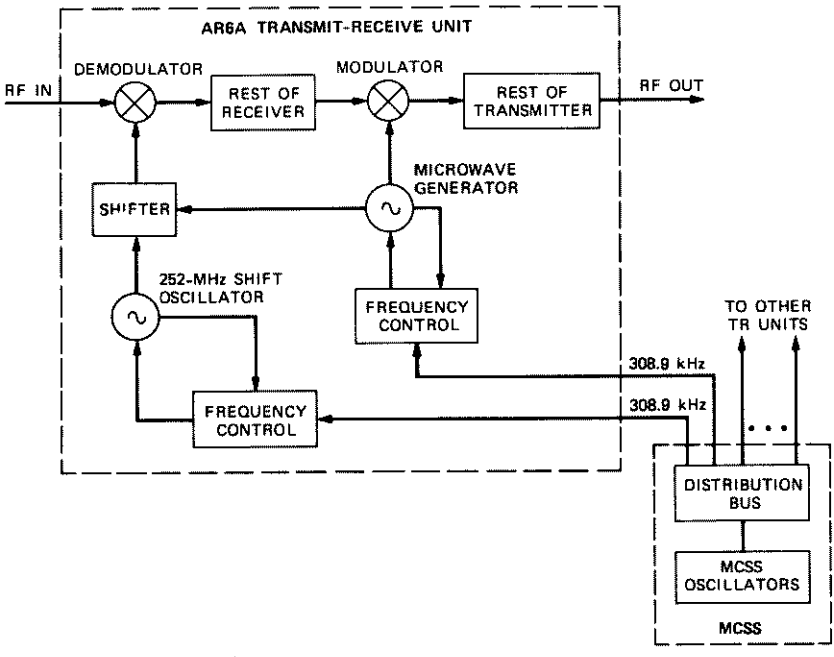



Fig. 1—Synchronization of AR6A TR units to MCSS.

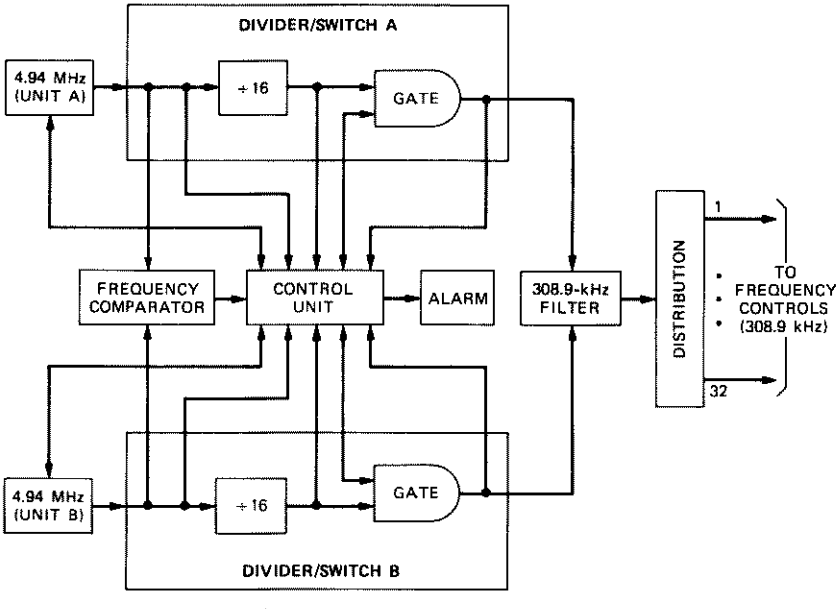



Fig. 2—Block diagram of MCSS.

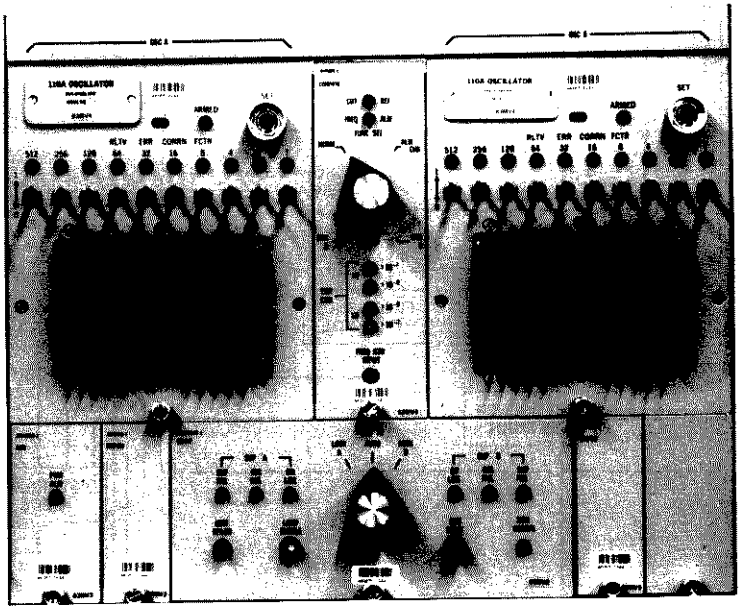


Fig. 3—Actual MCSS.

## III. DETAILED CIRCUIT DESCRIPTION

### *3.1 4.94-MHz oscillator**

The 4.94-MHz oscillator, shown in block diagram form in Fig. 4, consists of a crystal-controlled oscillator, RF amplifier, frequency-controlling network, and temperature-control network in a thermally stable environment. Because of repeater-station temperature variations, use of a stable oven is necessary to obtain the required frequency stability. Other peripheral circuitry such as the heater power amplifier, the binary frequency-controlling switches, and their display Light Emitting Diodes (LEDs) are exposed to ambient temperature.

#### *3.1.1 Radio Frequency (RF) oscillator and amplifier*

The RF oscillator is a modified Pierce-type crystal oscillator using a precision quartz crystal unit. A capacitance and varactor-diode network is in series with the crystal for frequency adjustment of ±3 ppm from the nominal frequency of 4.94197666 MHz. This stage has automatic gain control to ensure stable frequency operation. The crystal unit is a third-overtone AT-cut resonator having a turnover temperature between 76 degrees and 86 degrees Celsius. The output

* The MCSS oscillator was designed by M. W. Zuidervliet, Jr. of Bell Laboratories in Allentown, PA.

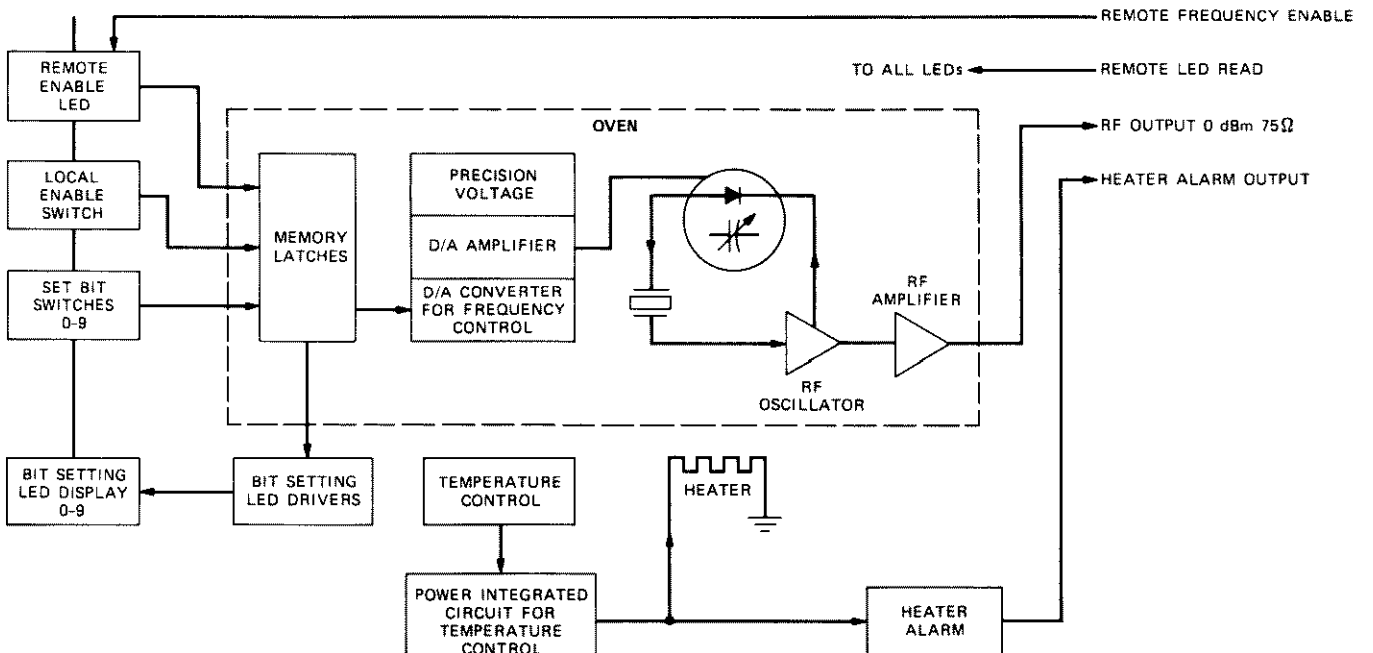



Fig. 4—Block diagram of MCSS oscillator.

of the oscillator is at the base of a low-noise transistor that uses the crystal unit to filter out any noise generated in the transistor that is more than a few hertz from the operating frequency. The output impedance of the oscillator is 50 ohms.

The RF amplifier consists of four transistor stages that buffer the oscillator from the output and boosts the output signal level of 0 dBm at an output impedance of 75 ohms.

### *3.1.2 Frequency tuning circuitry*

The varactor diode, in series with the crystal, is biased from the frequency adjust circuitry made up of a precision voltage reference source, Digital-to-Analog (D/A) converter, and current-to-voltage operational amplifier. Front-panel toggle switches control memory-latch circuits that, when enabled, supply digital input to the D/A converter. The D/A converts this input to a precision current value utilizing the precision voltage reference. A voltage proportional to this current is applied as bias to the varactor diode. The ten toggle switches are arranged for binary weighting with the least-significant bit giving a frequency change of one part in $10^8$. A new switch setting is input to the memory latches only when a remote enable signal is present and a load button on the oscillator is depressed. At other times the position of the toggle switches has no effect. This feature is present to prevent accidental detuning of the oscillator. Since the switch settings do not necessarily agree with the status of the memory latches, display LEDs are provided on the front panel that, when enabled, indicate the value of the digital input to the D/A.

### *3.1.3 Temperature control and heater alarm*

The temperature of the oven is maintained by a proportional temperature control using a Wien Bridge oscillator where the sensing elements are two thermistors in series. These thermistors are imbedded in the wall of the oven and are tightly coupled to the heater winding. The output of the Wien Bridge oscillator is buffered by an amplifier that has variable gain for adjustment of the sensitivity of the temperature-control circuit. This amplifier drives a power integrated circuit that provides the dc power to the heater winding. The power stage limits the current into the 15-ohm heater winding to approximately 1-1/3 amps.

There is an alarm circuit that senses the heater-winding control voltage and emits an alarm if this voltage moves beyond the window for normal operation. This is an indication that the oscillator already has or is about to lose temperature control and therefore change frequency.

### *3.2 Divide-by-16 circuit*

The main purpose of this circuit is to convert the 4.94-MHz input to 308.9 kHz and gate this signal to the distribution bus when the supply is on-line. There are two identical circuits corresponding to sides A and B.

Figure 5 shows a block diagram of the divide-by-16 circuit. Following the main signal path, the 4.94-MHz sinusoidal input signal is converted to a Transistor-Transistor Logic (TTL)-compatible square wave and divided by 16. The signal from the divider is input to one side of an AND gate. If this supply is to be selected as on-line, a select signal from the control unit enables the other gate input and the 308.9-kHz signal is output to the distribution bus.

Various monitors in the circuit detect improper operation. At the input, a detector monitors the level of the incoming 4.94-MHz signal and, if that level drops 2 dB, outputs an alarm to the control unit. A gradual 2-dB drop in level is not sufficient to cause the sine-wave-to-square-wave comparator to fail; thus, the alarm to switch to the other supply can be given before actual loss of 308.9 kHz occurs.

A coarse frequency monitor of the 308.9-kHz signal guards against a failure of the divider. This will detect a gross error that would occur if the divider malfunctioned to produce the wrong division ratio. A detected error of this type is output as an alarm to the control unit.

The level of the 308.9-kHz signal output to the distribution bus is monitored. The output gate for this signal is the last active circuit component of the reference signal path to the TR-bay frequency control units. The alarm indication for loss of level is output to the control unit but is meaningful only if the gate output is on-line.

The circuit buffers and provides a TTL line driver output of the 4.94-MHz signal to the frequency comparator unit where precise frequency comparisons between the two oscillators occur.

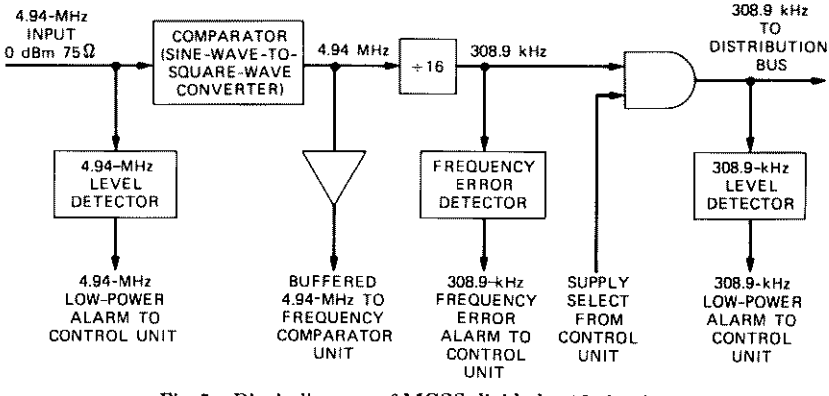



Fig. 5—Block diagram of MCSS divide-by-16 circuit.

### *3.3 308.9-kHz distribution*

The outputs of the two divider switch circuits are gated to the distribution circuit. Either circuit, but only one at a time, provides a 308.9-kHz TTL signal for the distribution bus. The two signal inputs are combined in a resistive power combiner followed by a 308.9-kHz bandpass filter that provides at least 40 dB of loss to frequencies of 618 kHz and higher with a passband loss of less than 1 dB.

The 308.9-kHz sine-wave output from the filter, at an impedance level of 135 ohms, is fed to a balanced transformer with a 135- to 8.06-ohm impedance transformation. This output drives a passive resistive distribution bus that provides 33 outputs balanced at an impedance of 135 ohms. Thirty-two of the outputs are for distribution to the TR bays and are terminated if not used. The remaining tap provides a monitor port for measurement of signal level at the bus output. There is 18 dB of signal loss from the filter output to any tap output, and the tap-to-tap isolation is 36 dB.

### *3.4 MCSS control unit*

The interconnection between the two 308.9-kHz sources of the MCSS is accomplished through the control unit. The control unit handles all MCSS alarms, as well as power and frequency sensor functions, and performs four separate major functions:

1. Selects (puts on-line) one of the two MCSS oscillators as the 308.9-kHz source.
2. Offers automatic switching to the other MCSS oscillator should the on-line unit develop any trouble.
3. Originates the general MCSS alarm together with visual indications pinpointing the specific alarm condition.
4. Furnishes the interface for the two oscillators, two divider and switch units, the frequency comparator unit, and the alarm unit.

All of the above-listed functions can be performed in either the auto or manual operating modes. The auto mode of operation is the normal operating condition of the control unit. In this mode, all sensors are monitored and an automatic switch will occur if there is a failure in the on-line supply. The sensors monitor the following items:

1. Power output of the 4.94-MHz oscillators
2. Operation of the oven within the oscillator unit
3. 308.9-kHz coarse output frerquency and power
4. Frequency difference between the two oscillators.

The sensors for items 1 and 3 are located on the divider and switch units and for item 4 on the frequency comparator unit.

The most severe alarm condition is the cut-reference alarm. When this alarm is generated the MCSS outputs to the TR bays are removed

and the microwave generators and shift oscillators become free running. This alarm can be generated for two reasons:

1. The frequency offset between the two oscillators is greater than 39.3 parts in $10^8$ (measured by the frequency comparator unit).

2. A failure has occurred first in the on-line section, causing an auto switch, and then in the former standby section, which was put on-line by the auto switch mentioned above. If the failure is in the standby supply, an oscillator fail alarm is generated and all switching between supplies is inhibited. Subsequent failure of the on-line supply will cause a cut-reference alarm.

The manual mode of operation is primarily for maintenance and repair of the MCSS. It permits a manual switch resulting in an interchange of the on-line and the standby section of the MCSS. However, consideration had to be given to the fact that whenever a manual switch to the standby supply is executed, a check has to be performed to determine if the newly selected supply does have a signal output of correct amplitude. Therefore, once a new supply is selected and on-line, enough time has to be allowed to permit a switchback to the previous on-line supply should the new one fail to produce a suitable output signal. Delay circuits permit this.

Since an auto switch will only occur as a result of a failure in the on-line section of the MCSS, a switchback is not possible and the check for a good output signal of the standby section is not necessary.

### *3.5 MCSS comparator unit*

#### *3.5.1 Functional description of the MCSS comparator unit*

The comparator unit performs two major functions in the MCSS.

1. For normal operation, there is continuous monitoring of the frequency difference between the two 4.94-MHz oscillators. An alarm signal is generated if the frequency difference exceeds 16.6 parts in $10^8$. A second alarm and cut-reference signal is generated when this difference increases to, or exceeds, 39.3 parts in $10^8$. The cut-reference signal removes the synchronization tone from the microwave generators and shift oscillators, and these units become free running.

2. Replacement or yearly routine maintenance requires frequency adjustment of the 4.94-MHz oscillator. In the frequency check mode of operation, the comparator aids in oscillator adjustment by indicating the frequency difference between the oscillator being adjusted and an external frequency calibration standard. In this mode, the comparator has a resolution of $\pm 1$ part in $10^8$, which is based on the settability of the 4.94-MHz oscillators whose smallest frequency adjustment increment is one part in $10^8$.

The position of the front-panel mounted function select switch, which can be in either the check A or check B position, determines

which 4.94-MHz oscillator is being checked against the external frequency standard. In the check B position, the external frequency standard takes the place of oscillator A and in the check A position the external frequency standard takes the place of oscillator B.

When in the check A or check B position, all frequency alarms are inhibited. The four light-emitting diodes on the front panel of the unit are enabled. These LEDs are labeled and arranged in the following manner:

$10^{-7}$
HIGH
$10^{-8}$
$10^{-8}$
LO
$10^{-7}$

The HIGH or LO indicates the direction of frequency deviation of the checked 4.94-MHz oscillator in comparison with the frequency standard. By observing the LEDs, the oscillator can be adjusted to within one part in $10^8$ of the 4.94-MHz signal.

Adjusting the oscillator is accomplished through ten toggle switches, labeled zero through 9, arranged in a binary weight fashion with the least significant bit being switch zero.

### *3.5.2 Method of detecting frequency offset*

The conventional method of detecting a frequency difference between two oscillators, in which one oscillator will generate a gate pulse during which the other oscillator is counted, results in large time intervals to detect small offsets. For example, to detect one part in $10^8$ for the MCSS 4.94-MHz oscillators would require a minimum time of $1/\Delta f = 20$ seconds, which is an unreasonable time interval.

The method chosen for measuring frequency differences in the MCSS is the same as that used on the Jumbogroup Frequency Supply (JFS) of the L5 System, which Fig. 6 shows utilizing the block diagram of the MCSS comparator. In this method, one of the two frequencies being compared is divided by D and then mixed with itself, with the sum signal A + A/D chosen by a bandpass filter. This signal is mixed with oscillator B and passed through a low-pass filter with the resulting signal difference of frequency

$$A + \frac{A}{D} - B = \frac{A}{D} + \Delta f.$$

This signal is used as a clock signal for an N divider that generates a gate pulse. During the gate-pulse interval, counts are accumulated by the C counter. This has the effect of multiplying the real fractional

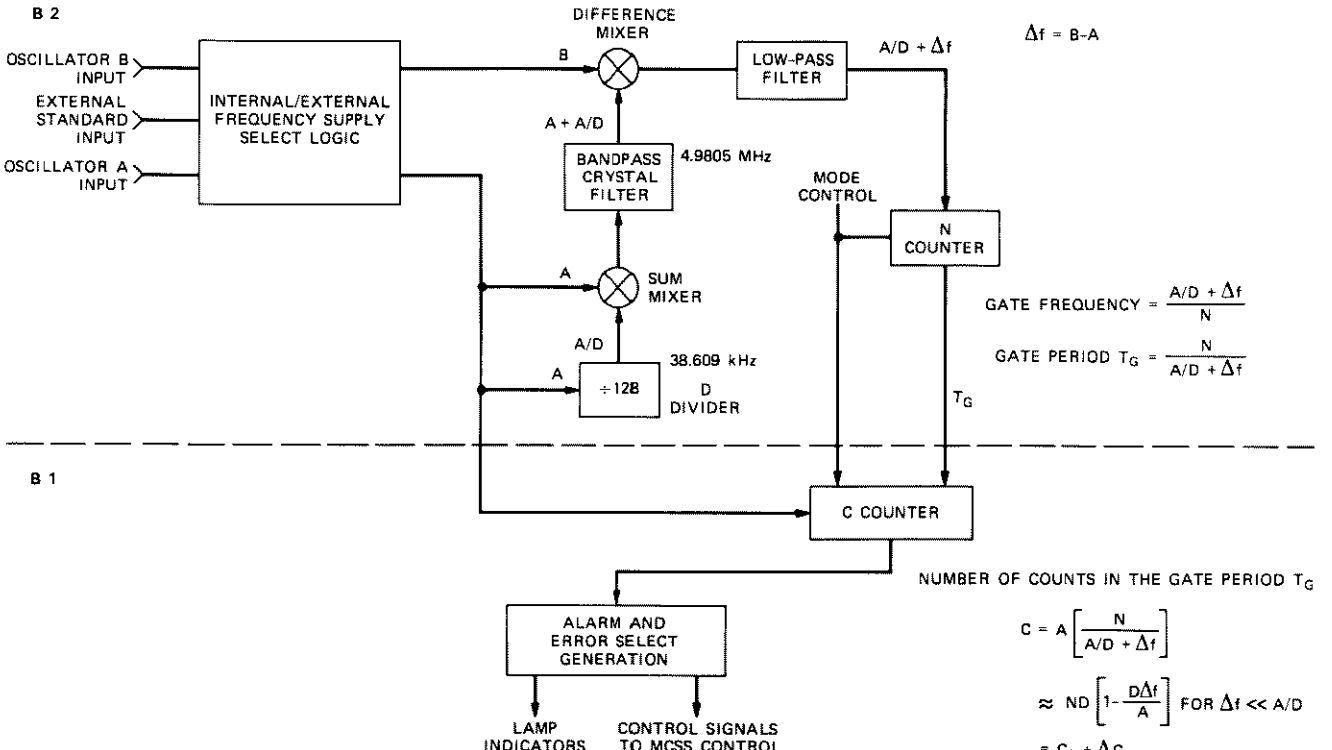



Fig. 6—Block diagram of MCSS comparator unit.

frequency error by a factor D. For the MCSS a value of D = 128 was chosen as realistic when considering the need for the bandpass filter that must pass A + A/D and reject A − A/D. Now the minimum time to detect an error of one part in $10^8$ is the time for $ND^2\Delta f/A$ to equal one, i.e., one extra or one less count than normal for no frequency error. Solving for N, one obtains

$$N = \frac{A}{D^2\Delta f}$$

The gate interval for the minimum count to occur is $T_G \approx ND/A = 1/D\Delta f$. This time interval is shorter than that for the conventional methods by a factor of 128.

#### *3.5.3 Block diagram and description*

Figure 6 shows the block diagram representation of the comparator unit. Two printed circuit boards (B1 and B2) are used to accommodate the large number of integrated circuits and the relatively large inductors comprising the low-pass filter. The inputs to the comparator unit, during normal operation, are two 4.94-MHz TTL signals derived from oscillators A and B. For routine maintenance and oscillator calibration, a third input signal, also TTL compatible, from an external frequency standard is also connected to the comparator unit.

The input signal frequency to the 14-stage N counter is A/D + Δf. The output of this counter is the gate pulse ($T_G$) for the C counter. The 22-stage C counter is preset to a value $C_0$ dependent on the operating mode. The clock input signal is from oscillator A, which counts down the C counter during the gate period $T_G$. If oscillators A and B are identical in frequency, the C counter will count down from $C_0$ to zero. The value of the C counter is determined by monitoring the Q signals of each counter stage. The Q signals are gated together to detect values of ΔC which correspond to frequency differences. The count can be either positive or negative, depending on whether the counter went through zero or not. For normal operation, only the magnitude of the ΔC count is significant. However, during frequency adjustment of the two oscillators, the sign of the count is also important because the frequency of the oscillator being adjusted is compared to an external frequency standard and the sign of the frequency error must be known for proper adjustment. In the frequency check modes, instead of generating alarms for excessive frequency differences, the comparison circuit uses the four LEDs on the front panel to indicate the magnitude and direction of frequency deviation of the oscillator.

#### *3.5.4 Other design considerations*

Except during oscillator adjustment and routine maintenance, the comparator unit operates in the normal mode. In this mode, only a

frequency alarm and cut-reference signal and alarm are generated when the frequency difference between MCSS oscillators is 16.6 parts in $10^8$ or 39.3 parts in $10^8$, respectively. The frequency comparison period in this mode of operation is only 78 milliseconds. It is possible for erroneous counts to occur because of plant battery hits or other office transients. To guard against false alarms occurring due to these erroneous counts, four-stage shift registers are used to output the frequency alarm and the cut-reference signal. To obtain an output from these shift registers, four consecutive readings exceeding the appropriate $\Delta C$ limit must occur. This, of course, delays the alarm signals by 312 milliseconds, but is quite acceptable in this application and does eliminate false alarms or, more seriously, loss of sync to an entire station because of an erroneous cut-reference signal.

### *3.6 MCSS dc power*

The MCSS dc power is provided by regulated dc-to-dc converters (−24 to +5 volts dc), which are current limited, over-voltage protected, and low-voltage alarmed. The two 4.94-MHz oscillators are powered by separate 5-volt dc converters. These same converters also individually power the divider/switch circuits. The comparator and control unit are dual fed from the two converters using diodes. These procedures are used in the MCSS to guard against reference signal loss due to failure of any single power unit.

### *3.7 Alarm unit*

The alarm unit in the MCSS provides visual and audible alarms of low power and other trouble conditions in the MCSS. It generates remote indications via telemetry systems to the alarm center, as well as local station alarms. Standard ACO features are available that can be activated at any miscellaneous key, jack, and lamp panel in the support bay or at any TR-bay display panel.

## IV. SUMMARY

The paper has described the important features of the AR6A MCSS. The MCSS provides a precise reference signal at 308.9 kHz to which the microwave generator and shift oscillator in a repeater bay are locked. The MCSS is settable to one part in $10^8$ and is stable to within five parts in $10^{10}$ per day. Extensive monitoring of performance and the provision of a standby reference signal ensure a high degree of reliability to the system.

## V. ACKNOWLEDGMENTS

The authors would like to acknowledge the efforts of M. W. Zuidervliet, who designed the 4.94-MHz oscillator, and N. B. Rowe and

J. R. Scoville, who contributed greatly in the MCSS design and evaluation.

## REFERENCE

1. J. M. Kiker, Jr. and J. Gonda, "The AR6A Single-Sideband Microwave Radio System: Microwave Carrier Supply," B.S.T.J., this issue.

## AUTHORS

**J. Morgan Kiker, Jr.,** B.S.E.E., 1962, University of California at Berkeley; M.E.E., 1964, New York University; Bell Laboratories, 1962—. Mr. Kiker began work on high-resolution radar systems. Since 1971 he has supervised work on ultralinear IF amplifiers and low-noise carrier and oscillator supplies for the AR6A Radio System. He is now working on digital radio systems. Member, Tau Beta Pi, Eta Kappa Nu.

**Siegfried B. Pirkau,** A.A.S., 1969, New Hampshire Vocational Technical College (NHVTC); Bell Laboratories, 1969—. Mr. Pirkau's first assignment was work involved with computer-simulated *Picturephone** coding. In 1971 he began work on single-sideband radio in the Repeater Evaluation Group, where he became involved with the design and construction of special test equipment. In 1980 he joined the Digital Radio Laboratory doing DR6 and DR11 development work.

* Registered service mark of AT&T.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Microwave Carrier Supply

By J. GONDA* and J. M. KIKER, Jr.*

(Manuscript received February 14, 1983)

In the AR6A repeater bay, microwave carrier power for both the receiver and the transmitter is derived from a single voltage-controlled crystal oscillator. For long-term stability, this oscillator is phase locked to an external-reference frequency, available at each radio station. From the output signal of this oscillator, an active frequency-multiplier chain generates about +21 dBm power in the 6-GHz band. Part of this power serves as the local oscillator signal for the transmitter directly. The other part is shifted in frequency, to produce the local oscillator signal for the receiver.

## I. INTRODUCTION

Two microwave carriers are needed in an AR6A† repeater bay for up and down conversion. These carriers have to be spaced 252 MHz apart and must have low noise, low jitter, and exceptional frequency stability.

In the microwave carrier supply, both carriers are derived from a single Voltage-Controlled Crystal Oscillator (VCXO)‡. An active frequency-multiplier chain generates the 6-GHz carrier for the up con-

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

‡ Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

verter directly. Part of this 6-GHz power is split off and shifted 252 MHz in frequency, to be used in the receiver down converter.

For frequency stability, the VCXO is phase locked to an external reference, available at each radio station. Steps are taken to ensure adequate frequency stability even during temporary loss of the external-reference signal.

Along with the detailed performance objectives, a general circuit description of the microwave generator is given in this paper, with the key circuits described in somewhat more detail. Typical test results and a brief description of the method used to measure FM noise close to the carrier concludes the paper.

## II. PERFORMANCE OBJECTIVES

Some of the performance objectives for the microwave carrier source designed for AR6A are similar to those required in FM radio systems; others are unique. The performance objectives for this new carrier source are enumerated below with discussion of those items dictated by application in the AR6A Radio System.

### *2.1 Frequencies*

The TR-bay microwave carrier supply has to provide two carriers separated by 252 MHz, in the 6-GHz band at any of the 10 Local Oscillator (LO) frequencies given in the AR6A frequency plan.[1]

### *2.2 Power output*

The available microwave carrier power requirement is +17 dBm for the transmitter and +11 dBm for the receiver at the shifted LO frequency.

### *2.3 Frequency stability*

The microwave carriers must be stable in frequency to about two parts in $10^7$ per year so that the equalizer pilots at IF will be located accurately with respect to the narrowband (12 kHz) crystal pick-off filters in each AR6A receiver. In addition, the accumulated frequency error from each repeater in a terminal section must be limited to a value that can be corrected by the multimastergroup terminal receiver. This stability objective of two parts in $10^7$ is about two orders of magnitude more accurate than for carriers for FM radio application. To meet the strict frequency-stability requirement, the microwave carriers are phase locked to an external-reference frequency,[2] having the required frequency accuracy and stability.

With conventional phase-lock control, if lock is lost the error-correction signal to the oscillator will go to zero and the frequency of

oscillation will shift to a free-running condition that will depend on temperature and oscillator aging since last tuning. In many cases this shift would be sufficient to render the radio channel unusable. To prevent this almost certain channel loss, an additional objective is specified for the phase-locked circuitry that on loss of lock the oscillator control voltage be maintained to keep the 6-GHz output frequency within ±200 Hz of the value when loss of lock occurred. Under these conditions the microwave generator free runs and drifts with temperature and time from this frequency.

Large temperature changes in a nonair-conditioned repeater station during a time that the microwave generator is free running could cause excessive frequency drift. Such large changes may occur from day-to-night temperature variations. To minimize these effects, a final objective on frequency stability is that the free-running frequency shall not change by more than 0.5 ppm over any 30-degrees F ambient temperature change within the limits of 40 to 140 degrees F. This objective is intended to provide acceptable frequency stability during the time the source is free running over a period of days if technicians cannot be immediately dispatched to clear the loss-of-lock condition.

The free-running frequency stability, together with the phase-locked error voltage memory, ensures continuous service on the AR6A route even in the unlikely event of the loss of the reference signal from the Microwave Carrier Synchronization Supply (MCSS), when every microwave source in the station is unlocked.

### *2.4 Phase noise*

The phase noise requirements for a microwave source used on an AM radio system are more stringent than for use on FM systems. For an AM system the microwave source phase-noise spectrum close to the carrier is dominant in the contribution of voice-circuit noise.

To demonstrate this fact, consider a baseband signal $a(t)$ that is up converted in frequency by a local oscillator with frequency $\omega_c$ and phase noise $\theta(t)$. If transmitted single sideband, the signal $s(t)$ can be represented as

$$s(t) = a(t)\cos[\omega_c t + \theta(t)] \pm \hat{a}(t)\sin[\omega_c t + \theta(t)], \tag{1}$$

where $\hat{a}(t)$ is the Hilbert transform of $a(t)$. The sign of the second term is negative if the upper sideband is selected and positive if lower sideband is selected. To evaluate the noise contribution from one source, assume the demodulation is done by a noise-free source. The applicable output terms, $r(t)$, at baseband are given by

$$r(t) = a(t)\cos[\theta(t)] \pm \hat{a}(t)\sin[\theta(t)]. \tag{2}$$

For the LO source $|\theta(t)| \ll 1$, so that $\cos[\theta(t)] \approx 1$ and $\sin[\theta(t)] \approx$

$\theta(t)$. In this case eq. (2) becomes

$$r(t) \approx a(t) \pm \hat{a}(t)\theta(t). \tag{3}$$

The output consists of the desired signal $a(t)$ plus a noise term given by $\hat{a}(t)\theta(t)$. Let $A(f)$ denote the power spectral density of the desired signal and $S_a(f)$ the spectral density of the phase noise. The spectral density of the Hilbert transform of a signal is equal to the spectral density of the signal itself; therefore, the spectral density, $N(f)$, of the noise term in eq. (3) is

$$N(f) = A(f) * S_a(f), \tag{4}$$

where $*$ denotes convolution.

The spectral density of the 6000 voice-circuit channel is noiselike and essentially flat over the 30-MHz frequency band. Since the phase spectral density of crystal oscillators as used in the microwave source increases as $1/f^3$ close to the carrier, the close-in noise components contribute the most to the convolution.

Since the phase jitter is present on the carrier recovery pilot, the Multimastergroup Translator for Radio (MMGT-R)[3] tracking receiver at the end of a terminal section will remove some of this noise. The remaining phase noise will not be reduced further by the following section since new carrier recovery pilots are inserted at the beginning of the next span. The total phase noise will accumulate as each terminal section is traversed. Taking into account the phase-noise reduction that can be obtained from the terminal, the phase-noise spectral density objective of the microwave generator necessary to meet its 4000-mile noise allocation was determined. This objective is shown as the upper curve in Fig. 1.

Though not part of the dBrnc0 noise-tree allocation, the phase noise of the microwave sources will affect voice-circuit phase-jitter objectives, which are important when the channels are used for voiceband data transmission.

## III. MICROWAVE-GENERATOR CIRCUIT DESCRIPTION

Figure 2 shows the block diagram of the microwave carrier supply for an AR6A repeater. A 21.5-dBm, 6-GHz signal is generated using a 1-GHz microwave generator and a 6X frequency multiplier. The 6-GHz power is split in the carrier distribution network, with +17 dBm going directly to the transmitter modulator; the remainder drives the shift modulator to generate the carrier for the receiver modulator. The shift oscillator frequency is 252 MHz.

Both the 1-GHz generator and the shift oscillator are phase-locked to a central 308.9-kHz reference signal through their respective frequency control units. In case of a reference-signal failure, the frequency

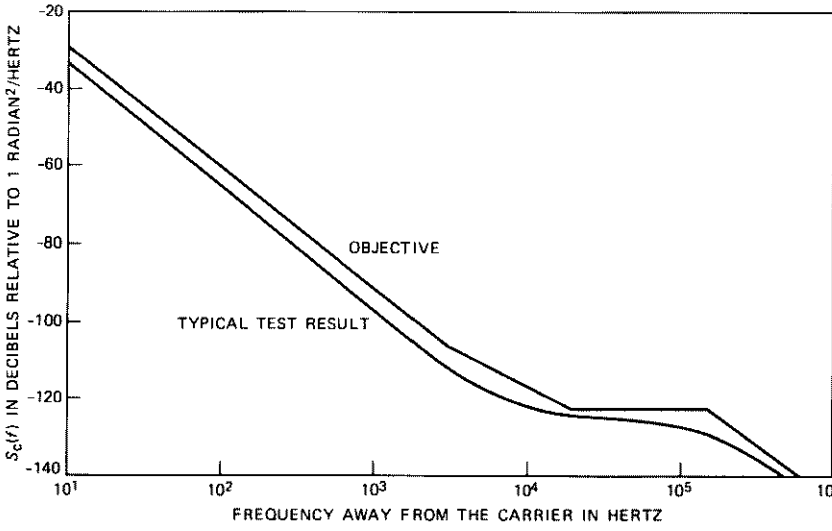



Fig. 1—Phase-noise objective and typical test result for the microwave generator.

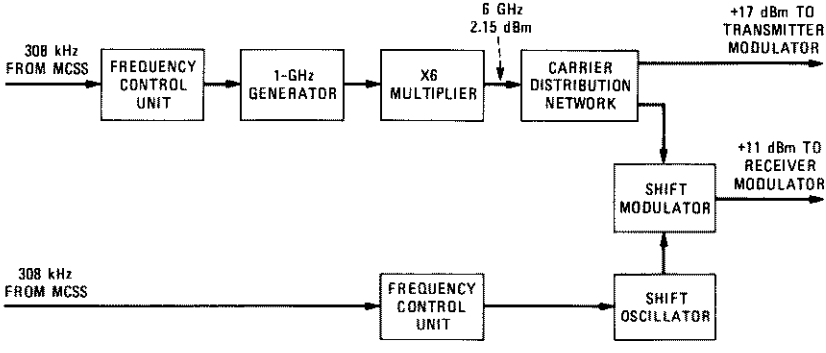



Fig. 2—Microwave carrier supply.

control units will put the oscillators on memory in order to continue to supply the microwave carrier and keep their frequencies within the required accuracy.

### *3.1 1-GHz generator*

At the beginning of the AR6A development, several candidates had been investigated for the 1-GHz generator, including general trade products. For stability, performance, and cost effectiveness the 1-GHz generator developed earlier for the TH-3 Radio System[4] was adopted and improved for AR6A radio use. Figure 3 shows the block diagram of the 1-GHz generator.

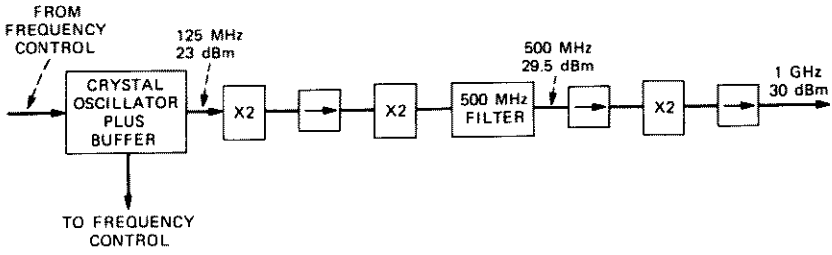



Fig. 3—1-GHz generator.

The generator is a straight multiplier chain of three active frequency doublers, driven by a crystal oscillator in the 125-MHz frequency band. Three isolators are used between successive stages to stabilize the doublers and facilitate tuning. A high Q INVAR cavity is used at 500 MHz to strip off multiplied white noise far from the carrier.

### *3.1.1 Oscillator and buffer amplifier*

The crystal oscillator is built with a Western Electric NPN transistor in the common base configuration. The third-overtone crystal operates in series resonance, and it is connected in the feedback loop between the emitter and collector of the transistor. A high Q varactor diode is connected in series with the crystal to provide voltage control of the oscillator frequency.

To meet the AR6A requirement for low noise close to the carrier, it was important to keep the $1/f$ noise of the transistor low by keeping the collector current low, to use a crystal with high unloaded Q, and to design the oscillator circuit (including the varactor) such that the Q of the total oscillator would approach the Q of the crystal itself. To keep intermodulation in the transistor at minimum, and to keep the conversion of $1/f$ noise into the 125-MHz frequency band low, limiter diodes are used to control the amplitude of oscillation, instead of relying on the nonlinearities of the transistor itself.

For long-term stability the crystal current is kept below 1 mA, which gives about +7 dBm power at the oscillator output.

Under normal operating conditions, the varactor voltage is regulated by the Phase-Locked Loop (PLL) in the frequency control unit to keep the crystal-oscillator frequency locked to a high-order harmonic (404th to 429th, depending on channel frequency) of the 308.9-kHz reference signal. In case of loss of lock (because of an MCSS failure, for example), the frequency control unit goes on memory, and the varactor voltage in the crystal oscillator is kept constant at the last (locked) value. In this free-running mode the crystal oscillator has to stay within 1 ppm of its nominal frequency until the unit is repaired (two days at most). To meet this requirement in stations without air

conditioning, the crystal is placed in an oven. The turnover temperature of the crystal is 65 degrees C.

A two-stage wideband buffer amplifier follows the crystal oscillator. A shunt-mounted PIN diode between the two stages of the buffer can be used to adjust levels for the entire microwave generator. The output level range of the buffer is 13 to 25 dBm.

A sample of about 12 dBm is decoupled from the buffer output to the frequency control unit for phase locking.

### *3.1.2 Frequency doublers*

The three active frequency doublers in the 1-GHz generator are built with Western Electric overlay transistors. Besides doubling their respective input frequencies, these stages have several decibels of gain each. Figure 4 shows a common simplified schematic of the doubler circuits.

Due to the ac short in the collector circuit at the input frequency, *f*, these stages have substantial current gain at that frequency. The increased collector current pumps the nonlinear capacitance of the collector-base junction, producing harmonic voltage components between collector and base. A second series resonant circuit between base and ground provides a current path for the generated second harmonic, bypassing both the input matching network and the base emitter junction of the transistor. The two matching networks transform the input and output impedance of the doubler circuit to 50 ohms.

Principally, all three doubler stages are as described. Only the impedance levels vary with frequency and the physical form of the circuit elements. In the first doubler from 125 to 250 MHz most elements are lumped; the third stage, however, from 500 MHz to 1 GHz is built entirely of distributed elements.

Nominal output power levels for the three doublers are: 27 dBm at 250 MHz, 30 dBm at 500 MHz, and 31 dBm at 1 GHz. With interstage losses taken into account, this corresponds to conversion gains of 4, 3, and 2 dB, respectively.

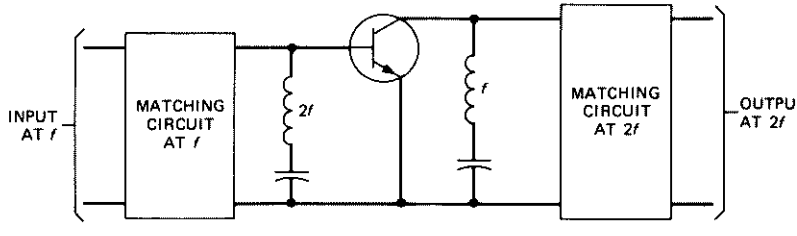



Fig. 4—Transistor doubler.

### 3.1.3 Isolators

All three isolators in the multiplier chain are of the lumped element design.[5] They have been developed at Bell Laboratories, Allentown, Pennsylvania. Their insertion loss is typically 0.5 dB forward, and greater than 20 dB in the reverse direction.

### 3.1.4 Noise-suppression filter

Although the excess noise of the frequency doublers is minimal, they increase the FM noise originating in the oscillator and buffer by a factor of 20 log $n$ along the multiplier chain. Far from the carrier, therefore, the multiplied noise could exceed system requirements.

To prevent this, a noise-suppression filter is inserted into the multiplier chain at 500 MHz.[6] The filter is a coaxial reentrant cavity with an unloaded Q of about 3000. Its 3-dB bandwidth is about 300 kHz, with about 2-dB insertion loss. For frequency stability the cavity is extruded from INVAR.

## 3.2 The 1- to 6-GHz multiplier

The 1- to 6-GHz multiplier is a broadband 6X frequency multiplier,[7] driven by the 1-GHz generator at a 29.5-dBm level. It delivers 21.5-dBm, 6-GHz power to the carrier distribution network. All unwanted harmonics are kept at least 80 dB below the carrier.

The multiplier circuit is built on alumina microstrip. The photograph in Fig. 5 shows the circuit details. Figure 6 gives the approximate lumped element equivalent circuit.

A self-biased commercial step recovery diode with zero bias capacitance of 3.3 pF is used for the nonlinear element. The diode is mounted into a hole in the microstrip circuit. One side is soldered to the ground plane on the back, the other side is thermocompression bonded to the microstrip pattern using a 20-mil wide gold ribbon.

Both the input and output filters are three-resonator bandpass, 0.1-dB Tchebyscheff type, designed to transform the diode impedance to 50 ohms in their respective passbands. The input filter is realized using quarter-wavelength resonators in an interdigital configuration. The output filter consists of side-coupled, half-wavelength resonators. The diode itself is part of the first resonator in the output filter. The radial lines at the input side of the diode are used to suppress unwanted harmonics; at the same time they act together as a capacitor for the pulse-forming circuit.

In order to meet the very strict spectral purity requirement for the AR6A System at the up and down converter inputs, as discussed earlier, all the unwanted harmonics had to be suppressed to at least 80 dB below the carrier at the 6X multiplier output. This requirement was met by adding a seven-resonator interdigital filter to the output

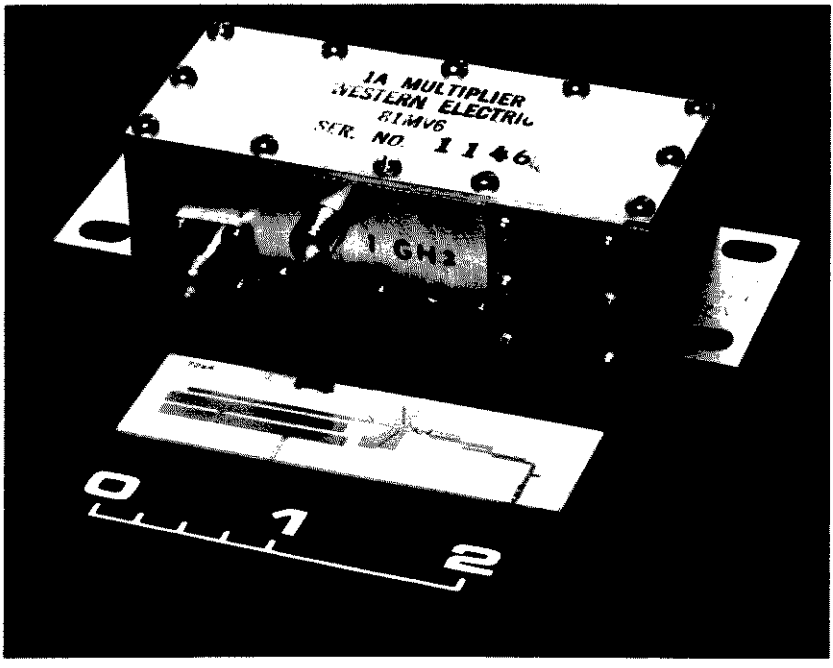



Fig. 5—Photograph of the 6X multiplier.

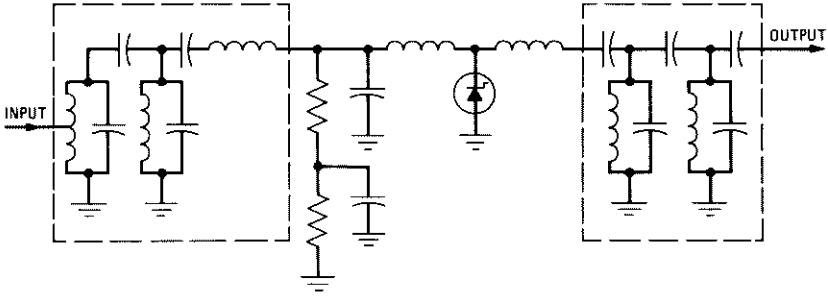



Fig. 6—Lumped element equivalent circuit of the 6X multiplier.

of the 6X multiplier. This filter is built into the back of the multiplier housing, and it has about 0.5-dB passband insertion loss. The overall size of the 6X multiplier, excluding the SMA connectors, is 3/4 by 1/4 by 3 inches. The typical swept response of the multiplier is shown in Fig. 7. One code covers the total AR6A bandwidth.

### *3.3 Carrier distribution network*

As indicated in Fig. 2, the carrier distribution network is used to split the output signal of the microwave generator between the trans-

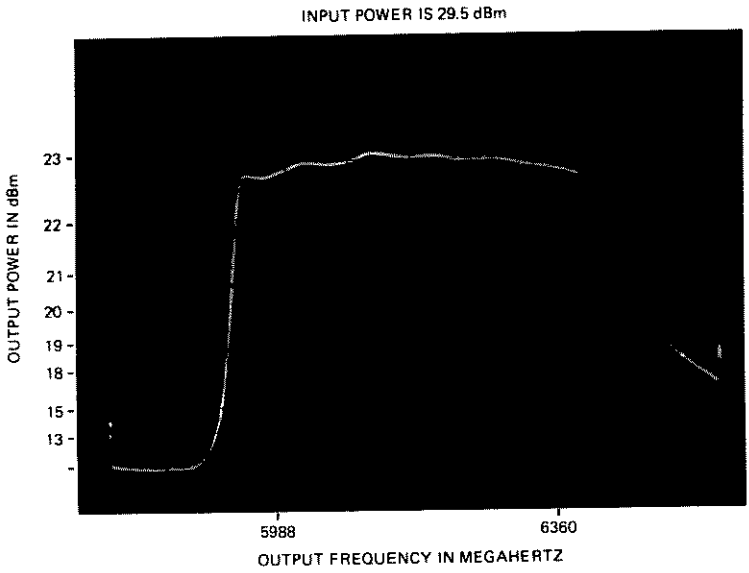



Fig. 7—Typical swept response of the 6X multiplier.

mitter and the shift modulator in a repeater station. The detailed description of this circuit can be found elsewhere.[8]

### *3.4 Shift modulator and shift oscillator*

The purpose of the shift modulator is to shift the frequency of the microwave generator by ±252 MHz and thus provide the local oscillator signal for the receiver modulator.

The shift oscillator generates the 252-MHz signal for the shift modulator using a 126-MHz crystal oscillator and a frequency-doubler circuit. As shown in Fig. 2, the shift oscillator is phase locked to the MCSS signal.

Reference 9 gives a more detailed description of both the shift modulator and the shift oscillator.

## IV. FREQUENCY CONTROL UNIT

The microwave generator and shift oscillator are phase locked to a harmonic of the microwave carrier synchronization supply frequency of 308.9 kHz[†] by means of two frequency control units. These units contain a comb generator to produce the desired harmonic of the 308.9-kHz signal; a phase-locked loop to synchronize the microwave generator or shift oscillator to the proper 308.9-kHz harmonic; a memory circuit that operates in conjunction with the phase-locked

[†] The actual frequency is 308.8735416 kHz.

loop to provide the control voltage to the oscillators; and alarm circuitry to provide bay alarms on loss of lock and memory end of range. The overall description of operation and some details of these individual circuits are given in the following sections.

### *4.1 Frequency control unit circuit description*

Figure 8 shows the block diagram of the frequency control unit. The 308.9-kHz reference signal from the MCSS is applied to the comb generator. The proper harmonic for locking the crystal oscillator is produced by the comb generator and applied to one input of the phase detector in the phase-locked circuit. A signal sample of the crystal oscillator to be controlled is applied to the other phase-detector input of the phase-locked loop. The output of the phase-locked loop is an analog error signal proportional to the phase difference between the reference and controlled oscillators. This error signal is applied as one input to the varactor of the controlled oscillator, with the other inputs from the memory circuit and a bias voltage source. The purpose of the fixed bias is to place operation of the varactor diode at the suitable point on its voltage-versus-capacitance characteristic and to offset the midrange digital error-correction voltage to zero. On initial alignment no digital error-voltage component (after offset) is applied to the varactor diode. In time, due to aging and temperature changes, the analog error voltage will change to maintain lock, and when the change reaches a certain value, the memory circuit will add a digital error-voltage component of a value equal to the change. In order to maintain the total error signal the same, the analog component will return back to its original value. As further changes occur, the memory circuit will continue to add or subtract steps as the analog signal either increases

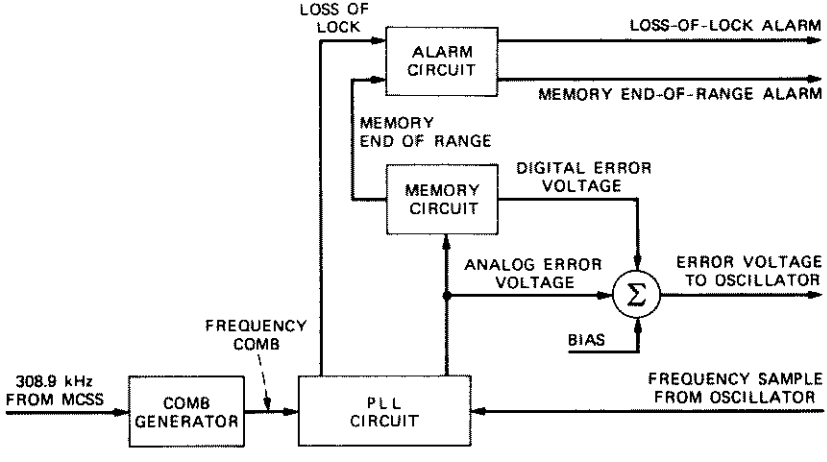



Fig. 8—Block diagram of frequency control unit.

or decreases. In this way the memory circuit provides the major part of the error signal with the phase-locked loop error signal varying an amount equal to the memory change threshold. If the reference signal from the MCSS is lost, the analog loop is opened and the analog error signal goes to zero. The memory continues to supply the digital error-signal component. In this way the generator is free running very near its last correct frequency. The maximum error is plus or minus the memory threshold value which is about 3.2 Hz at 125 MHz, which translates to a maximum error of about 155 Hz at 6 GHz.

There are 1024 steps of memory available with step 512 initially set when the oscillator is aligned. This provides for 512 steps in either direction to compensate for increase or decrease in frequency. This corresponds to ±1600 Hz at 125 MHz or about ±1.28 parts in $10^5$.

The alarm circuit monitors the memory step location and generates a bay alarm when 88 percent of the available range has been used. This is a warning to the operating personnel that manual retuning of the oscillator should be done. In addition, the alarm circuitry provides an alarm if phase lock between the reference signal and the controlled oscillator is lost.

### *4.2 Comb generator*

The comb generator[10] consists of three basic parts: (1) a sine-wave-to-square-wave converter, (2) a narrow pulse generator, and (3) a gated oscillator tunable in frequency and with output always beginning at the same phase.

The input 308.9-kHz sine-wave reference signal is converted into a TTL-level square wave using a high gain amplifier followed by a comparator. The comparator output is used to trigger a monostable multivibrator that generates 0.5-microsecond pulses at the reference-frequency rate. This narrow pulse is used to gate an oscillator that always starts at the same phase. The frequency of oscillation can be manually tuned. Let $f_r$ denote the frequency of the reference signal, A and $f_{osc}$ the amplitude and frequency of the gated oscillator, and $T_w$ the width of the gating pulse. A Fourier analysis shows that the output spectrum, $S_o(f)$, of the gated oscillator is given by

$$S_o(f) = Af_rT_w \sum_{n=-\infty}^{\infty} \frac{\sin \pi(f_{osc} - nf_r)T_w}{\pi(f_{osc} - nf_r)T_w} \delta(f - nf_r). \qquad (5)$$

The output is seen to consist of harmonics of the reference signal with amplitudes proportional to the sin x/x function. By tuning $f_{osc}$ at or near the desired harmonic, say $n_df_R$, the output of the comb generator will provide maximum output to the PLL at that frequency. Tuning of the oscillator is easily accomplished by observing the beat signal at the output of the PLL phase detector and tuning the oscillator

for maximum amplitude. The other harmonics that are present at the phase detector are not important since a given crystal oscillator can never be pulled far enough to lock to the wrong harmonic.

### *4.3 Phase-lock loop*

Figure 9 shows a block diagram of the phase-locked loop circuit. A sample from the controlled oscillator is buffered and amplified by a common base transistor amplifier and applied to one input of the phase detector. The frequency comb is the other input to this detector. The phase detector has a sinusoidal characteristic and is implemented using a double-balanced mixer. In normal operation, a small region around the zero crossing of this characteristic is used to approximate a linear phase detector. The output of the phase detector is amplified to provide the desired PLL loop gain and applied through a Complementary Metal-Oxide Semiconductor (CMOS) Single Pole Double Throw (SPDT) switch to the loop filter and the memory circuit. The output of this filter is the analog portion of the error voltage applied to the varactor.

The CMOS switch is under control of the loss-of-lock detector. As long as the loop is locked the control voltage holds the switch in the through-path position. A prolonged loss of lock will cause the switch to open, which reduces the analog portion of the error signal to zero. At this point only the memory circuit is providing error correction. The reset control forces the switch closed initially so that phase lock can be achieved.

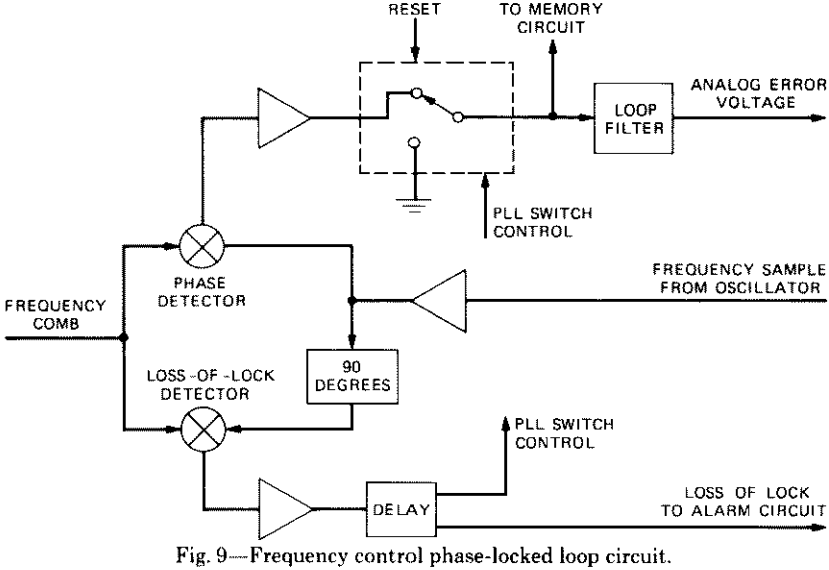



Fig. 9—Frequency control phase-locked loop circuit.

The loss-of-lock detector uses a second double-balanced mixer but with the frequency sample input shifted 90 degrees. When in phase lock, the loop phase detector is operating around the 0-degree point on the characteristic and the lock detector will be operating about the 90-degrees point providing a dc voltage output. This output is amplified and applied to a delay circuit. Loss of lock must occur for the entire period of this delay before the PLL switch is opened and a loss-of-lock alarm given. This delay is necessary to keep the loop closed long enough to reacquire lock lost because of a hit or by a switch in the MCSS from one reference source to another.

Dynamically, the PLL is a second-order type. Since its purpose is for long-term frequency stability, bandwidth is relatively narrow. The primary consideration for bandwidth is that it be adequate to reacquire lock when the MCSS reference switches and that the unit be fairly easy to lock initially when manual tuning of the controlled oscillator is required to bring the frequency within locking range. The pertinent frequency control parameters of the loop are as follows:

1. Natural frequency $\Omega_n$ is 51 rad/s.
2. Damping factor is 0.7.
3. Overall loop gain is $2.7 \times 10^3$ rad/s.

### *4.4 Memory circuit*

Figure 10 shows a block diagram of the memory circuit. The digital error voltage is derived from a 10-bit Digital-to-Analog (D/A) converter that is driven by a 10-stage up/down counter. On initial alignment the counter is preset to a count of 512, the midpoint of the counter. The output of the D/A converter for this count is offset by the bias source so that the effective digital error signal is zero.

The analog error-voltage sample from the PLL is heavily filtered and fed to two comparators. One comparator is set to switch when the

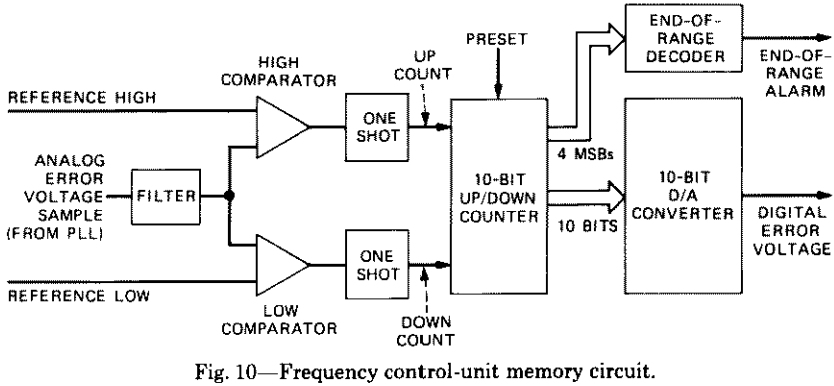



Fig. 10—Frequency control-unit memory circuit.

error voltage indicates a high-frequency correction, the other for a low-frequency correction. The references are set for thresholds corresponding to ±3.2 Hz at 125 MHz. The operation of the comparator activates a one shot, which generates one up-count or down-count pulse to the counter. The scaling of the D/A converter output is such that a 1-bit step will produce a change in output corresponding to a 3.2-Hz correction.

The four most significant bits of the counter are examined for an all 1's or all 0's condition. This indicates that 87.5 percent of the counter's range has been used in the upward or downward direction, respectively. This condition is decoded and output to the alarm circuit as an end-of-range alarm.

### *4.5 Alarm circuit*

The alarm circuit interfaces the frequency control unit to the TR-bay alarm circuits. The end-of-range indication from the memory circuit is latched before it is sent to the TR bay. The loss-of-lock signal is sent directly but buffered by TTL driver gates.

## V. PERFORMANCE

### *5.1 Power output variation with temperature*

Figure 11 shows the output power of the microwave generator, measured at the output of the sextupler, as a function of the ambient temperature. The total variation is about 1.6 dB. This is satisfactory for the AR6A System even in a nonair-conditioned environment.

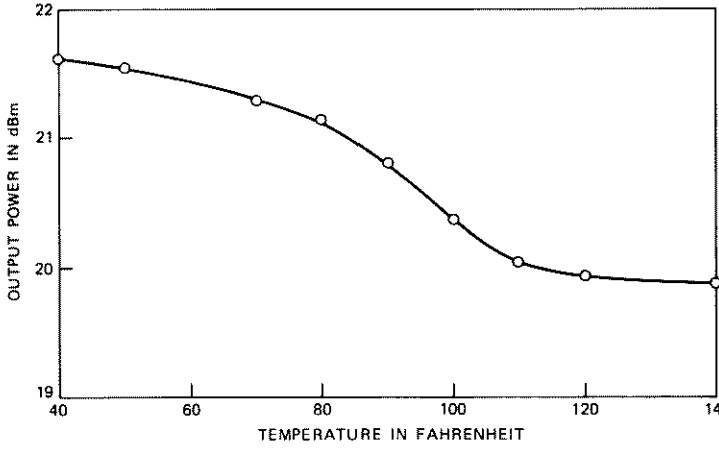



Fig. 11—Output power of the microwave generator as a function of ambient temperature.

### *5.2 Free-running frequency as a function of temperature*

Figure 12 plots the free-running frequency of the microwave generator as a function of the ambient temperature. This characteristic is important only when the oscillator is not locked to the MCSS (due to maintenance or MCSS failure). The variation is within the maximally allowed 0.5 ppm for any 30-degree F change in temperature.

### *5.3 Phase-noise performance*

#### *5.3.1 Measurement method*

There are several known methods (Refs. 11–13) to measure phase noise close to the carrier, each of them having its advantages and disadvantages. We have used all of the three methods referred to above during development. Figure 13 shows the block diagram of the phase-noise test set presently used in production. Two microwave generators are phase locked to each other by a narrowband phase-locked loop. The output signal of both are multiplied in frequency up into the 6-GHz band and then fed into a double-balanced mixer in phase quadrature. The output of the mixer goes into a spectrum analyzer after being amplified in a low noise amplifier. The output of the spectrum analyzer is plotted on an X-Y plotter directly as the noise spectral density, $S_\phi$(dB) versus frequency separation from the carrier.

#### *5.3.2 Measured phase noise*

Figure 1 shows the measured phase-noise spectral density of the microwave generator. The test results are typically 5 dB better than the requirement.

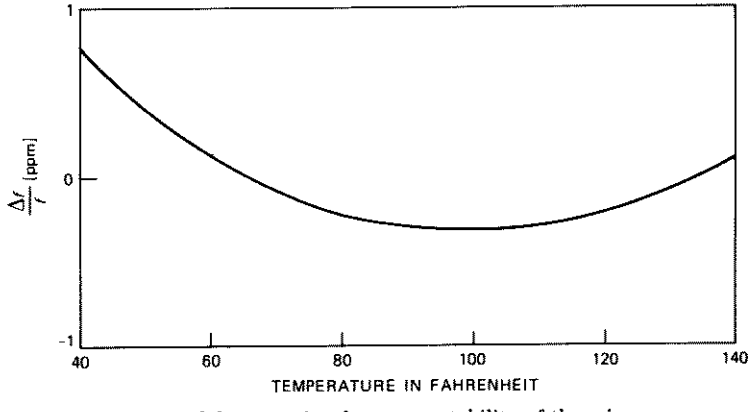



Fig. 12—The measured free-running frequency stability of the microwave generator as a function of the ambient temperature.

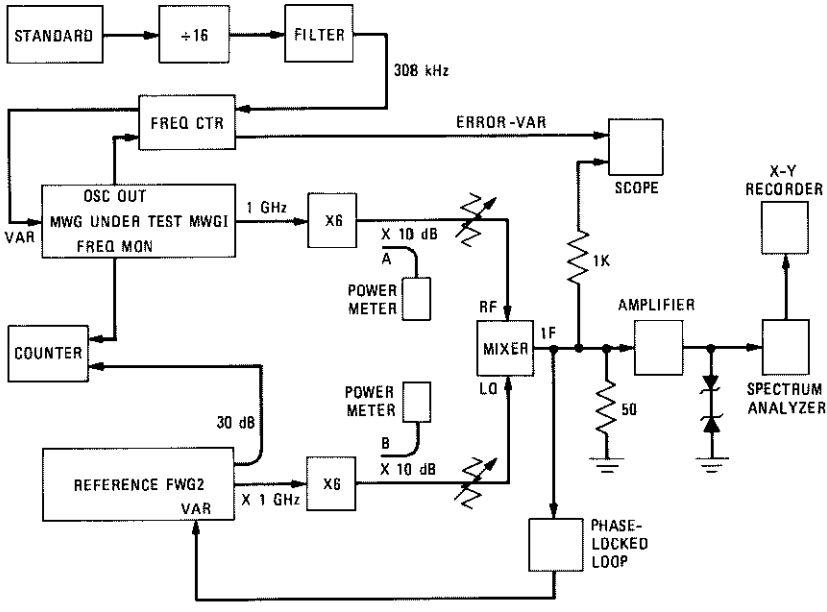



Fig. 13—Block diagram of the test set used to measure noise of the microwave generator.

### *5.4 Phase jitter*

The measured phase jitter of the microwave generator is 10 to 25 degrees peak to peak, using the filter recommended in Ref. 14, and about 0.5 to 0.7 degrees peak to peak with the jitter reduction provided by the MMGT-R and the weighting of the same filter.

### *5.5 Spurious tones*

There are low-level tones in the output spectrum of the microwave generator when the generator is operating in the bay. These tones are harmonics of 60 Hz, coming from the battery power plant, which is charged continuously, and 20- and 40-kHz tones generated by dc-to-dc converters in the power supplies in the bay. All of these tones are at least 40 dB below the carrier at the output of the microwave generator. Also present are 308.9-kHz sidebands from the phase-locked reference. These are better than 70 dB below the desired output.

## VI. SUMMARY

A low-noise, phase-locked microwave carrier supply has been developed for the AR6A Radio System with exceptional frequency stability. Steps were taken to ensure system survival even in the extreme case of reference-signal loss.

The architecture and key circuits of the carrier supply have been described, along with performance objectives and typical test results.

## VII. ACKNOWLEDGMENTS

The development of the microwave carrier supply was the combined accomplishment of many individuals. Among them the authors wish to acknowledge the contributions of T. J. Case, H. Goldstein, and J. R. Scoville.

## REFERENCES

1. J. Gammie, J. P. Moffatt, R. H. Moseley, and W. A. Robinson, "The AR6A Single-Sideband Microwave Radio System: System Design and Performance," B.S.T.J., this issue.
2. J. M. Kiker, Jr. and S. B. Pirkau, "The AR6A Single-Sideband Microwave Radio System: Frequency Control," B.S.T.J., this issue.
3. A. Dubois, D. N. Ritchie, and F. M. Smith, "The AR6A Single-Sideband Microwave Radio System: Terminal Multiplex Equipment," B.S.T.J., this issue.
4. H. R. Bedell, R. W. Judkins, and R. L. Lahlum, "Microwave Generator," B.S.T.J., *50*, No. 7 (September 1971), pp. 2205–21.
5. R. H. Knerr, "Series and Shunt Tuned Lumped Element Circulators," B.S.T.J., *59*, No. 5 (May-June 1980), pp. 723–30.
6. T. A. Abele and D. J. Leonard, "Microwave Generator," B.S.T.J., *47*, No. 7 (September 1968), pp. 1301–22.
7. J. Gonda, "Broadband Frequency Multiplication by Multitransition Operation of Step Recovery Diode," U. S. Patent 4-360-867, issued November 23, 1982.
8. N. R. Dietrich, "Microwave Integrated Circuits," B.S.T.J., *50*, No. 7 (September 1971), pp. 2175–94.
9. O. Giust, "Modulators," B.S.T.J., *50*, No. 7 (September 1971), pp. 2155–73.
10. M. Emshwiller, "Synchronizing Circuit," U. S. Patent 4-039-968, issued August 2, 1977.
11. J. R. Ashley, C. B. Searles, and F. N. Palka, "The Measurement of Oscillator Noise at Microwave Frequencies," IEEE Trans. Microw. Theory and Tech., *16*, No. 9 (September 1968), pp. 753–60.
12. J. H. Shoaf, D. Halford, and A. S. Risley, "Frequency Stability Specification and Measurement: High Frequency and Microwave Signals," National Bureau of Standards, Technical Note 632, January 1973.
13. L. Peregrino, and D. W. Ricci, "Phase Noise Measurement Using a High Resolution Counter with On-Line Date Processing," 30th Annual Frequency Control Symposium, June 2–4, 1976.
14. "Transmission Parameters Affecting Voiceband Data Transmission Measuring Techniques," Bell System Technical References, PUB 41009, May 1975.

## AUTHORS

**Joseph Gonda,** Dipl. Eng., EE, 1957, Dr. Techn., EE, 1962, Technical University of Budapest; Research Institute for Telecommunications, Hungary, 1957–1969; Bell Laboratories, 1969—. Mr. Gonda has been working on various microwave active circuits for microwave radio systems. His present responsibility is the development of dielectric resonator oscillators for digital radio applications.

**J. Morgan Kiker, Jr.,** B.S.E.E., 1962, University of California at Berkeley; M.E.E., 1964, New York University, Bell Laboratories, 1962—. Mr. Kiker began work on high-resolution radar systems. Since 1971 he has supervised work on ultralinear IF amplifiers and low-noise carrier and oscillator supplies for the AR6A Radio System. He is now working on digital radio systems. Member, Tau Beta Pi, Eta Kappa Nu.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Equalization for Multipath Fading

By N. O. BURGESS,* R. C. MacLEAN,* G. J. MANDEVILLE,* D. I. McLEAN,* M. E. SANDS,* and R. P. SNICER*

(Manuscript received February 25, 1983)

Multipath fading can introduce severe amplitude distortion and level changes in a radio channel. These must be dynamically equalized to meet the toll transmission requirements of AR6A. This article describes equalizer circuitry at intermediate frequencies, which continuously senses the level as a function of frequency in the transmission band and dynamically corrects the effects of multipath fading.

### I. FADING CHARACTERISTICS

In line-of-sight microwave radio transmission, the broadband radio channels (20 to 30 MHz) can exhibit the phenomena of both selective and nonselective fading. During nonselective fading, the signal power across the channel remains constant with frequency and simply decreases in level. This type of fade can be caused by attenuating effects of the atmosphere or it can be the precursor of selective fading. Some atmospheric conditions can cause propagation over two or more distinct paths, resulting in the reception of multipath components.[1] This event can cause the channel to experience selective fading. During selective fading, not only does the channel show a decrease in received signal power, but the signal level measured as a function of frequency (frequency response) also contains one or more minima.[2] Reference 2

* Bell Laboratories.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

contains graphs of measured field data showing typical time-varying frequency responses of a channel during selective fading.

A fade model consisting of four to six multipath components can match most observed channel characteristics.[3] However, a two-path fade model closely approximates a large number of observed channel characteristics.[4] The two-path model has the transfer function:

$$H(f) = 1 - re^{-j2\pi(f-f_0)\tau},$$

where $f_0$ is the frequency of maximum fade (i.e., minimum signal level) and $\tau$ is the delay difference in the two path lengths. The expression $-20 \log (1 - r)$ is the fade depth at frequency $f_0$.

Computer simulations were used to show that a two-path fade with a fade depth of 20 dB, a time-delay difference of 4 ns, and all possible frequencies of fade maximum ($f_0$) matches adequately the amplitude characteristics of the worst fades that would have to be equalized to meet system outage objectives for AR6A.*,† An equalizer that could compensate for such a fade to within ±2 dB of the nominal level for all values of the fade center frequency, $f_0$, will meet system specifications. Some examples of channel characteristics undergoing a two-path, 20-dB, 4-ns fade are shown in Fig. 1.

A good approximation of this selective fade characteristic can be obtained using a power series expansion truncated after the quadratic term: $H(f) = A_0 + A_1(f - f_m) + A_2(f - f_m)^2$, where $f_m$ is the midchannel frequency. The flat term, $A_0$, can be compensated using an Automatic Gain Control (AGC) amplifier leaving the shaped component alone to be equalized.

From the two-path model of the fading channel, the range of the linear term, $A_1$, must be ±18.5 dB and the range of the quadratic term, $A_2$, must be 17 dB. However, large values of the quadratic and linear shaping coefficients are not required simultaneously. This allowed the use of two Bode "bump" networks[5] with center frequencies at the channel ends to jointly realize the linear and quadratic correction terms.[6] It was further determined that a ±10 dB range on each of the Bode "bump" networks is sufficient to meet system requirements on shape equalization. Figures 2 and 3 depict the approximation of linear and quadratic correction functions by Bode "bump" networks. This technique has many advantages with respect to dynamic range and control circuitry, as well as noise figure and intermodulation distortion.

The remaining system equalization objective concerns the rate of change of the channel shape as a selective fade sweeps through the channel. Field data indicate that during deep selective fading, the rate

* Amplitude Modulation Radio at 6 GHz for the initial (A) version.

† Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

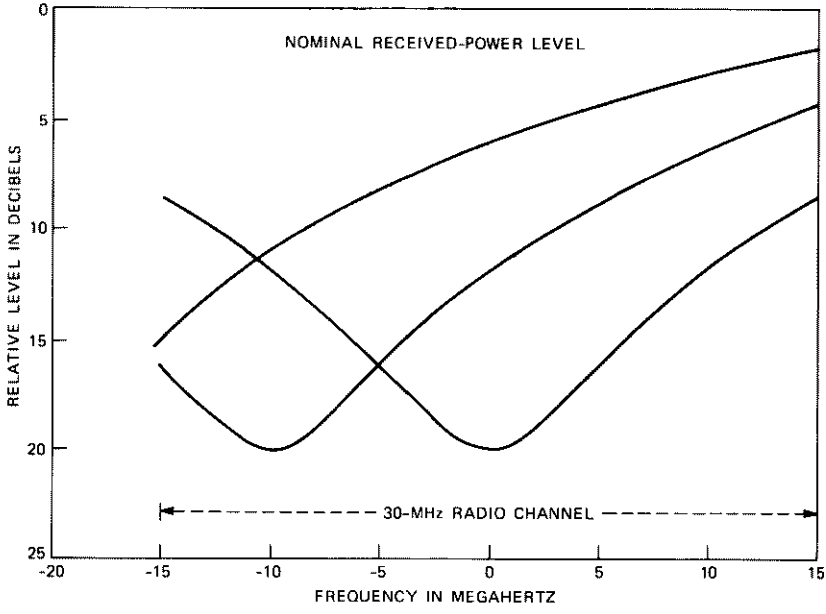



Fig. 1—Typical received-power frequency characteristic for the two-path model of a 20-dB, 4-ns fade in a 30-MHz channel.

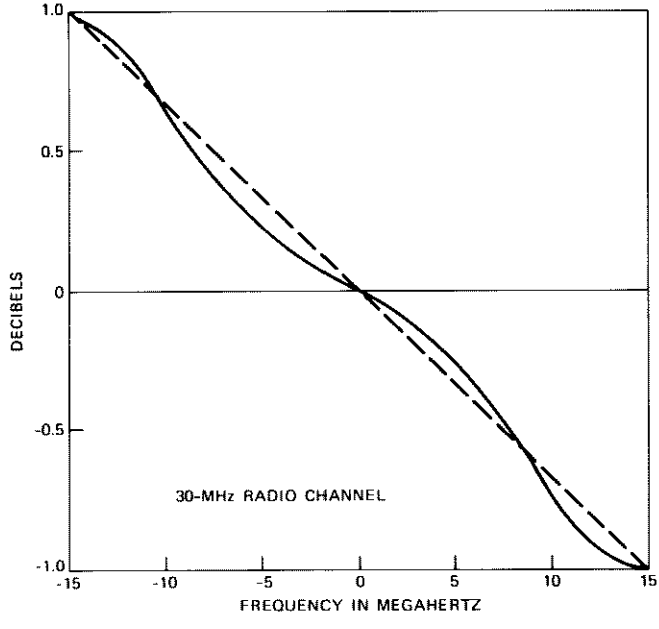



Fig. 2—Summation of two Bode "bump" networks to approximate a linear shape.

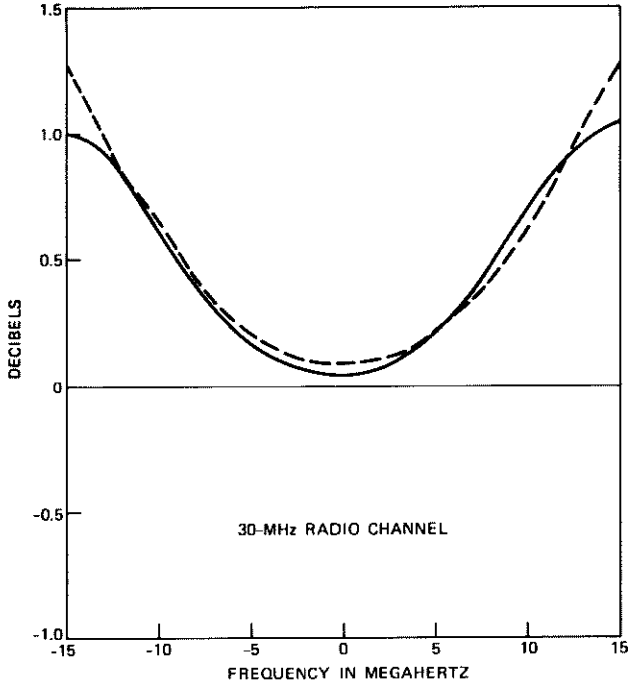



Fig. 3—Summation of two Bode "bump" networks to approximate a quadratic shape.

of change of signal level can be as high as 90 dB per second. Since the most severe fades will be eliminated by space-diversity switching, an objective of adapting to a rate of change of 50 dB per second was established.

## II. FUNCTIONAL DESCRIPTION

An error-detecting, zero-forcing technique was chosen to dynamically control the variation of the AGC and shape units. Three pilot tones are transmitted in the radio channel, one near each edge and one near the center. The pilot errors are determined by analog processing of the detected pilot levels. The errors thus determined are fed back to an analog circuit that varies the AGC and equalizer shape coefficients to force the detected errors to zero at the pilot frequencies. The control loop also detects when a space-diversity switch or pilot resupply should be initiated.

Figure 4 is a functional block diagram of the units comprising the dynamic equalizer.

### *2.1 Gain and equalization control*

Gain in the intermediate frequency (IF) channel is controlled by an AGC amplifier that is present at both repeater and main stations. The

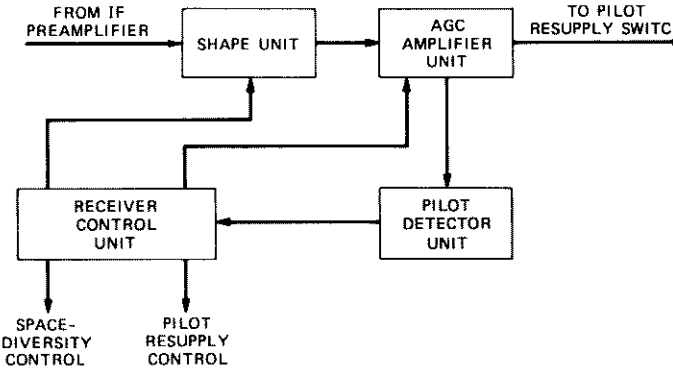



Fig. 4—Block diagram of the AR6A dynamic equalizer.

frequency characteristics of the IF channel are controlled by a shape unit that consists of electronically controlled Bode "bump" networks peaked at the high and low ends of the band. The shape unit is provided only at main stations.

The detector unit senses the levels of the three equalization control pilots and the pilot resupply enable pilot. Processing of the detected pilot levels is done within the receiver control unit. An error voltage for the low (high) bump is derived by subtracting a reference from the difference of the low (high) pilot level and the center pilot. An error voltage for the AGC is formed by subtracting a reference from the average of the three equalization pilot levels. These error voltages will all be zero when the three equalization pilots are at their nominal level. The error voltages are amplified, low-pass filtered, and used as the electronic control for the AGC amplifier and bump shapes.

Nonlinear shaping is provided in the control characteristics of the AGC gain and bump networks such that the gain (in dB) introduced is proportional to the drive voltages. A closed-loop gain of 100 is provided. A 1-Hz cutoff in the loop filter results in a closed loop response time of 10 ms.

The dynamic equalizer has been designed to regulate the pilot levels to within 0.5 dB of their nominal value under conditions of nominal input. For a 20-dB step change in input, the equalizer will settle to within 2 dB of its final value in under 10 ms. Also, a maximum dynamic tracking error of 2 percent of the fade depth in decibels is achieved for fade rates of less than 50 dB per second.

The dynamic equalizer has been designed to achieve a gain flatness of ±0.15 dB at nominal temperatures and ±0.5 dB over the temperature range of 40 to 120 degrees F.

The dynamic equalizer achieves a maximum noise figure of 20.7 dB and third-order intermodulation coefficient for $A + B - C$ products of −72.9 dB for main-station applications.

### 2.2 *Space-diversity and pilot resupply control*

The regulation accuracy requirements dictate that the pilot detector realize a precise measure of the nominal pilot level. A low-precision realization of the control loop would suffice were it not for a requirement to precisely determine the crossing of thresholds used to initiate a space-diversity switch.

A space-diversity switch will be initiated if any pilot at the *input* to the dynamic equalizer is more than 36 dB below its nominal level. To avoid the need for separate detectors to monitor this condition, the space-diversity control is derived within the dynamic equalizer circuitry. It turned out that this requirement limited the control circuitry design.

The detector output is an indication of the output pilot value relative to its nominal value. The control voltages to the AGC and bump networks are an indication of the amount of gain supplied between the equalizer input and output. Nonlinear shaping of the detector, AGC, and bump-network control characteristics yields a linear relationship in units of dB per volt; a weighted sum of the detector outputs and the control voltages yields a measure of the input pilot levels. These characteristics must be precisely controlled to provide accurate threshold comparisons.

In addition to the space-diversity control, the equalizer circuitry realizes the control signal that initiates a pilot resupply. This circuitry utilizes a fourth pilot detector, which senses the level of the resupply enable pilot.

## III. DETECTOR AND CONTROL CIRCUITS

### 3.1 *Detector*

The detector circuit contains four separate pilot detectors interconnected by a tree of hybrid transformers (see Fig. 5). Each pilot detector consists of a hybrid transformer, an IF bandpass filter, an IF amplifier detector, and an operational amplifier gain circuit.

The IF bandpass filter is a two-section, monolithic crystal filter with a 3-dB bandwidth of approximately ±6 kHz and a 60-dB bandwidth of approximately ±50 kHz. The midband loss is approximately 3 dB. Figure 6 shows a typical characteristic of these filters.

The IF amplifier (see Fig. 7) consists of two stages of a series-shunt feedback pair.[7]

The detector portion of the circuit consists of a Schottky diode. An R-C circuit serves as a filter for the detected signal. This circuit acts essentially as a peak detector that derives a dc voltage proportional to the peak of the input IF signal.

The operational amplifier circuit is used as a buffer between the

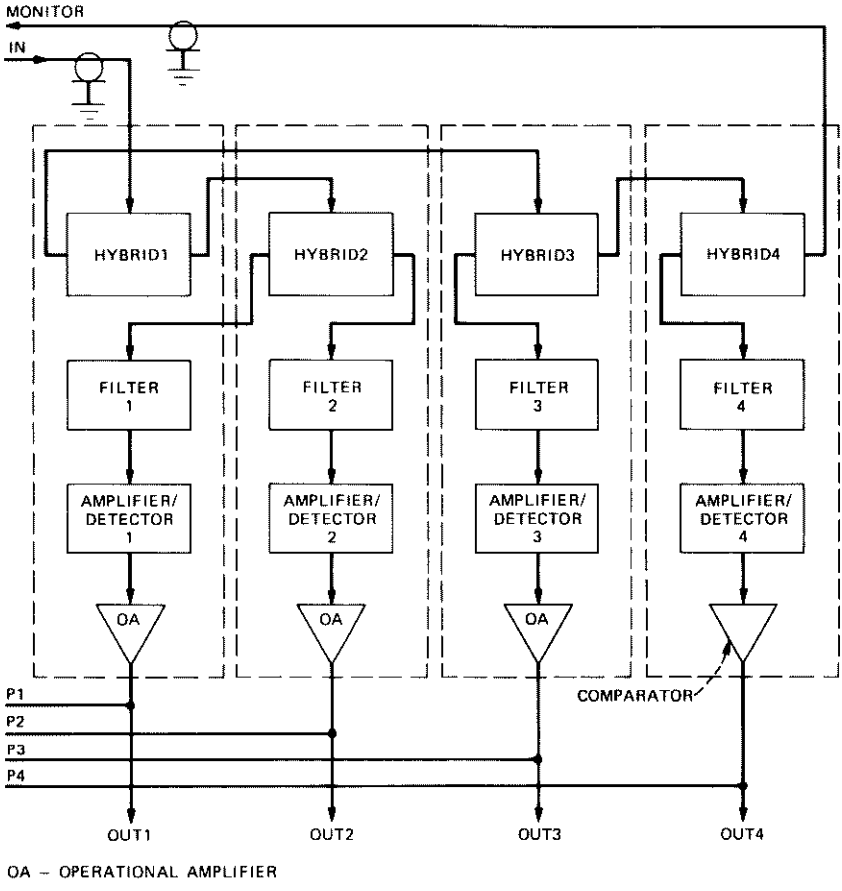



Fig. 5—Pilot detector circuit.

detector and the output of the circuit. This circuit has two adjustments. One is an offset adjustment used to adjust for zero output voltage with a nominal input signal level. The other is a gain adjustment that is set for a given dc output when the input level is 10 dB below nominal. This adjustment is required to compensate for variations in gain at the three pilot frequencies and also for variations in the Schottky diode.

The resupply-enable pilot detector is similar to the three other detectors except that the operational amplifier gain circuit is replaced by a comparator circuit, and the only adjustment is the comparator threshold voltage. A single adjustment is sufficient because only the detection of the presence of the resupply pilot above a certain level is required.

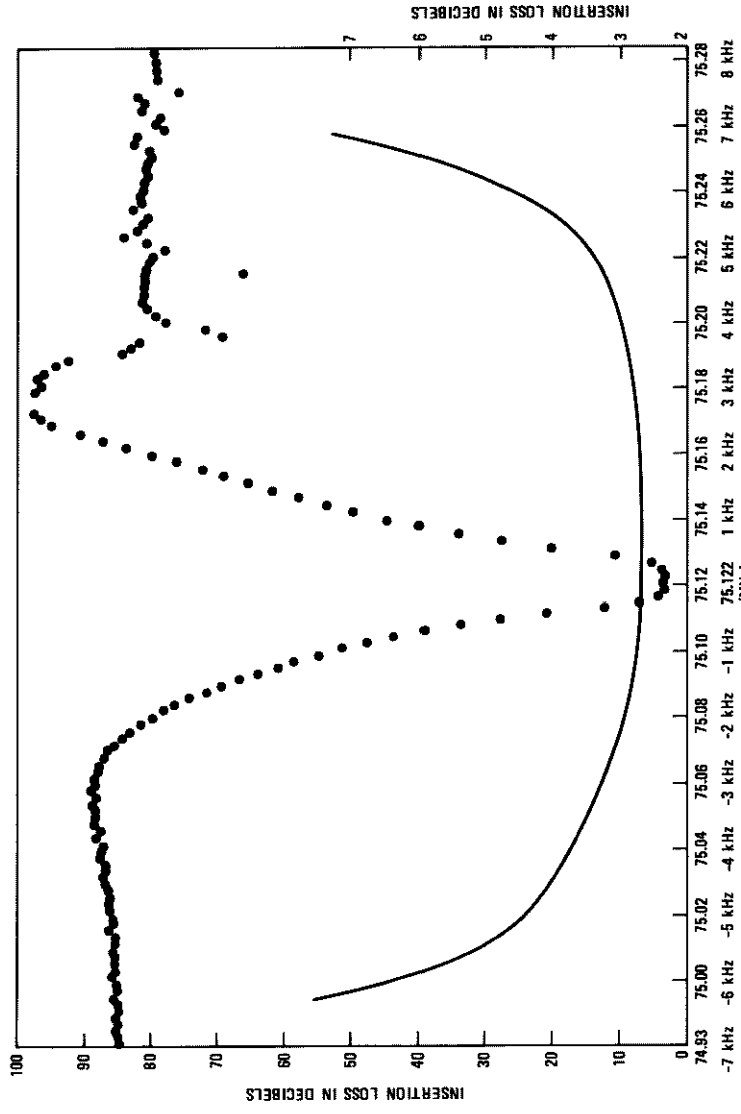



Fig. 6—Typical IF bandpass filter characteristic.

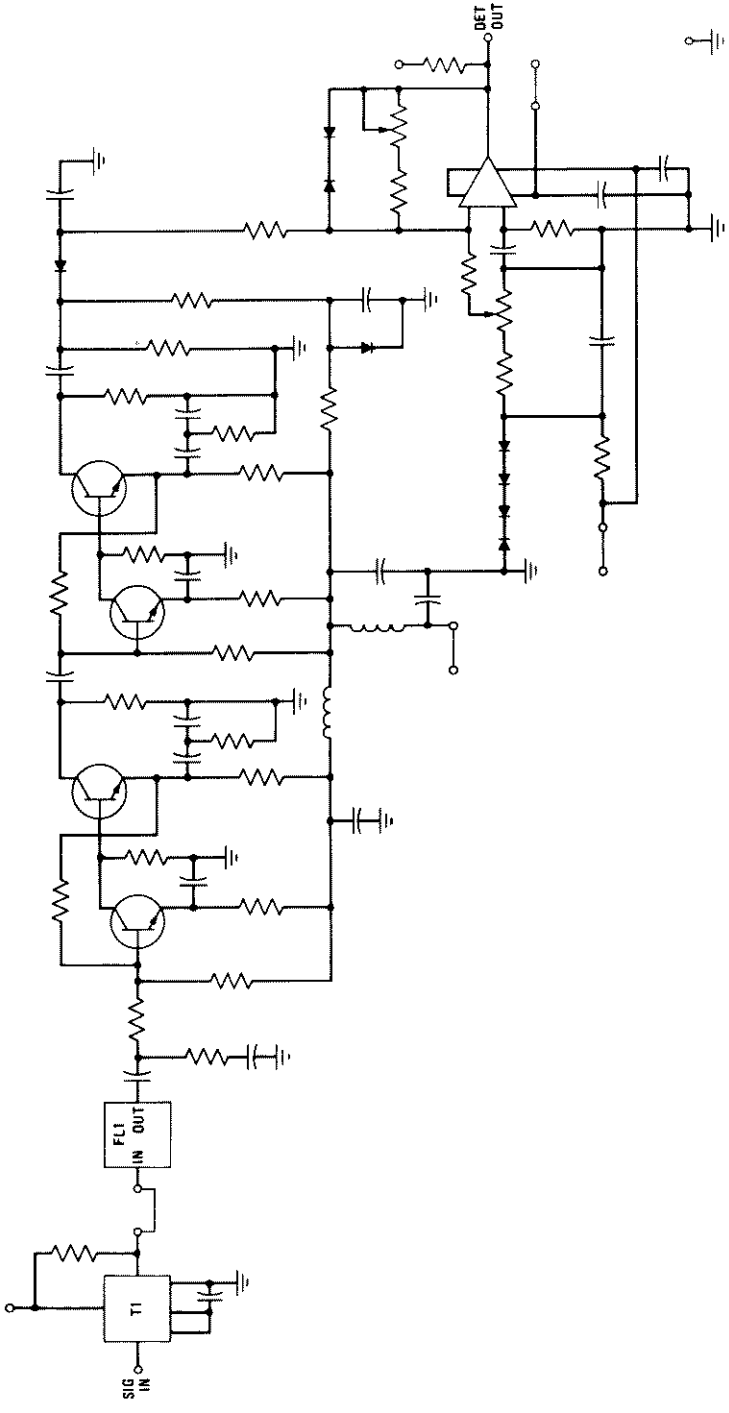



Fig. 7—Pilot detector circuit.

### *3.2 Receiver control circuit*

There are two types of receiver control units. The type that only controls the gain of the AGC amplifier is used at repeater stations. The type used only at main stations controls the gain of a shape unit as well as an AGC amplifier.

The receiver control circuit also generates the following initiation signals: (1) a signal to initiate a pilot resupply switch when the receiver is either underpowered or overpowered, and (2) a signal to initiate a space-diversity switch when the level of any one of the three line pilots drops below a certain threshold.

The circuit controls manual and remote receiver gain and remote pilot resupply, and also controls alarm initiating signals when the receiver is in other than its normal mode of operation.

Figure 8 is a block diagram of a main-station receiver control unit. The operation of the circuit is as follows.

To generate the space-diversity switch initiating signal, the inputs

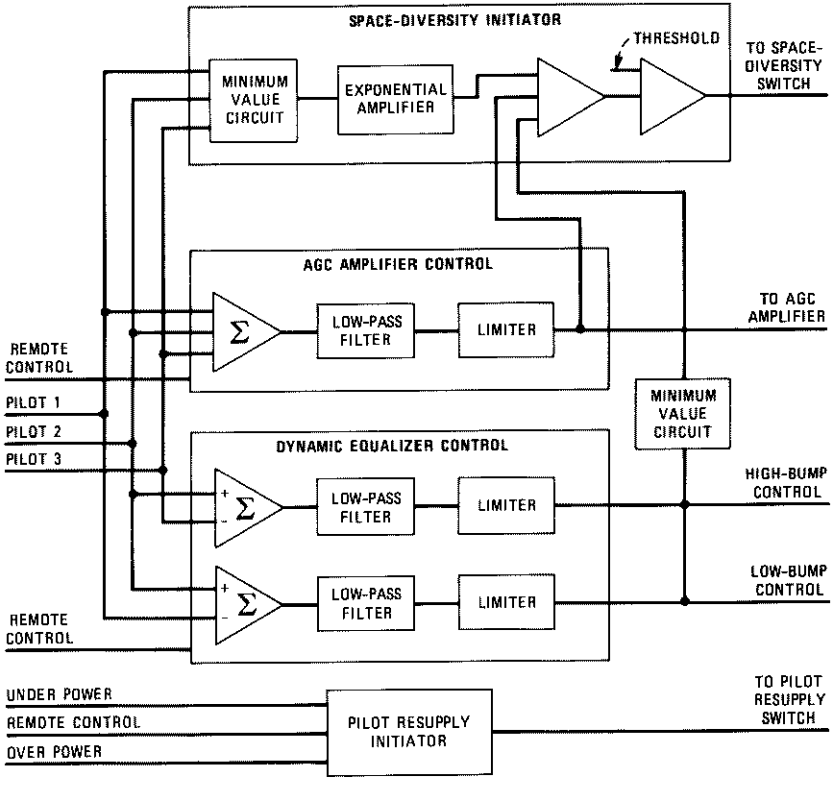



Fig. 8—Main-station receiver control circuit.

from the pilot detector circuits (PLT1–PLT3) are applied to a minimum value circuit. This circuit takes the level representing the lowest level pilot and applies it to an exponential amplifier. The output of this amplifier is a voltage that is linearly proportional to the decibel pilot level. The other input that determines the space-diversity threshold point is the AGC amplifier control voltage. The AGC amplifier is designed so that the gain in decibels is a linear function of the control voltage. The AGC control voltage is determined by the average of the three pilots. The comparator threshold is set to trip when the input voltages are such that the AGC control voltage and the voltage representing the level of the minimum-level pilot indicates that one of the pilots is 36 dB below nominal. The only difference between the repeater-station and the main-station receiver control circuits is that in the main-station circuit the control voltages for the shape unit are also used to determine the space-diversity initiation circuit trip point.

The pilot resupply is initiated in any one of three ways. It will be initiated when the power at the receiver output remains approximately 12 dB above nominal for 100 milliseconds. In this case, the functional input to the receiver control unit is from the resupply switch unit, which senses the total power and produces a control voltage proportional to the average power at the output of the AGC amplifier. The pilot resupply circuit will also be initiated if the average level of the three pilots drops 5 dB below nominal. It releases when the average level returns to less than 4 dB below nominal. This condition is derived from inputs from the detector unit. The pilot resupply can also be initiated remotely via the command and control system.

The AGC amplifier control section takes the voltage proportional to the level of the three pilots, averages them, and low-pass filters the resulting voltage to set the bandwidth of the loop. The output of the low-pass filter is applied to a limiter to limit the maximum gain of the AGC amplifier.

The shape unit control section of the receiver control unit takes the difference between the level of the center pilot and the upper or lower end pilots to control the appropriate bump equalizer in the shape unit. These voltages are also low-pass filtered and limited.

The repeater station receiver control unit does not contain a shape unit control section because a repeater station does not use a shape unit.

The receiver control unit also allows for the AGC amplifier to be set to nominal gain or manually adjusted over its entire gain range. Light-emitting diodes (LEDs) on the faceplate of the unit indicate when the dynamic equalizer is in other than its normal mode of operation.

## IV. THE SHAPE UNIT

The shape unit is composed of four Bode-type adjustable "bump" networks separated by amplifiers and controlled by four driver circuits as shown in Fig. 9. Two identical networks provide a bump-type characteristic at 59.8 MHz. The remaining two networks are also identical and peak at 88.5 MHz. Each network section is separated by amplifiers to provide both impedance buffering and gain to offset the flat loss introduced by the networks. High-frequency network sections are alternated with low-frequency sections to minimize interaction between like sections.

Each network section has an associated drive circuit. This provision allows for the independent operation of each network section so that varying p-i-n diode characteristics may be compensated in the driver element of the network section.

### *4.1 The bump equalizer sections*

Each "bump" network section is composed of a series-type, adjustable Bode network as shown in Fig. 10. The adjustable element is a p-i-n diode. The p-i-n diode acts as a resistor that varies in accordance with a bias voltage supplied by the drive circuit. As the resistance value of the p-i-n diode varies, the network shapes the amplitude of the transmission above and below the flat loss level. Measured amplitude shapes are shown in Fig. 11.

### *4.2 The buffer amplifiers*

As shown in Fig. 12, all of the amplifiers provide hybrid-type inputs and outputs. All amplifiers are designed to present 75-ohm input and output impedances. The amplifiers are, however, tuned to operate against the 301-ohm impedance presented by each "bump" network section. Each amplifier provides 7 dB of gain to compensate for the loss introduced by the bump sections.

### *4.3 The drive circuits*

The drive circuits receive the drive voltages from the receiver control unit. As shown in Fig. 9, two voltage follower circuits isolate the drive circuits from the receiver control unit. A schematic description of the drive circuits is shown in Fig. 13.

Each drive circuit is a unity gain dc amplifier. Potentiometer R26 allows for the adjustment of the offset voltage to the input of the amplifier. It is adjusted to provide flat transmission over the 59- to 89-MHz band with a drive voltage of 0V.

Elements R13 through R20, CR2, and CR3 function to make the insertion loss in decibels at the 62.448- and 85.856-MHz pilot frequencies proportional to the drive voltages. The circuit is adjusted so that

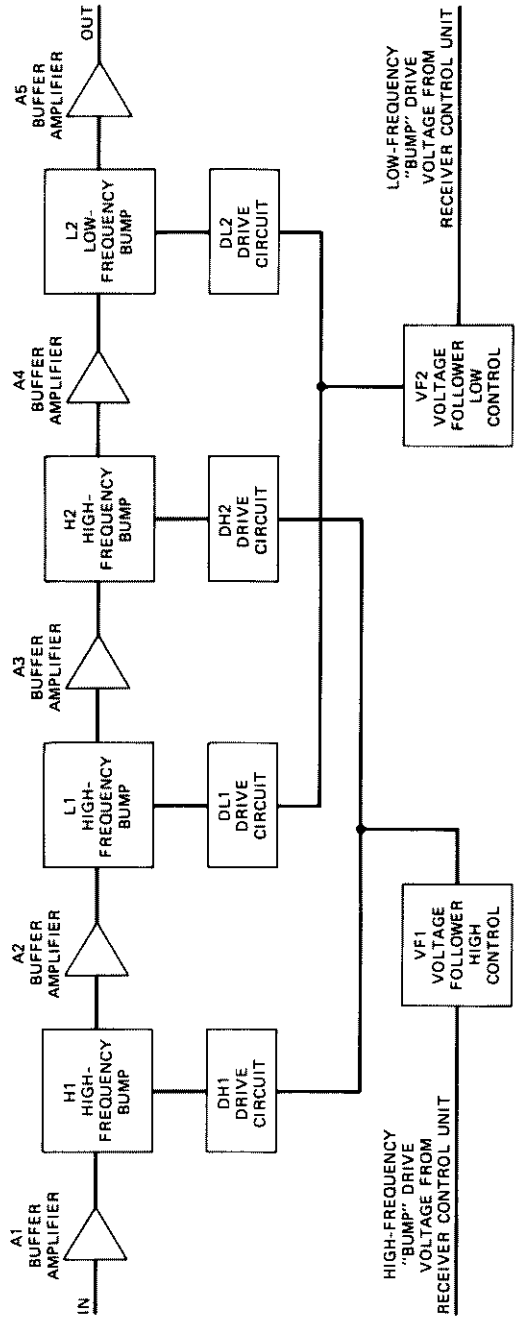



Fig. 9—Block diagram of the dynamic equalizer.

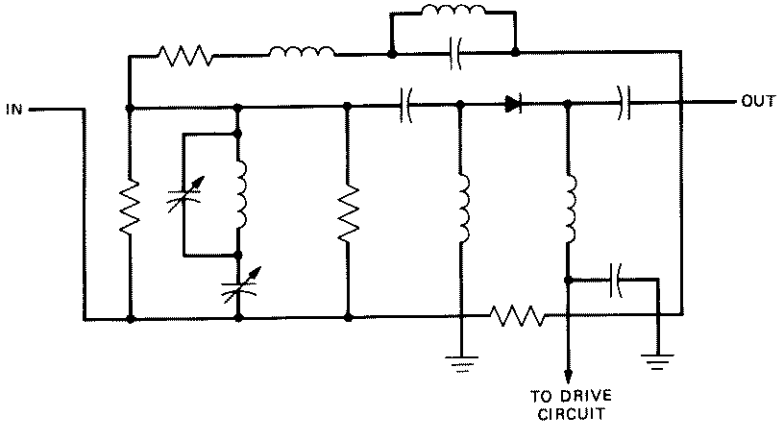



Fig. 10—Bode-type adjustable equalizer.

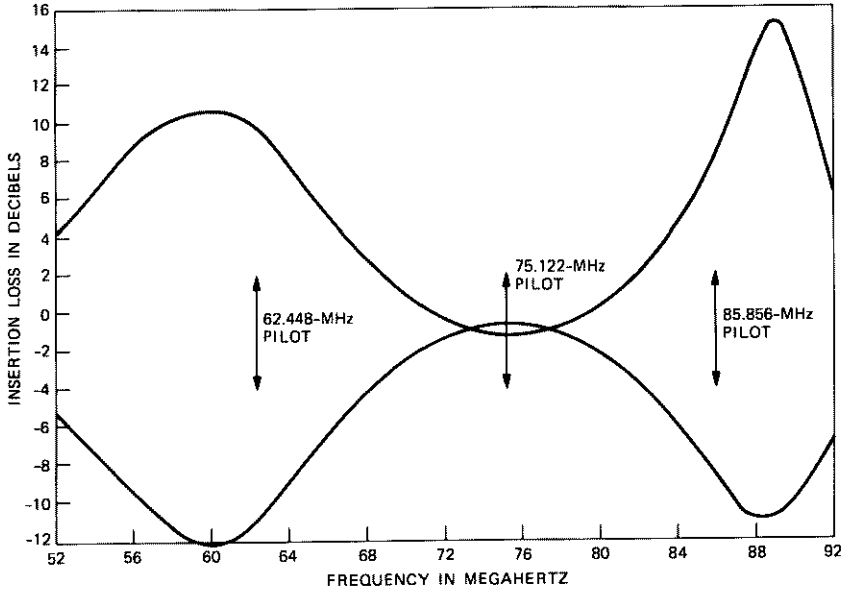



Fig. 11—Typical amplitude shapes generated by the dynamic equalizer for the AR6A System.

a change of 1V in drive voltage results in a 1-dB change in transmission for each network section. This linear relationship is the criterion upon which the space-diversity switching is based.

Diodes CR4 through CR6 act as temperature-sensing elements. Their purpose is to compensate for the temperature versus resistance characteristic of the p-i-n diode. Over the 40- to 140-degrees F temperature range, amplitude distortions in the transmission amount to

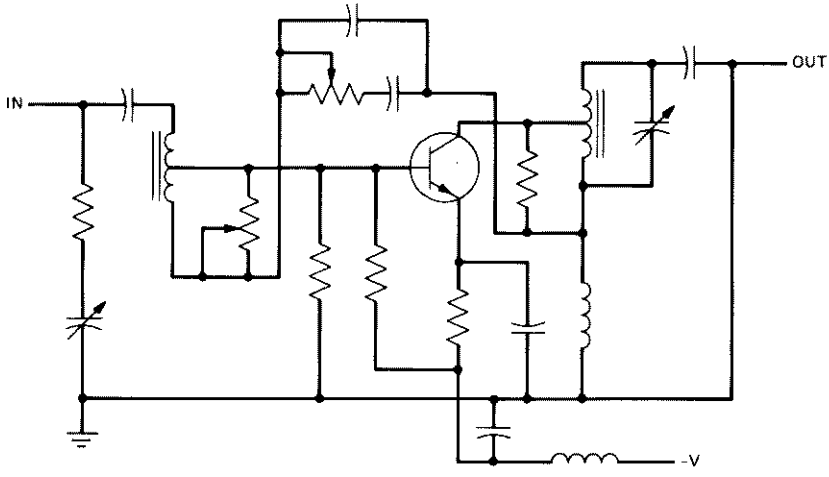



Fig. 12—Buffer amplifier.

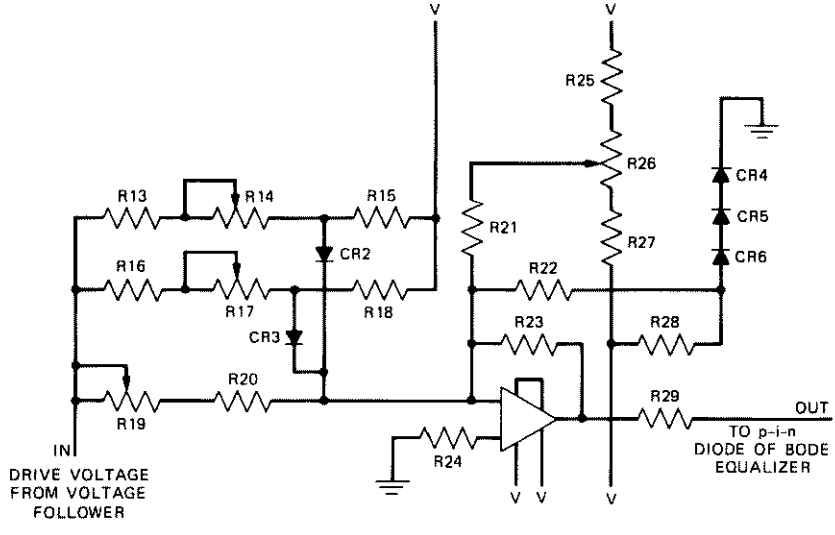



Fig. 13—Drive circuit.

as much as ±1.5 dB between pilot frequencies without compensation circuitry. With compensation these distortions are held to within ±0.25 dB.

## V. AUTOMATIC GAIN CONTROL AMPLIFIER

The AGC amplifier normally operates at a nominal gain of 15.7 dB and has a gain range of 61 dB. It is made up of a series of fixed-gain amplifier blocks alternating with variolossers (see Fig. 14). Each

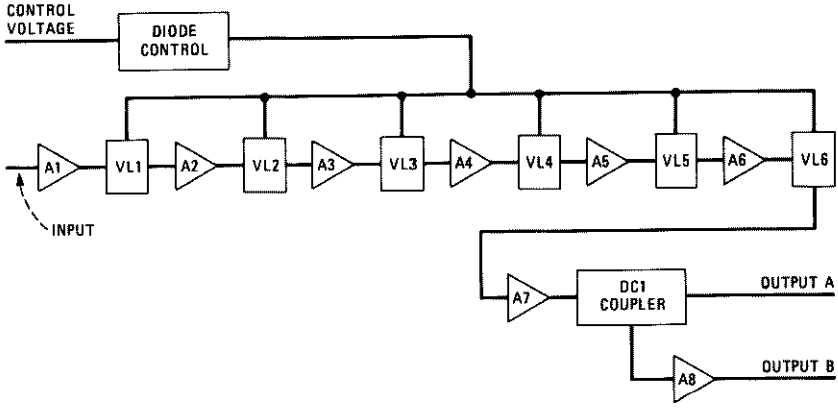



Fig. 14—Block diagram of an AGC amplifier.

variolosser has a dynamic range of approximately 10 dB, so six are necessary to provide the required 61-dB overall control range.

A control circuit is provided to convert the AGC input control voltage to a form that can be used to control the loss of the variolossers. This circuit is basically a voltage-controlled current source that linearizes the input control voltage versus overall AGC gain transfer curve. Another function of the control circuit is to stabilize the temperature of the AGC amplifier.

### *5.1 Amplifier blocks*

Eight fixed-gain amplifiers of the same basic design are required in the AGC amplifier.

Amplifiers A1 through A6 (see Fig. 14) are two-stage feedback amplifiers with a gain of 10.5 dB each. Amplifier A7 has a gain of 16 dB. This gain distribution was chosen to optimize the noise figure and intermodulation performance of the AGC amplifier. Very careful attention had to be paid to the physical design and layout since there is a total distributed gain of almost 80 dB.

The buffer amplifiers in the AGC saturate at a rather high power level due to the heavy biasing required to achieve low intermodulation distortion. Interstage clamping circuitry was added to the AGC to limit the output power to a safe level without affecting normal operation.

Directional coupler DC1 samples a portion of the main path signal and directs this signal to the amplifer A8. Amplifier A8 is a single-stage feedback amplifier with a gain of 8.5 dB. This output provides the signal to the pilot detector unit.

All eight of the amplifier blocks utilize a hybrid transformer impedance-matching configuration (see Fig. 15). This circuit can easily

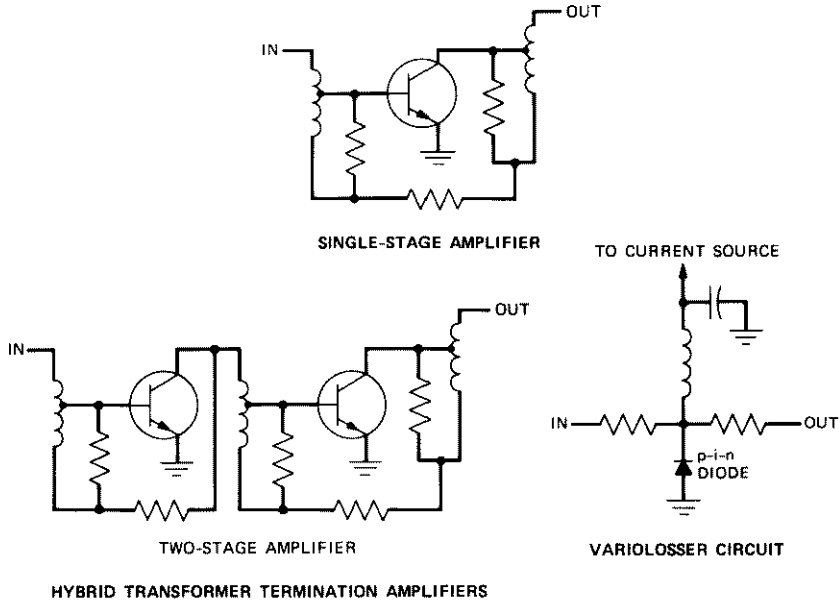



Fig. 15—Functional diagram of the AGC amplifier.

provide better than 30-dB return loss, which is important for the input and output ports of the AGC amplifier and for providing good buffering characteristics to mask the variolosser impedance variation.

The amplifiers in the AGC have been designed for optimum linearity performance to minimize intermodulation distortion. All eight amplifiers use a transistor that has been designed for optimum linearity. To utilize this transistor properly it must be operated at a collector voltage of 13V and an emitter current of 120 mA. Since 15 transistors are required for the AGC amplifier, the result is a relatively high total dc power consumption of 2 amps at −14.7V, or 30W. Consequently, a rather elaborate heat sink and ventilation scheme was required (see Fig. 16). The heat removal problem[8] was complicated by a very tight radio frequency interference requirement on the AGC amplifier.

### *5.2 Variolossers*

The variolosser is essentially a T-pad attenuator with a p-i-n diode as the shunt arm to ground (see Fig. 15). The impedance of the p-i-n diode is varied by changing the bias current through the diode. The p-i-n diode is a type that was developed specifically for this application.

The input AGC control voltage is common to all variolossers (see Fig. 14), but each p-i-n diode has its own voltage-controlled current source. The loss of each variolosser is variable from approximately 3 to 13 dB.

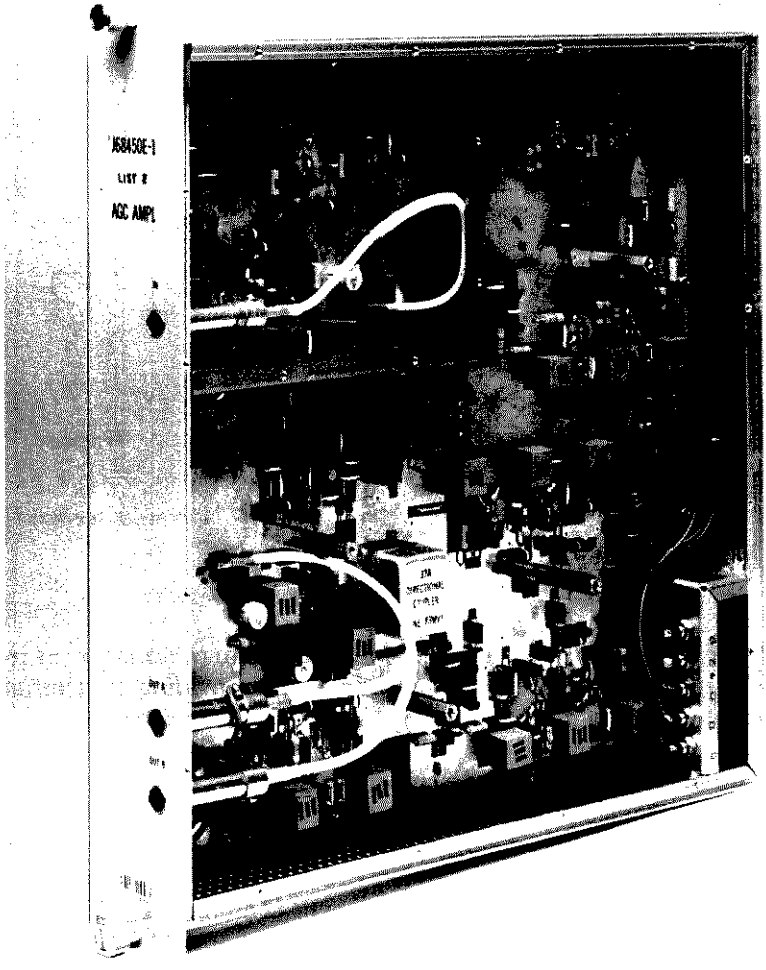


Fig. 16—AGC amplifier.

### *5.3 Control circuit*

The control circuit linearizes the control voltage versus decibel AGC gain curve and compensates for the temperature in the AGC amplifier.

It was necessary to linearize the control function since the p-i-n diode variolosser characteristics are very nonlinear. The AGC control section introduces a compensating distortion in the control function so that the overall transfer function is linear.

The p-i-n diodes in the variolossers are temperature sensitive so the ambient temperature variations must be compensated for. The ex-

pected ambient variation is from 40 to 140 degrees F. The voltage across a forward-biased silicon diode is used as a temperature-sensing element. This temperature-dependent voltage is used to modify the variolosser drive voltage in the proper way so that the AGC gain remains constant as the ambient temperature varies.

The decision was made to utilize a two-point "switched" temperature compensation concept rather than attempting the formidable task of tracking the constantly changing p-i-n diodes. With this method two points on the AGC gain curve are perfectly temperature compensated with an acceptable smooth transition in between.

## VI. CONCLUSION AND ACKNOWLEDGMENTS

A dynamic equalizer consisting of an AGC amplifier, a shape unit made up of Bode "bump" networks, and control circuitry has been designed to correct for the effects of selective fading in an AR6A channel. The use of Bode "bump" networks is key to the realization of a dynamic equalizer that is low in complexity and yet adequately meets system requirements.

Credit is due to many individuals associated with phases of this development. The authors specifically wish to acknowledge J. M. Kiker, Jr., for his guidance of the AGC amplifier development and P. D. Patel for the physical design of the dynamic equalizer units and IF shelf. The support and encouragement of F. J. Witt throughout the project is also gratefully acknowledged.

## REFERENCES

1. W. T. Barnett, "Multipath Propagation at 4, 6, and 11 GHz," B.S.T.J., *51*, No. 2 (February 1972), pp. 321–61.
2. G. M. Babler, "A Study of Frequency Selective Fading for a Microwave Line-of-Sight Narrowband Radio Channel," B.S.T.J., *51*, No. 3 (March 1972), pp. 731–57.
3. R. L. Taylor, "A Statistical Study of Selective Fading of Super-High-Frequency Radio Frequencies," B.S.T.J., *32*, No. 5 (September 1953), pp. 1187–202.
4. G. M. Babler, "Selectively Faded Nondiversity and Space Diversity Narrowband Microwave Radio Channels," B.S.T.J., *52*, No. 2 (February 1973), pp. 239–61.
5. H. W. Bode, "Variable Equalizers," B.S.T.J., *17*, No. 2 (April 1938), pp. 229–44.
6. United States Patent 4,003,006, G. M. Mandeville and D. I. McLean, issued January 11, 1977.
7. M. S. Ghausi, "Optimum Design of the Series-Shunt Feedback Pair with a Nominally Flat Magnitude Response," IRE Trans. Circuit Theory, *CT-8*, No. 4 (December 1961), pp. 448–53.
8. S. A. Harvey and P. D. Patel, "The AR6A Single-Sideband Microwave Radio System: Radio-Line Physical Design," B.S.T.J., this issue.

## AUTHORS

**Norman O. Burgess,** BS Ed, 1957, State College at Bridgewater, Massachusetts; Bell Laboratories, 1958—. Mr. Burgess' main responsibilities have been the design and development of passive analog filters and networks for analog and digital systems.

**Roderick C. MacLean,** ASEE, Franklin Technical Institute; Bell Laboratories, 1958—. Mr. MacLean was initially involved in the design of filters and equalizers for transmission systems. He then became a member of the Analog Computing group. From there he joined the Network and Subsystem Design group. In 1982 he joined a group that is responsible for the design of a multiplex for a lightwave system.

**Gordon J. Mandeville,** BSEE, 1968, Northeastern University; MSEE, 1970, Massachusetts Institute of Technology; Bell Laboratories, 1968—. Mr. Mandeville has worked on new equalizer structures for IF applications, automatic equalization for single-sideband radio systems, design aids for filter synthesis, wave digital filters, digital signal processing applications in transmission systems, and magnetic components. He is now Supervisor of the VLSI and DSP Applications Group. Member, IEEE, Sigma Xi, Tau Beta Pi, Eta Kappa Nu.

**Dale I. McLean,** BSEE, 1958, University of Maine; MSEE, 1961, Northeastern University; Western Electric, 1958–1959; Bell Laboratories, 1960—. Mr. McLean initially designed a variety of filters and networks for long- and short-haul carrier transmission systems and continued with work on subsystems. He is now concerned with hybrid integrated circuit design and laser adjustment of thin-film components. Member, IEEE, Tau Beta Pi.

**Melvin E. Sands,** BSEE, 1957, Virginia Tech; Bell Laboratories, 1962—. Mr. Sands was previously involved in the design and the development of radars for the U.S. Army's ballistic missile defense system. Since 1970 he has been involved in design and development work on the AR6A radio transmission system.

**Robert P. Snicer,** BSEE/MSEE, 1967, Massachusetts Institute of Technology; Bell Laboratories, 1967—. Mr. Snicer has been involved in digital transmission system design, computer-aided design, filter design, and subsystem design. He is presently Supervisor of a digital terminal applications software group. Member, Sigma Xi, Tau Beta Pi, Eta Kappa Nu.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# The Traveling-Wave-Tube Amplifier

By J. F. BALICKI,* E. F. COOK,* R. C. HEIDT,* and V. E. RUTTER*

(Manuscript received February 4, 1983)

This paper describes the amplifier for the AR6A radio transmitter, which consists of a Traveling-Wave Tube (TWT), its magnetic focus system, and driving power supply. The amplification of single-sideband modulated signals requires an unusually low intermodulation noise level and relatively low thermal noise power at the amplifier output. Further, owing to the use of predistortion to achieve the overall repeater intermodulation level objective, the amplifier must perform at high stability. These objectives have been met by choosing suitable TWT design parameters, operating the tube far below its saturated power capability, and devoting close attention to the details of tube construction. The power supply provides well-regulated heater, helix, and collector voltages to control the TWT gain. The supply also regulates the beam current. Current detection circuits protect the tube and power supply from tube internal arcs.

## I. GENERAL

The development of a traveling-wave-tube amplifier was undertaken to fulfill all proposed operating requirements for noise, signal intermodulation, gain, safety, and mechanical stability. Prior experience with Western Electric Traveling-Wave Tubes (TWTs)† indicated the

*Bell Laboratories.

†Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

direction of design. Subsequent application in the AR6A* Radio System has shown that the device, designated the 473A TWT, when operated with the power supply described in this article, meets all detailed repeater requirements. The use of this TWT was discontinued following the decision to terminate the manufacture of all Western Electric traveling-wave tubes. An equivalent TWT obtained from a commercial source and meeting the same requirements is now used in production of the AR6A Radio System.

## II. THE TRAVELING-WAVE TUBE

### *2.1 Requirements on the TWT*

The 473A TWT includes the vacuum tube and the focusing magnetic circuit (Fig. 1). This combination must be united at the manufacturing location, so that optimum focus and Radio Frequency (RF) operating parameters are achieved.

The following list of major requirements specified for the TWT includes the operating conditions specified over the frequency band of 5.925 to 6.425 GHz (with full-load output power of 24.8 dBm and gain reduced 2 dB below the optimum value by increasing the helix voltage):

1. RF gain—40 to 48 dB.
2. Gain plus noise figure—69.5 dB maximum.
3. Third-order intermodulation (three-tone test method)[1]—$M_{A+B-C}$ = −90.5 dB maximum, and constant within ±0.4 dB with the power per tone varied from 18 to 30 dBm.
4. Fifth-order intermodulation ($M_{2A+2B-C}$)— from 22- to 30-dBm output power, −180-dB maximum at 18-dBm output power, −160-dB maximum.

The device developed to meet these requirements consists of a helix-type, single-collector, convergent-flow, electron-gun traveling-wave tube, focused by a Periodic Permanent Magnetic (PPM) field.

As observed from the requirements, noise figure, signal intermodulation, and gain were the major operating parameters to be satisfied. Also, implied by the application, long operating life requires low cathode current loading and minimum interception of current from the beam by the helical slow-wave structure. Further, a critical need for this application is mechanical stability; it was verified that minute movement of the helix results in significant shift in intermodulation (IM) performance.

## III. TWT COMPONENT DESIGN APPROACH

### *3.1 Focus circuit and tube envelope*

The PPM focus circuit was adapted from an existing Western Electric manufactured circuit used on the 461A TWT. The existing

*Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

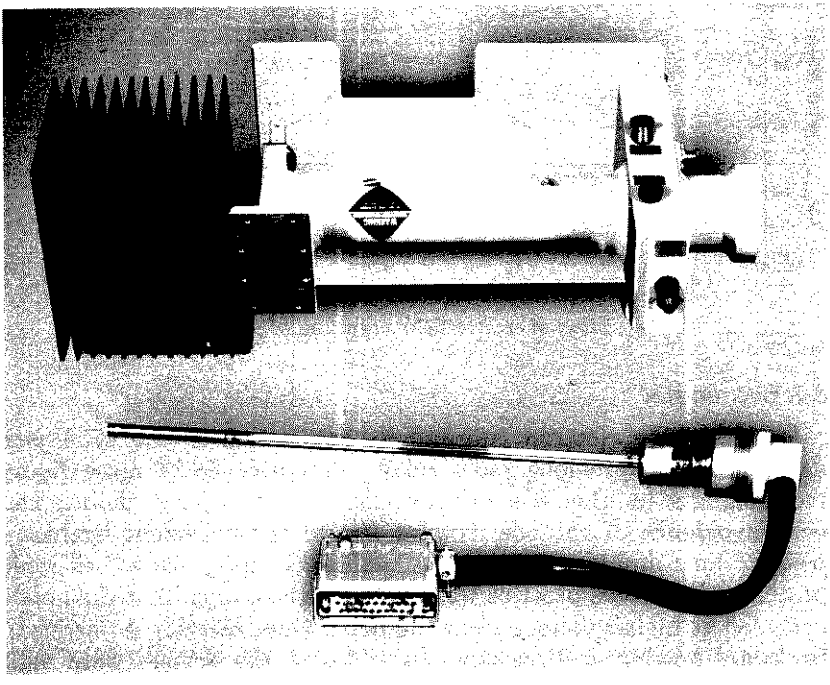


Fig. 1—Picture of AR6A 473A TWT.

glass envelope was also used so that it fit the magnet and pole piece parts. Since the TWT was required to be convection cooled by free air, a finned cooling structure was developed. An SMA coaxial RF input was specified and the coaxial-to-waveguide transition was made an integral part of the circuit.

Of major importance was the introduction of a cross-magnetic field in the collector region. The reason for this is that elastically reflected electrons or electrons emitted by secondary emission from the tube collector resulted in feedback to the input, which has a deleterious effect on intermodulation performance. The smoothing effect, with power variation, upon the third-order intermodulation, with and without the use of a cross field, is shown in Fig. 2.

The focus circuit is assembled from Alnico 8 magnets and soft iron pole pieces on precision mandrels so that no trimming of the magnetic field is necessary for tube focus. A movable pole piece near the input waveguide allows optimum focus to be reached. Cooling fins are coupled to the copper collector of the tube via a thin layer of conducting silicone grease. Adjustable tuning plungers in the waveguide optimize the RF impedance match between the input and output waveguides and helix.

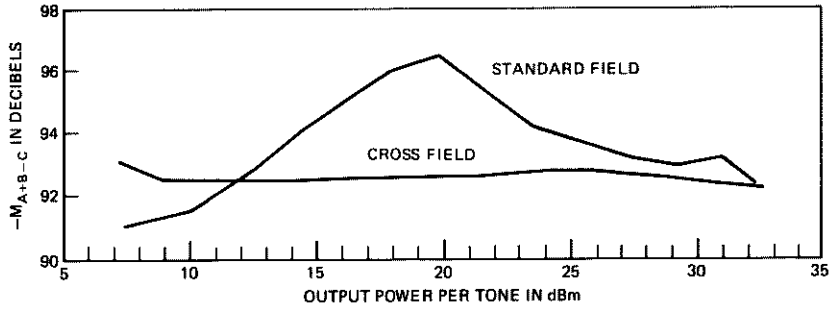



Fig. 2—Effect of crossed-field magnet on $M$ performance.

### *3.2 Electron gun*

Design of the electron gun and its application in the 473A is a departure from previous radio relay tubes because of noise figure requirements. In other Western Electric designs, major guidelines had been avoidance of cathode damage due to bombardment by heavy positive ions and use of moderate cathode loading. To afford ionic protection, previous practice was to design the gun so that the anode operated at higher voltage than the helix, thus, erecting a barrier to backward-flowing positive ions. In addition, the devices were built with highly convergent beam profiles, i.e., with the cathode diameter being much greater than the diameter of the beam projected through the helix. Typically, the emission density in such designs is $0.2\text{A/cm}^2$ of cathode area.

The effect on noise figure of both helix-voltage-to-anode-voltage ratio, $E_w/E_c$, and cathode current density, $J_C$, is shown in Fig. 3. The upper plot, evaluated at $J_C = 0.2$, shows the noise figures obtained with previous design rules. Clearly, lower noise figure can be obtained with $E_w/E_c > 1$ and with the cathode diameter smaller. A standard oxide cathode coating would be advantageous over other less proven, high-current types if the cathode loading could be reduced to allow its use. The anode voltage was set at 3150, nominal, so that $E_w/E_c = 1.3$. Diminishing benefit would be obtained from a higher ratio, and leakage problems on insulators would also increase. The resulting electron gun is a glass-rod supported triode that emits 70 mA dc from an oxide-coated cathode. The cathode-heater structure is designed to operate at 740 degrees Celsius (observed pyrometric temperature).

The life capability of the cathode has not been entirely confirmed. Based on calculation and extrapolation, greater than 30,000 hours is predicted. Limited facilities allowed five tubes to be run individually over times ranging from 3100 to 8500 hours. One of the five showed significant cathode degradation at 6000 hours; the other four were stable. No short-life problems have been encountered in application.

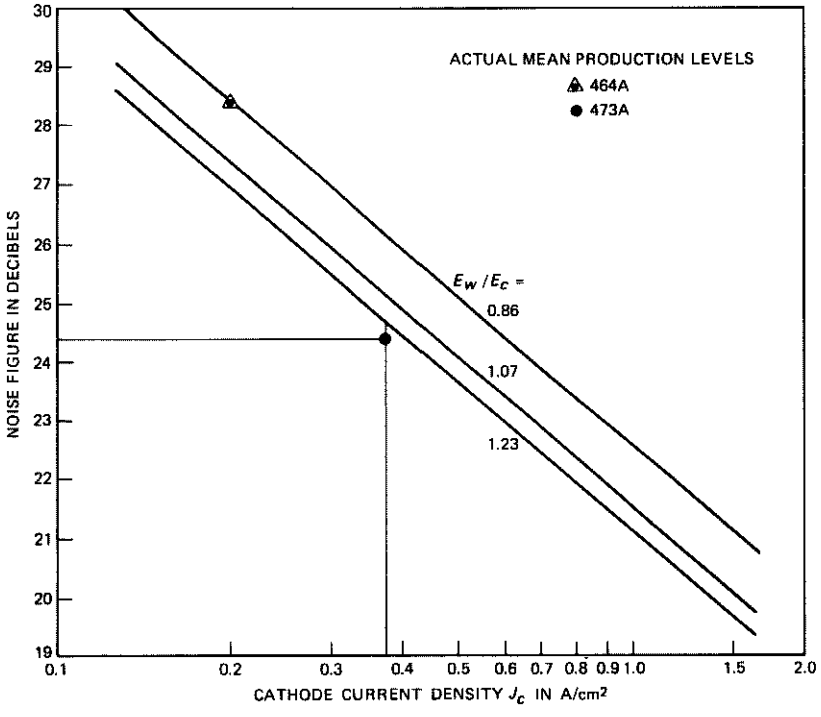



Fig. 3—Effect of cathode loading on noise figure for different voltage profiles.

### *3.3 Helical slow-wave circuit*

Initial tests confirmed the work of Curtice,[2] showing that for lowest intermodulation distortion the helix RF loss should be minimized and the gain parameter $C$ (which is proportional to the 1/3 power of beam-current-to-helix-voltage ratio) should be maximized.

Consideration of the effect of the gain parameter led to design of the helix to operate at 4100 volts. It is noted also that minimum helix loss is desirable for minimizing tube noise figure, although the effect is not as pronounced as it is on intermodulation. Low loss and mechanical rigidity tend to be opposed, because additional support material adds to RF losses.

The helix structure, therefore, utilized beryllium oxide rods. After refining the processing, rods with round cross section had nearly as low loss as those with modified T-shape cross section. The round cross-section rods were used and were bonded to the helix wire with glass frit (glaze). This arrangement was considered the best compromise from the standpoint of intermodulation and noise performance, mechanical rigidity, and heat dissipation.

The helix sever loss (i.e., the loss deliberately added to the midsec-

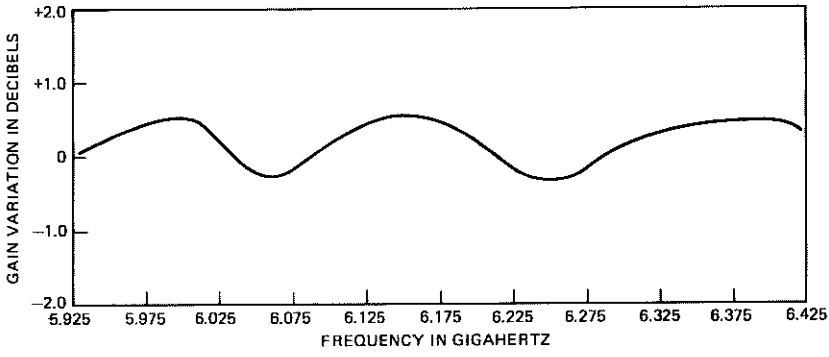



Fig. 4—Typical 473A traveling-wave-tube broadband gain variation.

tion of the helix to prevent self-oscillation and to provide good external RF matches) is a tantalum-aluminum sputtered film with Ta/Al ratio of 70/30 atomic percent.* This loss section is highly stable and not easily altered by bombardment with the 4000-volt electrons of the TWT beam.

The taper applied at the beginning and end of the sever loss is critical to achieving the desired broadband transmission shape. Figure 4 shows the typical broadband gain characteristic of the TWT. The required variation must be less than 1.2 dB peak to peak with no more than four cycles in the 500-MHz band.

### *3.4 Collector*

Construction similar to that employed in other Western Electric designs was used. The material is copper, of a length such that the modulated beam is well within the collector before significant beam spread occurs. This point is important from the standpoint of secondary emission. As noted previously, control of secondary electrons and reflected electrons is obtained by a cross-magnetic field over the collector.

### *3.5 Processing*

The design included careful attention to parts cleaning, degassing, and handling. Final exhaust was conducted with titanium gettering pumps and seal-off of each vacuum envelope was accomplished only after indicated internal pressure was in the region of $10^{-8}$ mm Hg.

## IV. TWT DEVELOPMENT PROBLEMS

Performance of the TWT was evaluated on a number of models. Problem areas that would likely result in low yield in manufacture were identified and changes were made.

*This film was originally developed for film resistors.[3]

### *4.1 Electron gun interelectrode leakage*

Initially, a gun structure using ceramic rod supports similar to that used in the Western Electric 461A[4] TWT was used. It was soon found that the higher voltages resulted in electrical breakdown across the support rods. A glass rod support structure was designed, leading to much improvement in the leakage problem. It was also necessary to maintain careful parts cleaning and processing. To keep electric fields low, adjacent parts, such as the gas getter and leads, were precisely placed and an initial voltage treatment of the gun was incorporated. The power supply was designed, as later described, to be insensitive to arcs that may occasionally occur. These actions resulted in attaining control over leakage and arcing problems.

### *4.2 Low-frequency noise*

The output of early tubes often showed noise at a few hertz and a few tenths of a decibel in amplitude. A rather elaborate computer-based study indicated that the noise was due to oscillation of ions trapped in potential depressions in the electron gun. The study also gave direction to gun design changes required. However, during the period of this study, the gas getter was changed from a nonevaporating type to an evaporating type; the change was actually made from the considerations of lowering cost and using less space within the envelope. It was found that the oscillation problem disappeared, presumably because the new getter reduced the pressure of the offending ion. Though attempts were made, the ion responsible for noise generation was never identified.

## V. TWT RESULTS

The gain and third-order intermodulation coefficient were characterized using a special three-tone intermodulation test set.[5] Several hundred tubes (including preproduction versions) were characterized. Typical results are shown in Figs. 5 and 6, which plot the gain and $M_{A+B-C}$ versus power and helix voltage, respectively. These measurements versus power were particularly important because of the requirement for constant third-order performance, which is necessary for the application of the predistorter to the AR6A transmitter.

Sample distributions of three major operating parameters from production—gain, third-order intermodulation, and noise figure plus gain sum—are shown in Figs. 7 through 9. Excellent control within the specification requirements is demonstrated. The remaining important parameter, fifth-order intermodulation, was measured only occasionally and was always well below the specification limits.

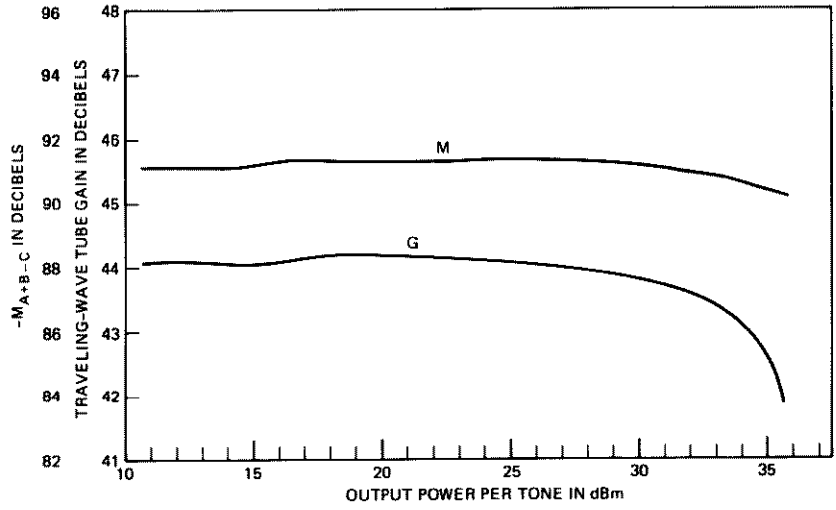



Fig. 5—Typical results of 472A TWT performance study showing third-order intermodulation coefficient and gain versus RF output.

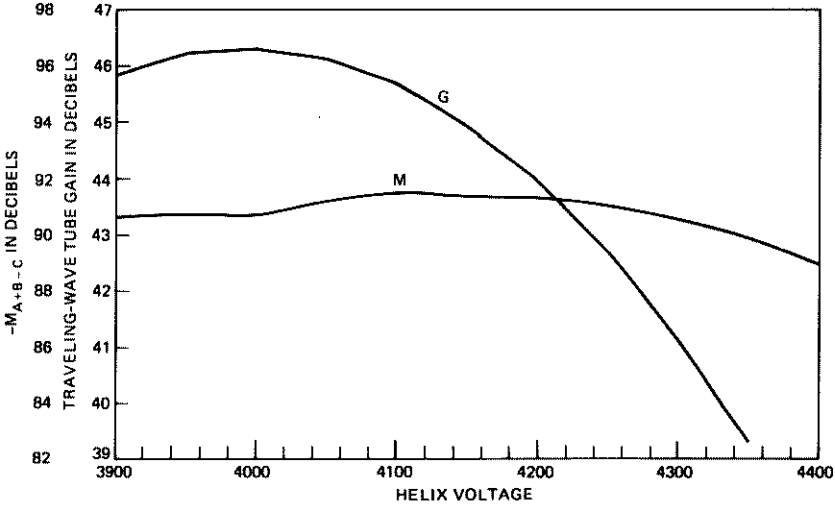



Fig. 6—Typical results of 472A TWT performance study showing third-order intermodulation coefficient and gain versus helix voltage.

## VI. REQUIREMENTS ON THE POWER SUPPLY

The requirements for the TWT power supply, with emphasis on those features that directly interact with the TWT, are summarized as follows:

1. Well-regulated heater and helix voltages and beam current are required to control the TWT gain and intermodulation noise.

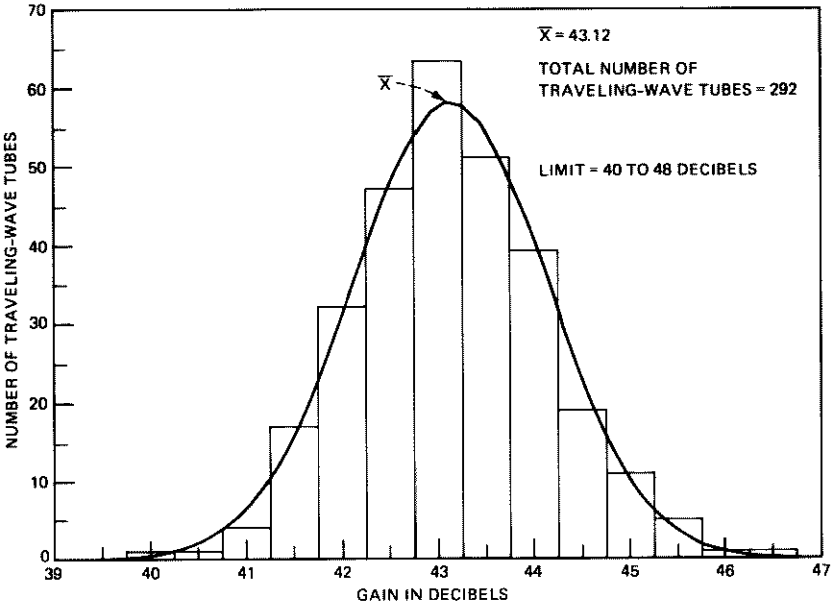



Fig. 7—473A TWT gain.

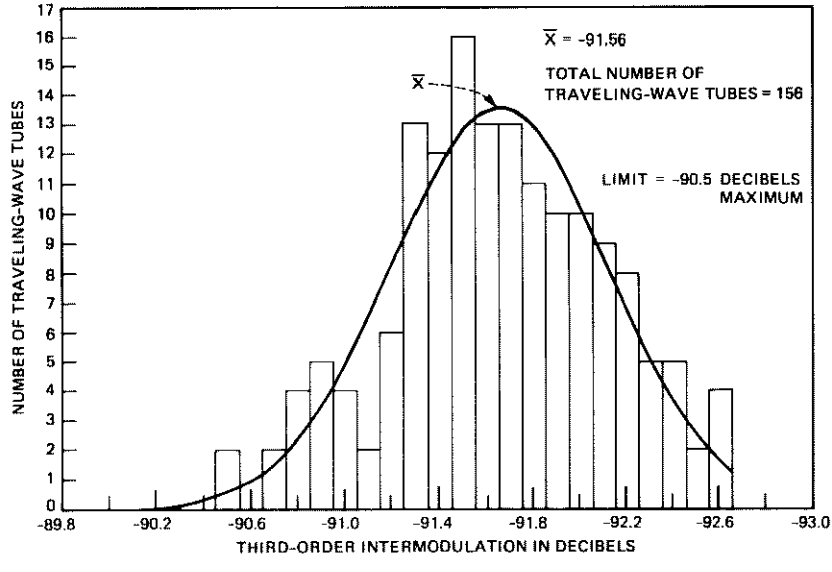



Fig. 8—473A TWT third-order intermodulation ($-M3$).

2. During turnon and turnoff, the electrode voltages must be sequenced so that damaging helix and anode currents do not occur.

3. Current detection circuits must protect the TWT and supply from tube internal arcs.

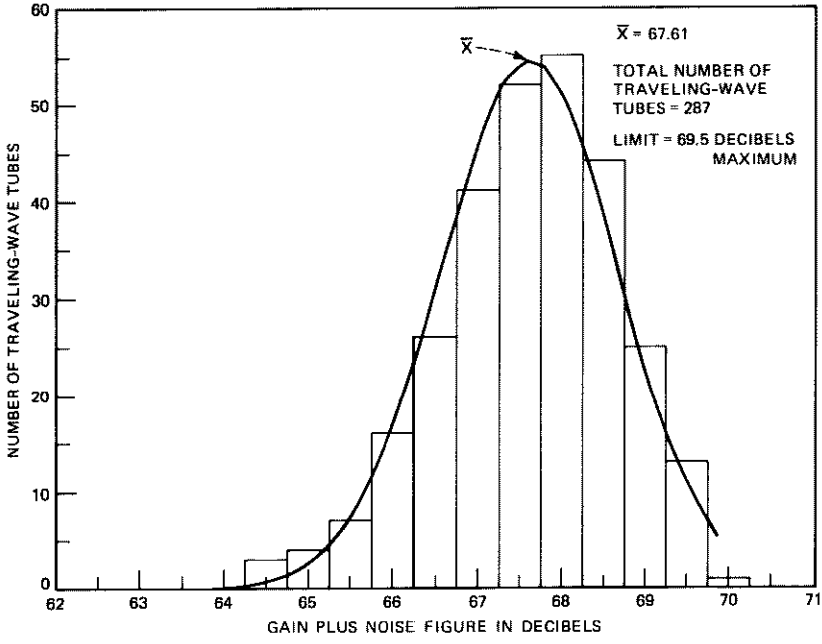



Fig. 9—473A TWT noise figure plus grain sum.

4. Side tones generated by the power supply and TWT combination must be more than 77 dB lower than a single-tone test signal.

## VII. POWER ARRANGEMENT AND REGULATION APPROACH

Figure 10 shows a block diagram of the power supply. The power supply and TWT parameters are summarized in Table I. The heater/Beam-Forming Electrode (BFE) dc-to-dc converter supplies 6.3 volts at 1.2 amperes to the TWT heater. The heater voltage is regulated to ±2 percent by controlling the duty cycle of the converter. During initial turnon, the current in the cold heater is limited to about 1.9 amperes by a limit circuit (not shown) within the converter.

The BFE supply is used to turn the TWT on and off electrically. When the TWT is running normally the BFE-to-cathode voltage is less than 1V (nominally 0V). During the time when turnon, turnoff, or internal TWT arcs occur, transistor Q1 is turned off. This causes the BFE potential to be approximately −450V, thus turning the TWT beam off. The BFE switching procedures are discussed in more detail in a later section.

Since the cathode is the common terminal of the TWT amplifier, electrode voltages are, therefore, normally specified with respect to the cathode (even though the collector is grounded). The collector

converter supplies +1800V. The 1800V output is controlled by the duty cycle of the converter to achieve ±2 percent regulation.

The anode converter and collector voltages are applied in series combination to provide from 2800 to 3500V to the anode. The beam current is monitored and the output voltage of the anode converter is regulated to control the beam current to 70 mA ±0.5 percent through control of the anode-to-collector voltage. The helix current ($I_w$) is also monitored. If $I_w$ is less than 1.75 mA, the anode converter regulates the beam current as discussed. Should $I_w$ reach 1.75 mA (corresponding to an unfocused TWT), the anode voltage ($E_c$) is controlled to hold $I_w$ at 1.75 mA. However, this causes the beam current to fall out of regulation, and when it is 2 percent lower than 70 mA, a trouble indication is generated. This indication occurs when a TWT cathode has become deactivated.

The helix converter and collector converter are applied in a series combination to provide from 4000 to 4550V to the helix. The helix voltage is adjustable using a resistive voltage divider. Since the TWT gain is extremely sensitive to changes in helix voltage, the helix converter includes a series regulator to control the helix voltage to ±0.5 percent. The converter being controlled supplies only the helix-to-ground voltage.

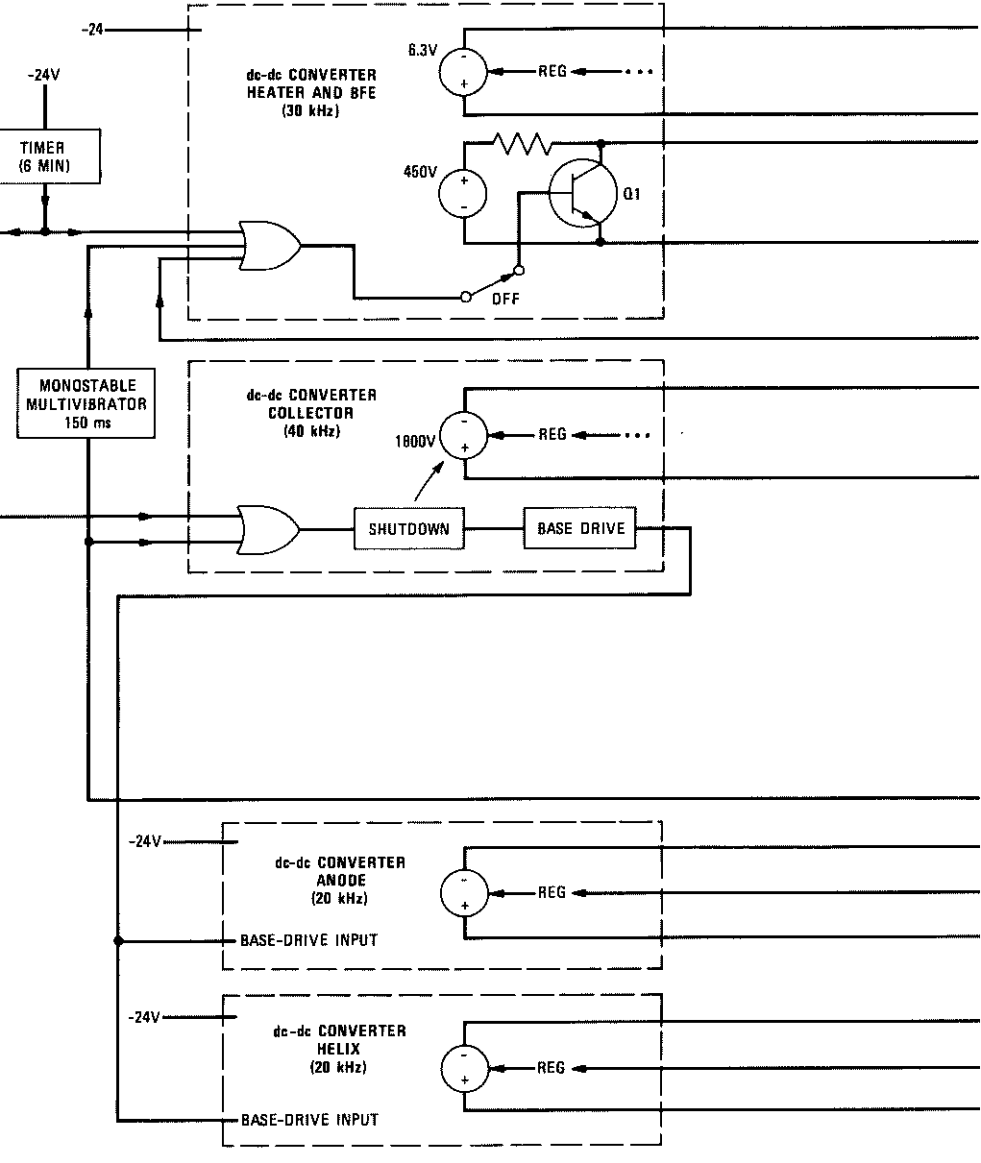
Both the helix converter and anode converter supply only a portion of their respective electrode voltages; the collector supply provides the remaining voltage. This arrangement is used for two reasons. First, the power transformer used in each converter is inexpensive since its secondary must withstand only the voltage it generates itself rather than approximately 1800V plus its self-generated voltage. Second, inexpensive monitoring circuitry may be used since it operates near ground (rather than cathode) potential.

## VIII. POWER SUPPLY OPERATION

### *8.1 Turnon and turnoff*

When −24V is connected to the input of the power supply through an external switch, the heater/BFE converter and the timer start immediately. The timer supplies a shutdown signal to the collector converter. Since the helix and anode converters depend on base-drive signals for their inverters from the collector converter, they are also shut down. The timer continues to supply a shutdown signal for a nominal 6 (minimum of 5) minutes to allow the TWT to warm up. During this preheat period the timer also supplies a signal that causes −450V to be applied to the BFE during warm-up to inhibit spurious emissions from the cathode.

At the end of the 6-minute warm-up period, the shutdown signal to


-24
-24V
TIMER
(6 MIN)
dc-dc CONVERTER
HEATER AND BFE
(30 kHz)
6.3V
REG
450V
Q1
OFF
MONOSTABLE
MULTIVIBRATOR
150 ms
dc-dc CONVERTER
COLLECTOR
(40 kHz)
1800V
REG
SHUTDOWN
BASE DRIVE
-24V
dc-dc CONVERTER
ANODE
(20 kHz)
REG
BASE-DRIVE INPUT
-24V
dc-dc CONVERTER
HELIX
(20 kHz)
REG
BASE-DRIVE INPUT

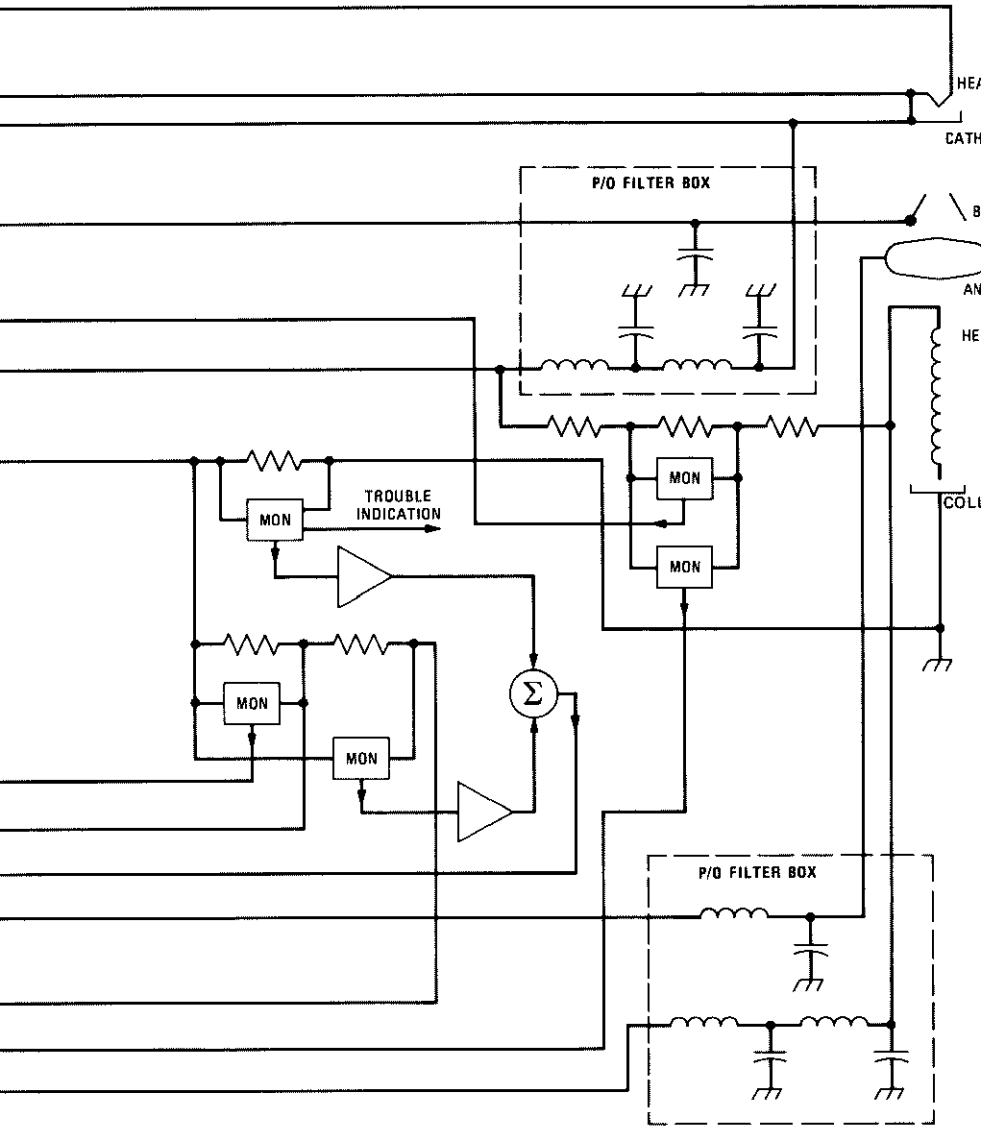



Fig. 10—Block diagram of the TWT power supply.

Table I—Parameters of TWT power supply requirements

| | |
|---|---|
| Heater current | 1.2 amps maximum |
| Heater voltage | 6.3 volts ±2 percent |
| BFE voltage | 0 |
| Anode current | 0.5 mA maximum |
| Anode voltage | As required for collector current |
| Helix current | 1.8 mA maximum |
| Helix voltage | Adjusted for 5 percent above synchronous operation; regulated to ±0.5 percent after adjustment |
| Collector current | 70 mA |
| Collector voltage | 1800 volts ±2 percent |

the collector converter is removed. The collector, helix, and anode converters then start. The time constants of each converter are designed so that these electrode voltages reach their respective final values in the order named above. The BFE voltage is not switched to zero until the collector converter has started, and the helix voltage has reached 4000V. This arrangement prevents excessive helix current while the TWT is turning on.

The external −24V switch is used for normal turnoff. On removal of −24V, the electrode voltages decay, the timer resets, and the BFE supply loses its continuous source of energy. As soon as the helix voltage reaches 4000V, the remaining BFE voltage is applied. There is still enough stored energy in the BFE supply's internal capacitor to inhibit the beam until the other electrode voltages fall safely below levels that would otherwise cause damaging values of helix and anode currents.

### *8.2 Arc detection*

On occasion a TWT arcs. Any combination of electrodes might be involved in the arc, but the direct or indirect result is almost always a large value of helix and/or anode current. When the helix plus anode current exceeds 10 mA, a shutdown signal is applied to the collector converter. Of course the helix and anode converters also shut down, since they depend on the collector converter for base drive signals. Decay of these voltages then causes the arch to extinguish. At this time the collector converter is again turned on and the tube voltages rise toward their respective normal values.

During the arc and subsequent shutdown, the RF output of the TWT is less than normal. When the TWT is turning back on, the process is controlled so that the AR6A protection switching system can operate properly. The RF output is held off by applying −450V to the BFE for at least 150 milliseconds, thus allowing AR6A Radio System transients, which may occur during the arcing process, to subside completely before reapplying RF signal to the system.

The arc detection arrangement described above not only protects

the TWT by turning it off for the prescribed time period, but also has proved to be successful in protecting the supply. One particularly arc-prone TWT was life tested for 14 days. A total of 110 arcs occurred, but no supply damage resulted. The TWT parameters gradually fell out of specified limits, but the TWT performance did not change significantly as the result of any one arc.

### *8.3 Tones*

The power-supply ripple requirements are not directly specified. Instead, they are implied by the requirement that the TWT sidetones must be more than 77 dB below the test tone.

Output filters are used in each converter. Additional Inductance Capacitance (LC) sections are mounted in a shielded steel box. A shielded cable connects the output of the filter box to the TWT.

The dominant sidetones occur at 20 kHz, the frequency of the helix and anode converters, and at 30 kHz, the switching frequency of the heater/BFE converter. These sidetones are caused by interwinding capacitance current that is generated in the power transformer in the helix, anode, and BFE converters. These currents are bypassed where appropriate, but they are still the dominant tone sources. In addition, special filtering is provided for transistor Q1 to prevent it from being turned off at a 30-kHz rate by interwinding capacitance current.

## IX. AMPLIFIER OPERATION

The TWT and power supply are electrically connected by a shielded cable. At the TWT end, the cable is potted into the end of the tube. The cable connector mates to the power supply connector, which is accessible through a slot cut into the top cover of the power supply. A safety arrangement prevents power-supply operation when the TWT connector is disconnected.

The RF signal connections to the TWT are via an SMA coaxial connector at the input and a reduced height WR159 waveguide at the output. Adjustable shorts are provided for RF matching at the input and output.

Meters on the front panel of the power supply are supplied to measure the helix and collector currents. On the front panel, a helix-voltage adjustment (labeled RF gain) is provided. This adjustment is made during alignment of the amplifier so that the TWT operates at a helix voltage higher than the synchronous value, thus providing gain that is approximately 2 dB lower than that achievable at the synchronous helix-voltage value.

All amplifiers were tested in a bay environment. The gain-versus-frequency characteristic (transmission characteristic) of the amplifier embedded in the AR6A transmitter was measured for each amplifier

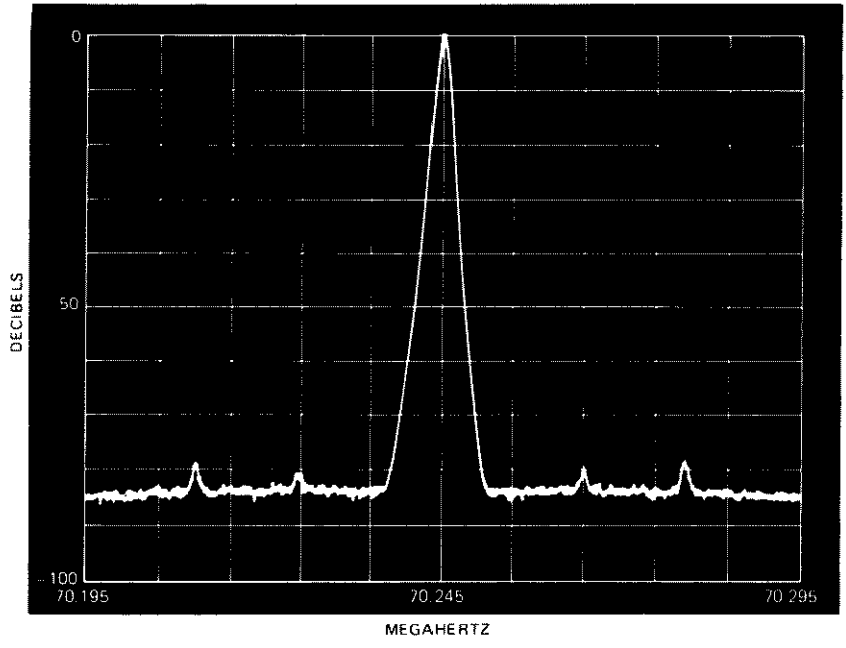



Fig. 11—Test tone with spurious modulation.

shipped for field application. The results of these tests were interpreted by subtracting out frequency characteristics attributable to other transmitter components. Such analysis shows that the average transmission characteristic for 100 amplifiers (with samples at every channel) has a bow of approximately 0.1 dB across the 30-MHz channel. (Note that any residual slope of the transmitter is equalized.) This is approximately what one would expect, considering the gain variation shown in Fig. 4.

Also, a test was made to check for a spurious modulation caused by very low ripple voltages at frequencies that are multiples of the 30- and 20-kHz switching rates of the power supply. A test tone was applied to the amplifier and sidetones were measured. The sidetones were required to be at least 77 dB below the level of the test tone. A graph extracted from a spectrum analyzer photograph shows a typical result (Fig. 11).

## X. ACKNOWLEDGMENTS

J. R. Meszar and C. A. Waldron were highly instrumental in the development of the power supply and TWT, respectively.

## REFERENCES


1. G. L. Heiter, "Characterization of Nonlinearities in Microwave Devices and Systems," IEEE Trans. Microw. Theory Tech., *21*, No. 12 (December 1973), pp. 797–805.
2. W. R. Curtice, "First-Order Non-Linear Effects in Traveling-Wave Tubes," Proc. IEE, *114* (August 1967), pp. 1048–50.
3. F. Huber and D. Jaffe, "High Stability Co-sputtered Ta-50 At. % Al Alloy Film Resistors," IEEE Trans. Parts, Hybrids, Packag., *8* (June 1972), pp. 4–9.
4. C. E. Bradford, "The TWT Amplifier for the Microwave Transmitter," B.S.T.J., *47*, No. 7 (September 1968), pp. 1379–95.
5. E. F. Cook, private communication, 1974.


## AUTHORS

**James F. Balicki,** B.E.E., 1960, University of Detroit; M.E.E., 1962, New York University; Bell Laboratories, 1960–1970; AT&T, 1970–1972; Bell Laboratories, 1972—. Mr. Balicki has planned power systems and developed power supplies for station, PBX, special customer, carrier, coaxial, multiplex, and microwave systems. Presently he is leading a team effort in the areas of automatic trimming testing and diagnostics for thick-film power modules. Member, Eta Kappa Nu, Tau Beta Pi.

**Edmund F. Cook,** B.S.M.A., 1965, Fairleigh Dickinson University; Bell Laboratories, 1954—. At Bell Laboratories, Mr. Cook was first engaged in the development of radio telemetry and electronic switching systems. Since 1971 he has participated in the development of the AR6A Microwave Radio System with particular emphasis on modulator and traveling-wave-tube studies.

**Richard C. Heidt,** A.S.E.E., 1956, Arlington State Junior College; B.S.E.E., 1959, Southern Methodist University; M.E.E., 1963, New York University; Bell Laboratories, 1961—. Mr. Heidt was first engaged in microwave transmission circuits and antennas. Since 1969 he has been involved in the design of transmitters for the single-sideband microwave radio system. He is currently Supervisor of a group developing digital microwave radio systems. Member, IEEE, Eta Kappa Nu, Sigma Tau, Kappa Mu Epsilon.

**V. E. Rutter,** B.E.E., 1951, Ohio State University; Bell Laboratories, 1951—. Mr. Rutter has been involved in studies concerning military equipment reliability and development of *Picturephone** and traveling-wave tubes. He has recently been involved in semiconductor development and currently supervises the submarine-cable integrated-circuit development effort.

* Registered service mark of AT&T.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

*The AR6A Single-Sideband Microwave Radio System:*

# Predistortion for the Traveling-Wave-Tube Amplifier

By R. P. HECKEN,* R. C. HEIDT,* and D. E. SANFORD*

(Manuscript received February 7, 1983)

To satisfy the single-sideband amplitude-modulated system noise requirements, the transmitter must be very linear. Predistortion reduces intermodulation distortion of the transmitter to acceptable levels. This paper deals with the design considerations, the model of the transmitter used to analyze third-order intermodulation products, and the methods applied to align the predistorter-transmitter combination for effective intermodulation reduction.

## I. INTRODUCTION

The frequency stability required for single-sideband amplitude-modulated (SSB-AM)† system operation results in the potential for in-phase addition of intermodulation noise from repeater to repeater. Traveling-Wave Tubes (TWTs) offer the best linearity (and, consequently, the lowest intermodulation distortion) available for currently existing 6-GHz power amplifiers. However, linearity must be improved to satisfy system noise requirements.

The concept of predistortion to reduce intermodulation distortion

*Bell Laboratories.

†Acronyms and abbreviations used in the text and figures of this paper are defined at the back of this *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

had originally been explored extensively at 4 GHz[1] and was found considerably less complex than feedforward.[2] In particular, predistortion at intermediate frequencies appeared attractive because of its simplicity and economy. Operating at intermediate frequency (IF), the predistorter is placed ahead of the up converter and is, therefore, able to cancel intermodulation products generated in the converter and the traveling-wave tube. Clearly, the latter is the major contributor to nonlinear distortions and will be treated accordingly.

Extensive measurements have shown that TWTs, if operated at low radio frequency (RF) power levels, i.e., in a domain of weak nonlinearity, behave essentially as first- and third-order devices and produce odd-order intermodulation products within the narrow frequency band of interest. In the AR6A* System the predistorter is designed to cancel distortions in this power region, and it is important to distinguish this mode of operation from other methods where predistortion is analyzed in a region of strong nonlinearity.[3,4]

In the following section we will proceed with a simple analysis of nonlinearities of the transmitter, which is based on a model that proved to be very accurate within the useful power range of the AR6A System. Section III will deal with practical design considerations, and in Section IV we will conclude with the results.

## II. NARROWBAND DISTORTION ANALYSIS

### *2.1 Nonlinear model of the traveling-wave tube*

Measurements on a large number of traveling-wave tubes having a saturated power-handling capability from 5 to 20W showed that even at very low RF power levels (<1W), the simple memory-free polynomial model describing gain nonlinearity of the TWT is inadequate. Two additional effects must be included in the analysis to achieve substantial intermodulation reduction of 20 dB or more: the amplitude and phase distortions due to reactive circuit elements and the amplitude modulated to phase modulated (AM-PM) conversion effect. A model of the TWT that was found very useful is shown in Fig. 1. This model characterizes a special class of nonlinear systems in which a memory-free, nonlinear two-port network is embedded between two linear passive two-port networks having the impulse responses $h_A(t)$ and $h_B(t)$, respectively. For narrowband systems the nonlinear two-port is well modeled by a finite odd-order power series in two variables with constant coefficients[5,6]

$$w = b_1' u + b_3' u^3 + b_5' u^5 + \cdots + b_1'' \dot{u} + b_3'' \dot{u}^3 + b_5'' \dot{u}^5 + \cdots, \quad (1)$$

*Amplitude Modulation Radio at 6 GHz for the initial (A) version.

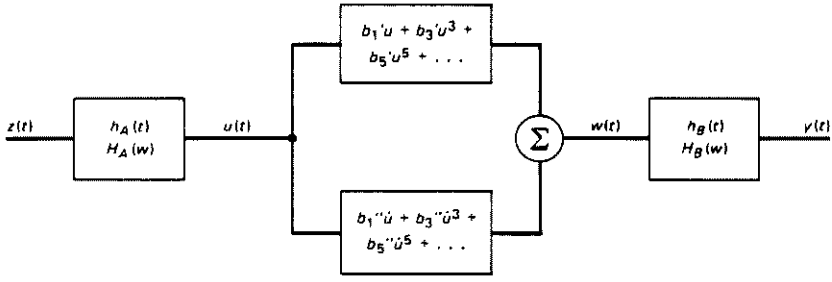



Fig. 1—Approximate model of a traveling-wave-tube amplifier with weak nonlinearities.

where $\dot{u} = du/dt$. The partial series in powers of $\dot{u}$ accounts for AM-PM conversion. An important aspect of this TWT model is the fact that the two interfaces between the three two-ports are, in general, not accessible.

For our analysis, we will limit the power series to first- and third-order terms. Applying the Fourier transformation to eq. (1), we obtain the complex nonlinear input-output relation

$$W = B_1 U + B_3 U^{3*} + \cdots, \tag{2}$$

with the notation $U^{3*}$ indicating the repeated convolution $U*U*U$. The complex coefficients are given by

$$B_1 = b_1' + j\omega_c b_1'', \tag{3a}$$

and

$$B_3 = b_3' - j\omega_c^3 b_3''. \tag{3b}$$

The narrowband assumption manifests itself in eq. (3), in that the frequency variable, $\omega$, has been replaced by the constant channel frequency, $\omega_c$, while performing the convolutions. With the input-output relations for the nonlinear portion of the TWT established, we can write the overall transfer function for the TWT model. Let $Z$ and $Y$ be the Fourier transforms of the input and output signals; then

$$Y = H_B B_1 H_Z Z + H_B B_3 (H_A Z)^{3*} + \cdots, \tag{4}$$

where $H_A$ and $H_B$ are the transfer functions of the input and output networks, respectively. Before we proceed with the distortion analysis of the transmitter, we need first to discuss some properties of the up converter.

### *2.2 The Schottky modulator*

Comparative measurements on Varactor Diode Converters (VDC) and Schottky Diode Converters (SDC) have shown the superiority of

the latter when considering its nonlinear distortion characteristics versus RF output power. These measurements revealed that the SDC, when pumped with high local oscillator power, behaves like a third-order device over a wide range of RF signal levels. This property is indeed useful because it allows the predistorter, tailored to compensate for third-order distortions, to improve the converter distortions as well. Moreover, because of its conversion loss, in contrast to the conversion gain of a VDC, a higher level of distortions can be tolerated in an SDC. This is so because, for a given performance, the TWT essentially limits the RF output power. To illustrate this, we compute the power of a three-tone, third-order product generated in the converter as it appears at the output of the TWT. If we refer to Fig. 2 this power is

$$P_{A+B-C} = M_{A+B-C} + 3P_A + G \qquad \text{(dBm)}, \tag{5}$$

where $M_{A+B-C}$ is the particular intermodulation coefficient[7] in decibels of the converter, $P_A$ is the power in dBm of each of the equal-level, three-tones at the output of the converter, and $G$ is the overall gain of the bandpass filter (BPF) and TWT. Since the single-tone power at the tube output is

$$P_{AO} = P_A + G \qquad \text{(dBm)}, \tag{6}$$

we find that

$$P_{A+B-C} = M_{A+B-C} - 2G + 3P_{AO} \qquad \text{(dBm)}. \tag{7}$$

Thus, for a specified performance ($P_{AO}$ fixed) the device distortion $M_{A+B-C}$ can be reduced by 2 dB for each decibel increase in TWT gain. Since the difference in conversion loss and gain can be as high as 12 dB, the Schottky diode converter may have an $M_{A+B-C}$ that could be 24 dB worse than that of a varactor diode converter, yet may still yield the same distortion performance. Typically, the SDC has an $M_{A+B-C}$ of about −25 dB compared to −40 dB of the VDC, which clearly shows the disadvantage of the varactor diode converter.

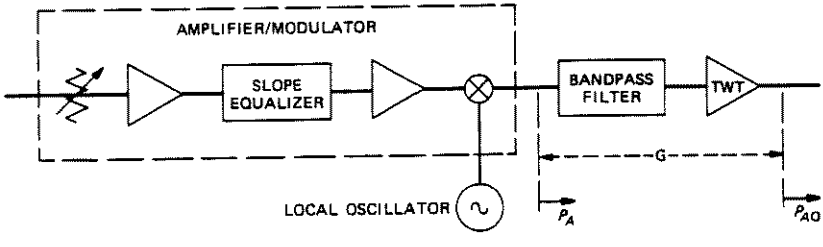



Fig. 2—Simplified block diagram of the AR6A amplifier/modulator with bandpass filter and traveling-wave tube.

It is useful to compare the distortion power generated by the converter to that of the TWT. At a total signal output power of +27 dBm for three equal-level tones, the single-tone power $P_{AO}$ is approximately +22 dBm. With $G = 45$ dB and $M_{A+B-C} = -25$ dB from eq. (7), $P_{A+B-C} = -49$ dBm. A TWT having an $M_{A+B-C} = -91$ dB generates intermodulation power of the same product at a level of −25 dBm. Obviously, one can ordinarily neglect the contribution of the converter since it increases the total intermodulation power by only 0.53 dB. However, if the predistorter can reduce distortions by more than 20 dB, the converter will contribute significantly to the resulting intermodulation noise unless the predistorter also reduces its nonlinearities. In spite of this, to keep our discussion simple we will not include the converter-generated distortions in our analysis.

### *2.3 Generalized nonlinear transmitter model*

The mathematical treatment of the nonlinearities of the complete transmitter is complicated, even when we neglect the converter nonlinearities, because of the operation in two frequency domains. As is shown in the appendix, the frequency translation from IF to RF requires each sideband to be considered separately. We will point out the differences where appropriate and develop the analysis for the upper sideband only. To arrive at a tractable model of the complete transmitter, we actually include the converter, bandpass filter, etc., in the TWT model as described in the appendix. Frequency characteristics at RF are then referred to the IF band, allowing the predistorter to be directly connected to this TWT model. The cascade of predistorter with IF amplifiers and the converter with bandpass filter, TWT, etc., is depicted in Fig. 3, where we used a nonlinear model of the predistorter that is similar to that of the TWT. Thus, we assume that the predistorter has the transfer function

$$Z = G_B A_1 G_A X + G_B A_3 (G_A X)^{3^*}, \tag{8}$$

which leads to the following overall transfer function for the upper sideband (USB):

$$Y_{\text{USB}} = H_B B_1 H_A G_B A_1 G_A X + H_B B_1 H_A G_B A_3 (G_A X)^{3^*} + H_B B_3 (H_A G_B A_1 G_A X)^{3^*} + \cdots. \tag{9}$$

In eq. (9) all terms requiring higher than third-order convolutions have been neglected. Two observations can be made immediately: the distortion terms depend directly or indirectly on the product $G_A X$ as part of the convolution process. Thus, the frequency characteristic of the input network of the predistorter will have no effect on the minimization of these terms. Also, $H_B$ is a factor in both distortion

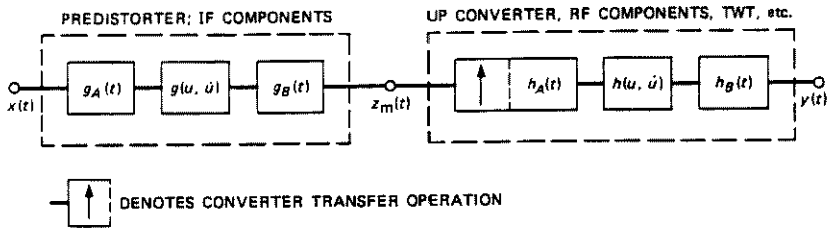



Fig. 3—A generalized model of the AR6A transmitter.

terms, indicating that the frequency characteristic of the TWT output network also has no effect on the intermodulation cancellation. For the purpose of this analysis we can, therefore, set $G_A = 1$ and $H_B = 1$. With these simplifications the third-order distortion term becomes, in the case of the upper sideband,

$$D_3 = B_1 H_A G_B A_3 (X)^{3^*} + B_3 (H_A G_B A_1 X)^{3^*} \qquad \text{(USB)}. \tag{10}$$

It can be shown that in order for $D_3$ to at least vanish over the frequency band where $X$ exists, it is necessary for the product $H_A G_B A_1$ to have constant magnitude and constant delay as function of frequency within that band, i.e., the condition must be met

$$H_A G_B A_1 = C e^{-j\omega\tau}, \tag{11}$$

where

$$C = |H_A G_B A_1| \qquad \text{and} \qquad \omega\tau = \sphericalangle(H_A G_B A_1).$$

This yields for $D_3$

$$D_3 = \left( B_1 \cdot \frac{A_3}{A_1} C e^{-j\omega\tau} + B_3 C^3 e^{-j\omega\tau} \right) X^{3^*}.$$

Hence, frequency-independent cancellation of the third-order distortions at the upper sideband is possible if, in addition to eq. (11), the following condition is fulfilled:

$$\frac{A_3}{A_1} = -\frac{B_3}{B_1} \cdot |H_A G_B A_1|^2 \qquad \text{(USB)}. \tag{12a}$$

The corresponding condition for the lower sideband (LSB) is

$$\frac{A_3}{A_1} = -\frac{\check{B}_3}{\check{B}_1} \cdot |H_A G_B A_1|^2 \qquad \text{(LSB)}, \tag{12b}$$

where $\check{B}$ is the complex conjugate of $B$. Thus, the required complex ratios $A_3/A_1$ for the LSB and USB are conjugate to each other. The

fundamental relationships for wideband distortion cancellation are described in eqs. (11) and (12). Therefore, design as well as alignment procedures are essentially based on these criteria. Obviously, eq. (11) requires amplitude and delay equalization for all components in the signal path between the nonlinear element of the predistorter and the nonlinear region of the TWT, unless these components have inherently flat response and constant delay. In the AR6A transmitter, however, a certain amount of slope equalization was required to achieve intermodulation improvement by more than 30 dB. This equalizer is integrated into the modulator/amplifier unit.

From eqs. (12a) and (12b) we derive the required magnitude and phase of the distortions to be generated in the predistorter. The magnitude is practically independent of the sideband used since the TWT generates approximately the same intermodulation level at both frequencies. Therefore,

$$|A_3| = \left| \frac{B_3}{B_1} \right| \cdot |A_1|^3 \cdot |H_A G_B|^2. \tag{13}$$

It is obvious that for stable operation of the predistorted transmitter, the magnitude of the transfer coefficients, $|A_1|$ of the predistorter and $|H_A G_B|$ of the interconnecting networks (which includes the up converter), must be very well controlled and remain stable over time and temperature. The same is true for the gain of the TWT, represented by $|B_1|$ in eq. (13).

The phase condition for the upper sideband is

$$(\sphericalangle A_3 - \sphericalangle A_1) = \pi + (\sphericalangle B_3 - \sphericalangle B_1) \qquad \text{(USB)}, \tag{14a}$$

and for the lower sideband is

$$(\sphericalangle A_3 - \sphericalangle A_1) = \pi - (\sphericalangle B_3 - \sphericalangle B_1) \qquad \text{(LSB)}. \tag{14b}$$

Thus, means have to be provided in the predistorter to achieve both phases since either the USB or LSB may be deployed, depending on the channel, in the AR6A System.

### *2.4 Interaction distortions*

In the derivation of the overall nonlinear transfer function for the transmitter, terms involving convolutions of higher than third order have been neglected. It is worthwhile, however, to discuss the next higher-order term, which is found to be

$$D_5 = 3B_3(H_A G_B A_1 X)^{2^*} * (H_A G_B A_3)(X)^{3^*}. \tag{15}$$

Assuming that the transmitter is aligned properly, such that conditions in eqs. (11) and (12) are fulfilled, the magnitude of $D_5$ becomes

$$|D_5| = 3|H_A G_B A_1|^5 \cdot \left| \frac{B_3^2}{B_1} \right| \cdot |X^{5^*}|. \tag{16}$$

We see from the above expressions that this fifth-order distortion is strictly due to interaction between the third-order terms of the predistorter and TWT and that the magnitude is very sensitive to the first- and third-order coefficients of the TWT. For $|D_5|$ to remain small, $|B_1|$ (i.e., TWT gain) should be large and $|B_3|$ of the device must be small at the outset.

### *2.5 Residual third-order distortions*

In practice it is not possible to fulfill the conditions given by eqs. (11) and (12) at all frequencies. It becomes necessary, therefore, to estimate the amount of residual distortions after imperfect predistortion. In particular, the effect of deviations from the ideal condition in eq. (11) is important because of the strong sensitivity of $D_3$, defined in eq. (10), to the transfer function $T = H_A G_B A_1$. We will, therefore, attempt to analyze $|D_3|$ as a function of small deviations $\delta_c$ and $\delta_\alpha$ of $|T|$ and $\alpha = \measuredangle T$, respectively. Thus, we express $T$ as

$$T = (C + \delta_c) \cdot e^{-j(\omega\tau + \delta_\alpha)}, \tag{17a}$$

which is approximately equal to

$$T \simeq Ce^{-j\omega\tau} + (\delta_c/C - j\delta_\alpha)Ce^{-j\omega\tau}$$

or

$$T \simeq T_o + E \cdot T_o. \tag{17b}$$

Obviously, $T_o = C \cdot e^{-j\omega\tau}$ represents the ideal transfer function, which we will assume to fulfill the condition in eq. (11). The error term $E \cdot T_o$ is proportional to the complex quantity

$$E = \frac{\delta_c}{C} - j\delta_\alpha, \tag{18}$$

and is a function of frequency. Introducing eq. (17b) into eq. (10) we obtain

$$D_3 = B_1 \frac{A_3}{A_1} (T_o + ET_o)(X)^{3^*} + B_3(T_o X + ET_o X)^{3^*}. \tag{19}$$

Performing the convolutions on the last term, eq. (19) can be expressed as

$$D_3 = B_1 \frac{A_3}{A_1} T_o(X)^{3^*} + B_3 C^2 T_o(X)^{3^*} + D_{3R}. \tag{20}$$

With the predistorter active and conditions in eqs. (11) and (12)

being met, the first two terms cancel. The residual distortions are then approximately given by

$$D_{3R} = B_1 \frac{A_3}{A_1} ET_oX^{3^*} + 3B_3(T_oX)^{2^*} * (ET_oX) + \cdots, \qquad (21)$$

where terms of higher order in $(ET_oX)$ have been neglected. By introducing eq. (12) and using the identity $(T_oX)^{2^*} = CT_o(X)^{2^*}$ we obtain

$$D_{3R} = -B_3C^2T_o[EX^{3^*} - 3X^{2^*} * (Ee^{-j\omega\tau}X)]. \qquad (22)$$

The power spectrum of $D_{3R}$ for given deviations $E$ can be computed using eq. (22). Unfortunately, this involves, in general, complex numerical calculations and results do not easily provide further insight. Instead, we have used eq. (22) to arrive at an upper bound for the magnitude of $D_{3R}$. By using the inequality

$$|D_{3R}| \leq |B_3| \cdot C^3[|E| \cdot |X^{3^*}| + 3|X^{2^*}| * |E \cdot X|] \qquad (23)$$

we have examined each term within the brackets individually for two typical, yet simplified, error functions $E$.

### *2.5.1 Case 1—Out-of-band residual distortions*

If we assume that $|E|$ and $|X|$ are as illustrated in Fig. 4, it becomes obvious that only the first term in eq. (23) contributes to residual distortions, because the product $E \cdot X$ vanishes everywhere. Typically, the bandpass filter between converter and TWT produces this type of residual out-of-band third-order distortion. In the AR6A System, their

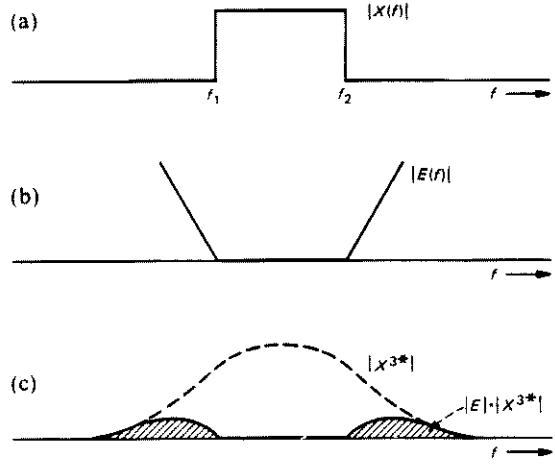



Fig. 4—Out-of-band distortions with perfect in-band predistortion: (a) input spectrum; (b) deviation from flatness; (c) distortion spectrum.

level is sufficiently small to cause no interference with other radio channels.

### 2.5.2 *Case 2—In-band residual third-order distortions*

This important case deals with parabolic in-band deviations $E$, as illustrated in Fig. 5. Satisfactory, but imperfect, tuning of the RF bandpass filter or insufficient flatness of the IF driver amplifier in the converter are primary causes for parabolic-type residual deviations. For simplicity, we approximate these by a triangular distribution, which causes the in-band residual distortions as shown shaded in Figs. 5c and 5d. It is important to recognize that the level of distortions due to the second term of eq. (23) varies in direct proportion to the fractional bandwidth $\Delta f$ over which $|E|$ assumes a substantial magnitude. This is in contrast to the contribution stemming from the first term, the maximum level of which is only proportional to the magnitude of $E$. As a rule of thumb, the first term dominates if

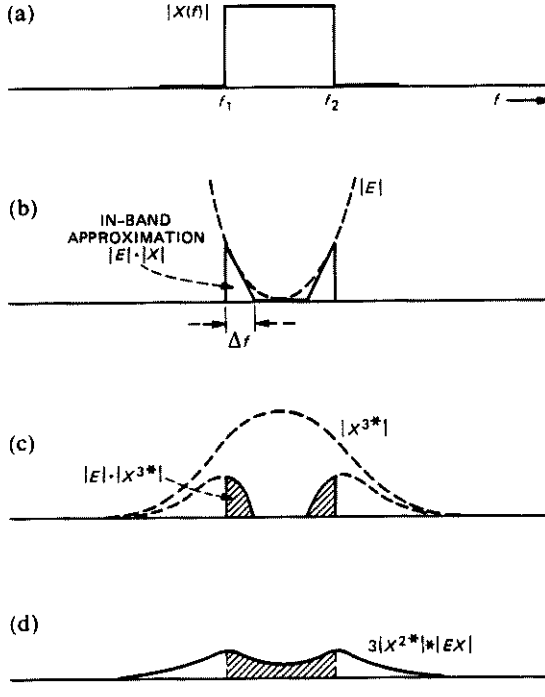



Fig. 5—Residual in-band third-order distortion spectrum: (a) input spectrum; (b) deviation from flatness; (c) residual distortion spectrum, first term, eq. (23); (d) residual distortion spectrum, second term, eq. (23).

$$3\Delta f < \tfrac{1}{4} \cdot BW,$$

where $BW$ is the bandwidth of the input spectrum, i.e., $BW = f_2 - f_1$. In the AR6A System a special slope equalizer has been incorporated into the driver amplifier of the up converter to ensure that essentially only parabolic residual deviations remain at the very edges of the signal band, thus, keeping $\Delta f$ as small as practicable.

The maximum tolerable amount of these deviations has been derived from eq. (23) by requiring that the ratio $|D_{3R}/D_{3o}|$ is below a specified level, where $|D_{3o}| = |B_3|C^3|X^{3^*}|$ represents the unpredistorted intermodulation spectrum within the message band. Thus, this ratio is a direct measure of distortion reduction. The predistortion improvement (PDI) in decibels is given by

$$PDI = -20 \log|D_{3R}/D_{3o}| \qquad \text{(dB)}. \tag{24}$$

If only the first term in eq. (23) contributes to residual distortions, the predistortion improvement becomes

$$PDI = -10 \log\left[\left(\frac{\delta_c}{C}\right)^2 + \delta_x^2\right] \qquad \text{(dB)}.$$

Figure 6 shows plots of constant PDI values as a function of phase deviation (in degrees) and flatness, defined as $20 \log(1 \pm \delta_c/C)$. We see that to achieve a PDI $\geq 40$ dB, the flatness must be better than $\pm 0.06$ dB and phase linearity is required to be within $\pm 0.4$ degree, provided both deviations contribute equal amounts.

Deviations in other coefficients describing the transmitter model, in particular $A_1$, $A_3$ and $B_1$, $B_3$, have been treated similarly. Except for $A_1$, their effect is less complicated because they are not convolved. It can be shown that as long as the residual distortions are caused only by nonconvoluted terms, the total amount of residual distortions is bound by the inequality

$$\left|\frac{D_{3R}}{D_{3o}}\right| \leq \sqrt{\left(\sum_{i=1}^{N} |m_i|\right)^2 + \left(\sum_{i=1}^{N} |\Delta\psi_i|\right)^2}, \tag{25}$$

where $m_i$ represents the deviation from flat magnitude and $\Delta\psi_i$ the deviation from phase linearity of the $i$th contributing coefficient. This upper bound estimate given by eq. (25) is very conservative because it is based on the assumption that at any frequency all deviations of the different component parts are causing in-phase addition. This is, in general, not true and a more useful estimate based on statistical addition is

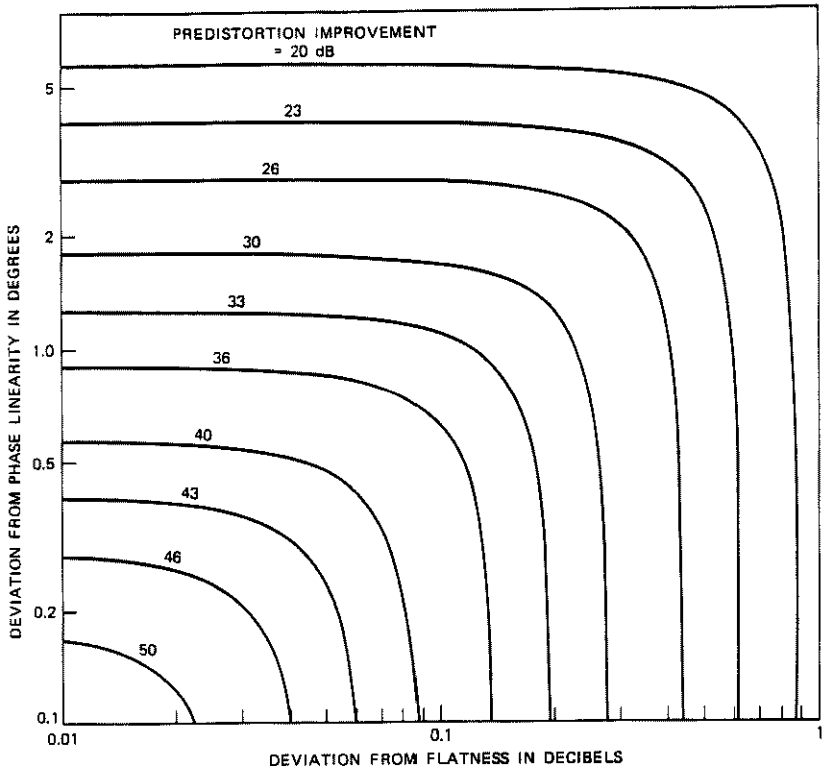



Fig. 6—Contours of constant predistortion improvement.

$$\left| \frac{D_{3R}}{D_{3o}} \right| \equiv \sqrt{\sum_{i=1}^{N} |m_i|^2 + \sum_{i=1}^{N} |\Delta\psi_i|^2}. \tag{26}$$

For the AR6A transmitter, under nominal testing conditions, the objective was to achieve a PDI of at least 30 dB over the full 30-MHz band. Allowing an additional margin of 3 dB and allocating equal deviations to each of the five possible contributors in the transmitter model (namely, $A_1$, $A_3$, $B_1$, $B_3$, and the product $H_A G_B$) results in a flatness requirement within ±0.06 dB and phase linearity requirement within ±0.4 degree. (Using the conservative upper bound would yield ±0.03 dB and ±0.18 degree as requirements on the model coefficients.) Since the coefficients $A_1$ and $A_3$ are only representative of the various components in the predistorter, its actual circuit requirements have been set at nominally ±0.02 dB for flatness. The corresponding requirements for phase linearity are then automatically met because of the minimum phase property of the circuits used.

## III. PRACTICAL DESIGN CONSIDERATIONS

### *3.1 Components*

The predistorter comprises components operating over the 30-MHz bandwidth centered at 74.1 MHz. The predistorter is described in the block diagram shown in Fig. 7. The key component is the nonlinear generator referred to as the cuber.[8] It consists of four back diodes arranged in a bridge configuration so that signals incident at the input of the network are attenuated by approximately 60 dB at the output. The amplitude of each third-order complementary distortion term at the cuber output is related to the amplitudes of the input signals at frequencies A, B, and C by

$$P_{A+B-C} = 46.6 + P_A + P_B + P_C \quad \text{(dBm)}. \tag{27}$$

The driver amplifier at the input to the predistorter provides gain so that the input signal to the cuber is at the optimum level. The optimum is dictated by choosing the drive such that the cuber does not generate fifth-order distortion terms that would exceed the level assumed in eq. (9). The distortion amplifier provides gain so that the complementary distortion terms generated by the cuber have the required amplitude. The gain in the distortion amplifier must be limited because of the addition of thermal noise back into the main path. The distortion adjustment potentiometer in the output coupler is used to fine tune the amplitude to provide the exact value of $|A_3|$ required for each individual transmitter.

The phase of the third-order complementary distortion terms generated in the cuber is established by the phase resolver network.[9] This network is designed to provide a constant-phase linear-phase difference between the two output ports and has the adjustment capability to cover the range of phase angles required. The output ports of the

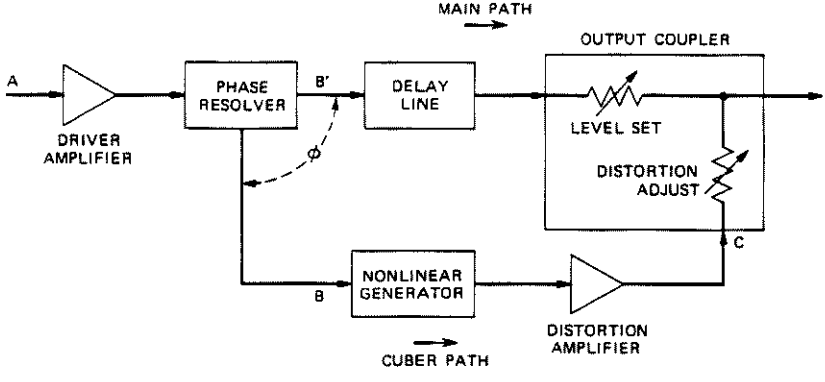



Fig. 7—Predistorter block diagram.

resolver network are arranged so that the connections to these ports can be interchanged. By doing so, the predistorter is adaptable for either USB or LSB operation.

The delay line in the main path is added to match the delay of the cuber path. To account for tolerances of the components, a delay adjustment is required during the manufacturing procedure so that the delays of the two paths are matched as closely as possible.

Because of the large amount of delay required, mismatches at the ends of the delay line result in shape over the 30-MHz channel that is difficult to equalize. Thus, the return loss of the appropriate ports of the delay line, phase resolver, and output coupler is carefully controlled.

### *3.2 Equalization*

Transmission deviations from a flat shape in either the main or cuber path result in residual third-order distortion. Two slope equalizers in the predistorter are employed to control the linear portion of the deviations.

The first slope equalizer adjusts the path from point $A$ to point $B$, as shown in Fig. 7. This slope equalizer is part of the driver amplifier. The end points over the 30-MHz channel are equalized within ±0.02 dB. This procedure imposes strict shape requirements on the resolver network so that

$$|H_{AB}| \cong |H_{AB}'|. \tag{28}$$

The second slope equalizer adjusts the path from point $B$ to point $C$. This slope equalizer is part of the distortion amplifier. The equalization process is accomplished by first disabling the bridge network in the cuber, then equalizing the end points over the 30-MHz channel to within ±0.02 dB.

### *3.3 Alignment of transmitter*

Since each predistorter is equalized during the manufacturing process as previously discussed, the rest of the transmitter must be equalized after all the parts are assembled. From eq. (11) these separate equalizations are equivalent to requiring that $H_A$ be aligned in the transmitter configuration while $G_B A_1$ is aligned separately.

From Figs. 2 and 3, we see that $H_A$ includes the amplifier/modulator, bandpass filter, and the input network of the TWT. A slope equalizer is incorporated in the amplifier/modulator. It is clear that the interface of this portion of the transmitter is somewhere within the active portion of the TWT and is, therefore, not accessible using standard methods. A special scanning intermodulation (IM) test set[10] was designed to permit this measurement to be made. The scanning

capability of the test set permits the transmission path to be slope equalized over the 30-MHz channel.

The overall result of the equalization process is measurable using the scanning IM test set to measure the predistortion improvement across the band. The results of a number of such measurements are summarized in Fig. 8. Two types of TWTs have been used in the AR6A transmitter. For the transmitters using the WE 473A TWT, the average PDI for 24 TWTs indicates a minimum value of 33 dB. To obtain this result, the deviation from flatness must be approximately 0.1 dB. For the transmitters using the KS-22469 TWT, the average PDI for 27 TWTs indicates a minimum value of 35 dB with the corresponding deviation from flatness being approximately 0.07 dB.

Final measurements are made using the scanning IM test set on each transmitter. At this time approximately 3000 units have been produced. The results still show an average minimum PDI of approximately 35 dB.

## IV. PERFORMANCE

### *4.1 Variation with time*

Predistortion of the traveling-wave tube has been demonstrated in the laboratory and manufacturing environment. Transmitter perform-

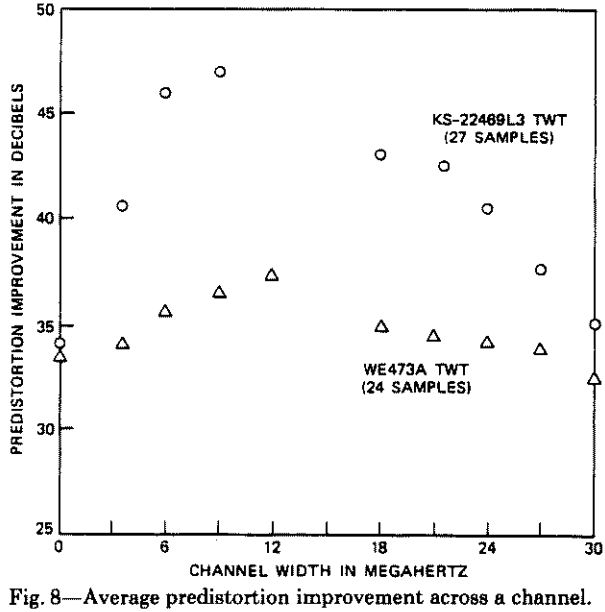



Fig. 8—Average predistortion improvement across a channel.

ance in the field with time and temperature is being tracked by observing the AR6A System in its operating environment.

Data accumulated over a period of several years for 25 TWT amplifiers indicate that the performance with time largely depends upon the stability of the TWT and power supply. The change of gain of the TWT amplifier has been, in general, less than 0.35 dB per year (corresponding to a PDI of 22 dB). Exceptions exist; some amplifiers have demonstrated very large (>2 dB) gain changes. Although no regular monitoring plan has been established, occasional evaluation of selected channels has corroborated the stability estimates.

### *4.2 Variation with temperature*

Temperature changes also cause transmitter gain variation and, therefore, cause the PDI to vary. The AR6A transmitter was tested over an ambient temperature range from 40 to 120°F. These tests revealed that the variation of the PDI was due to transmitter gain changes. The largest contributor to the gain change is the TWT amplifier. Separate tests of five amplifiers established an average change of 0.35 dB over the temperature range. For a transmitter aligned at 80°F, this type of gain variation results in a worst-case PDI of 27 dB.

### *4.3 Summary*

In some cases, gain variations with time and temperature significantly exceed the typical value expected. The AR6A System design compensates for these extreme cases by allowing sufficient margin. The overall intermodulation level in a switch section is monitored. When limits are exceeded, the monitoring system can identify the transmitter or transmitters that require adjustment. In a typical switch section, the random aging and temperature conditions result in an overall switch section performance that satisfies objectives.

## V. ACKNOWLEDGMENTS

The authors wish to acknowledge D. R. Rice, H. Miedema, and R. I. Felsberg for their important early contributions.

## REFERENCES

1. H. Miedema and R. I. Felsberg, private communication.
2. H. Seidel, "A Microwave Feed-Forward Experiment," B.S.T.J., *50*, No. 9 (November 1971), pp. 2879–916.
3. P. Hetracul and D. P. Taylor, "Compensators for Bandpass Nonlinearities in Satellite Communications," IEEE Trans. Aerosp. Electron. Sys., *12*, No. 4 (July 1976), pp. 509–14.
4. A. R. Kaye, D. A. George, and M. J. Eric, "Analysis and Compensation of Bandpass Nonlinearities for Communications," IEEE Trans. Commun., *20*, No. 5 (October 1972), pp. 965–72.

5. Y. L. Kuo, private communication.
6. R. P. Hecken and R. C. Heidt, "Predistortion Linearization of the AR 6A Transmitter," Int. Conf. Commun., June 8–12, 1980.
7. G. L. Heiter, "Characterization of Nonlinearities in Microwave Devices and Systems," IEEE Trans. Microwave Theory and Techniques, *21*, No. 12 (December 1973), pp. 797–805.
8. R. P. Hecken, U.S. Patent No. 4,157,508, issued June 5, 1979.
9. R. L. Adams, J. L. Donoghue, A. N. Georgiades, J. .R Sundquist, R. E. Sheehey, and C. F. Walker, "The AR6A Single-Sideband Microwave Radio System: System Networks," B.S.T.J., this issue.
10. R. I. Felsberg and M. E. Sands, "The AR6A Single-Sideband Microwave Radio System: Special Test Equipment," B.S.T.J., this issue.

## APPENDIX

### *On the Transfer Function for the Up Converter With a Traveling-Wave Tube*

The conversion characteristic of a Schottky diode converter is, for our purposes, most conveniently approximated by the ideal multiplier function

$$z(t) = \mathrm{K} \cdot s(t) \cdot z_{\mathrm{m}}(t), \tag{29}$$

where K is a frequency-independent constant, $s(t)$ the local oscillator (pump) signal, and $z_{\mathrm{m}}(t)$ the modulating (IF) signal. With $s(t) = \cos \omega_o t$, one can see that eq. (29) represents a double-sideband signal with suppressed carrier. Since the bandpass filter following the converter in the actual transmitter suppresses the unwanted sideband, as well as any residual carrier, the use of eq. (29) is indeed justified. Applying a single-tone modulating signal $z_{\mathrm{m}}(t) = A_{\mathrm{m}} \cos(\omega_{\mathrm{m}} t + \phi_{\mathrm{m}})$, one obtains from eq. (29) the lower single-sideband (SSB) signal

$$z(t) = \frac{A_{\mathrm{m}}}{2} \mathrm{K}\ Re\{e^{j(\omega_o - \omega_{\mathrm{m}})t} \cdot e^{-j\phi_{\mathrm{m}}}\}, \tag{30a}$$

and the upper SSB signal

$$z(t) = \frac{A_{\mathrm{m}}}{2} \mathrm{K}\ Re\{e^{j(\omega_o + \omega_{\mathrm{m}})t} \cdot e^{+j\phi_{\mathrm{m}}}\}. \tag{30b}$$

The converter, thus, not only translates the modulating frequency to the desired RF band, but also causes the conjugation of the modulating signal phase. One may, therefore, define two ideal converter transfer operations linking the two frequency domains as follows:

$$Z_{\mathrm{LSB}} = Z(\omega_o - \omega_{\mathrm{m}}) = \mathrm{K_c} \cdot \check{Z}_{\mathrm{m}}(\omega_{\mathrm{m}}) \qquad \text{(LSB)} \tag{31a}$$

and

$$Z_{\mathrm{USB}} = Z(\omega_o + \omega_{\mathrm{m}}) = \mathrm{K_c} \cdot Z_{\mathrm{m}}(\omega_{\mathrm{m}}) \qquad \text{(USB)}, \tag{31b}$$

where $\check{Z}$ is the complex conjugate of $Z$, and $\mathrm{K_c}$ a frequency-independent constant. Note that these operations describe the combined functions

of up converter and bandpass filter. Introducing eq. (31a) into the nonlinear transfer function, eq. (4) of the main text, yields (with some simplifications)

$$Y_{\mathrm{LSB}} = H_B(B_1 H_A \check{Z}_{\mathrm{m}} + B_3(H_A \check{Z}_{\mathrm{m}})^{3^*}) \tag{32a}$$

and

$$Y_{\mathrm{USB}} = H_B(B_1 H_A Z_{\mathrm{m}} + B_3(H_A Z_{\mathrm{m}})^{3^*}). \tag{32b}$$

With these expressions we have derived nonlinear transfer functions for the cascade of up converter, bandpass filter, and TWT. Note that for the LSB and the USB case, the modulating signal is $Z_{\mathrm{m}}$ and it is the converter that performs the conjugation. It also must be mentioned that for the intentions of this model, the converter constant $K_c$ and the transfer function of the bandpass filter have tacitly been absorbed in $H_A$.

## AUTHORS

**Rudolf P. Hecken,** M.S.E.E., 1959, Ph.D., 1964, Technische Hochshule Aachen, West Germany; Institute for High Frequency Technology, Aachen, 1959–1964; CERN, Geneva, Switzerland, 1964–1968; Bell Laboratories, 1968—. Mr. Hecken has been involved in the design of networks for the L5 coaxial system, FM microwave radio systems, and AR6A Radio System. Since 1982 he has been Director of the Radio Systems and Terminals Laboratory.

**Richard C. Heidt,** A.S.E.E., Arlington State Jr. College, 1956; B.S.E.E., Southern Methodist University, 1959; M.E.E., New York University, 1963; Bell Laboratories, 1961—. Mr. Heidt was first engaged in microwave transmission circuits and antennas. Since 1969 he has been involved in the design of transmitters for the single-sideband microwave radio system. He is currently Supervisor of a group developing digital microwave radio systems. Member, IEEE, Eta Kappa Nu, Sigma Tau, Kappa Mu Epsilon.

**David E. Sanford,** E.T., RCA Institutes, 1968; Fairleigh Dickinson University, 1969–1971; Merrimack College, 1982—; Bell Laboratories, 1968—. Mr. Sanford initially worked in the field of plasma physics and was involved in development of transmitter and receiver circuitry for single-sideband radio. Recently he has developed circuits for the receiver portion of digital radio systems.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10, December 1983
Printed in U.S.A.

# *The AR6A Single-Sideband Microwave Radio System:*

# System Networks

By R. L. ADAMS,* J. L. DONOGHUE,* A. N. GEORGIADES,* R. E. SHEEHEY,* J. R. SUNDQUIST,* and C. F. WALKER*

(Manuscript received February 14, 1983)

The paper describes some of the key networks needed for the AR6A System for the predistorter, Intermediate Frequency (IF) filtering, and terminal multiplex. The design of a passive broadband adjustable phase splitter for the predistorter is described. Network designs using inductors, capacitors, and, in some cases, crystals for IF filtering, equalization, and multimastergroup connection are also described.

## I. INTRODUCTION

A large number of filters and equalizers, and an adjustable phase splitter are needed in AR6A for pilot insertion and removal, band limiting, amplitude correction, and predistortion circuitry in repeaters and in multiplex equipment. Technology developed for TD and TH radio[1,2] and L4 and L5 coaxial-carrier transmission systems[3–5] was enhanced to meet the specific and demanding needs of the AR6A† System filters and equalizers; however, a special approach was required for the design of an adjustable phase resolver (phase splitter) for the predistorter. The phase resolver is an adjustable Inductor Capacitor (LC)‡ network that allows matching the phase of the predistortion

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version of the system.

‡ Acronyms and abbreviations used in the text and figures of this paper are defined at the back of the *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

signal to the opposite of the phase of the distortion component of the main signal. The filters provide wider bandwidths and higher discrimination than their FM radio counterparts, and amplitude equalization, rather than delay equalization, is required. Extensive use was made of computer aids for design and analysis. A high-speed, precision interactive mini Computer-Operated Transmission Measurement Set (COTMS) proved necessary for laboratory development of networks. More than 700 models of about 140 network designs were needed for development and final design of the AR6A System.

The theory and design of the phase resolver for the AR6A predistorter is described in some detail. Descriptions of the Intermediate Frequency (IF) filter and equalizer, the upper and lower IF bandpass filters, and the multimastergroup connector filters follow.

## II. PHASE RESOLVER

### *2.1 Function*

The phase resolver is a passive three-port network used in the AR6A predistorter.[6] It splits the predistorter input signal into two equal-amplitude signals that have a constant phase difference in the IF band between 59 and 89 MHz. The phase difference is manually adjustable. This provides adjustment of the phase predistortion component to cancel the distortion products generated in the traveling-wave tube.

### *2.2 Theory*

The phase-resolver network consists of a quadrature hybrid network, two attenuators, and a combining hybrid transformer connected as shown in Fig. 1. The outputs can be shown to be

$$E_B = 0.5\ E_A(G_1 + jG_2), \qquad (1)$$

$$E_C = 0.5\ E_A(-G_1 + jG_2).$$

Thus, the magnitude of either output is

$$|E_{B,C}| = 0.5\ |E_A|\ \sqrt{G_1^2 + G_2^2}, \qquad (2)$$

and the phase difference is

$$\Theta_B - \Theta_C = 2\ \tan^{-1}(G_2/G_1). \qquad (3)$$

A key element of this network is the broadband quadrature hybrid network.

To achieve a wideband quadrature hybrid network, a Cauer configuration[7] was chosen. This is a four-port network consisting of two hybrid transformers connected to two lossless LC networks, $N$ and $N'$, as shown in Fig. 2. With port A driven, ports B and C will have quadrature phase, and port D will be isolated if $N$ and $N'$ are

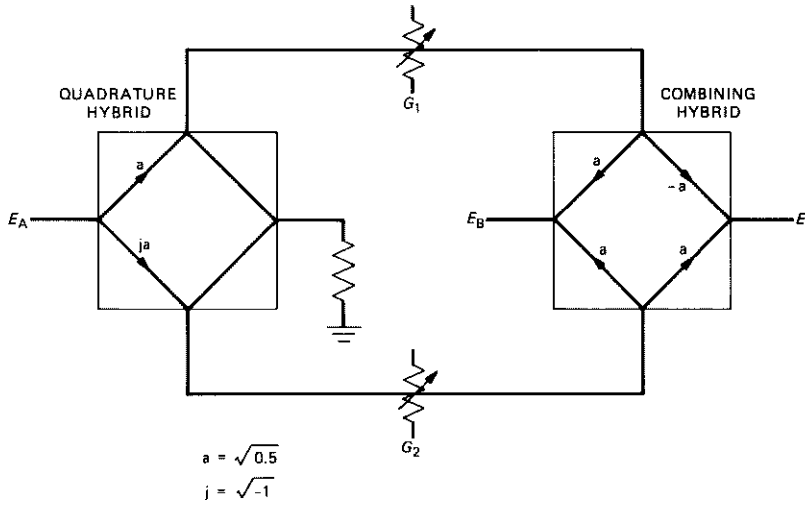



Fig. 1—Phase-resolver block diagram.

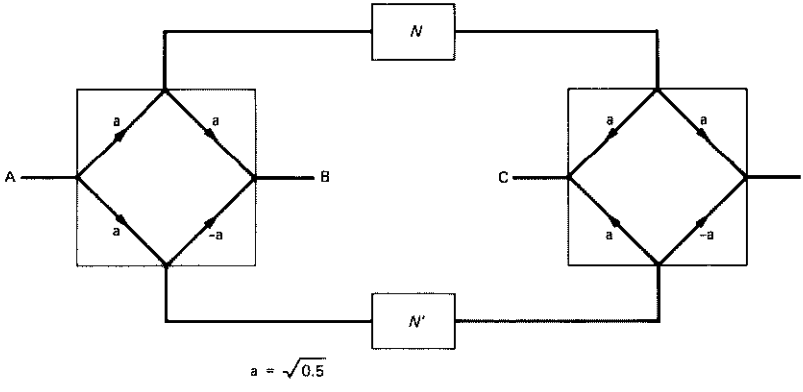



Fig. 2—Cauer network.

dual symmetric networks. This can be determined if the scattering parameters for the complete network are derived in terms of the scattering parameters of $N$ and $N'$.

$$s_{AA} = (s_{11} + s'_{11})/2$$
$$s_{BA} = (s_{11} - s'_{11})/2$$
$$s_{CA} = (s_{21} + s'_{21})/2$$
$$s_{DA} = (s_{21} - s'_{21})/2. \quad (4)$$

If $N$ and $N'$ are dual networks, then

$$s_{11} = -s'_{11}, \quad (5)$$

and

$$s_{21} = s'_{21}.$$

Then,

$$\begin{aligned} s_{AA} &= 0, \\ s_{BA} &= s_{11}, \\ s_{CA} &= s_{21}, \\ s_{DA} &= 0. \end{aligned} \tag{6}$$

The other twelve scattering parameters for the four-port network have a corresponding form. These relations describe a hybrid network with port A isolated from port D, and port B isolated from port C. It is a quadrature network if $N$ and $N'$ are not only dual but also symmetric lossless LC networks. This can be shown as follows.

It is convenient to define a split ratio, $\Gamma$, as

$$\Gamma = \frac{s_{BA}}{s_{CA}} = \frac{s_{11}}{s_{21}}. \tag{7}$$

For a quadrature network, $\Gamma$ must have the following form:

$$\Gamma(s) = s\,\frac{P_{11}(s^2)}{P_{21}(s^2)} \quad \text{or} \quad \frac{P_{11}(s^2)}{sP_{21}(s^2)}. \tag{8}$$

Here, $P_{11}$ and $P_{21}$ are polynomials. With $s = j\omega$, phase quadrature can be observed. To avoid right-half plane transmission zeros requiring right-half plane zeros, $\Gamma$ is restricted to the following form:

$$\Gamma(s) = \frac{P_n(s^2)}{s^{2m+1}}. \tag{9}$$

Here, $P_n$ is an $n$th-order polynomial and $m$ is an integer. Then the networks, $N$ and $N'$, are simple ladder networks.

The design of a quadrature Cauer network consists of first finding a suitable polynomial, $P_n(-\omega^2)$, such that $|\Gamma|$ approximates unity over the frequency band of interest. A corresponding symmetric ladder network and its dual are then synthesized to realize $N$ and $N'$. The synthesis procedure can be simplified by taking advantage of the fact that the networks are symmetric.

### *2.3 Realization*

Extensive use was made of computer aids and computer-operated transmission measuring sets during the design and development. The hybrid transformers were characterized on COTMS.[8] With this characterization used in a circuit analysis program, an optimization routine

could be used to modify $N$ and $N'$ to compensate for parasitics. For model characterization and laboratory correction of parasitics, a high-accuracy, high-speed computer-operated transmission measurement set proved necessary. This set, "Mini COTMS," has a measurement speed fast enough for computer-corrected simultaneous real-time display of loss, phase or delay, and input and output return loss.

Figure 3 is a photograph of the phase resolver.

## III. IF FILTER/EQUALIZER

### *3.1 Function*

This circuit provides the 74.13-MHz center frequency IF filtering and amplitude equalization. Figure 4 shows that four networks are used. The bandpass filter limits adjacent channel interference and image interference in the Multimastergroup Translator for Radio (MMGT-R). One basic equalizer and two mop-up equalizers provide static amplitude equalization. The basic equalizer corrects for systematic amplitude variations in the overall Transmitter-Receiver (TR) bay transmission characteristic. Two mop-up equalizers correct for unit-to-unit amplitude deviations.

### *3.2 Design*

Design was facilitated by two interactive computer programs. One does the rational function approximation and insertion loss synthesis; the other performs Norton transformations to optimize filter topology and element values. A general-purpose linear network analysis program allowed simulation of parasitic effects in any of the designs.

An 11th-order filter was synthesized to meet the objectives. Two

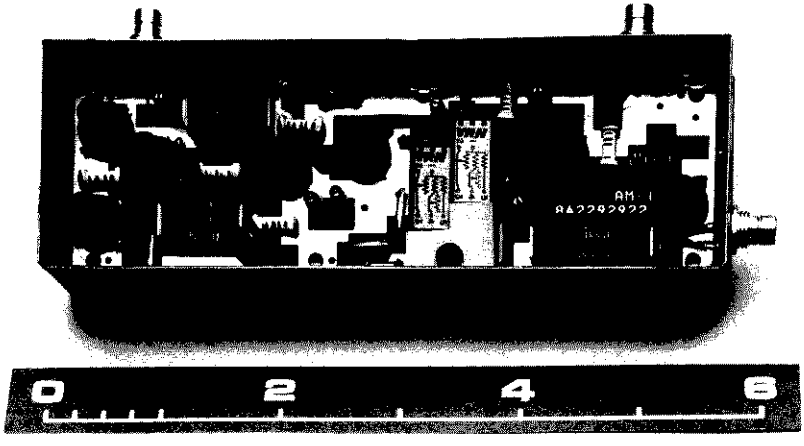



Fig. 3—Photograph of phase resolver.

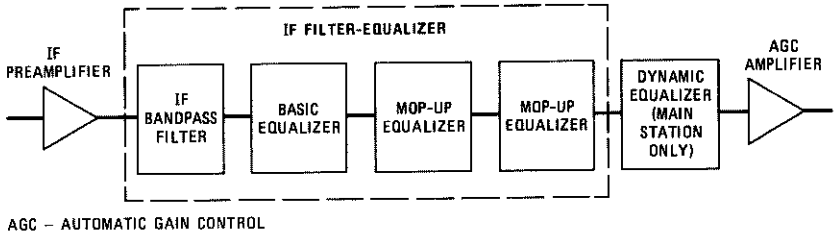



Fig. 4—IF filter-equalizer block diagram.

bridged-T sections are needed to compensate for amplitude variations caused by the finite $Q$ of the filter inductors. The filter is assembled on a printed wiring board that is attached to a cast-aluminum housing as shown in Fig. 5. Electrostatic shields are contained within the housing. Field tests have confirmed that the filter meets system requirements.

Four mop-up equalizer designs achieve positive-slope, negative-slope, positive-quadratic-shape, and negative-quadratic-shape amplitude response. Bridged-T sections are used to realize each of the required shapes. Frequency response measurements of the AR6A switch section involved are used to choose the shapes used. When a mop-up equalizer is not needed, a 3.5-dB pad is used.

## IV. LOWER AND UPPER IF BANDPASS FILTERS

### *4.1 Function*

The transmitting-output and the receiving-input spectrum of the MMGT-R each cover the 59.844- to 88.460-MHz band and contain ten mastergroups. This frequency spectrum is split into two bands of five mastergroups each. The Lower-band Intermediate Frequency (LIF) is from 59.844 to 73.116 MHz, and the Upper-band Intermediate Frequency (UIF) is from 75.188 to 88.460 MHz.

Two bandpass filters (BPFs), the LIF and the UIF filters, are used in the MMGT-R to select each of these bands in the modulator and in the demodulator. The filters reject the carrier frequency, the upper sideband, and second and third harmonics of the carrier. Lower sideband loss keeps crosstalk and tone interference into adjacent radio channels, due to undesired modulator products, at acceptable levels.

### *4.2 Design*

The designs of the two filters are similar, so only the LIF BPF will be covered. The stopbands cover a wide frequency range and very high losses are required. To achieve reasonable element values, to reduce the effect of parasitics for this wideband, and to ease tuning, a low-pass filter and a high-pass filter in tandem are used rather than a

Fig. 5—Photograph of IF filter.

bandpass design. Computer-aided insertion loss methods (Section 3.2) were used to synthesize a 13th-order Cauer elliptic high-pass filter and a ninth-order equal ripple passband with arbitrary stopband low-pass filter. Norton and pi-delta transformations were applied to achieve equal termination resistances and desirable element values. Two bridged-T constant resistance sections are required to compensate for amplitude variations in the passband caused by the finite $Q$ of the inductors.

The filter is assembled on two printed wiring boards and housed in a drawn steel enclosure (see Fig. 6). One board contains the equalizer, pad, and high-pass sections; the other board contains the low-pass section. Each filter inductor is shielded, and there is a shield between the printed wiring boards to reduce magnetic coupling between inductors.

## V. MULTIMASTERGROUP CONNECTOR FILTERS

### *5.1 Function*

The AR6A multimastergroup (MMG) connector is a precision passive network which allows a direct connection from the MMGT-R receiver to the MMGT-R transmitter when separation of the various components of the MMG signal is not required for routing or other purposes. In this application, the type-B Mastergroup Translators (MGTB) and other multiplexing equipment are not required. The key elements of the MMG connector are a highly selective bandpass filter and a crystal band-elimination filter. Figure 7 is a photograph of the two filters. The bandpass filter selects the 8.628- to 21.9-MHz MMG

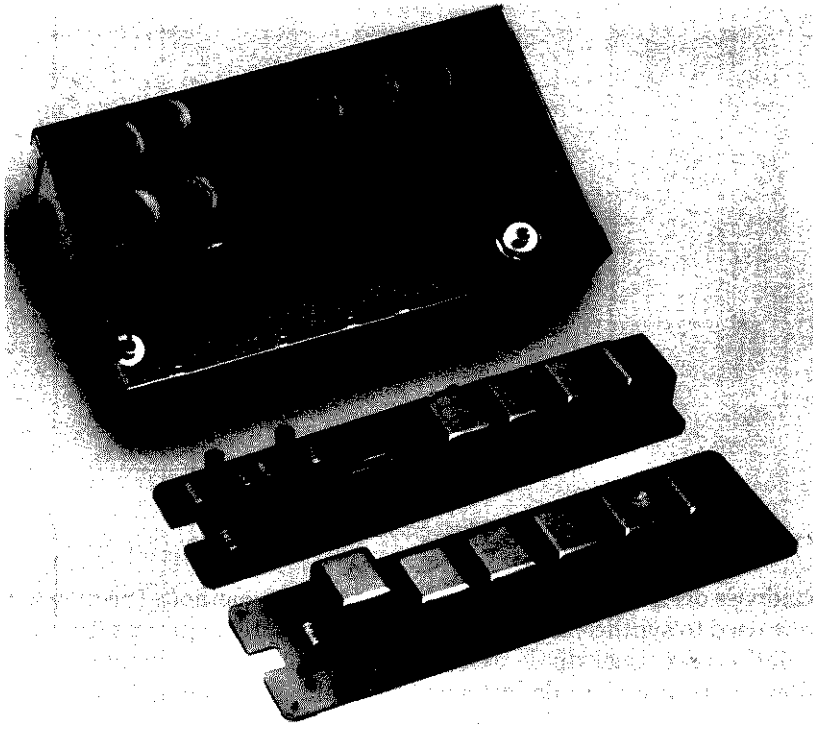


Fig. 6—Photograph of LIF BPF.

spectrum and rejects other frequencies to avoid crosstalk between radio channels. It also suppresses two pilot tones. The Band-Elimination Filter (BEF) suppresses four additional pilot tones.

The filters must have minimum effect on the message spectrum in spite of the high selectivity required. Wide skirts on the pilot suppression characteristic are needed to reduce noise near the pilots. The skirts increase the complexity of the crystal BEF significantly.

### *5.2 Design*

The electrical design of the bandpass filter was achieved with the help of two interactive computer programs (Section 3.2). The inductor values were chosen to match a family of precision high-$Q$ adjustable inductors originally designed for MGTB.[5] A complex arm bridged-T equalizer section was added to compensate for loss variations due to finite $Q$ effects. The result is a 22nd-order loss-equalized bandpass filter. It was also possible to suppress two of the pilots by placing six crystal units at each frequency in shunt arms of the filter.

Four filters in tandem are used to suppress the remaining four pilots. For each frequency, a low-pass filter is designed to pass the desired

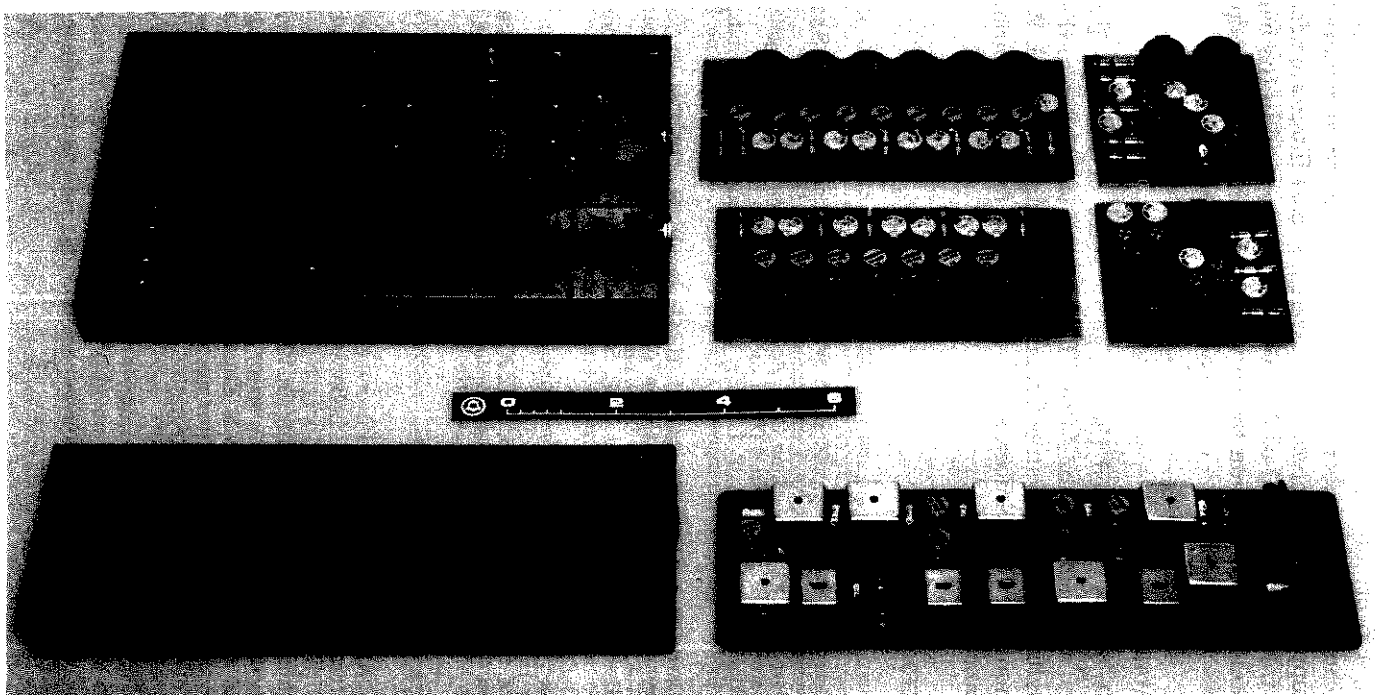


Fig. 7—Photograph of MMG connector filters with bandpass filter at top and crysal band-elimination filter at bottom.

spectrum, and crystal units are placed in shunt arms to reject the pilot. Because of the number of crystal units required to achieve the desired suppression characteristic, separate low-pass filters are needed for each frequency. A total of 28 crystal units are used in the band-elimination filter in addition to the 12 crystal units in the bandpass filter.

## VI. SUMMARY

The paper has described the design of some of the key filters, equalizers, and networks for AR6A. The authors would like to thank S. Darlington for his innovative Cauer coupler designs, and F. J. Witt, T. H. Simmonds, R. P. Snicer, and J. L. Garrison for their inspiration and guidance.

## REFERENCES

1. E. J. Drazy, R. C. MacLean, and R. E. Sheehey, "TD-3 Microwave Radio Relay System: Networks," B.S.T.J., *47*, No. 7 (September 1968), pp. 1397–422.
2. E. J. Drazy, R. E. Sheehey, and H. C. Wang, "TH-3 Microwave Radio System: Networks," B.S.T.J., *50*, No. 7 (September 1971), pp. 2137–54.
3. J. L. Garrison, L. P. Labbe, and C. C. Rock, "L-4 System: Basic and Regulating Repeaters," B.S.T.J., *48*, No. 4 (April 1969), pp. 841–88.
4. W. G. Albert, J. B. Evans, Jr., T. J. Haley, T. B. Merrick, and T. H. Simmonds, Jr., "L-4 System: Terminal Arrangements," B.S.T.J., *48*, No. 4 (April 1969), pp. 993–1040.
5. J. L. Garrison, A. Olsen, Jr., and T. H. Simmonds, Jr., "L5 Coaxial-Carrier Transmission System: Transmission Networks and Magnetic Components," B.S.T.J., *53*, No. 10 (December 1974), pp. 2203–48.
6. R. P. Hecken, R. C. Heidt, and D. E. Sandford, "The AR6A Single-Sideband Microwave Radio System: Predistortion for the Traveling-Wave-Tube Amplifier," B.S.T.J., this issue.
7. W. Cauer, *Synthesis of Linear Communication Networks*, Vol. 1, New York: McGraw-Hill, 1958, Chapter 9.
8. R. M.-M. Chen, C. F. Hempstead, Y. L. Kuo, M. L. Liou, R. P. Snicer, and E. D. Walsh, "L5 Coaxial-Carrier Transmission System: Role of Computing and Precision Measurements," B.S.T.J., *53*, No. 10 (December 1974), pp. 2249–68.

## AUTHORS

**Robert L. Adams,** BSEE, 1967, Purdue University; MSEE, 1969, Massachusetts Institute of Technology; Bell Laboratories, 1967—. Mr. Adams has been involved with the reduced sensitivity active filter design and computer-assisted inductor-capacitor (LC) filter tuning. In 1975 he became Supervisor of a group that designed thin-film Resistor Capacitor RC active filters, passive LC filters, and crystal filters. Presently, he is Supervisor of a group concerned with adaptive systems, switched-capacitor networks, and telecommunications applications of Very Large-Scale Integration (VLSI) silicon integrated circuits.

**James L. Donoghue,** Bell Laboratories, 1956—. Mr. Donoghue has been involved in the design and development of transmission networks for various projects for telephone operating companies including A5 channel banks and LMX3 and L5E coaxial system.

**Andrew N. Georgiades,** Bell Laboratories, 1958—. Mr. Georgiades spent

most of his career designing communication networks for such projects as the K and O carrier systems, the L3, L4, and L5 coaxial systems, *Picturephone*,* DMG and MGT projects. Lately he participated in the design and development of a digital channel-bank controller simulator. He is presently a member of the technical staff in the Subsystems Analysis and Design Group engaged in the realization of a digital echo canceller. He has written extensively on the use of crystal networks, and he is a coholder of a patent for inventing the monolithic band-elimination filter.

**Robert E. Sheehey,** Assoc. EE, 1960, Wentworth Institute, Boston, MA; Bell Laboratories, 1960—. Mr. Sheehey has worked in the development of electrical networks for radio relay and coaxial carrier systems, in addition to the realization of analog networks for data transmission using hybrid integrated-circuits technology. His present responsibilities include the development of adaptive notch equalizers for use in digital radio systems.

**J. Robert Sundquist,** Wentworth Institute, Boston, MA, 1951; Raytheon Company, 1951–1957; Bell Laboratories, 1957—. Mr. Sundquist has been engaged in the design and development of a variety of inductors and transformers. He is presently concerned with the design and development of filters and various other networks for transmission systems.

**Charles F. Walker,** ASEE, 1967, Franklin Institute; BET, 1971, Northeastern University; Bell Laboratories, 1967—. Mr. Walker has worked on the development of analog filters for data sets, digital terminals, and radio relay systems. His present responsibilities include the design of circuits for digital channel banks using hybrid integrated circuit technology.

* Registered service mark of AT&T.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 10 December 1983
Printed in U.S.A.

## *The AR6A Single-Sideband Microwave Radio System:*

# Special Test Equipment

By R. I. FELSBERG* and M. E. SANDS*

(Manuscript received January 22, 1983)

This paper describes two specialized test sets designed specifically for AR6A transmit-receive radio equipment alignment. The scanning intermodulation (IM) test set can adjust the transmitter for proper gain shape and optimum predistorter performance. It can also confirm that the transmitter is meeting its IM requirements. The pilot selection test set is used to test the radio intermediate frequency receiver units for proper operation during changing received signal conditions. This test set provides signals to simulate either flat or selective fades and can be used to check the pilot resupply function or space-diversity switch transition levels.

## I. INTRODUCTION

It was realized early in the development of the AR6A† Radio System that special test equipment would be needed to ensure that performance objectives of this new single-sideband (SSB)‡ system were met. Two specialized test sets have been designed for this purpose: (1) the scanning intermodulation (IM) test set, and (2) the pilot selection test set. Both sets are self-contained in carrying cases with handles and

* Bell Laboratories.

† Amplitude Modulation Radio at 6 GHz for the initial (A) version.

‡ Acronyms and abbreviations used in the text and figures of this paper are defined at the back of the *Journal*.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

are provided with the necessary cables and other accessories for use in the field.

The scanning IM test set has been developed to adjust the transmitters in the AR6A Radio System to meet the critical linearity requirements necessary for a multihop radio route. Swept wideband measurement of IM single-sideband amplitude-modulated (SSBAM) radio is required for predistorter and up-converter alignment and to verify that the high degree of linearity necessary for this type of transmission has been obtained. The transmitter alignment is accomplished by one single-tone level measurement/adjustment and three separate three-tone, third-order IM measurement/adjustments made with the test set. An auxiliary oscilloscope (the display section of the radio transmission test set) is used to display the resulting transmitter intermodulation characteristics over the frequency band of interest.

The AR6A receiver conditions the received signal prior to retransmission to the next station. The circuits that perform this function require special test signals to verify the correct functioning of this portion of the AR6A System. The pilot selection test set was developed to supply the required signals in a convenient manner.

## II. SCANNING INTERMODULATION TEST SET

### *2.1 General*

The single-sideband amplitude-modulated transmitters in AR6A are required to be extremely linear so that the third-order IM noise that accumulates over a long route is held to an acceptable level. Transmission with very low IM noise is accomplished by using a traveling-wave tube (TWT) operated well below its maximum power output capability and by the use of predistortion. Use of the predistorter requires flat transmitter third-order IM characteristics and precise alignment to obtain maximum third-order IM reduction. Predistortion operates by generating third-order IM noise that is equal in amplitude and opposite in phase to the third-order IM noise generated by the nonlinearities in the transmitter. These nonlinearities result in cancellation (nulling) of the IM noise. To obtain wideband predistortion, the overall third-order IM from the transmitter must have a flat IM frequency characteristic. The IM test set is used to adjust the up converter and predistorter in the transmitter for the required third-order IM flatness and third-order IM distortion cancellation. In addition to the linearity requirement, the transmitter output power level is required to be held within certain limits. The alignment of the transmitter requiring IM slope adjustment and predistorter IM null adjustment interacts with the transmitter gain. As a result, the output power must be monitored as adjustments are made to maintain the required power level.

A portable scanning IM test set has been developed to provide easy and rapid alignment of transmitters in the field where the three-tone, third-order IM distortion product from the transmitter can be rapidly scanned and displayed over the entire frequency band of interest. The set is divided into two modules (Figs. 1 and 2) to reduce the weight

Fig. 1—Test module of scanning IM test set.

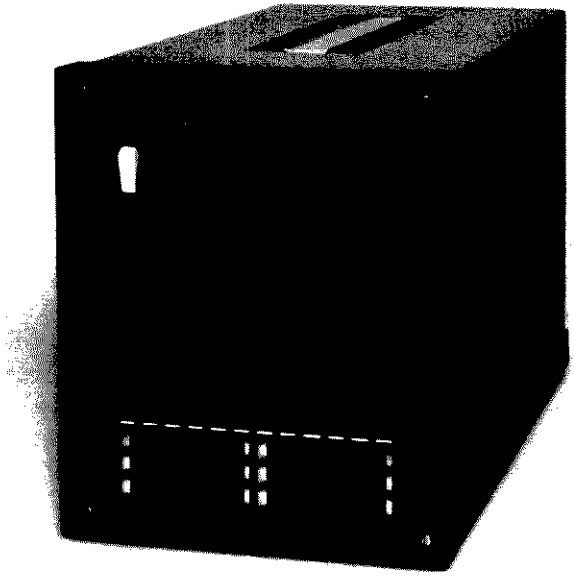


Fig. 2—Power module of scanning IM test set.

and size to acceptable values. The units are housed in metallic cases with retractable carrying handles. Separate lightweight soft-pack and hard-pack shipping cases are available to protect the modules during transportation. Space has been provided in the power module soft-pack and hard-pack to store the ten cables required for operation of the test set. An auxiliary oscilloscope is required for use with the test set to display the IM characteristics as a function of frequency. An auxiliary power meter is also required to monitor the output power during alignment.

### *2.2 Principle of operation*

The test set can be divided into two basic sections consisting of a tone-generating section and a receiving- or signal-processing section. Referring to the simplified block diagram of Fig. 3, the three-tone generating section is located on the upper left. A manually swept, voltage-controlled oscillator (VCO) tone is split and applied to two mixers. Also applied to the two mixers are tones from crystal oscillators with frequencies that differ by 1 MHz. The two down-converted tones fall within, and can be manually swept over, the required 30-MHz intermediate frequency (IF) band. In addition, a third tone fixed in level and frequency near the center of the IF band is combined with the two manually swept tones. The three tones are sampled at the output coupler and applied to the automatic gain control (AGC) unit and then to the variolosser to maintain the level of the swept tones constant across the 30-MHz band. The three-leveled tones are applied to the device under test (DUT), i.e., the transmitter. The three-tone, third-order IM distortion product $P_{a-b+c}$ resulting from nonlinearity in the transmitter is fixed in frequency since the difference in frequency between tone (a) and tone (b) remains fixed (1 MHz apart) although the tones are swept in frequency and the third tone (c) is also fixed in frequency. As a result, the IM product, although fixed in frequency, reflects any change in IM distortion versus frequency as a change in its amplitude when the two tones are swept over the frequency band. The tones and the resulting IM products from transmitter nonlinearities are coherently down converted from 6 GHz to IF in the signaling processing circuits shown on the right of Fig. 3. The IF signals are down converted again to 315,920 Hz so that only the single IM distortion product $P_{a-b+c}$ from the transmitter falls within the very narrow passband (50 Hz) of the monolithic crystal filters employed in the test set. As a result, all other tones and IM products are excluded, thus preventing any spurious three-tone IM from being generated in the gain stages following the filter. In addition, a signal-to-noise advantage is gained by use of the filters. This is important, since the IM product signal level at the input to the filter is very low and therefore requires high amplification. Very stable crystal oscilla-

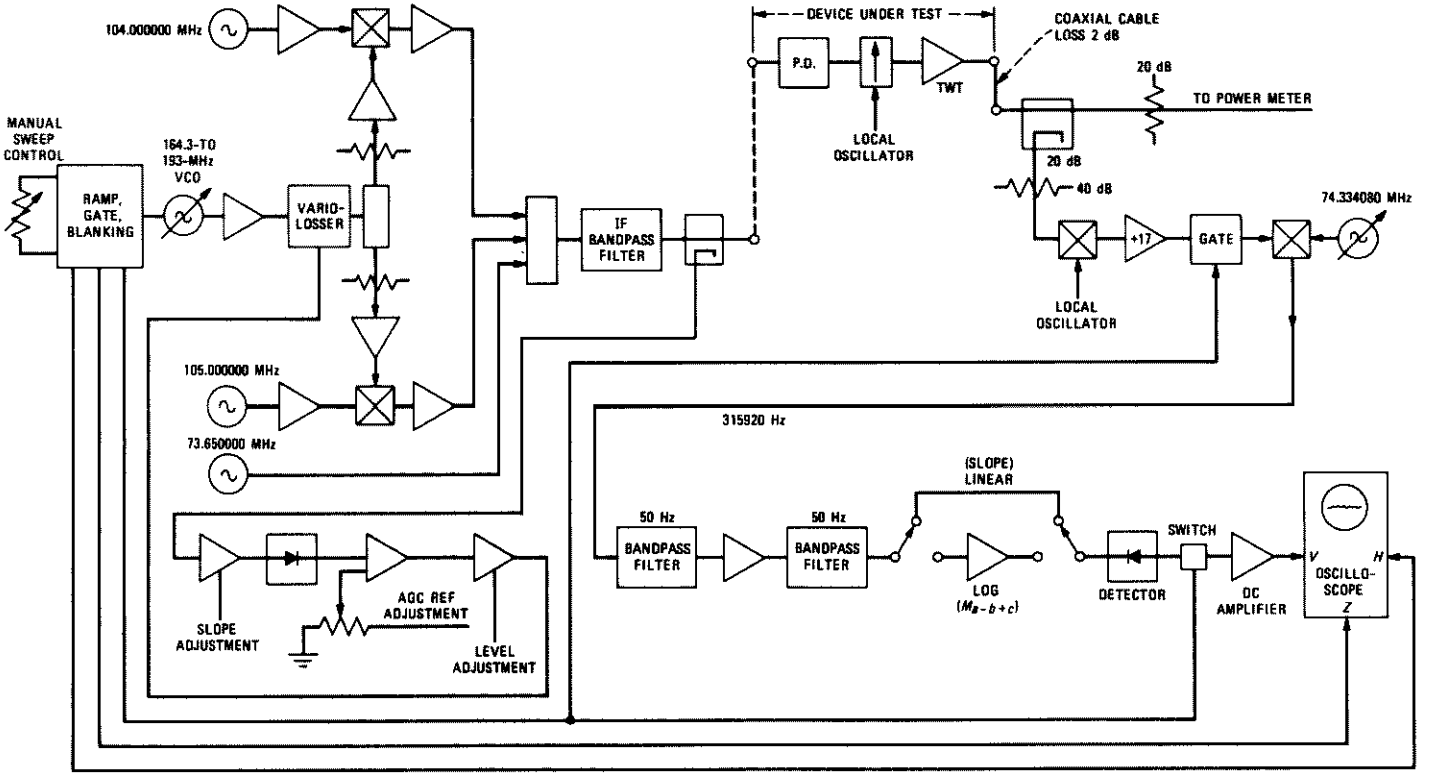
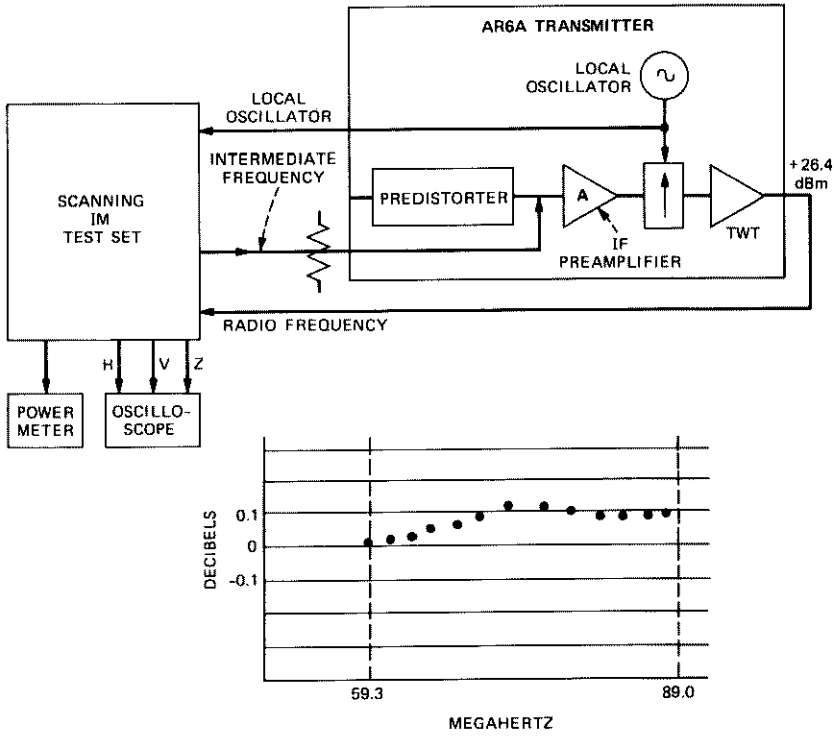



Fig. 3—Simplified block diagram of a scanning IM test set.

tors are employed to ensure that the IM product remains within the very narrow passband of the filters. Four oven-stabilized crystal oscillators are used for this purpose and have a stability of better than one part in $10^7$ per day. One oscillator, employed by the signal processing section, is made adjustable by means of a knob on the front panel of the test set to center the IM product in the filter passband when necessary to compensate for crystal oscillator drift.

### *2.3 Tests performed*

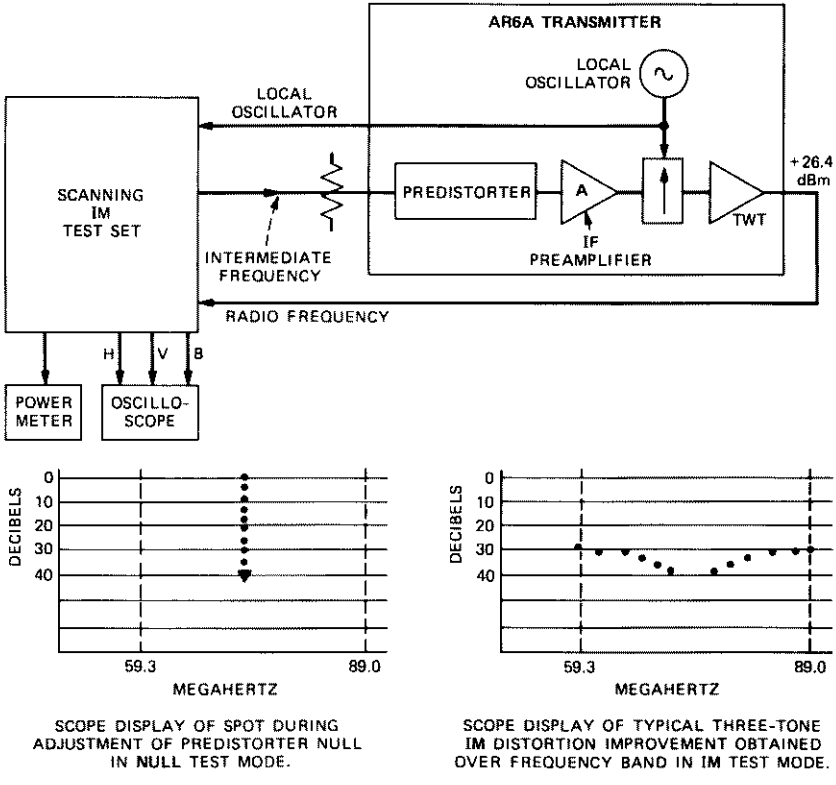
The first test (LEVEL) provides a calibrated tone to the input of the transmitter so that the power output may be set to the required level. The remaining three tests are three-tone, third-order IM distortion tests. The test set supplies three IF tones, two manually swept and one fixed in frequency, to the input of the transmitter for this purpose. In the second test (SLOPE) shown in Fig. 4, a three-tone,




SCOPE DISPLAY OF TYPICAL ADJUSTED THREE-TONE IM TRANSMITTER SLOPE OBTAINED OVER THE FREQUENCY BAND IN SLOPE TEST MODE.

Fig. 4—Transmitter slope adjustment.

third-order IM distortion product, resulting from distortion in the transmitter, is adjusted for flatness over the 30-MHz band by means of a slope adjustment control located in the up-converter preamplifier (A) in the transmitter. In the third test (NULL) shown in Fig. 5, the predistorter in the transmitter is now included in the transmitter circuit and is switched on and three tones are held fixed in frequency. The IM distortion product is minimized (nulled as shown diagrammatically in the figure as a spot moving downward) by adjustment of the phase and amplitude controls on the predistorter in the transmitter. The fourth test (IM) also shown in Fig. 5 displays typical reduction of manually swept three-tone IM distortion versus frequency obtained by predistortion. It can then be determined on the calibrated display if the IM distortion reduction meets or exceeds requirements over the entire frequency band and, if so, this indicates that the preceding tests and alignments have been satisfactorily completed.




SCOPE DISPLAY OF SPOT DURING ADJUSTMENT OF PREDISTORTER NULL IN NULL TEST MODE.

SCOPE DISPLAY OF TYPICAL THREE-TONE IM DISTORTION IMPROVEMENT OBTAINED OVER FREQUENCY BAND IN IM TEST MODE.

Fig. 5—Predistorter adjustment and IM improvement obtained.

### 2.4 Requirements for the scanning IM test set

#### 2.4.1 Tone requirements

The test set is required to supply a single IF tone of a specified power level for the LEVEL test mode, and three tones, one fixed and two that can be manually swept in frequency, of specified power levels for the other three IM test modes. The single-tone level has been specified to have a power level equal to the full-load signal level and is used to adjust the transmitter gain to the required level. The total power of the three tones has been specified to be the same as that for the single tone. Intermodulation testing is required over the entire frequency range of each transmitter to ensure linear operation over the entire frequency band of each channel. The test set is required to produce tones that can be manually swept over the 30-MHz IF band (59.3 to 89.0 MHz). A worst-case maximum intermodulation condition has been found to exist when the three tones are held fixed in proximity to each other. The test set is designed with the three tones held fixed in frequency near the center of the band in the NULL mode so that the predistorter is nulled under worst-case conditions.

#### 2.4.2 Spurious tone requirement

All spurious tones from the test set are required to be at least 40 dB below the power levels of the three tones. Any spurious tones that coincide with the fundamental tones as they are swept should be at least 50 dB below the power levels of the three tones so that they will add less than 0.025 dB to any of the fundamental tones. This will ensure that spurious signals will be held low enough that they will have negligible effect on the IM product. However, with swept measurements, this effect is almost unnoticeable since the spurious signals move at two or more times the rate of the fundamental tones and pass rapidly through them unnoticed, allowing the 50-dB criteria to be relaxed by 10 dB to the 40-dB requirement.

#### 2.4.3 Tone flatness requirement

The tones from the test set are required to be flat with frequency to within 0.03 dB as they are swept over the 30-MHz frequency band. This requirement ensures that the tone levels and consequently the IM derived from these tones in the transmitter will reflect mainly the IM characteristics of the transmitter with minimum contribution of IM from the test set. The tones are set to be flat (0-dB variation) at the band edges with the maximum allowed variation of 0.03 dB at midband. The maximum variation of each of the two swept tones contributes 0.06-dB error to the CRT display of transmitter slope at midband. This small error is of little consequence, particularly since the slope can be adjusted using only the extreme frequency points.

### *2.4.4 Test set IM requirements*

The three-tone, third-order IM generated by the test set is required to be at least 50 dB below that of the transmitter without predistortion. This requirement ensures that any three-tone, third-order IM originating in the test set is much less than the IM originating in the transmitter (with predistorter on) that is being measured. Therefore, the IM contributed by the test set to the measurement (less than 1 dB) would be barely discernible on a cathode ray tube (CRT) display of a 25-dB PDI (predistorter IM improvement in dB). Test set tone flatness and thermal noise from the transmitter and test set limit the amount of PDI that can be measured with reasonable accuracy to about 40 dB.

## III. PILOT SELECTION TEST SET

### *3.1 General*

The IF shelf of the AR6A Radio System contains a number of circuits to condition the received signal prior to retransmission. It was necessary to provide a way to check these circuits for proper operation when they were connected together in the system. A test set was designed that would condition test signals fed to the AR6A IF shelf to be representative of the type that would be experienced during normal system operation.

### *3.2 Principle of operation*

The required test signal is composed of three pilot tones at frequencies of 62.448, 75.122, and 85.856 MHz. These three tones are varied in amplitude in certain prescribed ways to exercise the IF shelf signal correction circuits. This signal simulation technique checks for the proper functioning of the shelf, resupply switch, pilot detector, receiver control, AGC amplifier, and the dynamic equalizer.

### *3.3 Description of test set*

The pilot selection test set (PSTS) is a portable unit that can be easily transported for field testing of the AR6A System. The unit is completely passive and requires no source of power. The PSTS must be used in conjunction with a suitable power meter.

The PSTS is made up of narrowband filters, shape networks, continuously variable attenuators, and a precision step attenuator. These components are interconnected in various ways by means of patch connectors on the front panel of the test set. All attenuators, shape networks, and filters can be used as individual items by access from the front panel jacks (see Fig. 6).

The PSTS does not contain the oscillators that supply the three required pilot tones (see Fig. 7). Each AR6A support bay contains a

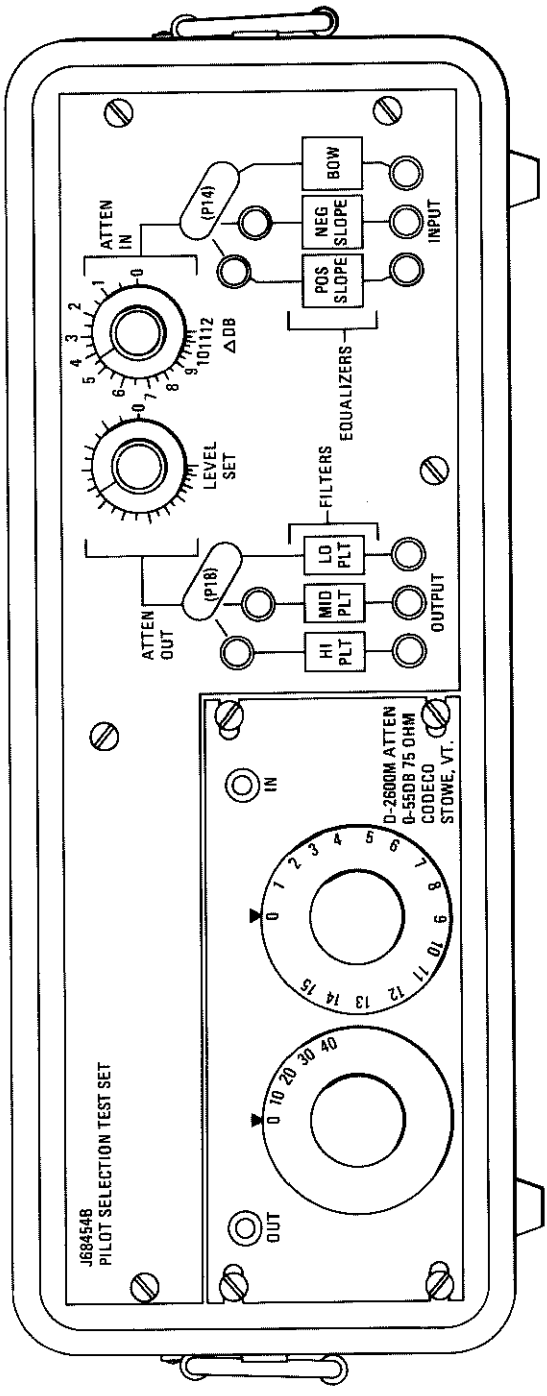



Fig. 6—Pilot selection test set.

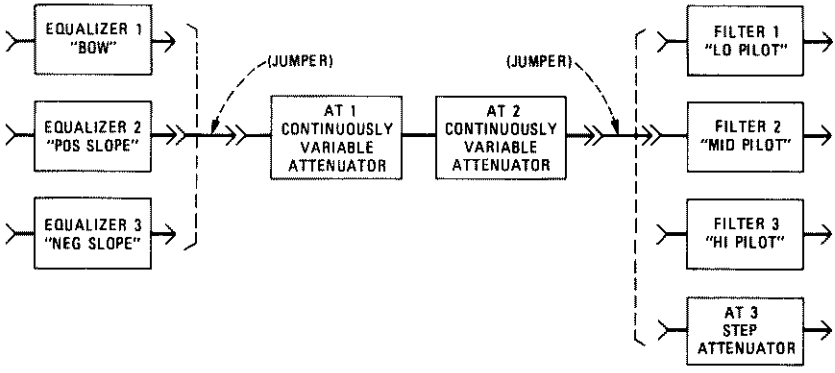



Fig. 7—Block diagram of the pilot selection test set.

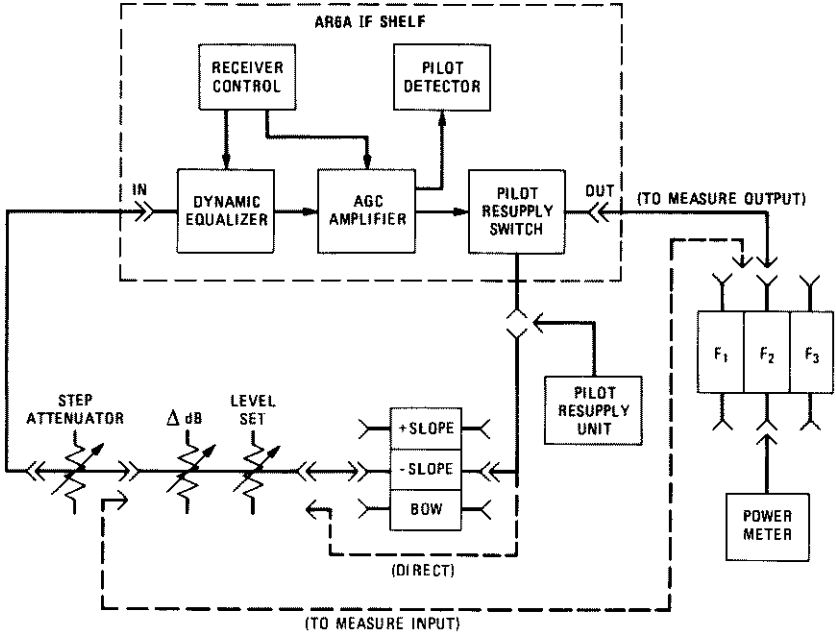



Fig. 8—Shelf-test functional diagram.

pilot source that is used for the pilot resupply mode of operation of the TR bay. The pilot resupply unit supplies the fixed amplitude pilot tones and the PSTS then modifies their amplitudes as required to test the AR6A bay (see Fig. 8).

The output from the pilot resupply unit contains all three pilot tones mixed together. The PSTS contains a narrowband monolithic crystal filter for each of the tones. These filters, when used in conjunction with a power meter, allow each tone to be measured individ-

ually. This arrangement is used to set the pilot reference power levels initially and then to observe the pilot levels after being acted upon by the AR6A receiver under test.

The PSTS checks the performance of the AR6A dynamic equalizer by first establishing equal pilot signal levels and then introducing known level differences in the pilots by the use of shape networks. Signal correction by the dynamic equalizer can then be observed.

The three shape networks are a positive slope, a negative slope, and a bow shape. All three shape networks attenuate the center frequency pilot 10 dB. The positive slope network attenuates the low-frequency pilot 18 dB and the high-frequency pilot 2 dB. The negative slope network attenuates the low-frequency pilot 2 dB and the high-frequency pilot 18 dB. Thus, the two networks can exercise the end pilots relative to the center pilot by ±8 dB. The bow network attenuates the center pilot frequency 10 dB and both the upper and lower pilot frequencies 2 dB. The resulting signal transmission through the network will be a bow with the end pilots 8 dB higher than the center pilot.

The PSTS checks the performance of the AR6A AGC amplifier by varying the amplitude of all three pilots together with a step attenuator while monitoring the AGC function. A precision step attenuator is provided to adjust signal levels over a 55-dB range in 1-dB steps. This step attenuator can also be used with the linearity test set if desired. The pilot resupply and space-diversity threshold levels are checked by varying the amplitude of all three pilots together with a continuously variable attenuator while watching for trip levels to occur.

Two continuously variable attenuators are provided. One is used to set the input power level accurately while the other is used to determine trip levels. The use of two variable attenuators allows the input power level adjustment to remain undisturbed while the second attenuator searches for trip levels. The original test set input power level can then easily be restored by turning the second attenuator back to zero. The use of continuously variable attenuators eliminates the transient effects on trip levels associated with step attenuators.

For convenience, hairpin patch plugs simplify the interconnection of the shape networks to the variable attenuator input. Also, a second hairpin plug interconnects the variable attenuator output to the pilot filters.

## IV. CONCLUSION

The scanning IM test set has proven to be essential in the field for the initial transmitter predistorter linearity alignment and for the continued confirmation and maintenance of the high degree of linearity required by the AR6A Radio System.

The pilot selection test set is essential in the field to verify the correct functioning of the IF receiver circuits and to provide simulated fading for checking trip levels of the pilot resupply and space-diversity circuits.

## AUTHORS

**Robert I. Felsberg,** B.S.E.E., 1963, Fairleigh Dickinson University; Bell Laboratories, 1953—. Mr. Felsberg has been involved in military projects at Whippany, N.J., and recently in SSBAM radio. Currenty, he is working on digital radio at Merrimack Valley.

**Melvin E. Sands,** B.S.E.E., 1957, Virginia Tech; Bell Laboratories, 1962—. Mr. Sands was previously involved in the design and the development of radars for the U.S. Army's ballistic missile defense system. Since 1970 he has been involved in design and development work on the AR6A radio transmission system.

## ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ACO | alarm cutoff |
| AGC | automatic gain control |
| AM | amplitude modulated |
| AR6A | amplitude modulation radio at 6 GHz, version A |
| BEF | band-elimination filter |
| BFE | beam-forming electrode |
| BPF | bandpass filter |
| BSP | Bell System Practices |
| BSR | Bell System reference |
| BSRF | Bell System reference frequency |
| BSRT | Bell System reference tone |
| BSRTU | Bell System reference transmitting unit |
| CMOS | complementary metal-oxide semiconductor |
| CRT | cathode ray tube |
| D/A | digital-to-analog |
| DAS | Data Acquisition System |
| DDD | direct distance dialing |
| DUT | device under test |
| EMI/RFI | electromagnetic interference/radio frequency interference |
| EQ | equalizer |
| FB | front-to-back ratio |
| FET | field-effect transistor |
| FSK | frequency shift-keyed |
| G/T | gain/temperature |
| HPF | high-pass filter |
| IF | intermediate frequency |
| IM | intermodulation |
| JFS | jumbogroup frequency supply |
| LC | inductor capacitor |
| LED | light-emitting diode |
| LIF | lower-band intermediate frequency |
| LO | local oscillator |
| LO | locally generated microwave carrier |
| LOA | law of addition |
| LPF | low-pass filter |
| LSB | lower sideband |
| MCSS | microwave carrier synchronization supply |
| MG | mastergroup |
| MGDF | mastergroup distributing frame |
| MGTB | mastergroup translator, type B |
| MMG | multimastergroup |

| | |
|---|---|
| MMGT–R | multimastergroup translator–radio |
| OMFS | office master frequency supply |
| PAL | phase alternation line |
| PFS | primary frequency supply |
| PLL | phase-locked loop |
| PM | phase modulated |
| PSTS | pilot selection test set |
| R-C | resistor-capacitor |
| RF | radio frequency |
| RFS | regional frequency supply |
| SC | suppressed carrier |
| SCOTS | Surveillance and Control of Transmission Systems |
| SDC | Shottky diode converters |
| SDS | Software Development System |
| SECAM | sequential with memory |
| SL | section loss |
| SPDT | single-pole double throw |
| SSB | single sideband |
| SSBAM | single-sideband amplitude modulation |
| STMS | selective transmission measuring set |
| TASC | telecommunication alarm surveillance and control |
| TDMA | time division multiple access |
| TL | transmission level |
| TR | transmitter-receiver |
| TSS–R | Transmission Surveillance System–Radio |
| TTL | transistor-transistor logic |
| TWT | traveling-wave tube |
| UIF | upper-band intermediate frequency |
| USB | upper sideband |
| VCO | voltage-controlled oscillator |
| VDC | varactor diode converters |
| XPD | cross-polarization discrimination |