ESD-TR-68-149

THIRD QUARTERLY TECHNICAL REPORT EXPERIMENTAL SIGNAL PROCESSING SYSTEM

August 1967

ESLE

DIRECTORATE OF PLANNING AND TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Sponsored by: Advanced Research Projects Agency
Washington, D. C.

ARPA Order No. 800

This document has been approved for public release and sale; its distribution is unlimited.

(Prepared under Contract No. F19628-67-C-0198 by International Business Machines Corp., 18100 Frederick Pike, Gaithersburg, Maryland 20760)

AD670866

LEGAL NOTICE

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

ESD-TR-68-149

THIRD QUARTERLY TECHNICAL REPORT
EXPERIMENTAL SIGNAL PROCESSING SYSTEM

August 1967

DIRECTORATE OF PLANNING AND TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Sponsored by: Advanced Research Projects Agency
Washington, D. C.

ARPA Order No. 800

(Prepared under Contract No. F19628-67-C-0198 by International Business Machines Corp., 18100 Frederick Pike, Gaithersburg, Maryland 20760)

FOREWORD

This research is supported by the Advanced Research Projects Agency. The Electronics Systems Division technical project officer for Contract No. F19628-67-C-0198 is Major Cleve P. Malone, Jr. (ESLE). This report covers the period from 1 May 1967 through 31 July 1967.

This technical report has been reviewed and is approved.

William R. Lauterbach
Lt Colonel, USAF
Chief, Seismic Array Program Office
Development Engineering Division
Directorate of Planning and Technology

ABSTRACT

The Third Quarterly Technical Report for the LASA Experimental Signal Processing System identifies the effort expended to provide the hardware and software in support of research and development directed toward the study of seismic signal processing. It also delineates work tasks planned for the next quarter. This document presents detailed information related to machine configurations, time delay correlation, event location accuracy, optimum processing, fast Fourier transform, array design, steering delay library, magnitude estimation, and travel time characterization.

CONTENTS

ILLUSTRATIONS

Figure Page

TABLES

Section 1

INTRODUCTION

The work reported herein was performed under the "LASA Experimental Signal Processing System" Contract Number F19628-67-C-0198, and is a continuation and extension of the "Large Aperture Seismic Array Signal Processing Study,"[1] Contract Number SD-296, and the "Large Aperture Seismic Array Signal Processing Communications and Simulation Study,"[2,3,4] Contract Number AF19(628)-5948.

The purpose of this contract is to design, develop, and implement the LASA Experimental Signal Processing System (ESPS), including the hardware and software, to provide an experimental capability to:

a. Evaluate performance of the system in accordance with system requirements

b. Demonstrate the capability to meet basic LASA signal processing objectives

c. Conduct research to develop means of improving and extending the capability of the system

d. Perform seismic and signal processing experiments of interest.

During the first[5] and second[6] quarters, effort was primarily directed toward establishing the Seismic Array Analysis Center (SAAC), Washington, D.C., installation and operation of the Detection and Event Processors as System/360 Model G and H general purpose digital computers, respectively, and the continuation of system studies in seismic signal processing.

Work in the present quarter has focused on such signal processing studies as time delay correlation, event location accuracy, optimum processing, fixed-point Fourier transform, and array design programming for system development. In addition, activity in event magnitude estimating, time delay library formulation, and travel time characterization supporting event processing has been initiated.

Section 2

RESULTS AND SUMMARY OF WORK ACCOMPLISHMENTS

The following subsections correspond to the functional disciplines of operations, systems, and programming.

2.1 OPERATIONS

The principal effort during this quarter was expended on the machine system, Experimental Display and computer operations.

Installation and checkout of the SAAC hardware configuration was completed. The current and planned configurations are described in Appendix I. The IBM 2250-1 CRT Display was installed, and debug of the software support packages was initiated.

Modification and integration of the Experimental Display was completed for both the off-line and on-line modes. The strip chart recorder interface, reported previously,[2] has been deferred due to priority work in other areas.

2.2 SYSTEMS

During this quarter, effort pertinent to overall process improvement and event processing structure has been undertaken.

As reported in Appendix II-1, a method of automatically determining time delays from the event data has been formulated. Preliminary testing indicates process stability and that the resulting set of time delays is consistent even when the signal-to-noise ratio at the seismometer level is near unity.

The effects of time delay and system calibration errors on large array event location accuracy are presented in Appendix II-2. It is shown that although the processed output record signal-to-noise ratio and system calibration errors contribute to location error, the significant location errors are due to array aperture and incorrect time delays.

A study to determine the effectiveness of fidelity optimum processing to discriminate against one of two simultaneously occurring events has been undertaken. The results presented in Appendix II-3 relate the effect of signal-to-noise ratio and array geometry on processing performance.

Current interest in the Fast Fourier Transform algorithm has prompted investigation of its fixed arithmetic execution precision to determine the applicability of microprogramming. It is shown in Appendix II-4 that the errors in fixed point execution can be reasonably bounded by scaling the process when an overflow occurs and that microcoded execution speed factors are about ten over standard assembly language programming.

A method of array geometry optimization has been established and tested and is described in Appendix II-5. Since the technique employs interelement distance dependent correlation coefficient data as input, its principal advantage appears to be comparative in nature.

In the phase delay activity, effort was concentrated on defining process structure, automatic acquisition, anomaly characterization, and subarray beam assignment.

With respect to process structure and automatic acquisition, the chief accomplishment has been the development of a library organization for handling the phase delays for both detection and event processing. Progress on this effort is described in Appendix III-1.

A procedure based on the use of sets of correction factors where each phase delay set is valid over a specified geographic area called a region has been developed. The procedure also involves the use of interpolation techniques to calculate corrections relative to areas between regions. Appendix III-2 describes the regions and their associated correction factors as well as a test which was performed to evaluate the effectiveness of the region groupings. The results of the test indicate that regional corrections improve system performance.

The subarray beam assignment investigation addresses the method of covering an arbitrary location in inverse velocity space by a judicious choice of subarray beams. During this quarter, formulas were developed for calculating subarray steering losses relative to a specified configuration of preformed beams and the associated LASA steering loss resulting from the use of such preformed beams. In addition, a rule was specified for computing the beam from each subarray which is to be used in forming the LASA beam, when given a desired LASA beam location. Details regarding the subarray assignment investigation may be found in Appendix III-3.

In Appendix IV-1, the empirical relationships between the classical event magnitude measurement and the kinetic energy "density" method of magnitude estimation are given.

A significant portion of event processing assumes the use of travel time data to sort wave arrivals into consistent arrival families and to use this data in estimating event location. The travel time information in its original tabular form is bulky and would require interrelation with epicentral range and hazardous differentiation of discrete data. In Appendix IV-2, orthogonal polynomial fitting in the least squares error sense was applied to a portion of the Jeffreys-Bullen Seismological Tables to explore the existence of explicit functions relating body wave travel time and horizontal inverse phase velocity to epicentral event range at a known depth.

2.3 PROGRAMMING

Programming effort during this quarter was concentrated in Detection Processing, Supervisor/Monitor, Plotting Routines, Display Programs, and Experimental Programs. Accomplishments in each of these areas will be presented in the following subsections.

2.3.1 Detection Processing

At the completion of this quarter's work, three operational versions of the Detection Controller were in existence. Two of these were oriented towards a real-time operating environment, while the third was tailored to support analysis activity.

The difference between the two real-time versions lies in the area of input/output control. One operates using the standard IBM System/360 Disk Operating System (DOS), while the other operates using a specially written input/output controller. Some experiments were made to determine if there was any timing advantage of one method or the other. The latter was faster by about 0.5 percent and appears to have no significant effect on the total number of array beams that can be performed in a given time interval.

The data analysis version of the Detection Controller uses standard DOS input/output, but it is intended for operation under slightly different conditions. In particular, the signal rectification and integration process has been set at 5 Hz, rather than 1.67 Hz, and the threshold detection and noise integration functions have been removed. The capability to produce intermediate tapes of subarray beams (unfiltered and filtered), array beams, and short time average values has been included in this version. The primary use of this program, thus far, has been to produce input tapes for both beam pattern correlation and experimental display work.

All three of these programs have the same basic structure and design philosophy. All accept the standard SAAC Edit Format, as input, although they do not presently interpret the array status data. A unique feature of these programs is a storage allocation routine which allows maximum use of available core storage for each run of the Detection Controller. This has enabled us to process as many as 1024 LASA beams in one run of the program.

An additional program capability includes the dispersed subarray beam capability (sce Appendix III-3). An input card supplied with the steering delays for each array beam indicates to the Detection Controller which of the subarray beams are to be used for array beamforming. On the basis of this information, the delays are properly indexed for use by the beamforming microprogram.

2.3.2 Supervisor/Monitor

Systems generation and maintenance activity was accented upon availability of the System/360 2040H early in thc quarter. A multiprogramming Disk Operating System was provided. Various versions were generated as configuration changes took place. At this point, the system contained no alterations, but it was gencrated with a greatly expended supervisor area in anticipation of the future core memory requirement. Special attention was given to output spooling techniques which would allow reasonably efficient program development using a multisystem including just one line printer and one card punch.

2.3.2.1 Dctection Subsystem

A test which measured the overhead incurred by using a standard Disk Operating System supervisor in a detection processing function was conducted. The results underscored the advantages in choosing this system as a superstructure around which the Detection Monitor could bc built.

Work in this quarter also included detail design, coding, and simulated testing of the first Detection Monitor package. This package embraced capabilities originally planned as two models. First, the Disk Operating System was modified to cnable multiprogramming operation in a machine without storage protcction so that the data acquisition function could operate asynchronously with the detcction processing function, while having common access to certain portions of core memory. The monitor was designed so that inputs could bc rcceived in real time from any source, with the detection processing programs relieved of device-dependent considerations. The same design also cnabled the setting of modc indicators from external sources to allow dynamic entry by thc detection processing program into various outputting modes for display purposes.

Thc design approach is notable in that the Detection Controller may bc integrated with the Detection Monitor into a working system merely by assembling macros with thc program. The code generated by the macros rcpresents thc Detection Monitor and becomes a separate program only at thc timc thc Detection Subsystem is loaded and initialized for execution. This was tested by using a simulated Detection Controller and a program which was cxecuted in thc other processor to play thc role of the real-time interfacc function.

As the quarter closed, work began on a third Detection Monitor model. DEMON III represents a structural rather than a functional change over previous versions. The Detection Subsystem programs in DEMON III will be entirely core-resident at all times. In previous versions, programs were loaded from disk memory to perform infrequent functions such as handling I/O device errors and operator communication. The delays introduced by such an arrangement cannot be tolerated in a real-time system. Therefore an appropriate subset of the Disk Operating System transient functions will be made a part of the permanently resident nucleus.

2.3.2.2 Event Subsystem

Significant progress was made toward providing operating sytems tailored specifically to SAAC needs. These systems take the form of modifications to the Disk Operating System. The manner in which the modifications are provided allows the generation of a standard DOS, an SAAC tailored model, or anything in between, by specifying parametric values associated with one set of macro source statements. That is, only one system source copy must be maintained.

The first step in providing suitable systems was to include the support of devices in the SAAC configuration which are not supported in DOS. This fulfills our needs for program debugging at SAAC and serves as the basis upon which other monitors may be developed.

These changes to the supervisor were implemented and tested during this quarter, and the package was provided for daily general-purpose operation in place of the standard system. It includes input/output macros and error checking for the following devices:

1627 Plotter
Channel-to-Channel
2250 Model I Display
Experimental Display/Strip Chart Recorder

Also, an effort was made to provide a higher level of 2250 Display support by moving a capability from the IBM/360 Operating System into the SAAC system.

A fundamental requirement for the monitor needed to support development of event processing programs and to support the ultimate automatic operation is the capability to schedule and execute tasks automatically, as previously outlined. The capability to dynamically schedule tasks into the background job queue has been designed and coded as an extensive set of modifications to the Disk Operating System. This scheduling program must also coordinate the spooling of outputs generated by two programs executing simultaneously and intended for the same destination (e.g., printing). These functions were undergoing test as the quarter ended. An extension of this will allow a list of jobs to be created and managed dynamically for the foreground partition as well as the background partition. This work is in the design phase.

Other support includes the specification for a possible remote terminal operation which can be built into the monitor discussed above.

2.3.3 Plotting Routines

The SAAC machine configuration contains an on-line IBM 1627 plotter. The step size of this particular digital plotter is 0.01 inch. The pen can be moved in three directions: horizontally, vertically and diagonally. Moves which are not in one of these basic directions are accomplished by a combination of the basic movements in such a way that the approximation is always within one-half the step size.

To facilitate the use of the plotter, several subroutines were written for use with FORTRAN programs. Instructions for the plotter are written on magnetic tape by these routines and later sent to the plotter by a utility program which may be run simultaneously with other programs.

Three basic routines are available for line generation and annotation of plots. Their FORTRAN names are PLOT, SYMBOL, and NUMBER.

The PLOT subroutine calculates the necessary plotter instructions to move from one coordinate to another with the pen raised or lowered. The coordinates are given in inches of deflection from a present origin. The algorithm to generate lines which are not in one of the basic directions is an integral part of the routine as is the handling of all output instructions.

The SYMBOL subroutine accomplishes the translation of alphameric information into sets of pen movements. Titles and other annotation, as well as symbols drawn at specific data points, can be drawn from lists of letters on special codes.

Numeric data is translated to its alphameric equivalent, with the appropriate number of decimal places, by the NUMBER subroutine and then sent to the SYMBOL routine.

2.3.4 Display Program

Design and implementation of an initial program capability for both the Experimental Display and the IBM 2250 Visual Display unit were accomplished during this quarter.

The 2250 display program accepts as input the seismometer level edited tapes and displays up to eight channels on the display. The channels are selectable through the 2250's console typewriter. Scaling is controlled through the functional keyboard of the 2250. Appropriate time and channel identification information is displayed in addition to the seismometer waveforms.

The Experiment Display program accepts as input the rectified and integrated beam tapes produced by the Detection Controller and displays the traces as a 32x32 matrix on the cathode ray tube experimental display. The beams can be displayed in a forward or reverse direction, in real time or at a fast scanning rate, in a single step mode in which the display is updated whenever an interrupt button is pushed, or in a stationary time state. The scaling and pedestal values which dynamically control the presentation are continuously adjustable from the Experimental Display Unit.

2.3.5 Experimental Programs

The implementation of two experimental programs—Time Delay Correlation and Beam Pattern Correlation—began during this quarter, and an initial capability was developed for both areas. Continued modification of these programs is anticipated as experimental results evolve. They are discussed briefly in the following paragraphs.

The Time Delay Correlation program was written to provide an automatic means for determining subarray beam steering delays. Using an iterative process, the program performs a cross-correlation between an array beam and each of the subarray beams or sensors in turn, adjusting each subarray beam delay based upon the incidence of the maximum cross-correlation. A more detailed description will be found in Appendix II-1 of this report.

A Beam Pattern Correlation program was developed to scan the Detection Processor Short Time Average (STA) output for seismic events. The program accepts a reference pattern as input along with the STA data and generates a display tape as output as well as the printed output. The correlation is performed over each STA time sample of 1024 Beams to generate a set of correlation indexes. These indexes represent the correlations for each position of the reference pattern in the beam field. These values are then adjusted and placed on the display tape in such a way that the positions with the largest correlations will be shown as high intensity areas on the display. Each input STA time sample generates one display time sample. This work is currently in progress.

Section 3

PLANS

The following activities will be initiated during the next quarter. Discussions correspond to the functional disciplines. Each task is presented in sufficient perspective to identify its interrelation and contributions to the project.

3.1 OPERATIONS

Activity for the next reporting period centers upon completing plans for SAAC machine expansion to three computers, as indicated in Appendix I-1.

Plans call for machine utilization to increase with a partial second shift operation scheduled throughout the period. In addition to program verification and debug efforts continuing at an accelerated level, two other activities will utilize major blocks of computer time. These activities include the generation of a modest library of edited LASA event tapes and an expanded effort in data analysis of these events.

The availability of applications programs will allow investigations of the means of utilization, limitations, and capabilities of both the IBM 2250 CRT Display and the Experimental Beam Display. The evaluation of these displays is expected to have a major effect on the definition of requirements for system maintenance and operations consoles.

3.2 SYSTEMS

The time delay correlation program reported in Appendix II-1 appears to warrant extension. Emphasis will be placed on experimentation for its intended purpose and upon applying the technique to event location. In the area of computational signal processing techniques, null steering development and spatial beam detection will be investigated as potential system functional components.

With respect to the time delay library process structure, the chief tasks ahead are the following:

a. Specify, in detail, a maintenance procedure and schedule for checking the library of phase delays

b. Specify the conditions under which the correlation program will be used for updating, correcting, and enlarging the library

c. Establish criteria for identifying events to be used in library updating.

Programming of the sector interpolation formulas will be undertaken. It is hoped that the effectiveness of these formulas in obtaining accurate steering delays will be tested by applying them to several events located outside the 35 regions.

The magnitude-kinetic energy "density" relationship will be used to find event magnitudes for a number of events using unfiltered and filtered LASA beams and various window sizes. Comparisons will be made with event magnitudes calculated by the classical method. The differences in magnitude estimates of each LASA subarray to that obtained on an array basis should be explored for different values of magnitude.

An attempt will be made to find polynomials which fit body wave travel time and horizontal inverse phase velocity as functions of both epicentral event range and event depth.

In addition, it appears desirable to investigate certain system instrumentation configurations to establish possible interim on-line operating objectives and requirements.

3.3 PROGRAMMING

The planned effort includes activity in the Detection and Event Processor areas, as well as in the continued development of experimental and data analysis programs.

Primary emphasis in the Detection Processor area will be on extending the existing capability of the real-time version of the Detection Controller to operate with varying array configurations. This extension will provide the mechanism to evaluate storage and timing considerations as well as detection capability for different magnitude events. In the Detection Monitor area, design and implementation will proceed on DEMON III, the core resident version. A version of the Detection Controller will be merged with the first Detection Monitor Model and tested by simulation.

In the Event Processor area, the present supervisor will be extended to allow the experimental display to function more efficiently with the processor. Currently, when the experimental display program is executing, the Event Processor must be dedicated to this function, thereby preventing the normal multiprogramming mode of operation.

In the experimental and data analysis programming areas, efforts will be continued in developing and extending the experimental and IBM 2250 display capability, time delay correlation, noise correlation, and beam pattern correlation areas.

Appendix I

SAAC COMPUTER EQUIPMENT

Figure 1 depicts the configuration of SAAC machines at the end of this reporting period. Changes made to the configuration since the last report fall into two categories:

a. Additions made to implement the planned growth of the laboratory. Included in this category are the addition of the IBM 2250-1 Display Unit, the IBM 2740 Remote Printer Keyboard, the IBM 029 Interpreting Keypunch, and the expanded capability Experimental Display.

b. Additicns and/or reconfigurations undertaken to better adapt the system to its evolving utilization. Because of the heavy emphasis on program evaluation and debug, independent operation of each System/360 Model 40 became increasingly desirable. As a result a 2841-1 Disk Control Unit and 2311-1 Disk Unit were added to the Detection Processor so that each processor could operate under an independent Disk Operating System.

Integration of the Experimental Display with the Event Processor has released a 2403-2 Tape Drive and Control Unit. These were connected to the Detection Processor Channel No. 2 to provide a tape dump for print-out. The resulting tape is printed by the Event Processor on the 1403-2 line printer operating in a foreground mode and overlapped with normal Event Processor operation.

Figure 2 shows the machine system developments planned during the next reporting period. The addition of the 1403-2 Line Printer to the 40G will eliminate the need for the 2403 Tape Control and Tape Unit and allow it to be deleted from the system. The replacement of the 2520 by the 2540 Reader-Punch will improve the efficiency of card operations.

Figure 3 illustrates the planned Auxiliary System configuration.

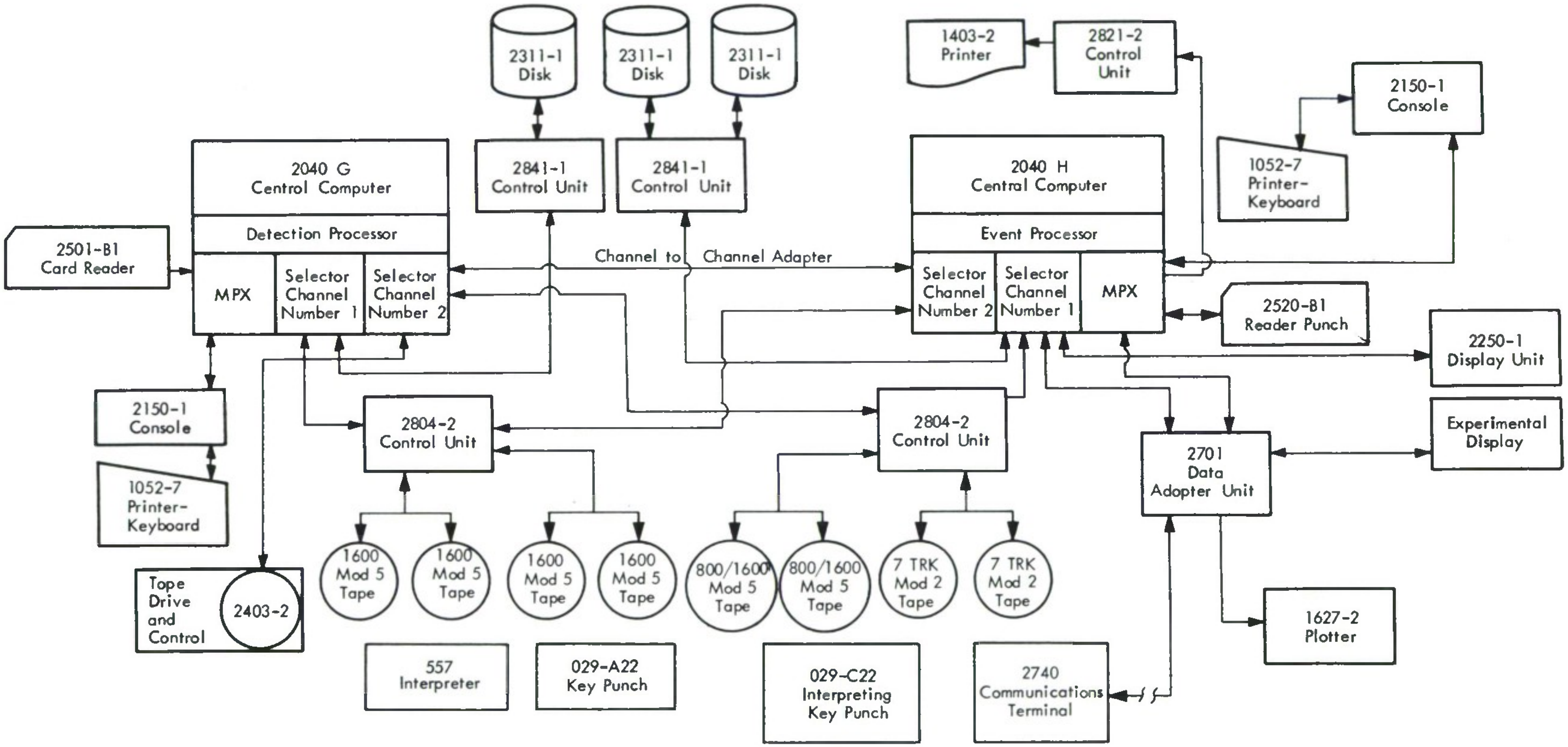
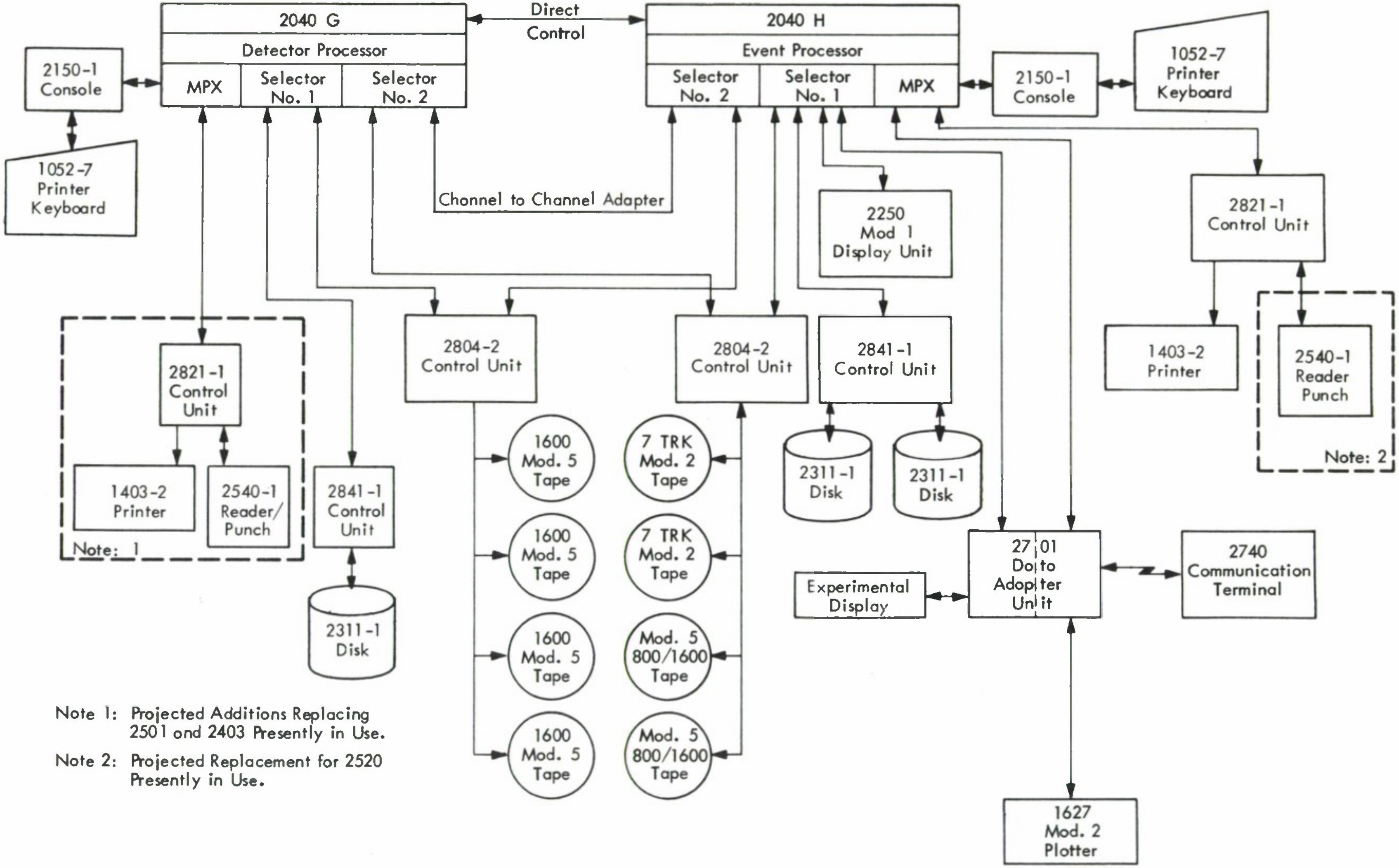


Figure 1. Current Equipment Configuration



Note 1: Projected Additions Replacing 2501 and 2403 Presently in Use.

Note 2: Projected Replacement for 2520 Presently in Use.

Figure 2. Planned Equipment Configuration

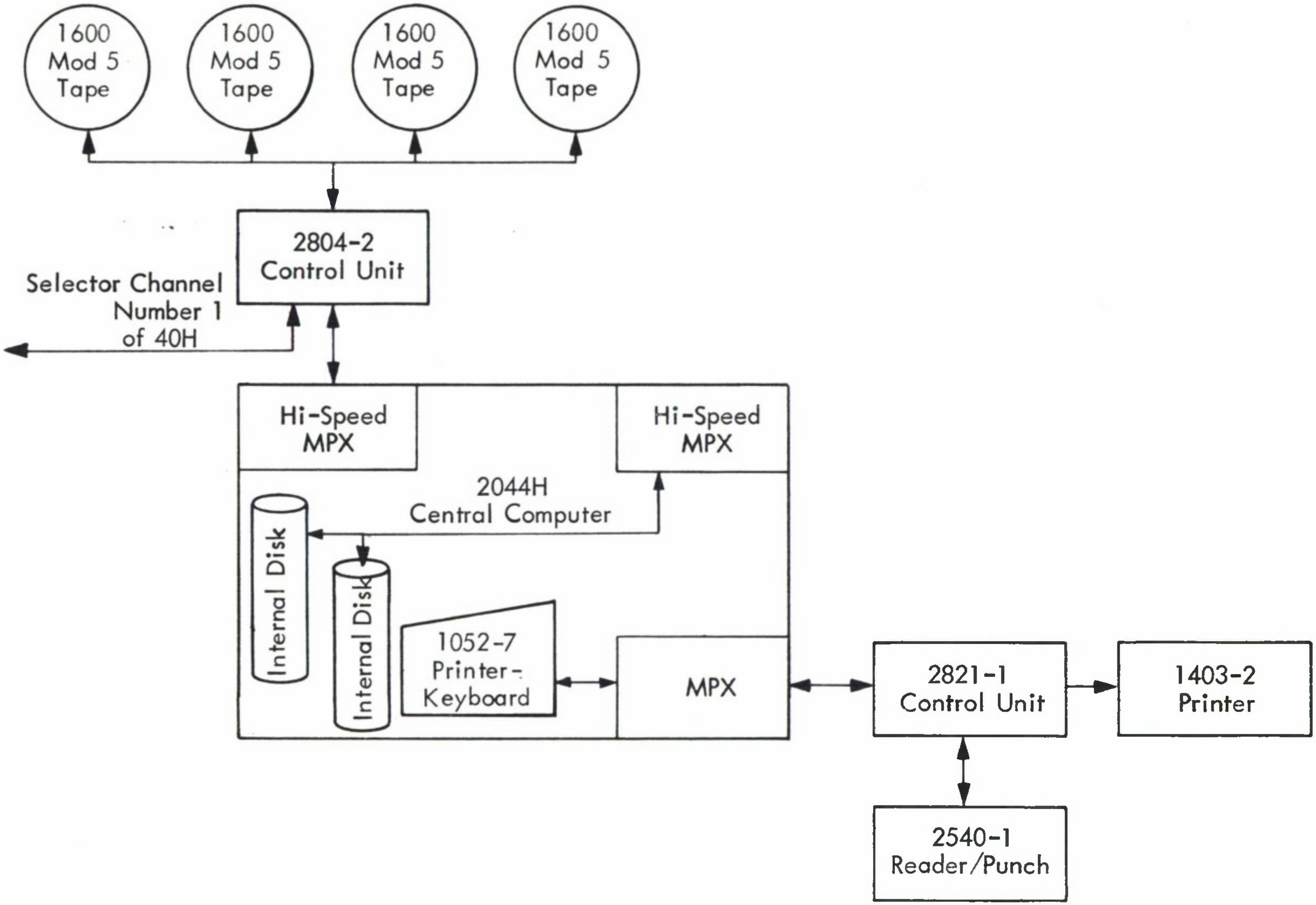


Figure 3. Auxiliary Equipment Configuration

Appendix II

SIGNAL PROCESSING

In this appendix, the technique of automatic time delay generation and the estimated event location accuracy of large arrays are presented. The results of a study on the use of fidelity optimum processing against one of two simultaneously occurring events appear promising and warrant both study continuation and initial experimentation. The accuracy of fixed point arithmetic execution and the estimated execution speed improvement obtained by microprogramming the Fast Fourier Transform is discussed. The results of studies in array optimization are also included.

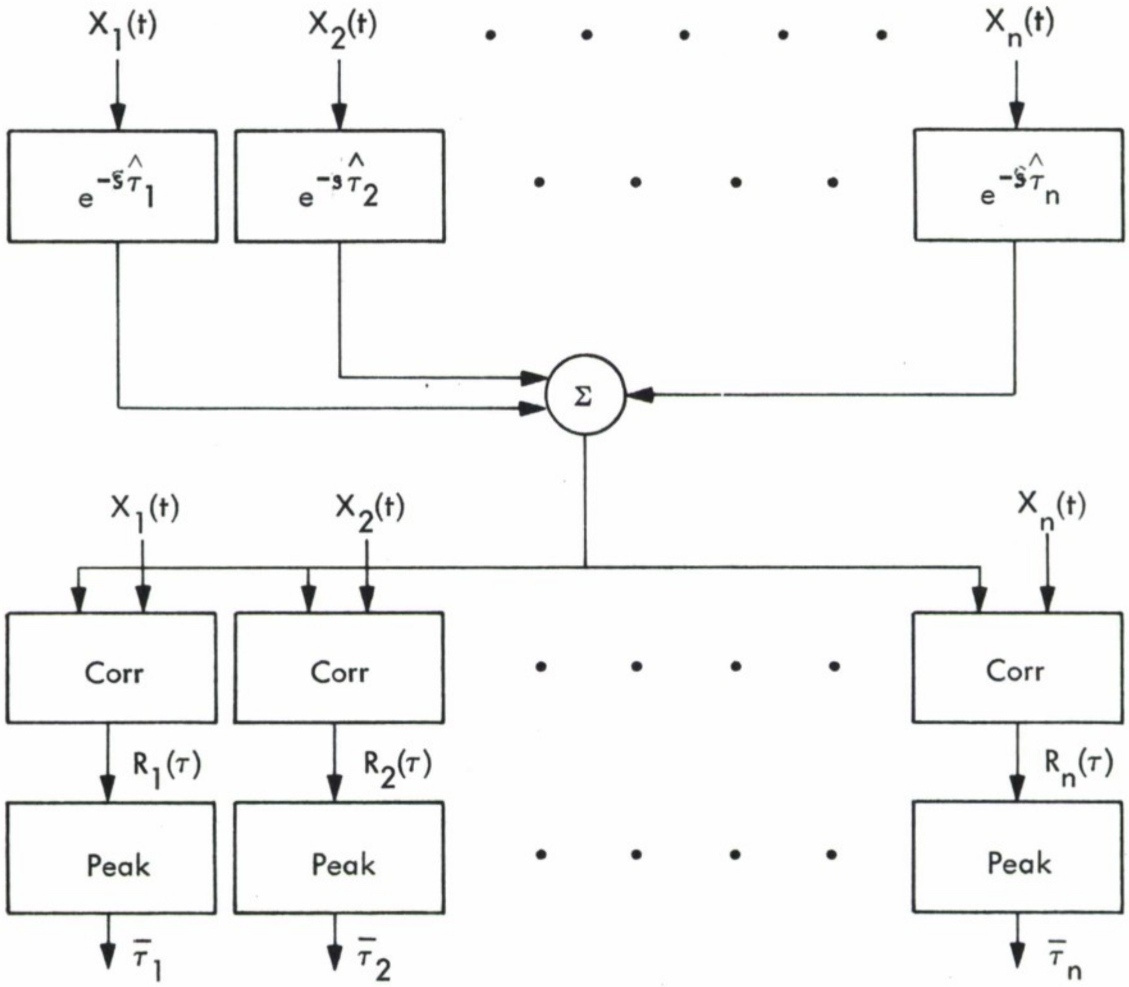
II.1 TIME DELAY CORRELATION

II.1.1 Introduction

The time delay correlation procedure described herein was suggested as a topic of investigation because it was felt that the large volume of time delay data anticipated during system operations could not be efficiently and precisely handled manually. The results of this study and especially its implications on event processing warrant further priority considerations. In the present section, the method and some experimental results are given and the effects of incorporating wavefront analysis are also discussed. The potential effects of the findings on system performance and recommendations are given in the conclusions.

II.1.2 Procedure

Depicted in Figure 4 is the formulation of the time delay correlation procedure. It employs plane wave delays with regional corrections (see Appendix III-2) as initial values for the time delays ($\tau_1, \tau_2, ---, \tau_n$). The Laplace notation is used to signify the delay process from which the beam [B(t)] is formed. Each input [$X_1(t)$, $X_2(t)$,---,$X_n(t)$] is correlated with the beam. The time ($\bar{\tau}_n$) at which the peak value of the correlation function [$R_n(\tau)$] occurs is selected and used to estimate the correction to the initial time delay values. The quantity α has been introduced to provide a process stability control parameter (i.e. when $\alpha = 1$, full value corrections are applied). In the process programming, operational coding notation and floating point arithmetic have been employed.



$$B(t) = (1/N)\sum_{n=1}^{N} X_n(t - \hat{\tau}_n).$$

$$R_n(\tau) = (1/T)\int_0^T B(t)\, X_n(t - \tau)dt$$

$$R_n(\bar{\tau}_n) = \text{Max } R_n(\tau)$$

$$\hat{\tau}_n(K+1) = \hat{\tau}_n(K) + \propto\{\bar{\tau}_n - \hat{\tau}_n(K)\}$$

Figure 4. Time Delay Correlation

Once initialized, the process is allowed to iterate K times. As soon as $\hat{\tau}_n(K+1)$ is not significantly different from $\hat{\tau}_n(K)$, it is conjectured that the beam power is near maximum and the time delays are optimal.

For the limited testing performed to date, the procedure has been found to be susceptible to false nulls. As expected, it has been observed that the correlation function is sinusoidal and that suboptimal convergence occasionally occurs at an integral number of periods away from maximum. The system's narrow band response can be employed to remove this ambiguity by using wave front analysis techniques to test the observed nulls for credibility. Since the false nulls are large with respect to wave front corrugations, the procedure is expected to yield satisfactory results.

II.1.3 Experimental Results

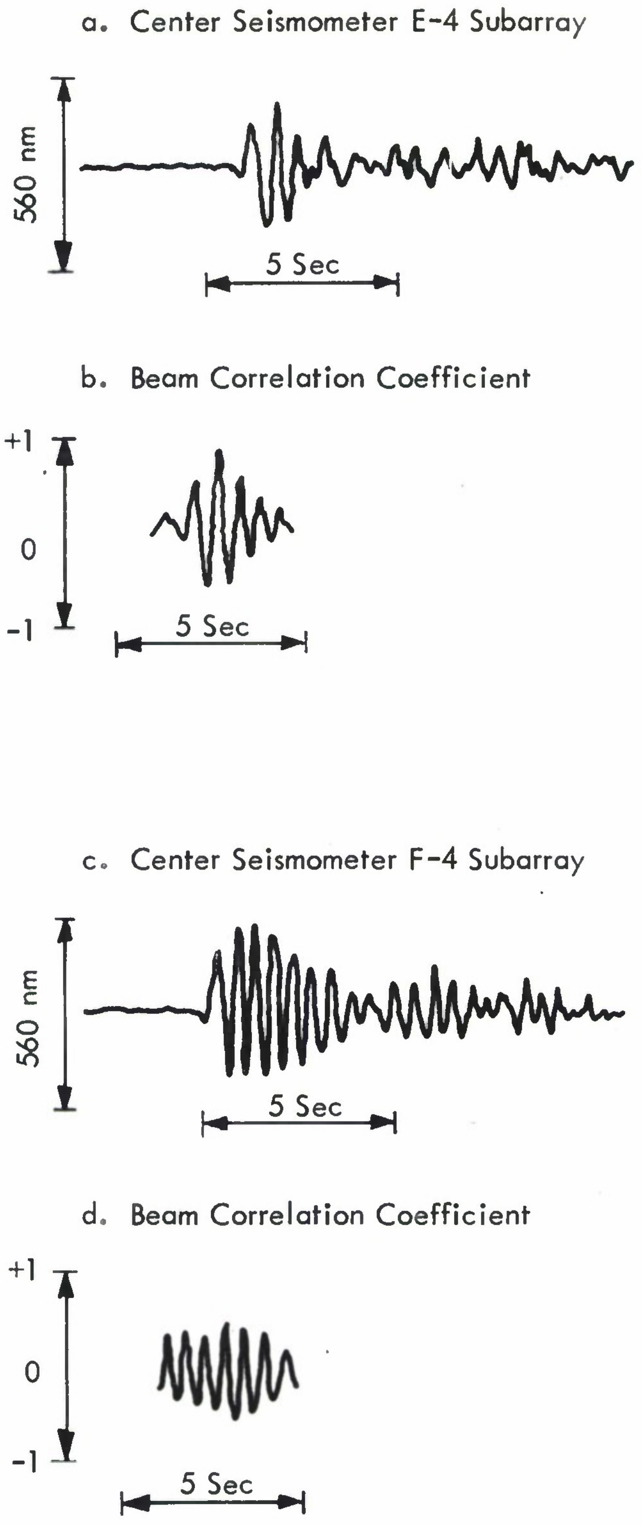
To demonstrate technique, a limited sample of certain experimental results is given. Although use of subarray beams would no doubt yield better results, single seismometers were employed for computational convenience. Depicted in the upper portion of Figure 5 is a well-behaved element trace (a) and its beam correlation function (b). A poor element signal (c) and its beam correlation function (d) are shown by the lower pair. Note that the sensitivity of (b) is much greater than (d) and, hence, for this data, the confidence of estimating the time delay from (b) is significantly greater than that for (d).

The effect of signal-to-noise ratio was also investigated by applying the correlation procedure to scaled Longshot data for which the average input signal-to-noise ratio was significantly reduced. The average of the magnitude of the differences between the correlation time delays and the time delays determined from the unscaled Longshot traces was computed. Denoted by Δ in Figure 6 are the results for three scaled sets of data with input signal-to-noise ratios of 10dB, 5dB and -1.5dB. The effects of repeating correlation after employing wavefront corrections are indicated by □ , in Figure 6.

Typical waveforms of the scaled and unscaled traces are shown in Figure 7. For the latter, the signal-to-noise ratio is almost 35dB and the time delay can be determined with little difficulty. However, when the signal-to-noise ratio is reduced by 36dB so that the input signal-to-noise ratio is negative, and even when the traces are appropriately time-phased as is the case in Figure 7, it is obvious that a manual time delay measurement is impractical.

II.1.4 Conclusions

The importance of accurate time delays cannot be over emphasized. First, a loss in beam gain is experienced[1,4] and the resulting beam defocusing will degrade array/processing performance. Second, event location errors (see Appendix II.2)



a. Center Seismometer E-4 Subarray

b. Beam Correlation Coefficient

c. Center Seismometer F-4 Subarray

d. Beam Correlation Coefficient

Figure 5. Waveform Correlation Traces

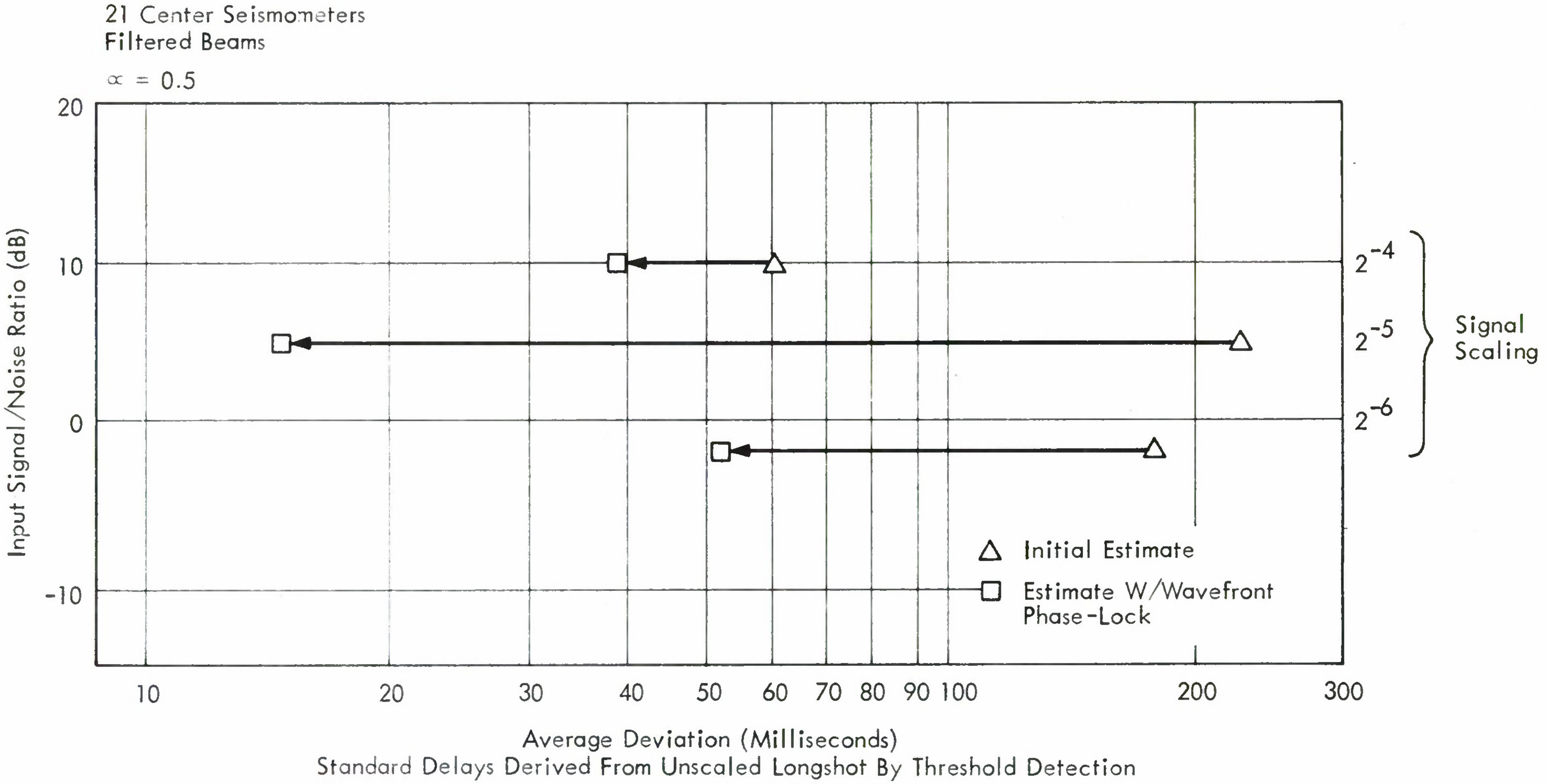


Figure 6. Correlation Estimation of Phase Delays

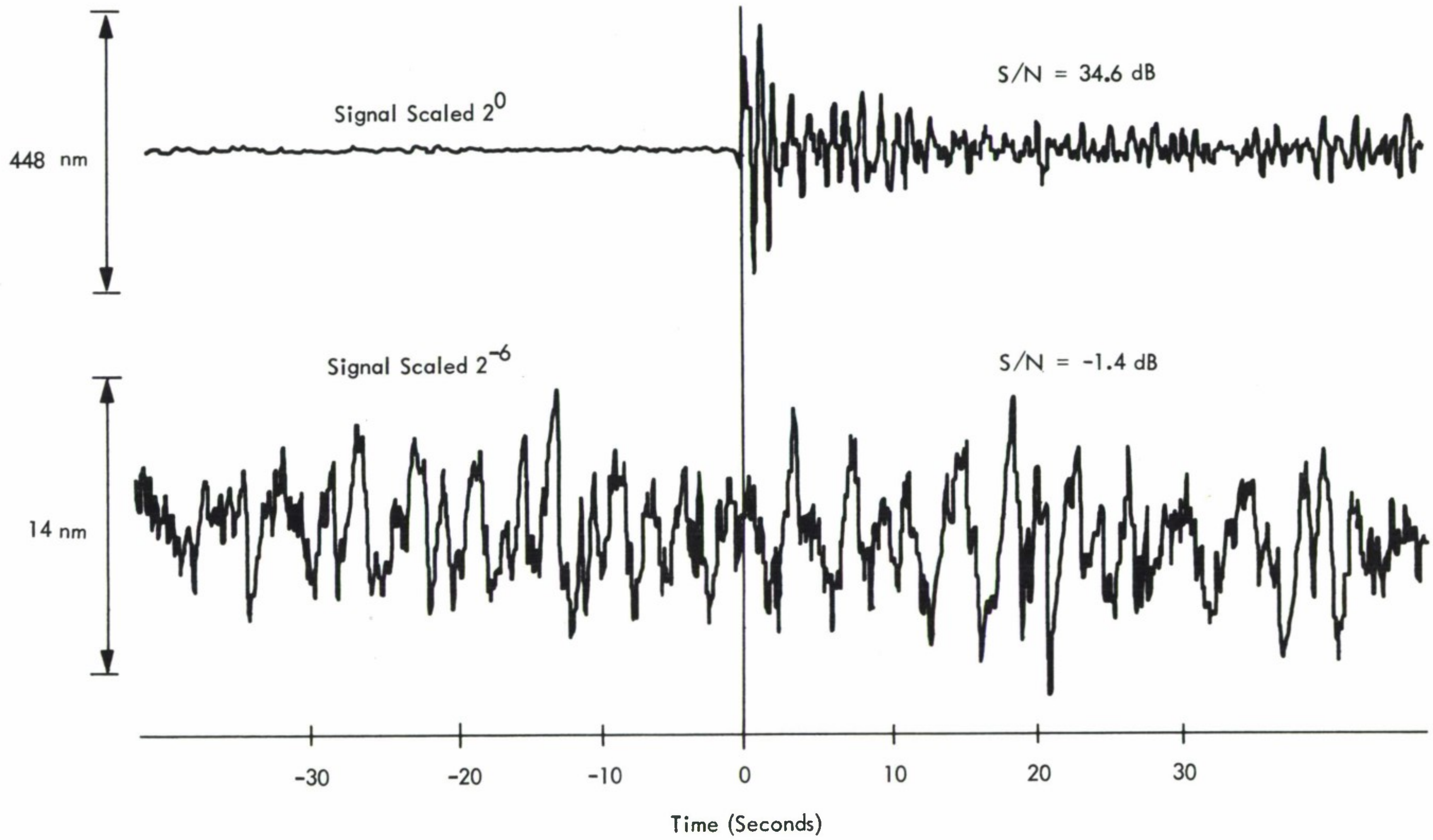


Figure 7. Seismometer Record of Longshot

are strongly dependent on the time delays. Third, the maintenance of both the time delay and location conversion libraries in a form suitable for updating implies an automatic processing procedure for both time delay determination and use.

The method discussed above provides the desired results. The sparse experimental results available to date warrant consideration of the technique as an integral part of event processing.

II.2 EVENT LOCATION ACCURACY

II.2.1 Introduction

The error generated when estimating the location of a seismic event received on a large array can be attributed, in part, to the presence of seismic noise at the time of arrival of the event signal (i.e., to the fact that the signal-to-noise ratio is finite), and, in part, to imperfect "calibration." In the present context the term "calibration" refers to the method and data used to convert wavefront measurements (such as the reciprocal wave velocities u_x, u_y) to a geographic location for the event.

In Section 2, an idealized model is used to examine the effect of seismic noise on the accuracy of the estimated event location. In Section 3, an attempt is made to incorporate the errors resulting from imperfect calibration.

II.2.2 The Effect of Seismic Noise on Estimated Event Location

Because of the presence of seismic noise at the time of arrival of a seismic event at the array site, errors are generated in the estimated values of the reciprocal phase speeds ($\hat{u}_x$, $\hat{u}_y$) of the event wavefront. Hence, even with perfect "calibration", errors in the geographic event location will result. These errors tend to increase with a decreasing signal-to-noise ratio.

To assess the effect of seismic noise in a quantitative manner, we consider an idealized large array model, wherein array beams are formed from subarray beams for which the following behavior is postulated:

a. Seismic noise received at any one subarray is statistically independent from that received at any other subarray

b. The signal waveforms received at all subarrays are identical except for (frequency independent) scale factors and for travel time differences

c. The wavefront shape is accurately known and the time delay for the k-th subarray can be represented as $t_k + x_k u_x + y_k u_y$, where t_k is the (accurately known) station correction for the k-th subarray and (x_k, y_k) are the position coordinates of the center of that subarray.

The first assumption is in reasonable accord with present experience. The second assumption is not always fulfilled: substantial decorrelation has been observed for signal waveforms received on different subarrays. However, by and large, the correlation is satisfactory for a few seconds following the signal onset. The third assumption postulates perfect "calibration" of the wavefront shape. This assumption is realistic if the seismic event originated in close geographic proximity of previously recorded strong seismic events. Where such previously received events are lacking, the assumption is not always valid.

With the postulates, the estimated reciprocal phase speeds ($\hat{u}_x$, $\hat{u}_y$) approximate Gaussian variables with the true quantities (u_x, u_y) as mean values. Furthermore, assuming application of (frequency independent) weighting of the subarray beams to achieve maximum array beam output signal-to-noise ratio, the contours of constant probability density for (u_x, u_y) coincide with the constant-loss contours of the beam pattern centered at (u_x, u_y). Such weighting is desirable, for example, when the subarrays do not all have the same number of seismometers.

Denoting by p the probability for ($\hat{u}_x$, $\hat{u}_y$) to fall outside the beam contour corresponding to a loss of L dB for a center frequency of f Hz, the following approximation holds:

$$p \approx 10^{-0.05 \cdot L \cdot (2\pi f\sigma_0)^{-2}}, \tag{1}$$

where σ_0 is the standard deviation of the timing error for the signal waveform in the LASA beam output and is given by:

$$(2\pi f\sigma_0)^2 = \frac{N}{S} \cdot \frac{1}{2WT}\,. \tag{2}$$

Here S/N is the output signal-to-noise ratio of the array beam steered to the event location (u_x, u_y), while WT is the time-bandwidth product (for LASA, WT ≈ 1). The above formulae provide minimum error bounds and tend to under estimate the errors in (u_x, u_y) particularly for low S/N.

The preceding results conform to the intuitively obvious fact that the location error ($\hat{u}_x$, $\hat{u}_y$) can be minimized by positioning the subarrays in such a manner that the resulting beam pattern has minimal main-lobe width. For the particular case where the subarrays are constrained to be located in a circular area of radius R, they should be positioned on the circumference. In that case formula (1) reduces to:

$$p \approx \exp\left[-(\pi f r R)^2 \cdot \frac{S}{N} \cdot 2WT\right], \tag{3}$$

where r is the distance between the estimated location ($\hat{u}_x$, $\hat{u}_y$) and the true location (u_x, u_y). The root-mean-square of r is given by:

$$\sqrt{E\left[r^2\right]} = \frac{1}{\pi f R}\sqrt{\frac{N}{S} \cdot \frac{1}{2WT}}\,. \tag{4}$$

The error, r, is measured in sec/Km and is related to the (u_x, u_y) plane. To translate it as a geographic location error, it must be multiplied by a sensitivity factor C which generally depends on the range from the event location to array as well as on the azimuth from the true event location to estimated event loeation. After averaging over this azimuth, the factor C varies from about 1.2 x 10^5 to 2.0 x 10^5 Km^2/see. Thus, C = 1.6 x 10^5 Km^2/see appears to be a satisfactory compromise partieularly suitable for ranges from the event location to the array in exeess of 65°.

As an example, eonsider the case where f = 1 Hz, WT = 1. Then, the root-mean-square geographic error is given by:

$$\text{RMS geographic error} \approx \frac{0.36 \text{ x } 10^5}{R} \sqrt{\frac{N}{S}} \text{ Km}, \qquad (5)$$

with the array radius R being measured in Km. This result is depicted graphically in Figure 8.

We emphasize that minimization of the main-lobe width may not be a praetieal design criterion since it can lead to a poor side-lobe structure.

The expressions (3) and (4) presuppose that all subarrays show the same output signal-to-noise ratio. When this is not the case, the value of S/N in (3) and (4) must be derated. For example, when all subarrays have the same number of seismometers, except for one subarray whieh has a larger number, the value of S/N must be derated by less than 0.4 dB, provided that the large subarray has no more than half of all the seismometers in the array and provided that the subarrays are suitably positioned relative to eaeh other.

II.2.3 The Effeet of Imperfect "Calibration" on Location Error

The considerations in Section 2 have assumed perfect "ealibration" (i.e., have disregarded the errors arising when converting the estimated reciproeal phase speeds ($\hat{u}_x$, $\hat{u}_y$) to geographic event location). These errors exist even when the event depth is known or postulated.

The main eause for "calibration" errors lies in the fact that strong seismic events, needed to establish the data base for the "ealibration" teehnique, are rather rare; favor specific geographie areas to the virtual exclusion of other large regions; have finite signal-to-noise ratio; and are sometimes difficult to pinpoint in depth and geographie loeation. These eircumstanees not only impede precise determination of the relative station eorrections required for beamforming, but also hamper the establishment of an aeeurate eonversion method from (u_x, u_y) to geographic loeation.

Although extensive data analyses have been undertaken to obtain the relative station eorreetions for LASA, no explicit study has yet been made to calibrate a conversion method from (u_x, u_y) to geographic location. Presently, ray tracing

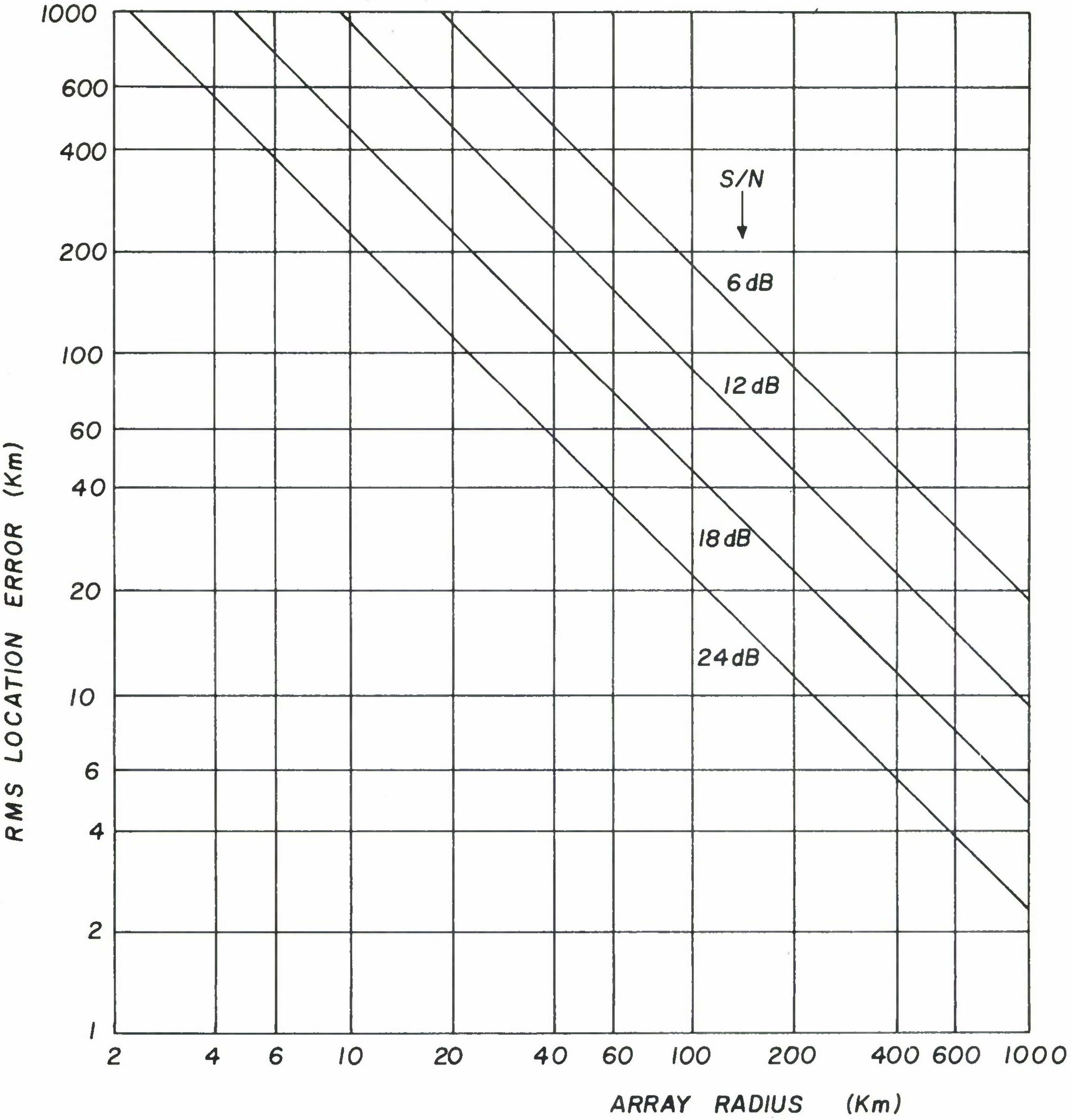


Figure 8. Location Error with Perfect "Calibration"

teehniques, based on the Jeffreys-Bullen tables and others, are used for this conversion, but no systematie effort has been undertaken to compare the ealculated event locations with those obtained from other obse:vations (such as from a network of seismic stations) and to establish corrections for possible bias in the event locations calculated from LASA data. In partieular, little information appears to exist regarding the event location accuracy presently achieved by LASA.

One may conjeeture that the "ealibration" error will eonsist of two portions: one part being proportional to the reciproeal of the array diameter, 2R, and refleeting the effect of the finite array aperture; the second part being independent of aperture size and reflecting the location inaccuracies for seismie events used in the data base. We shall assume that both portions ean be looked upon as random errors when considered on a worldwide basis. Hence, combining these errors in RMS fashion with the error presented by formula (4), we obtain the following expression for the total geographic RMS location error:

$$\text{RMS location error} = \sqrt{\left(\frac{C}{\pi f R}\right)^2 \cdot \frac{N}{S} \cdot \frac{1}{2WT} + \frac{A}{R^2} + B}\ \text{Km}, \tag{6}$$

where the eonstants A and B will depend on the "ealibration" accuracy.

Although in the initial phases of the operation of an array, large "ealibration" errors may exist (reflected in large values of A and B in formula (6)), as time progresses and the data base broadens, the "calibration" errors should decrease and the values of A and B should become smaller. Also, A and B will tend to be less for geographic areas where seismic events are relatively prevalent than for areas where they are rare.

The results of formula (6) have been plotted in Figure 9 using the same data as in Section 2 (i.e., taking f=1 Hz, WT=1, and $C=1.6\times10^5$ Km^2/sec) and choosing $A=10^8$ Km^4 and B=0. This choice for A and B implies a calibration error of 100 Km for an array radius of 100 Km and has been somewhat arbitrary.

It is interesting to eompare this error with the loeation error which would be incurred when the sole contributing eause is assumed to be the fact that the sampling rate is finite, thus limiting the aecuracies with which the time delays for array beamforming ean be implemented. This location error ean be ealeulated from formula (1) provided that the standard deviation σ_0 be defined by:

$$\sigma_0^2 = \frac{(\Delta T)^2}{12.K} \tag{7}$$

(based on the assumption that timing errors due to time sampling are uniformly distributed over the sampling interval ΔT and are uneorrelated between subarrays). Here, K is the number of subarrays. For an array whose subarrays are of equal size and are positioned on a circle with radius R, the RMS error is found to be $2\sigma_0$ C/R. For a sampling rate of ten samples per seeond (ΔT=0.1 sec), K=21 and

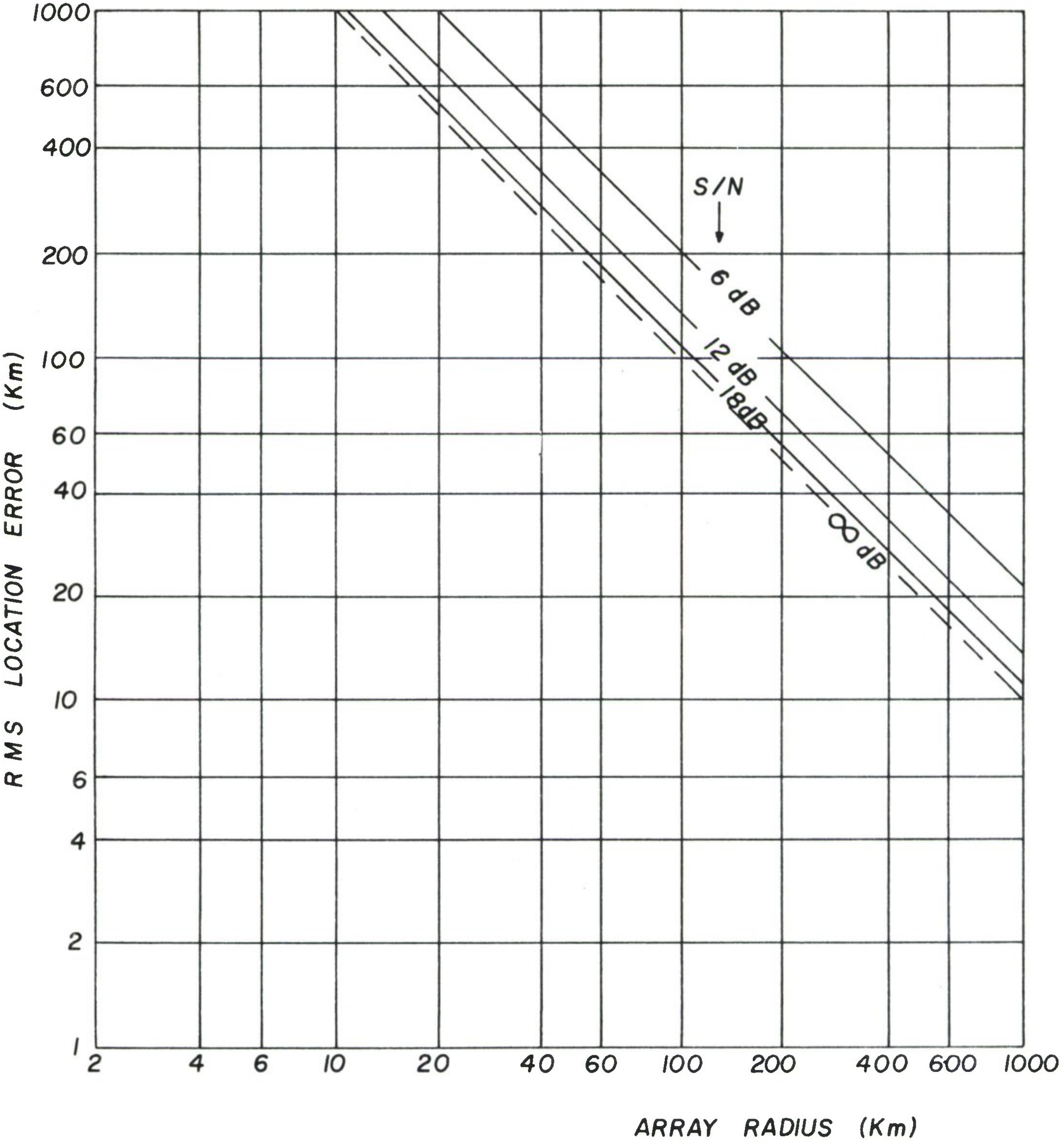


Figure 9. Location Error with Imperfect "Calibration"

$C=2x10^5$ Km^2sec, one finds a location error of 20 Km when $R=100$ Km. Thus, we assumed for Figure 9 a "calibration" error five times larger than the error caused by the sampling rate. For LASA, formula 1 leads to an RMS error of 40 Km, caused by sampling rate.

II.3 FIDELITY OPTIMUM PROCESSING

II.3.1 Introduction and Summary

This appendix contains the results of a theoretical study to estimate the ability of fidelity optimum processing to discriminate against one of two simultaneous events. It compares the signal-to-noise ratio of conventional processing and fidelity optimum processing when the noise consists, in part, of a second, interfering seismic event. Antenna patterns of the resulting optimum processor are also obtained.

Specifically, the noise is assumed to be the sum of an uncorrelated background and an interfering event. The background is assumed to have the same power spectrum at each seismometer. The interfering event is also assumed to be a second order stochastic process with a power spectrum identical to that of the background noise.

In this situation, the gain from optimum processing is a function of the array configuration, the radius of the array, the ratio of the power of the interfering event to the background noise power, and the magnitude of the difference in inverse phase velocity between the event of interest and the interfering event. Three array configurations were considered: The LASA configuration, 21 seismometers equally spaced on a circle and five seismometers equally spaced on a circle. If we let R be the radius of the array and $|\vec{\Delta u}|$ be the magnitude of the difference in inverse phase velocity between the two events, then the gain is a continuous function of $R\,|\vec{\Delta u}|$ and we determined this function over the range of interest. Ratios of interfering event noise to background noise varying from 0.05 to 100 were considered.

Substantial gains were found to occur for $R|\vec{\Delta u}| \geq 0.2$ for the LASA configuration and for $R|\vec{\Delta u}| \geq 0.15$ for the other configurations. In the above ranges of $R\,|\vec{\Delta u}|$ and for large ratios of interfering event noise power to background noise power, the optimum processor essentially eliminated the interfering event.

II.3.2 The Power Output of a Linear Processor

We suppose we have K seismometers and designate their sampled outputs by $X_1(t),\ldots, X_K(t)$. We assume:

$$X_k(t) = S(t) + N_k(t), \qquad k = 1, \ldots, K; \tag{8}$$

where each seismometer output is the sum of a signal, which we are interested in estimating, and noise. The signals are assumed to be lined up or in phase from seismometer to seismometer. We suppose the noises, $N_k(t)$, to form a multivariate second order process with a spectral matrix $P_{kk'}(f)$. That is,

$$E\{N_k(t)\ N_{k'}(t+\tau)\} = \phi_{kk'}(\tau)$$

and

$$P_{kk'}(f) = \Delta t \sum_{\tau=-\infty}^{\infty} \phi_{kk'}(\tau)\ e^{-2\pi jf\tau}, \tag{9}$$

where Δt = sampling interval.

Now we will consider the case where a linear operator with impulse response $\theta_k(s)$ and transfer function $A_k(f)$ is applied to the kth seismometer output for $k=1, \ldots, K$. These filter outputs are summed to obtain the output of the array, $Y(t)$. This is illustrated in Figure 10.

$\theta_k(s)$ and $A_k(f)$ are related by:

$$\theta_k(s) = \int_{-w}^{w} A_k(f) e^{2\pi jfs}\ df,$$

and

$$A_k(f) = \Delta t \sum_{s=-\infty}^{\infty} \theta_k(s)\ e^{-2\pi jfs} \tag{10}$$

where $w = \frac{1}{2\Delta t}$.

We will be interested in the output noise power of an array with such processing. The output noise power spectrum (see Figure 11) is given by:

$$\sum_{k,k'} \widetilde{A_k(f)}\ A_{k'}(f)\ P_{kk'}(f). \tag{11}$$

By $\widetilde{A_k(f)}$, we mean the complex conjugate of $A_k(f)$.

Hence, the output noise power is given by:

$$\int_{-w}^{w} \sum_{k,k'} \widetilde{A_k(f)} A_{k'}(f) P_{kk'}(f)\ df. \tag{12}$$

II.3.3 Form of the Fidelity Optimum Processor

In fidelity optimum processing, the goal is to minimize the output noise power subject to the constraint that the signal is undistorted. Now, the transfer function seen by the signal is $\sum_{k=1}^{K} A_k(f)$; hence, the goal of fidelity optimum processing is to minimize:

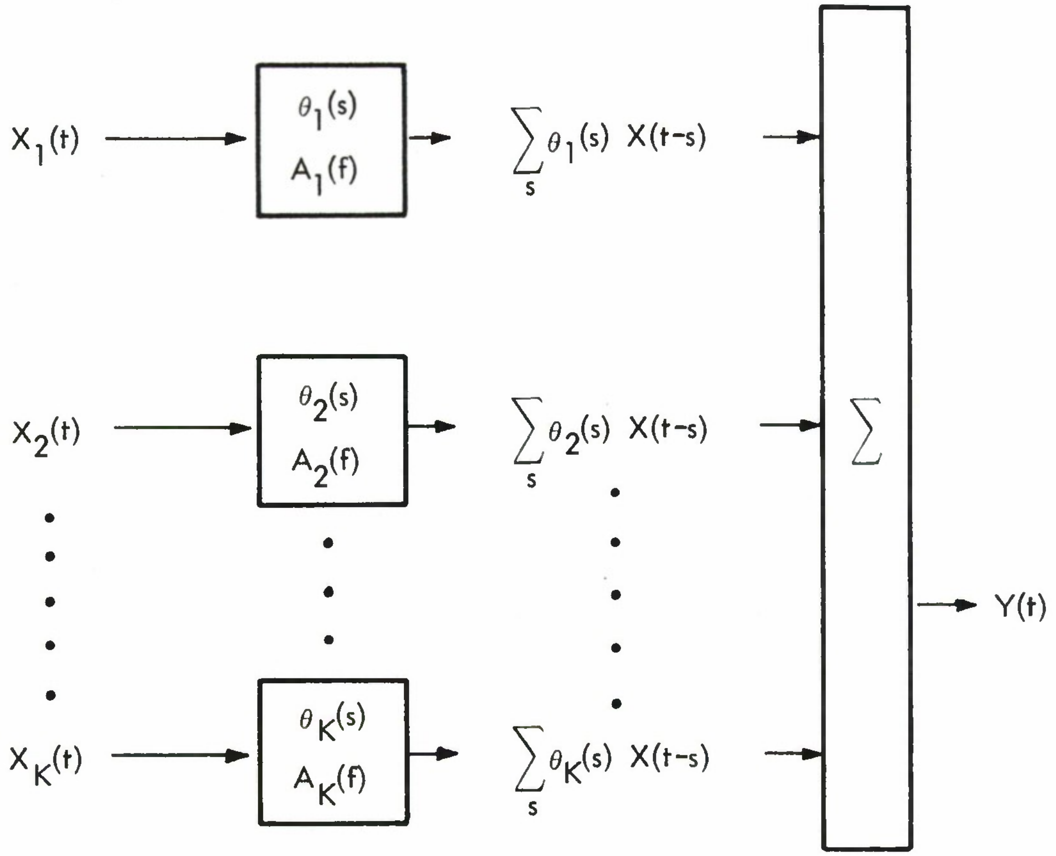


Figure 10. Linear Processor

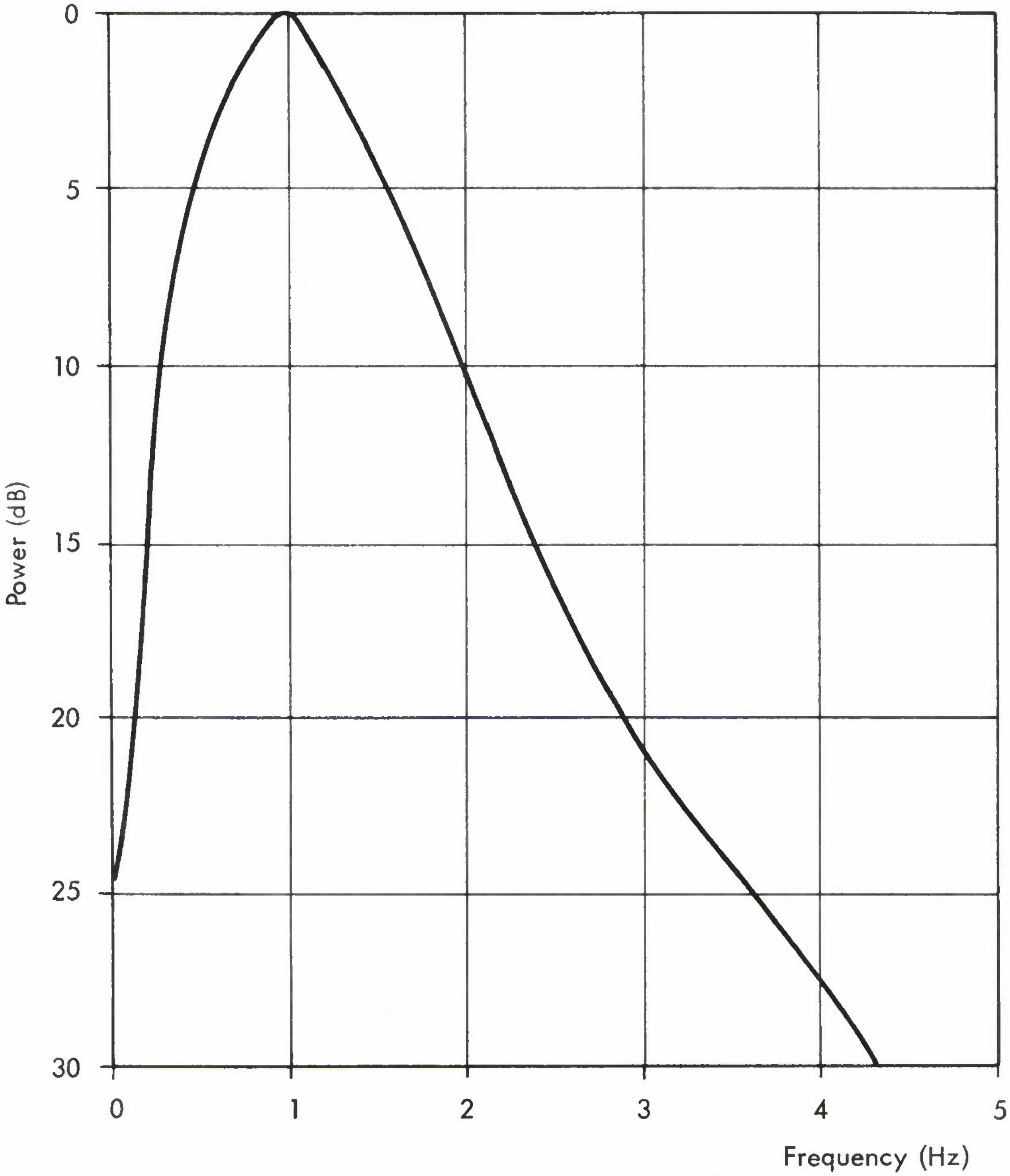


Figure 11. Noise Spectrum Used

$$\int_{-W}^{W} \sum_{k,k'} \widetilde{A_k(f)}\ A_{k'}(f)\, P_{kk'}(f)\ df, \tag{13}$$

subject to the condition:

$$\sum_{k} A_k(f) = 1. \tag{14}$$

It can be shown (see Capon[7] or Vanderkulk[2,8]) that the $A_k(f)$ satisfying the above are given by:

$$A_k(f) = \frac{\sum_{k'} P^{-1}_{kk'}(f)}{\sum_{k,k'} P^{-1}_{kk'}(f)}\ , \tag{15}$$

where $P^{-1}_{k,k'}(f)$ is the k,k' element of the inverse of the KxK matrix $P_{kk'}(f)$.

II.3.4 Inverting the Spectral Matrix

$P_{kk'}(f)$ for fixed f is a matrix of complex numbers. If we let:

$$P_{kk'}(f) = C_{kk'}(f) + j\ Q_{kk'}(f) \tag{16}$$

and

$$P^{-1}_{kk'}(f) = D_{kk'}(f) + j\ E_{kk'}(f),$$

then it is easily seen that the following relationship holds between real 2K x 2K matrices

$$\begin{pmatrix} C & -Q \\ Q & C \end{pmatrix}^{-1} = \begin{pmatrix} D & -E \\ E & D \end{pmatrix}. \tag{17}$$

This relationship can be used to find $P^{-1}_{kk'}(f)$ with a real matrix inversion routine.

We can also use the above form to show that

$P^{-1}_{kk'}(f) = \widetilde{P^{-1}_{k'k}}(f)$ and, hence, that:

$$\sum_{k,k'} P^{-1}_{kk'}(f) = \sum_{k,k'} \text{Re}\left\{P_{kk'}(f)\right\} = \sum_{k,k'} D_{kk'}(f) \tag{18}$$

Consider first the matrix $\begin{pmatrix} C & -Q \\ Q & C \end{pmatrix}$. Since C is symmetric and Q is skew symmetric, this matrix is symmetric. Hence, its inverse $\begin{pmatrix} D & -E \\ E & D \end{pmatrix}$, is symmetric which means that $E^T = -E$.

By E^T we mean the transpose of E. This is equivalent to $P^{-1}_{kk'}(f) = \widetilde{P^{-1}_{k'k}}(f)$.

II.3.5 A Convenient Form for the Output Noise Power with Optimum Processing

We saw earlier that the output noise power in optimum processing was given by:

$$\int_{-w}^{w} \cdot \sum_{kk'} \widetilde{A_k(f)}\, A_{k'}(f)\, P_{kk'}(f)\, df. \tag{19}$$

This is an integral over a quadratic form which can be written as:

$$\sum_{kk'} \widetilde{A_k}(f)\, A_{k'}(f)\, P_{kk'}(f) = \tilde{A}^T\, P\, A,$$

$$\text{where } A = \begin{pmatrix} A_1(f) \\ \vdots \\ A_K(f) \end{pmatrix}, \tag{20}$$

and P is the cross spectral matrix. We have suppressed f for simplicity. However, we also saw that:

$$A_k(f) = \frac{\sum_{k'} P^{-1}_{kk'}(f)}{\sum_{kk'} P^{-1}_{kk'}(f)} = P^{-1}\Lambda\ , \tag{21}$$

where

$$\Lambda = \frac{1}{\sum_{kk'} P^{-1}_{kk'}(f)} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Hence, $\tilde{A}^{\tau} \; P \; A = \tilde{\Lambda}^{\tau}\tilde{P}^{\tau^{-1}} \; P \; P^{-1} \; \Lambda = \tilde{\Lambda}^{\tau}\tilde{P}^{\tau^{-1}} \; \Lambda$

However, since $P^{-1}_{kk'}(f) = \widetilde{P^{-1}_{kk'}(f)}$, we have $\tilde{\Lambda} = \Lambda$ and $\tilde{P}^{\tau^{-1}} = P^{-1}$

Hence, finally :

$$\tilde{A}^{\tau} \; P \; A = \Lambda^{\tau} \; P^{-1} \; \Lambda = \frac{\sum_{kk'} P^{-1}_{kk'}(f)}{\left(\sum_{kk'} P^{-1}_{kk'}(f)\right)^{2}} = \left(\sum_{kk'} P^{-1}_{kk'}(f)\right)^{-1} \tag{22}$$

or

$$\tilde{A}^{\tau} \; P \; A = \left(\sum_{kk'} \mathrm{Re}\{P^{-1}_{kk'}(f)\}\right)^{-1}. \tag{23}$$

This yields:

$$\text{output noise power} = \int_{-W}^{W} \sum_{kk'} \widetilde{A_k(f)} \; A_{k'}(f) \; P_{kk'}(f) \; df \tag{24}$$

or

$$= \int_{-W}^{W} \left(\sum_{kk'} \mathrm{Re}\{P^{-1}_{kk-1}(f)\}\right)^{-1} df. \tag{25}$$

II.3.6 The Output Noise Power in Conventional Processing

In conventional processing the output noise power is given by:

$$\int_{-W}^{W} (1/K^2) \sum_{kk'} \mathrm{Re}\{P_{kk'}(f)\} \; df$$

II.3.7 The Special Case of Noise Consisting of a Single Seismic Disturbance Plus an Uncorrelated Background

Suppose the noise $n_k(t)$ to be the sum of two noises $n_{b,k}(t)$ and $n_{c,k}(t)$. Further suppose $n_{c,k}(t)$ to be a seismic disturbance with delays $\tau_1, \ldots, \tau_K$ relative to the delays of S(t). Then:

$$n_{c,k}(t) = n_c(t - \tau_k). \tag{26}$$

Further suppose that n_c (t) is a second order process with power spectrum P_c (f) and autocovariance function ϕ (τ). Then:

$$E\{n_{c,k}(t)\ n_{c,k}(t+\tau)\} = E\{n_c(t-\tau_k)\ n_c(t-\tau_{k'}+\tau)\} \tag{27}$$

$$= \phi(\tau+\tau_k-\tau_{k'})$$

Taking the Fourier transform we obtain P_c (f) $e^{2\pi jf(\tau_k - \tau_{k'})}$ as the contribution to the cross spectrum $P_{kk'}$ (f).

Now $n_{b,k}(t)$ is assumed to be a background noise which is uncorrelated from seismometer to seismometer and uncorrelated with the seismic disturbance. If it has a power spectrum P_b (f) then the cross spectra are given by

$$P_{kk'}(f) = P_b(f)\ \delta\ (k,k') + P_c(f)\ e^{2\pi jf(\tau_k - \tau_{k'})}; \tag{28}$$

or if we let $P_c(f) / P_b(f) = M(f)$,

$$P_{kk'}(f) = P_b(f)\left\{ \delta\ (k,k') + M(f)\ e^{2\pi jf(\tau_k - \tau_{k'})}\right\}, \tag{29}$$

where δ (kk') is the Kronecker delta. From Vanderkulk[8] and Kelly and Levin[23] we have:

$$P^{-1}_{kk'}(f) = \frac{1}{P_b(f)}\ \{\delta\ (kk') - M(f)\ e^{2\pi jf(\tau_k - \tau_{k'})} / (1 + KM(f))\}. \tag{30}$$

Using the above we can write down expressions for the transfer functions A_k (f) and the noise power output for optimal and conventional processing. The transfer function is:

$$A_k(f) = \frac{1 - \dfrac{M(f)}{1+KM(f)}\ e^{2\pi jf\tau_k} \displaystyle\sum_{k'} e^{-2\pi jf\tau_{k'}}}{K - \dfrac{M(f)}{1+KM(f).} \displaystyle\sum_{kk'} e^{2\pi jf(\tau_k - \tau_{k'})}}\ . \tag{31}$$

The output noise (optimum processing) is:

$$\int_{-W}^{W} P_b(f) \left(K - \frac{M(f)}{1+KM(f)} \sum_{kk'} e^{2\pi jf(\tau_k - \tau_{k'})} \right)^{-1} df, \tag{32}$$

and the output noise power (conventional processing) is:

$$\int_{-W}^{W} P_b\,(f)\;\{K+M\,(f) \sum_{kk'} e^{2\pi jf\,(\tau_k\; -\; \tau_{k'})}\} \;/\; K^2 \;\; df. \qquad (33)$$

The calculations can be simplified by noticing that:

$$\sum_{kk'} e^{2\pi jf\,(\tau_k\; -\; \tau_{k'})} = \left|\sum_{k} e^{2\pi jf\,\tau_k}\right|^2 . \qquad (34)$$

II.3.8 Determination of Delays

We suppose K seismometers are located by the vectors $\vec{p}_k$ with polar coordinates (Rr_k, γ_k). R is a parameter which controls the overall radius of the array. We let $\vec{u}_1$ be the inverse phase velocity of the event of interest, $\vec{u}_2$ be the inverse phase velocity of the interfering event and $\vec{\Delta u}=\vec{u}_1-\vec{u}_2$. Further we let $\vec{\Delta u}$ have polar coordinates $(\,|\vec{\Delta u}\,|,\ \phi)$. With these definitions the delays, τ_k, of the interfering event with respect to the event of interest are given by

$$\tau_k = \vec{\Delta u}\ .\ \vec{p}_k \qquad (35)$$

$$= R\ |\vec{\Delta u}|\ r_k\ \cos\,(\gamma_k\ -\ \phi)$$

In our investigation the r_k will be normalized so that their maximum is approximately equal to one. Hence, the delays are a function of R $|\vec{\Delta u}|$ and ϕ.

II.3.9 The Antenna Pattern

For the class of linearly processed arrays under consideration, the antenna gain in the direction with delays $\tau_1, \ldots, \tau_k$ is:

$$\text{antenna gain } (f,\ \tau_1, \ldots, \tau_k) = \left|\sum_{k} A_k\ (f)\ e^{-2\pi jf\,\tau_k}\right|^2 \qquad (36)$$

This is a power gain, and if we have an input from this direction with power spectrum P(f), the power output will be:

$$\int_{-W}^{W} P(f) \left| \sum_{k} A_k(f)\ e^{-2\pi jf\tau_k} \right|^2 df. \tag{37}$$

II.3.10 Specific Calculations

We now give the results of some calculations to determine the ability of optimum processing to discriminate against the noise described in Section II.3.7 (i.e., noise consisting of a single seismic disturbance plus an uncorrelated background).

In these calculation, we assumed P_c (f) and P_b(f) to have the same shape: i.e., $P_c(f) = P(f) = MP_b(f)$. We took as the shape of P(f) a smooth approximation to the spectrum of the output of an individual seismometer for the Kamchatka earthquake (April 8, 1966) as given in Figure 3-43, Page 3-130 of Ref. 4.

As the ratio M of the spectra P_c (f) and P_b(f), we took values from 0.05 to 100. We considered three configurations of seismometers. The first was that for the 21 subarrays of the present LASA. The r_k and γ_k are given in Table 1. The second was an array of 21 seismometers located equally spaced on the circumference of a circle. The third was an array of five seismometers located equally spaced on the circumference of a circle.

For these configurations and for a wide range of directions of arrival of the unwanted seismic event, we obtained the noise power output for optimum processing and for conventional processing. We also computed the ratio of the conventional noise power output to the optimum noise power output. In addition, we calculated antenna patterns for both a sample case of optimum processing and conventional processing.

In these calculations, we assumed that the array was pointed in the direction corresponding to all delays equal to zero. The noise event will then have a set of delays, not all equal to zero, which we saw in Section II.3.8 are functions of $R\ |\vec{\Delta u}|$ and ϕ, where R is the radius of the array and ($\vec{\Delta u}$, ϕ) are polar coordinates of the difference in inverse phase velocity of the event of interest and the interfering event. We did the calculations for various values of ϕ and generally for a range of $R|\vec{\Delta u}| \geq 0$. The results are plotted against $R|\vec{\Delta u}|$ with ϕ fixed.

II.3.10.1 Results of the LASA Configuration

In Figure 12, we have plotted the noise power output for conventional processing and, in Figure 13, the noise power output for optimum processing as a function of the direction of the interfering seismic event. This is done for values of M from 0.05 to 100. Specifically, the noise spectral matrix is assumed to be:

Table 1

LASA COORDINATES

Instrument Group	Normalized Position	
	r_k	γ_k
A0	0.000	
B1	0.123	55.2°
B2	0.007	141.7°
B3	0.082	246.1°
B4	0.090	346.7°
C1	0.183	23.8°
C2	0.163	97.8°
C3	0.131	192.5°
C4	0.128	293.7°
D1	0.305	56.8°
D2	0.263	141.9°
D3	0.252	231.9°
D4	0.308	336.1°
E1	0.542	13.6°
E2	0.686	106.9°
E3	0.607	183.3°
E4	0.538	278.3°
F1	1.093	46.4°
F2	1.037	147.4°
F3	1.036	219.4°
F4	0.973	235.4°

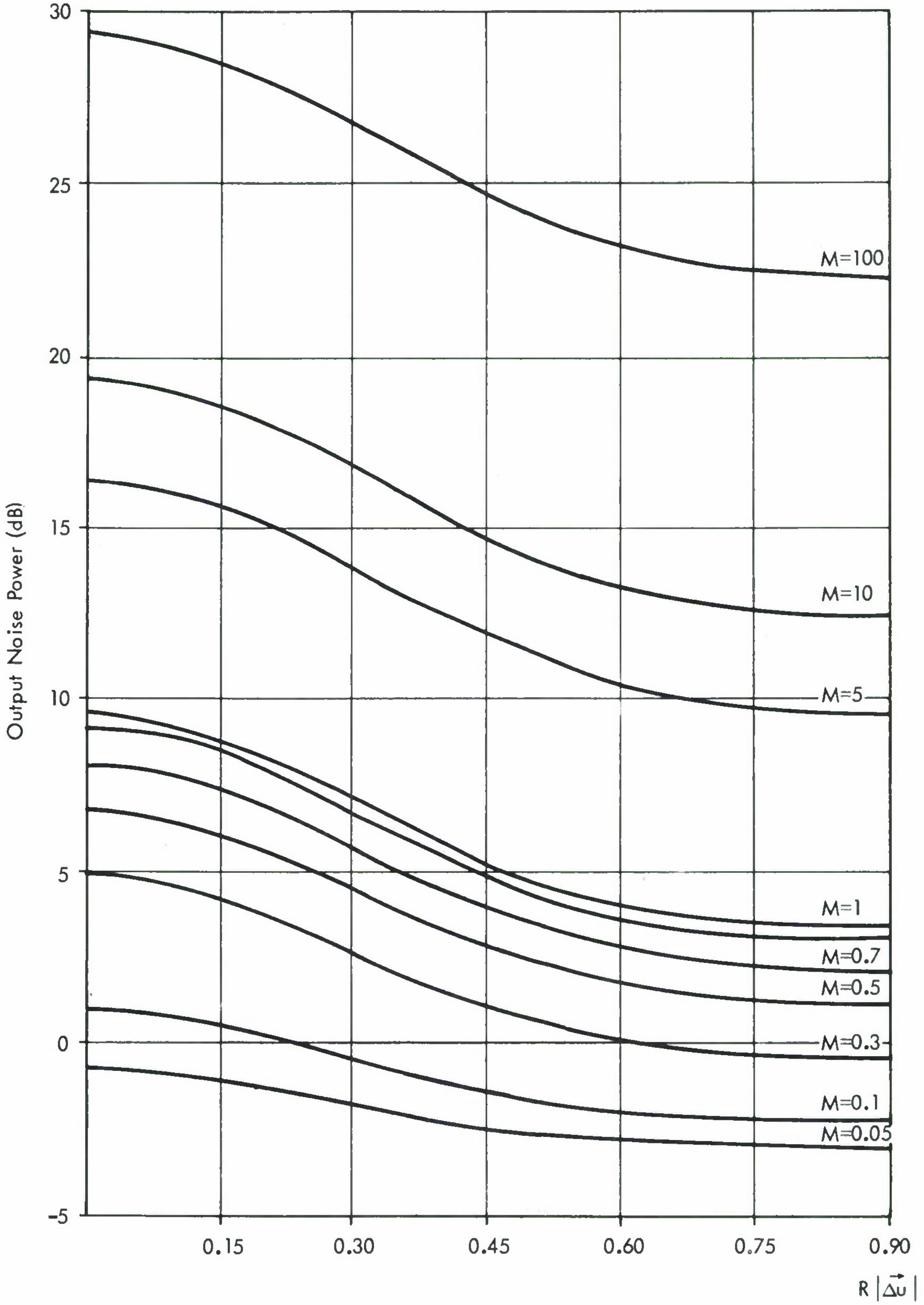


Figure 12. Output Noise Power, Conventional Processing, LASA

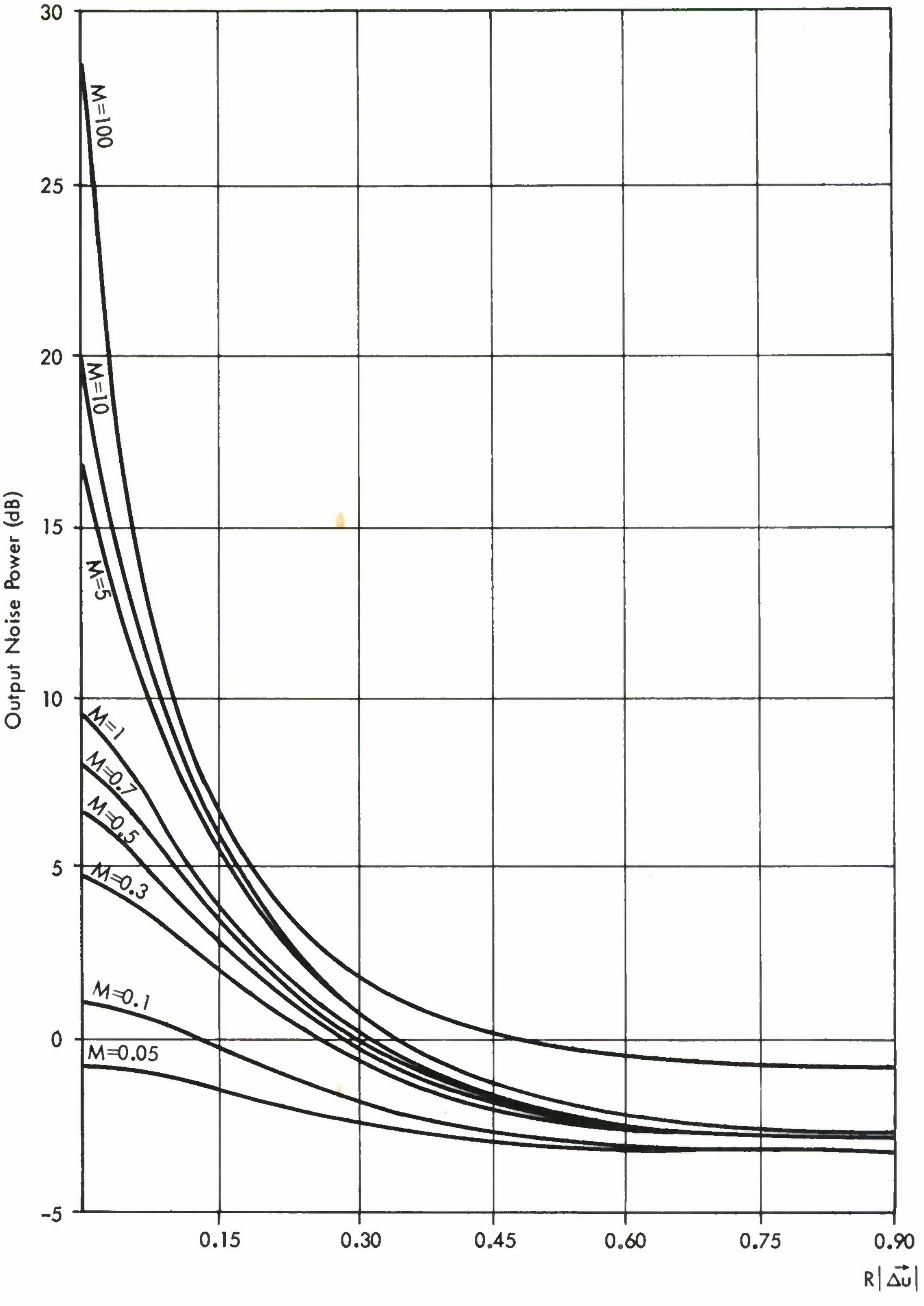


Figure 13. Output Noise Power, Optimum Processing, LASA

$$P_{kk'}(f) = P(f)\ \{\delta\ (kk') + Me^{2\pi jf\ (\tau_k - \tau_{k'})}\}\ , \qquad (38)$$

where the τ_k and $\tau_{k'}$ are determined by the direction of the interfering seismic event relative to the event of interest as described in Section II.3.8. We have assumed $\phi = 0$ (i.e., all events are arriving from the same relative azimuthal direction) and varied the $R|\overrightarrow{\Delta u}|$ for different values of M.

In Figure 14, we have plotted the dB gain in noise rejection of optimum processing over conventional processing for the LASA configuration and values of M from 1 to 100. Notice that substantial gains occur for $R|\overrightarrow{\Delta u}| \geq 0.2$.

In Figure 15, we have plotted the antenna gain of the optimum processor for an event with $R|\overrightarrow{\Delta u}| = 0.3$, $\phi = 0$ and $M = 10$. Cuts radially out from the center are plotted for = 0, $\pi/8, \ldots, \pi$. Notice the sharp null in the direction of the interfering event. Figure 16 gives several radial cuts of the conventional antenna pattern.

II.3.10.2 Results for 21 Seismometers Located Equally spaced on a Circle

Figure 17 contains the noise power output for conventional processing and Figure 18 the noise output power for optimal processing for the case for 21 seismometers equally spaced on a circle for M from 0.05 to 10. In Figure 19, we have plotted the dB gain in noise rejection from optimum processing for M from 1 to 10. There is a significant improvement over the 21 seismometer configuration of the LASA array. Figure 20 contains the radial cut of the optimum processor antenna pattern through the interfering event for the same interfering event depicted in Figure 15.

II.3.10.3 Results for Five Seismometers Located Equally Spaced on a Circle

Figures 21, 22, 23, and 24 give some results for a configuration of five seismometers located equally spaced on a circle. Figure 23 gives the antenna pattern of the optimum processor again for the same interfering event as in Figures 15 and 20. Figure 24 gives the antenna pattern of the conventional processor for $0 \leq \phi \leq \pi/5$. The remainder of this pattern can be obtained by symmetry.

II.3.11 Optimum Processing Consisting of a Single Weighting of Each Seismometer Output Before Addition

Suppose we wish to restrict our processing to simple weighted sum of the seismometer outputs. That is, we wish our output to be of the form:

$$y(t) = \sum_k c_k\ x_k(t), \qquad (39)$$

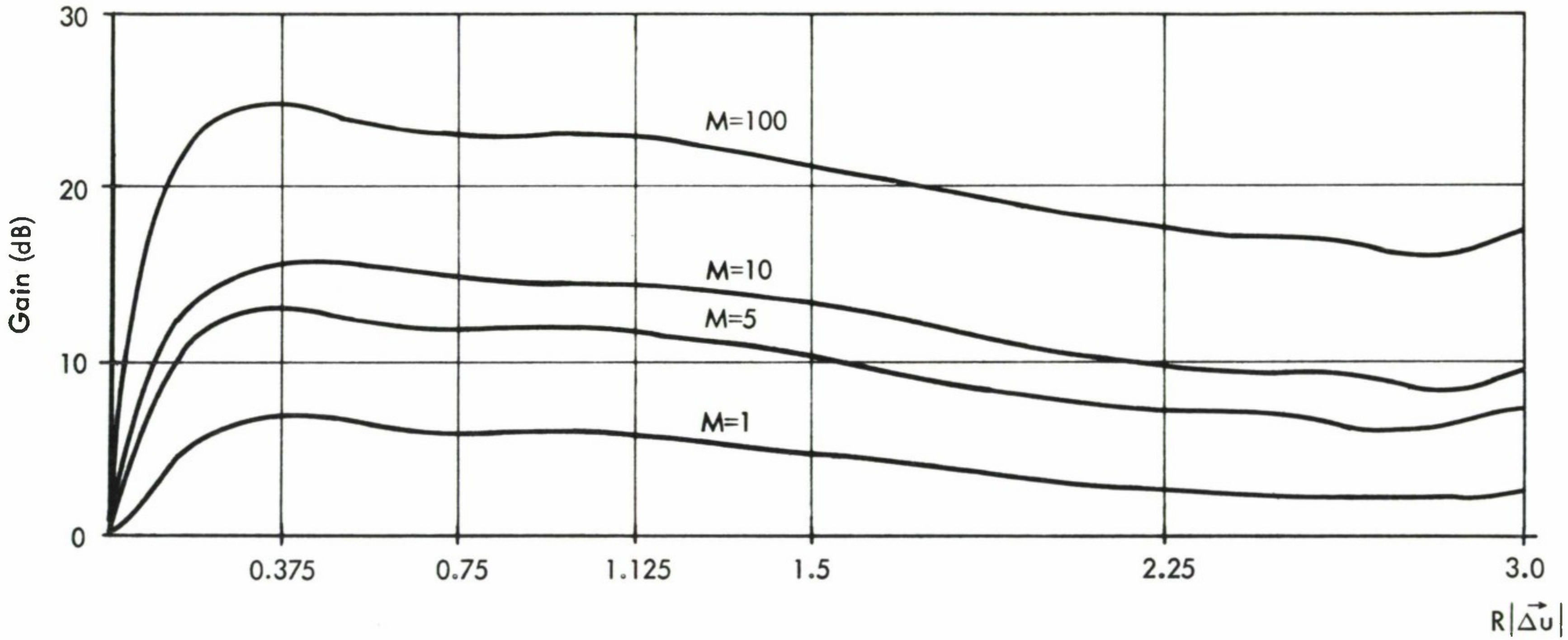


Figure 14. dB Gain Due to Optimum Processing, LASA Configuration

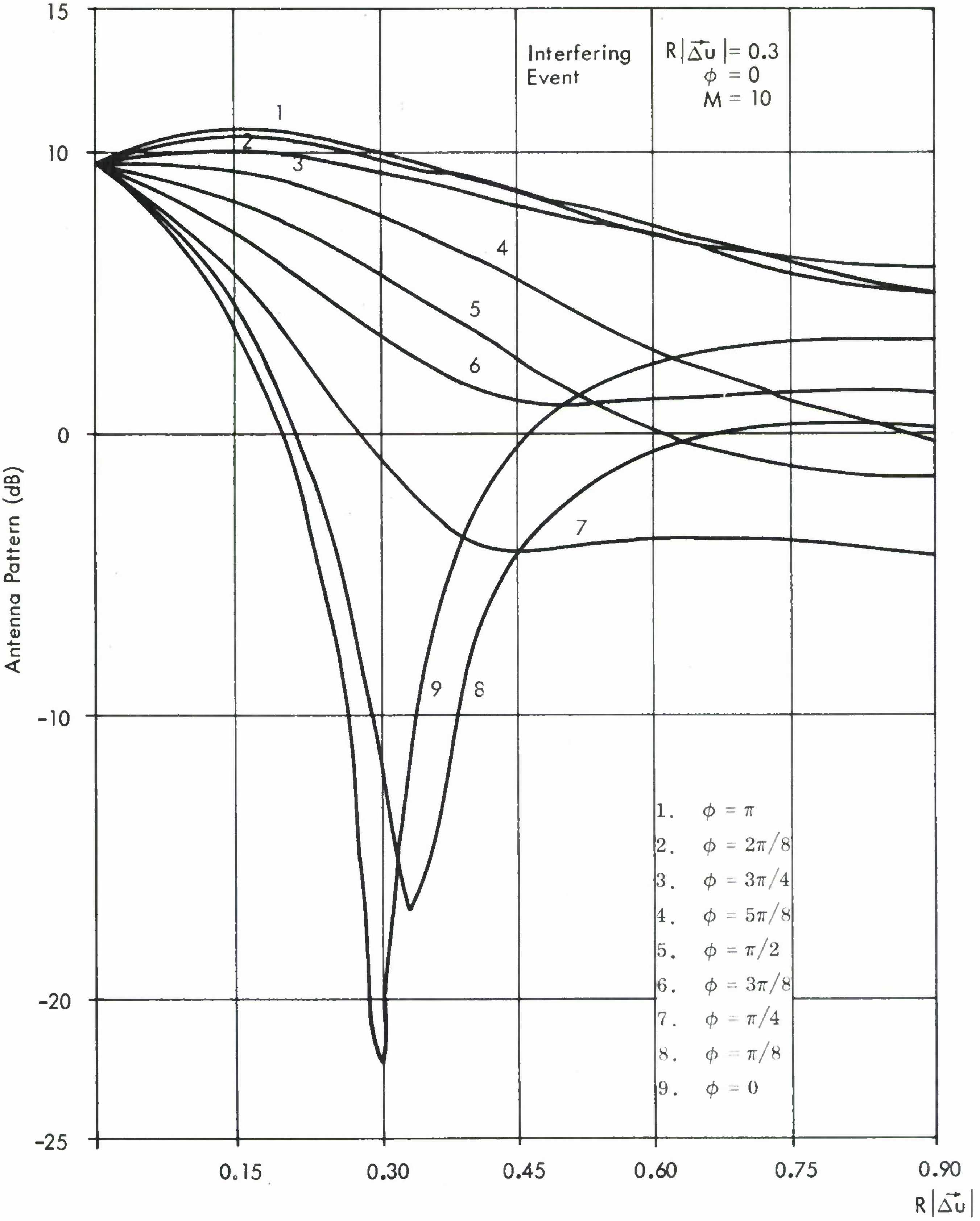


Figure 15. Antenna Pattern
Optimum Processing
Interfering Event

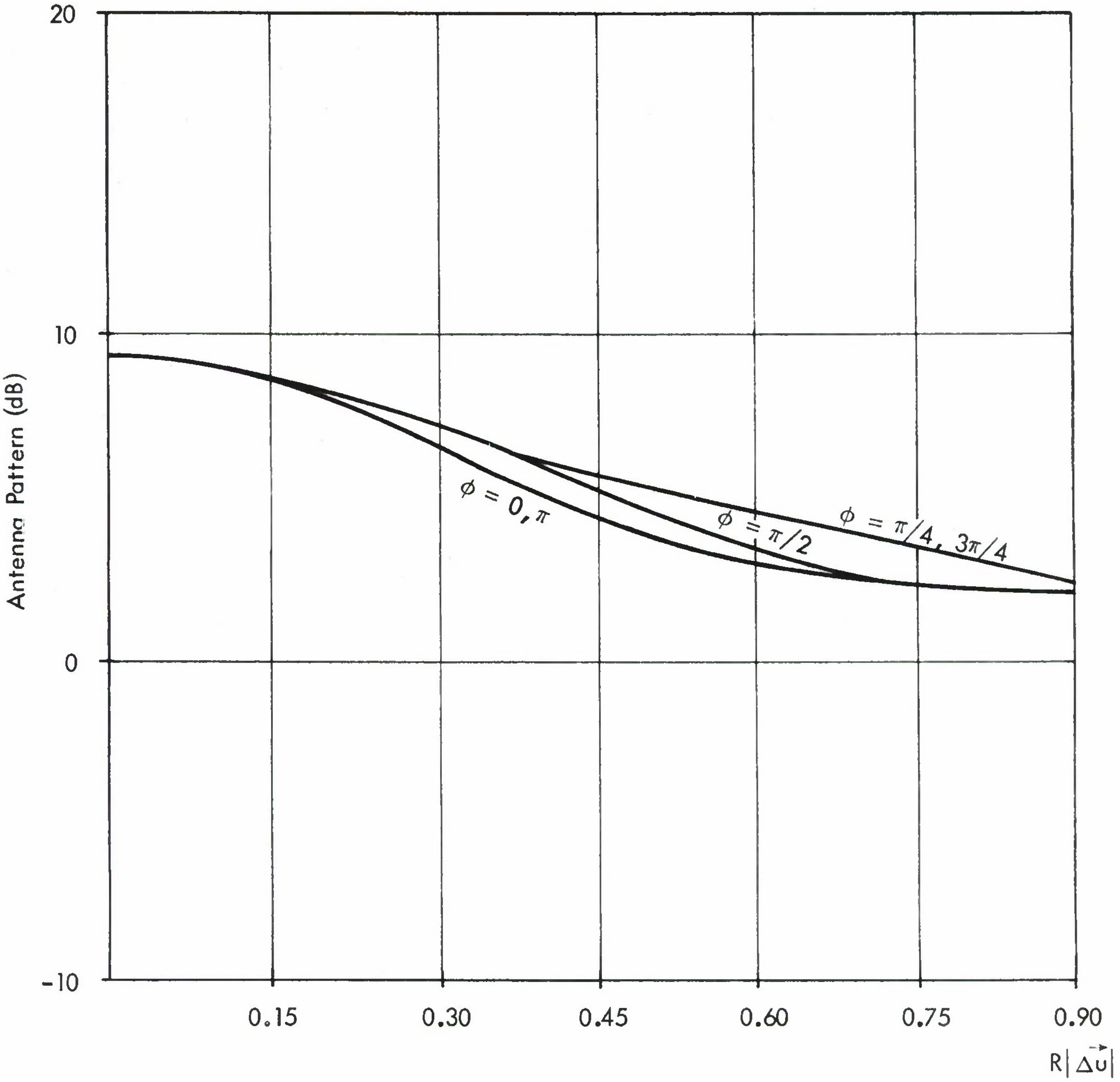


Figure 16. Antenna Pattern
Conventional Processing
LASA Configuration

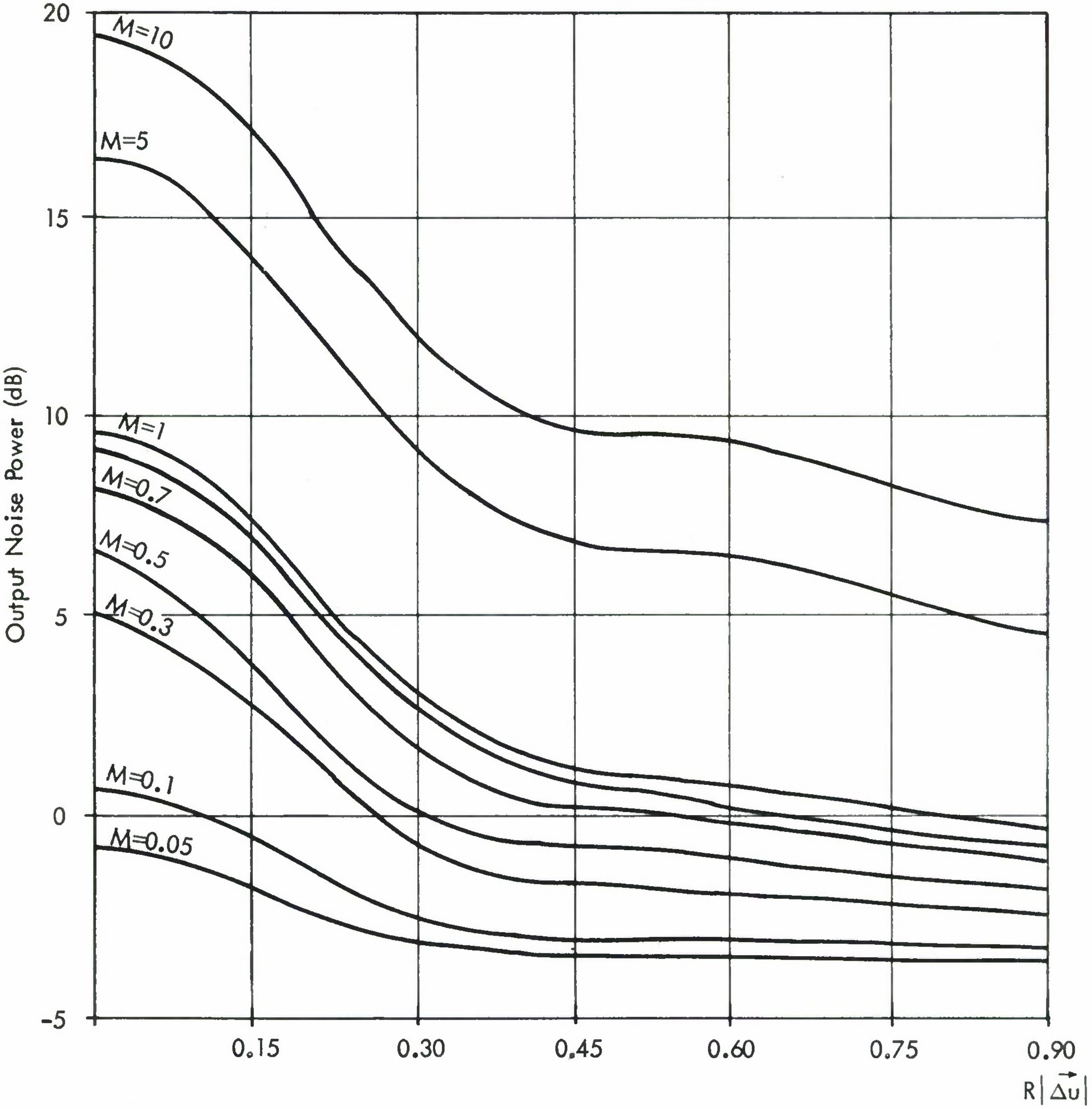


Figure 17. Output Noise Power
Conventional Processing
21 Seismometers on a Circle

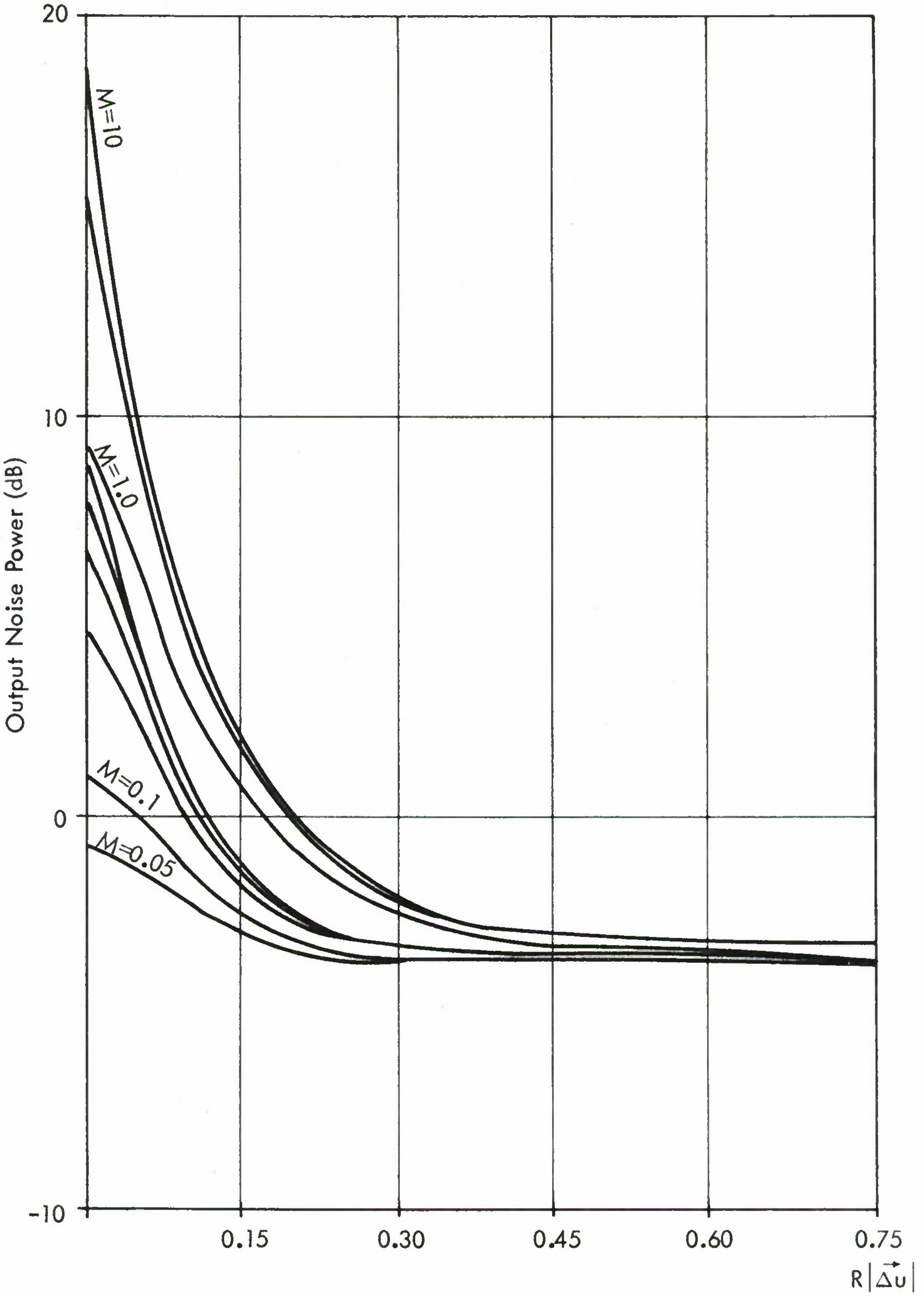


Figure 18. Output Noise Power
Optimum Processing
21 Seismometers on a Circle

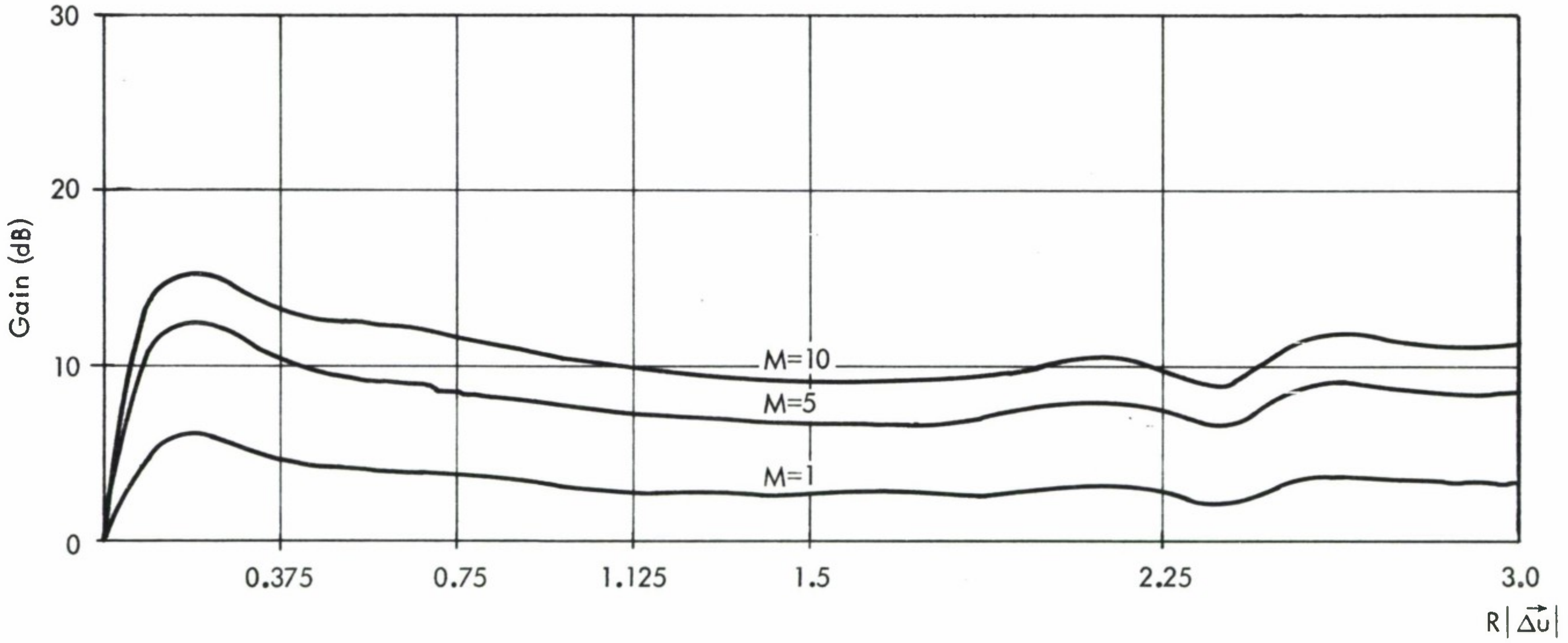


Figure 19. dB Gain Due to Optimum Processing
21 Seismometers on a Circle

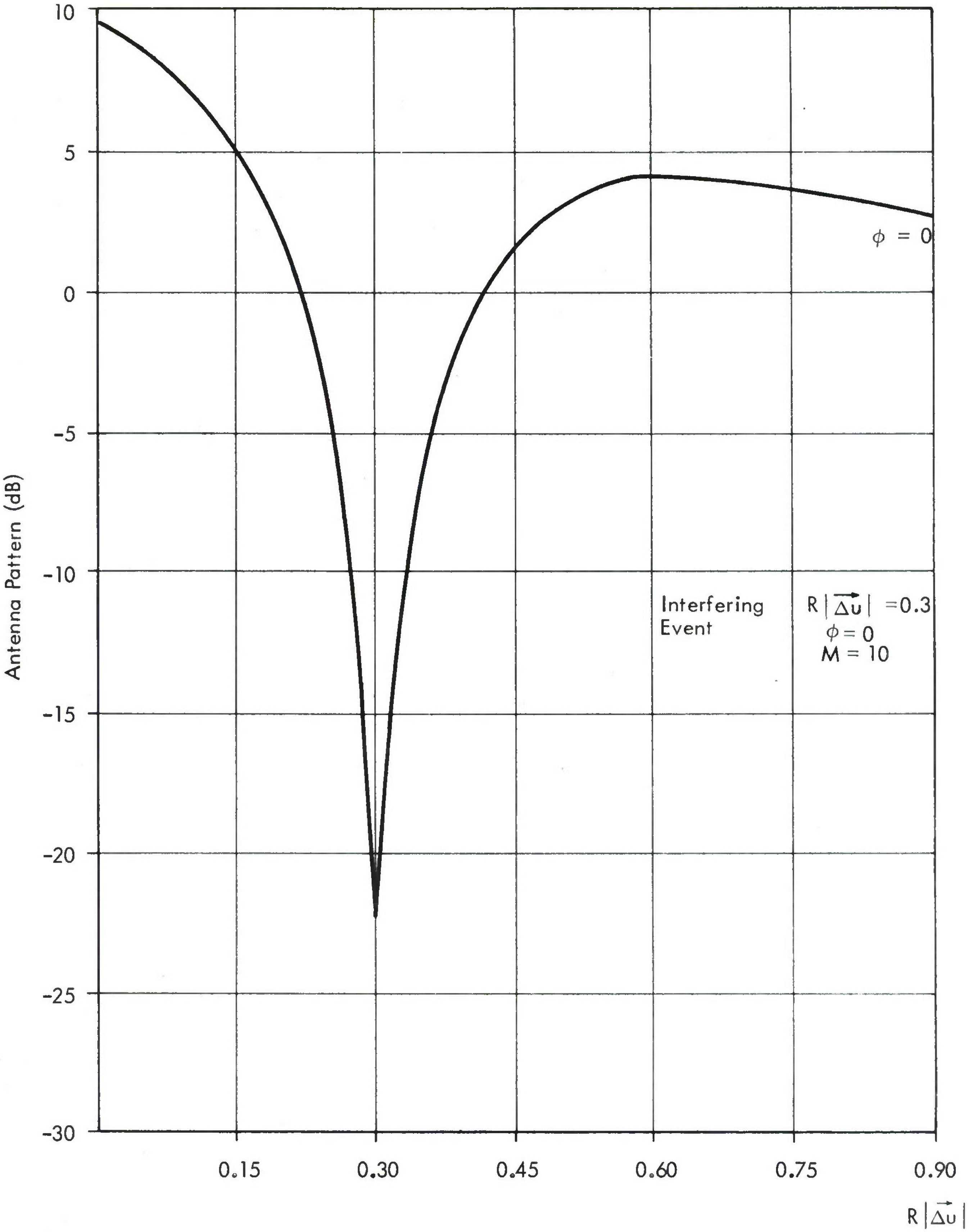


Figure 20. Antenna Pattern, Optimum Processing 21 Seismometers on a Circle Interfering Event

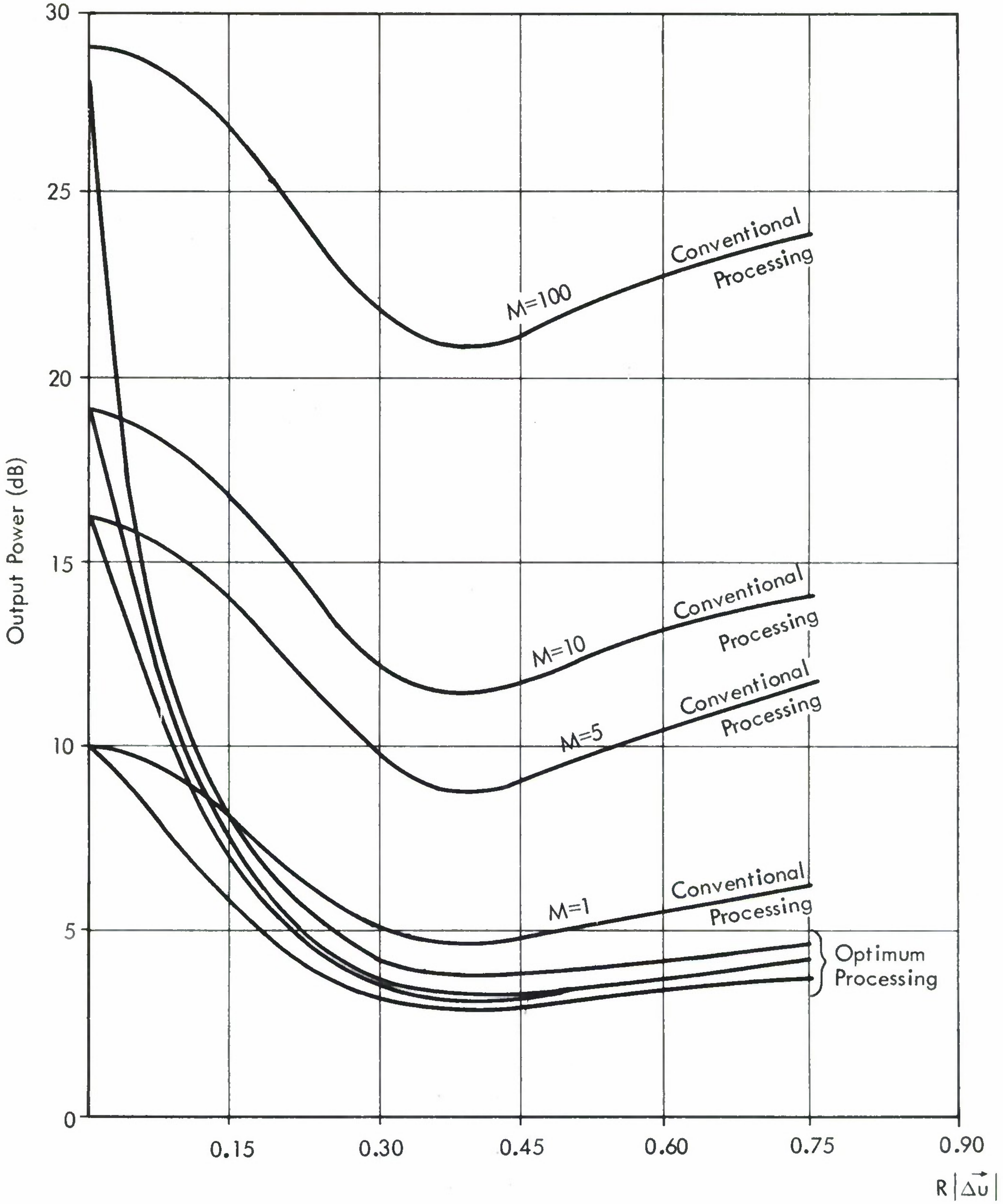


Figure 21. Output Noise Power
5 Seismometers on a Circle

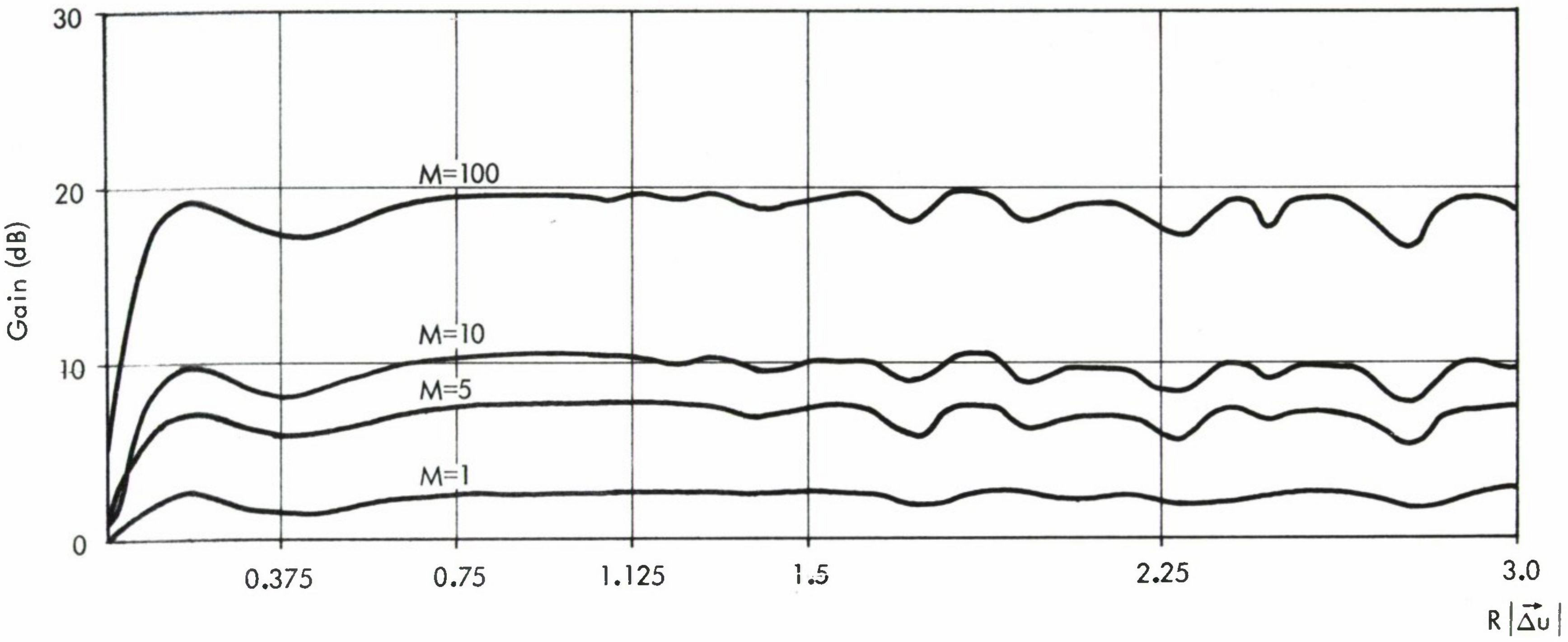


Figure 22. dB Gain Due to Optimum Processing
5 Seismometers on a Circle

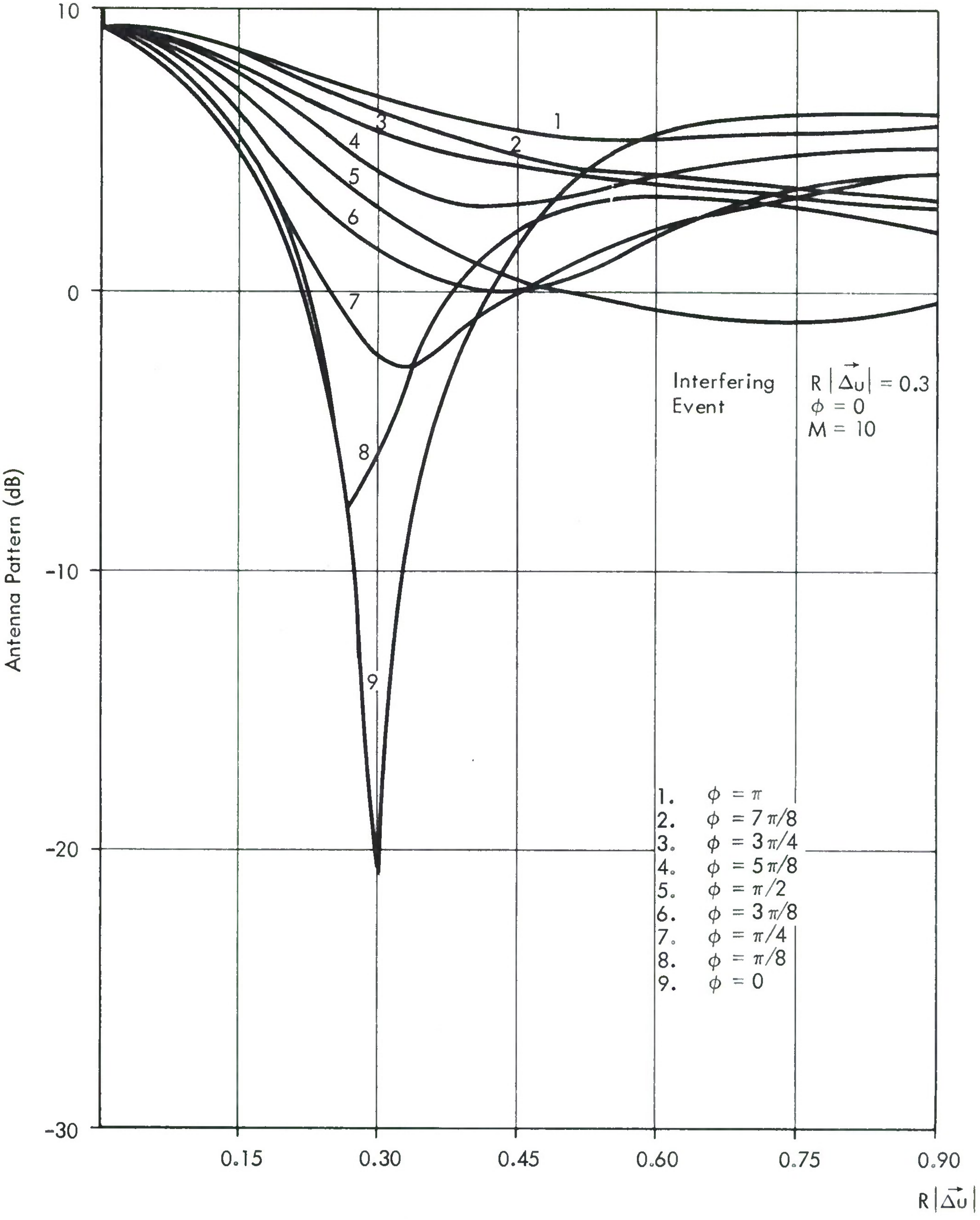


Figure 23. Antenna Pattern
Optimum Processing
5 Seismometers on a Circle
Interfering Event

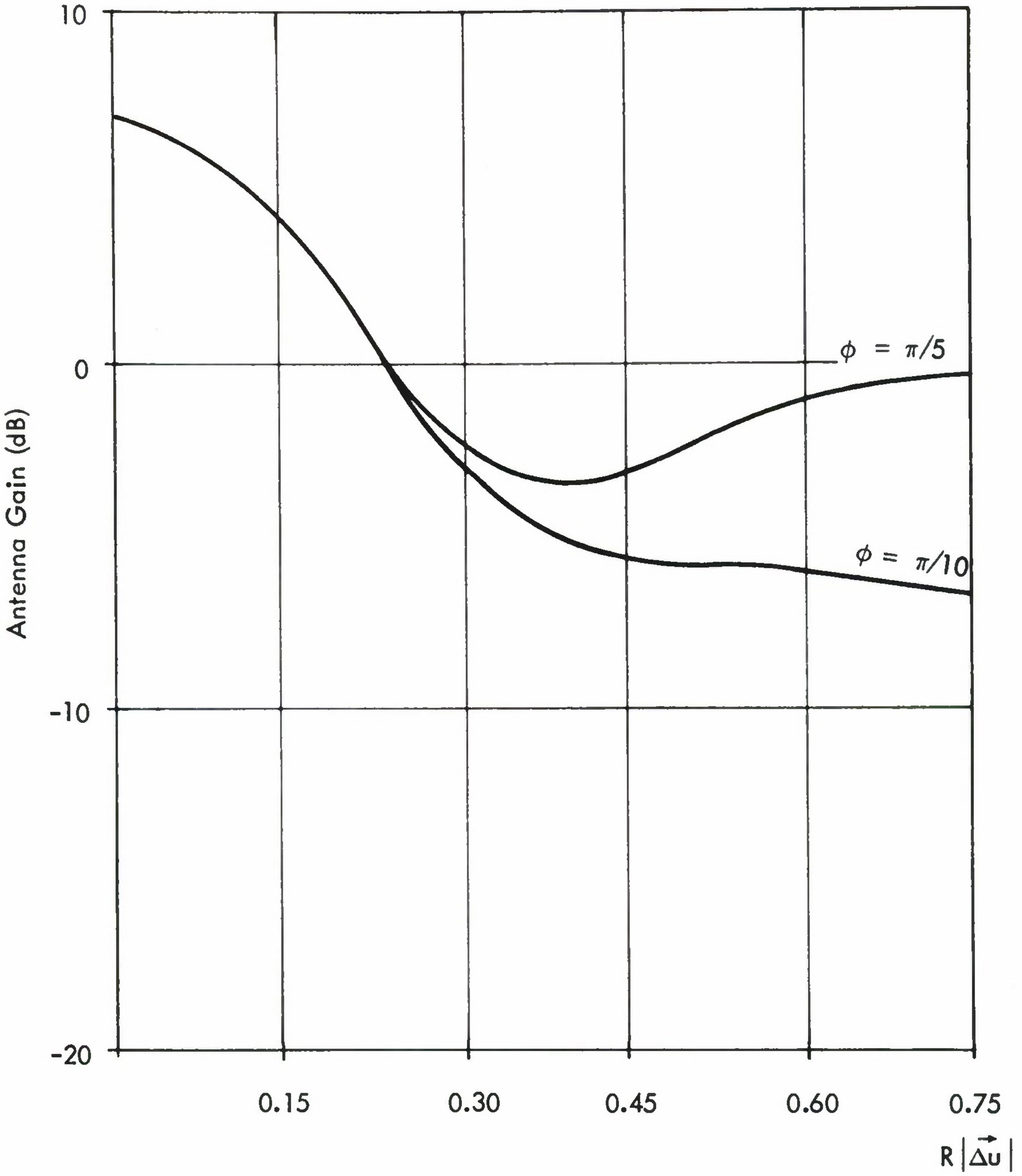


Figure 24. Antenna Gain
Conventional Processing
5 Seismometers on a Circle

with the fidelity constraint:

$$\sum_k c_k = 1. \tag{40}$$

This is the same as restricting the $A_k(f)$ of sections II.3.2 and II.3.3 to be of the form $A_k(f) = c_k$. With this restriction the output noise power is given by:

$$\text{output noise power} = \int_{-w}^{w} \sum_{k,k'} c_k \, c_{k'} \, P_{kk'}(f) \; df \tag{41}$$

$$= \sum_{kk'}^{w} c_k \, c_{k'} \int_{-w}^{w} P_{kk'}(f) \; df \tag{42}$$

$$= \sum_{kk'} c_k \, c_{k'} \int_{-w}^{w} \mathrm{Re}\left\{P_{kk'}(f)\right\} \; df. \tag{43}$$

Now if we let:

$$\int_{-w}^{w} \mathrm{Re}\{P_{kk'}(f)\} \; df = \Pi_{kk'}. \tag{44}$$

Then we wish to minimize

$$\sum_{kk'} c_k \, c_{k'} \, \Pi_{kk'} \quad , \tag{45}$$

subject to the constraint:

$$\sum_k c_k = 1. \tag{46}$$

This is the same abstract problem described in Section II.3.3 and the optimum set of c_k are given by:

$$c_k = \frac{\sum_{k'} \Pi^{-1}_{kk'}}{\sum_{k,k'} \Pi^{-1}_{kk'}} \quad . \tag{47}$$

Further as in Section II.3.5:

$$\text{output noise power} = \sum_{k,k'} c_k \, c_{k'} \, \Pi_{kk'} \quad , \tag{48}$$

$$= \left(\sum_{kk'} \Pi^{-1}_{kk'} \right)^{-1} \quad ; \tag{49}$$

and from Section II.3.9 the antenna gain is:

$$\text{antenna gain (opt. processing)} = \left| \sum_k c_k \, e^{-2\pi j f \tau_k} \right|^2 . \tag{50}$$

For the case of present interest, a noise consisting of an interfering event with spectrum P_c (f) and delays τ_k and a uncorrelated background with spectrum P_b (f), we have

$$\Pi_{kk'} = \int P_{kk'} (f) \, df, \tag{51}$$

$$= \delta \, (k,k') \int P_b (f) \, df + \int P_c (f) \cos 2\pi f (\tau_k - \tau_{k'}) \, df \; ;$$

or if $P_b (f) = P(f) = P_c (f) / M$,

$$\Pi_{kk'} = \delta(k,k') \int P(f) \, df + M \int P(f) \cos 2\pi \, f(\tau_k - \tau_{k'}) \, df. \tag{52}$$

We made some calculations comparing the gain in noise rejection using this simple processing with the gain from complete optimum processing. Figures 25 and 26 give the noise power output for the LASA configuration and for 21 seismometers on a circle for single multiplication optimum processing. There is considerable improvement over conventional processing but substantially less than with complete optimum processing. For five seismometers on a circle, a single calculation for the val ue M=5 showed single multiplication optimum processing to give almost no improvement over conventional processing.

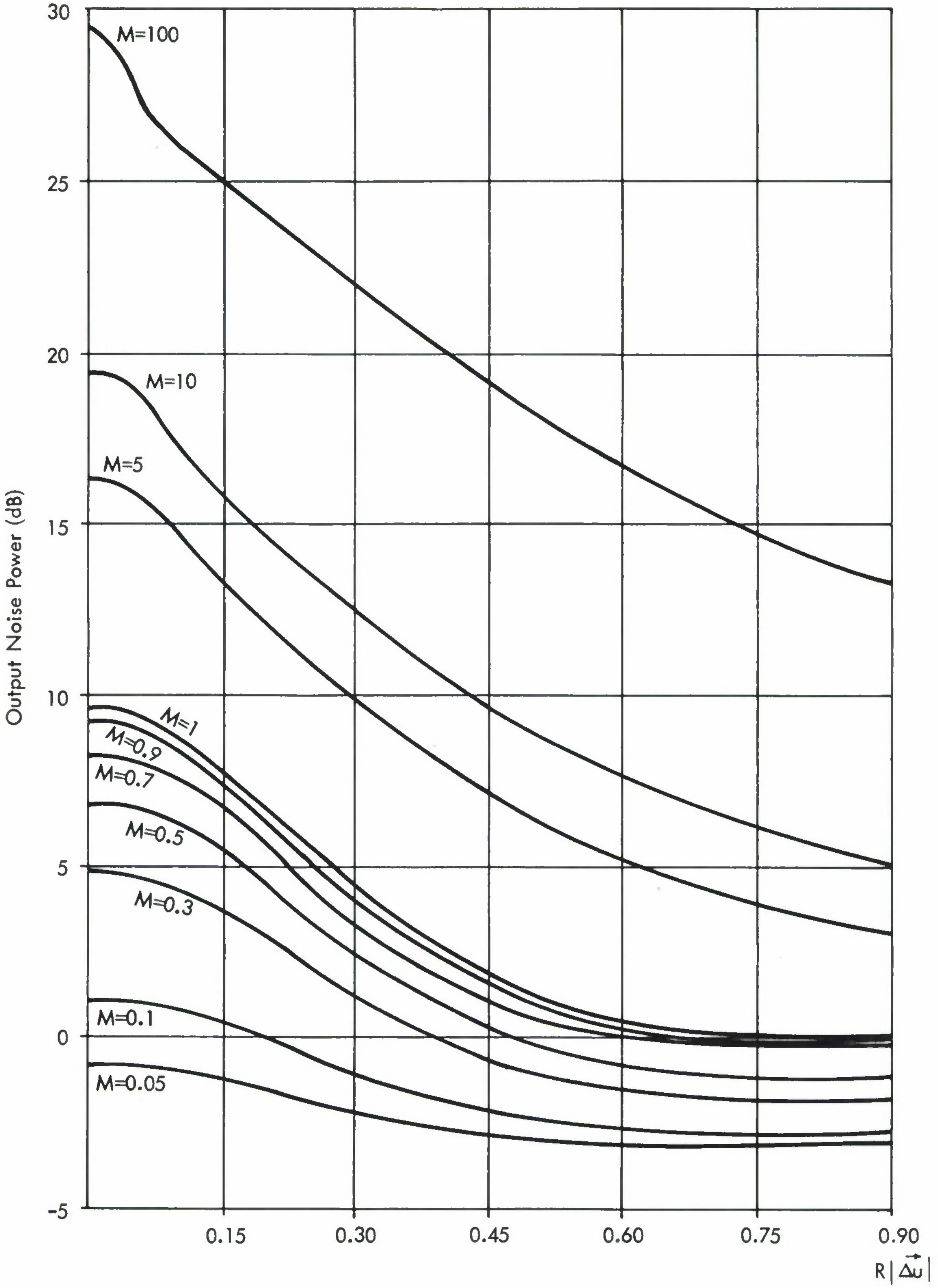


Figure 25. Output Noise Power, Single Multiplication Optimum Processing, LASA Configuration

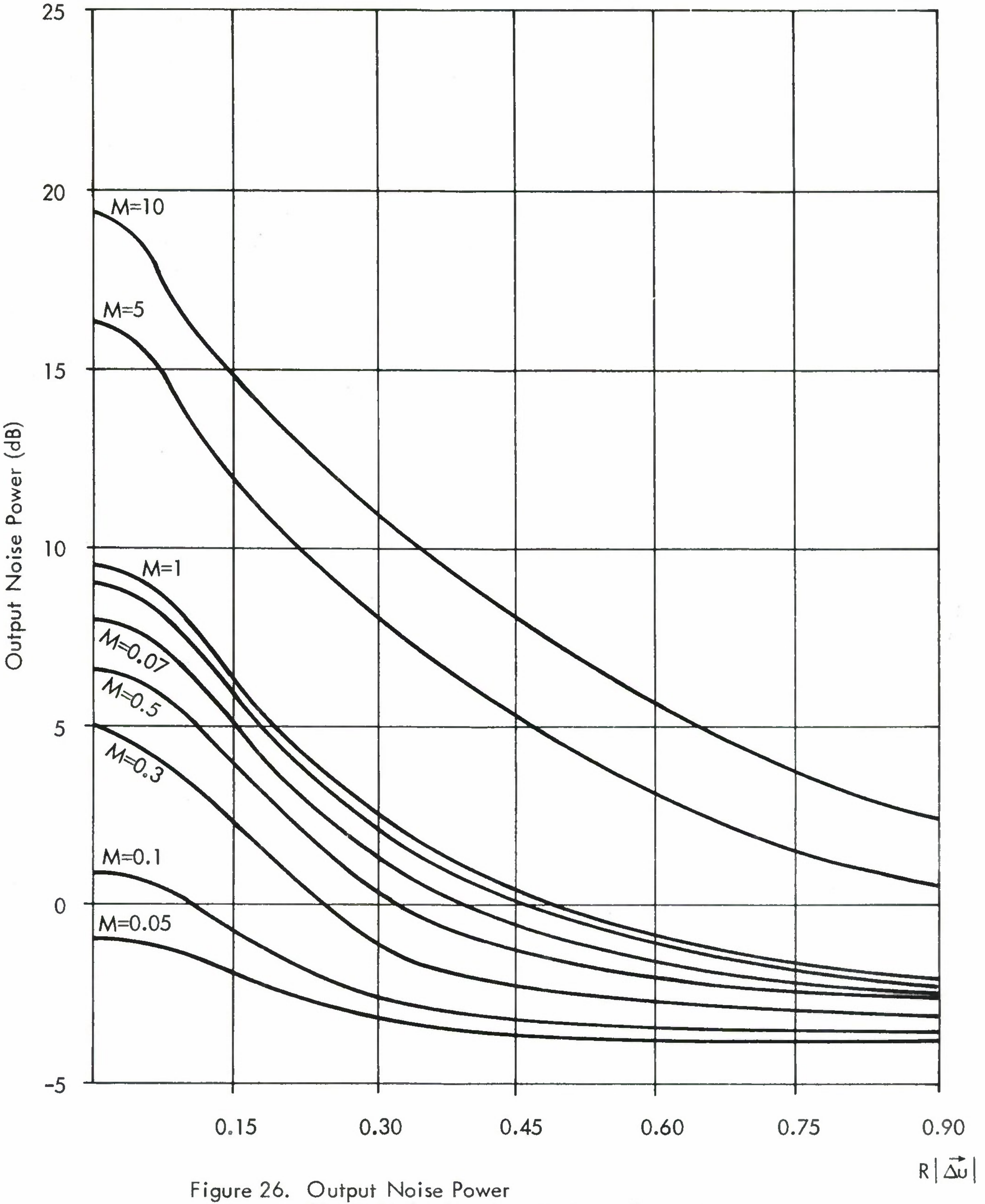


Figure 26. Output Noise Power
Single Multiplication Optimum Processing
21 Seismometers on a Circle

II.4. FIXED POINT FFT EXECUTION

II.4.1 Introduction

II.4.1.1 Summary

This appendix contains the results of an effort directed at estimating the impact of microcoding on the performance of the fast Fourier transform algorithm (FFT) on the IBM System/360 Model 40. There were two major aspects to this effort: estimating the speed of a fixed point, Model 40, microcode and estimating the error in the fixed point calculation. The microcode and its speed are discussed in detail in Section II.4.3.

If N is the number of complex points to be transformed (N is assumed to be a power of two), then the following estimate is obtained for the speed of a fixed point microcode operating on 16 bit numbers:

Transform Time $\approx 52 \text{ N} \log_2 \text{N} \ \mu$ secs.

This is approximately ten times faster than both floating and fixed point machine language codes for the algorithm on the Model 40.

The accuracy of the fixed point calculation is discussed, in detail, in Section II.4.2. The fixed point calculation we considered, both for microcoding and in the accuracy estimation, involved keeping the array properly scaled by rescaling at any stage in which there occurred an overflow. With this scheme, the error is a function of the numbers and timing of these overflows. However, an upper bound to the error can be obtained by assuming an overflow and rescaling at each stage of the calculation. Again, if we assume we are transforming $N = 2^M$ complex points, then the upper bound of the ratio of the root mean square error to the root mean square of the resulting transform increases as the square root of N or by $\sqrt{2}$ at each of the M stages of the calculation. Specifically for the Model 40 microcode which operates on 16 bit numbers, we have:

$$\frac{\text{R M S (error)}}{\text{R M S (result)}} \leq \frac{2^{(M+3)/2} \ 2^{-15} \ (0.3)}{\text{R M S (initial array)}}$$

This is well within the accuracy requirements of the FFT applications envisaged for LASA data. This point is discussed, in detail, in subsection II.2.4.5.

II.4.1.2 The Finite Fourier Transform

If X(j) j = 0, 1, . . . , N-1 is a sequence of complex numbers then the finite Fourier transform of X(j) is the sequence:

$$A(m) = (1/N) \sum_{j=o}^{N-1} X(j)e^{-2\pi ijm/N} \qquad m=0, 1, \ldots, N-1. \tag{53}$$

The inverse transform is

$$X(j) = \sum_{m=0}^{N-1} A(m)e^{2\pi ijm/N} \tag{54}$$

In both of the above equations $i = (-1)^{1/2}$. We will be considering a fixed point calculation of these transforms using the Fast Fourier Transform algorithm[9, 10]. In connection with Equation (53), we will actually obtain N A (m) from X (j). We will then give this result with N^{-1} as part of an overall scale factor. Considering the calculation of N A (m) from X (j) or X (j) from A (m) Parseval's theorem states:

$$\sum_{j=o}^{N-1} |X(j)|^2 = N \sum_{m=o}^{N-1} |A(m)|^2$$

$$\text{or} \quad \sum_{m=o}^{N-1} |NA(m)|^2 = N \sum_{j=o}^{N-1} |X(j)|^2 \tag{55}$$

and we see that the mean square value of the result is N times the mean square value of the initial sequence. This fact will be used below.

II.4.1.3 The Inner Loop of the Fast Fourier Transform Algorithm; Step by Step Scaling

The inner loop of the FFT algorithm operates on two complex numbers from the array. It takes these two numbers and produces two new complex numbers which replace the original ones in the array. Let $X_m(i)$ and $X_m(j)$ be the original complex numbers. Then, the new pair $X_{m+1}(i)$, $X_{m+1}(j)$ are given by:

$$\begin{aligned} X_{m+1}(i) &= X_m(i) + X_m(j)W \\ X_{m+1}(j) &= X_m(i) - X_m(j)W \end{aligned} \tag{56}$$

where W is a complex root of unity. If we write these equations out in terms of their real and imaginary parts, we get:

$$\begin{aligned} \mathrm{Re}\{X_{m+1}(i)\} &= \mathrm{Re}\{X_m(i)\} + \mathrm{Re}\{X_m(i)\}\ \mathrm{Re}\{W\} - \mathrm{Im}\{X_m(i)\}\ \mathrm{Im}\{W\} \\ &\ \vdots \\ \mathrm{Im}\{X_{m+1}(j)\} &= \mathrm{Im}\{X_m(i)\} - \mathrm{Re}\{X_m(i)\}\ \mathrm{Im}\{W\} - \mathrm{Im}\{X_m(j)\}\ \mathrm{Re}\{W\} \end{aligned} \tag{57}$$

At each stage the algorithm goes through the entire array of N numbers in this fashion, two at a time, If $N = 2^M$, then the number of such stages in the computation is M.

As we move from stage to stage through the calculation, the magnitudes of the numbers in the array generally increase which means that the array can be kept properly scaled by overflow tests and right shifts. Consider first the root mean square of the complex numbers. From equations (56) we have

$$\sqrt{\frac{|X_{m+1}(i)|^2 + |X_{m+1}(j)|^2}{2}} = \sqrt{2}\sqrt{\frac{|X_m(i)|^2 + |X_m(j)|^2}{2}} \tag{58}$$

Hence, in the root mean square sense, the numbers (both real and complex) are increasing by $\sqrt{2}$ at each stage. Consider next the maximum modulus of the complex numbers. From equations (56) one can easily show that

$$\max\{|X_m(i)|, |X_m(j)|\} \leq \max\{|X_{m+1}(i)|, |X_{m+1}(j)|\} \leq 2 \max\{|X_m(i)|, |X_m(j)|\} \tag{59}$$

Hence, the maximum modulus of the array of complex numbers is non-decreasing.

Both of the above results suggest that the fixed point calculation be done as follows: the entire original array and the W's are scaled with the numbers moved as far to the left as possible; after the multiplication, the most significant bits are retained; after the addition, there is a test for overflow; if overflow is detected, the entire array is shifted right one bit and the calculation is continued. Equation (59) shows that the maximum modulus of the complex numbers cannot increase by more than a factor of 2. Hence, if scaling were controlled by this modulus, there would be no more than one shift per stage. However, the method we have outlined above would control the scaling on the basis of overflow of the calculation of the real and imaginary parts. It can be shown that, in this case, it is possible (although unlikely) to get more than one overflow per stage or a single overflow of more than one bit. It is impossible to get an overflow of more than two bits. (This can be seen from equations (57) since there are only two additions).

II.4.2 An Accuracy Analysis of the Fixed Point Calculation

II.4.2.1 Introduction

We will assume, in this analysis, that the inputs (i.e., the real and imaginary parts of X(j) or A(m)) are represented by B bits plus a sign. We assume the binary point lies to the left of the leftmost bit. We showed earlier that the magnitudes of these members of the array would generally increase as we moved from stage to stage in the calculation. Hence, the method of operation is to test for overflow within the inner loop. If there is no overflow, the calculation proceeds as usual. If there is an overflow, then the two inputs producing the overflow are shifted right until there is no overflow. The amount of the shift is recorded (it will be either one or two bits) and the entire array is shifted right this same amount. In this scheme, we shift not only those elements we've already calculated but also those yet to be done. (This latter shifting could be done in subsequent calculations if this were more efficient). The total number of shifts is accumulated and the power of two, raised to the negative of this total number of shifts, constitutes an overall scale factor to be applied to the final array.

Our numbers in most applications will come from an A-D converter and will contain a quantization error. We will not consider the propagation of this quantization error. Our purpose is to compare the accuracy of a short word fixed point computation with a longer word floating point computation. In practice, the quantization error would be common to both alternatives.

There are two operations which produce errors which are propagated through the calculation.

a. When two B bit numbers are multiplied together a 2B bit product results. If this product is rounded to B bits, an error whose variance is:

$$\Delta_1^2 = 2^{-2B}/12 \tag{60}$$

is created. This error has a standard deviation:

$$\Delta_1 = 2^{-B}/\sqrt{12} \approx 0.3(2^{-B}) \tag{61}$$

b. When two B bit numbers are added together and there is an overflow, then the sum must be shifted right and a bit lost. If this bit is a zero, there is no error. If it is a one, there is an error of $\pm 2^{-B}$ depending upon whether the number is positive or negative. The variance of this error (it is unbiased assuming there are an equal number of positive and negative numbers) is:

$$\Delta_2^2 = 2^{-2B}/2 . \tag{62}$$

It has a standard deviation

$$\Delta_2 = 2^{-B-1} \approx 0.7(2^{-B}) \tag{63}$$

II.4.2.2 Upper Bound Analysis

In this section, we give an upper bound analysis of the ratio of the R M S error to the R M S of the answer. This upper bound is obtained by assuming that during each step of the calculation there is an overflow and a need to rescale. We let X_k be a typical real element at the kth stage (i.e., the real or imaginary part of a complex element) and let

$$V(X_k) = \text{variance of } X_k. \tag{64}$$

We will, in this analysis, replace Δ_2^2 by $6\Delta_1^2$. We will also let $\Delta^2 = \Delta_1^2$.

Since the first stage gives an overflow, the original data must be rescaled or truncated by one bit. Hence,

$$V(X_0) = 6\Delta^2. \tag{65}$$

In going from the original data to the results of the first stage, W=1 and, hence there is no multiplication and we either add or subtract. Further, we assume that the next stage will result in an overflow and hence we will have to rescale. This gives

$$\begin{aligned} V(X_1) &= 2V(X_0) + 4 \cdot 6\Delta^2 \\ V(X_1) &= 2(6\Delta^2) + 4 \cdot 6\Delta^2. \end{aligned} \tag{66}$$

In these expressions and this entire discussion, we are assuming all errors to be independent and, hence, that the variance of the sum is the sum of the variances. Going from the first stage to the second stage, we have $W = (-1)^{1/2}$ and again there are only additions and subtractions. Thus, with the rescaling

$$\begin{aligned} V(X_2) &= 2V(X_1) + 4^2 \cdot 6\Delta^2 \\ &= 2^2(6\Delta^2) + 2(4 \cdot 6\Delta^2) + 4^2 \cdot 6\Delta^2. \end{aligned} \tag{67}$$

In going from the second stage to the third stage, we have multiplications and we have them in all subsequent stages. In generating the third stage, half the inner loops have multiplications. Consider the first equation of equations (57). All the other equations are identical in terms of error propagation. Remember that X_3 (i) is complex:

$$\mathrm{Re}\,\{X_3(i)\} = \mathrm{Re}\,\{X_2(i)\} + \mathrm{Re}\{X_2(j)\}\,\mathrm{Re}\{W\} - \mathrm{Im}\{X_2(j)\}\,\mathrm{Im}\{W\} \tag{68}$$

Equation (68) yields, with rounding to B bits after the addition and with rescaling,

$$\begin{aligned} V'(X_3) &= V(X_2) + [\mathrm{Re}^2\{X_2(j)\} + \mathrm{Im}^2\{X_2(j)\}]\; V(W) \\ &\quad + [\mathrm{Re}^2(W) + \mathrm{Im}^2(W)]\; V(X_2) \\ &\quad + (4^3\Delta^2) + 4^3 \cdot 6\Delta^2 \\ &= V(X_2) + |X_2(j)|^2\Delta^2 + V(X_2) + (4^3\Delta^2) + 4^3 \cdot 6\Delta^2 \end{aligned} \tag{69}$$

In Equation (69), the first term is the variance of the first term of (68). The second and third terms of (69) are the variance of the full 2B bit products given by the second and third terms of (68). The fourth term of (69) is the result of rounding after the addition. The fifth term is the rescaling term. Finally, we saw in Equation (58) that the average modulus squared of the complex numbers is increasing by a factor of 2 every stage. Hence, if we let K equal the average modulus squared of the initial array, i.e.,

$$K = \frac{1}{N}\sum_{i=0}^{N-1} |X_0(i)|^2$$

then we have:

$$V'(X_3) = 2V(X_2) + 2^2K\Delta^2 + (4^3\Delta^2) + 4^3 \cdot 6 \cdot \Delta^2 \tag{70}$$

Equation (70) would be correct for $V(X_3)$ if all the inner loops involved multiplications. However, at this stage only half of them do and hence:

$$\begin{aligned} V(X_3) &= 2V(X_2) + 2K\Delta^2 + 4^3\Delta^2 + 4^3 \cdot 6\Delta^2 \\ &= 2^3(6\Delta^2) + \mathbf{2}^2(4 \cdot 6\Delta^2) + 2(4^2 \cdot 6\Delta^2)\; 4^3 \cdot 6\Delta^2 + 2K\Delta^2 + 4^3\Delta^2/2. \end{aligned} \tag{71}$$

In the next stage, three quarters of the inner loops require multiplications and these multiplications get progressively more numerous as the stages increase. Hence, from here on, we will assume all stages have multiplications in all the inner loops. Thus, applying the above techniques, we get:

$$V(X_4) = 2^4(6\Delta^2) + 2^3(6 \cdot 4\Delta^2) + 2^2(6 \cdot 4^2\Delta^2) + 2(6 \cdot 4^3\Delta^2) + 6 \cdot 4^4\Delta^2$$
$$+ 2^2K\Delta^2 + 2^3K\Delta^2 + 4^3\Delta^2 + 4^4\Delta^2; \qquad (72)$$

and generally if M is the last stage:

$$V(X_M) = 2^M(6\Delta^2) + 2^{M-1}(6 \cdot 4\Delta^2) + \ldots + 2(6 \cdot 4^{M-1}\Delta^2)$$
$$+ 2^{M+2}K\Delta^2 + (M-3)2^{M-1}K\Delta^2 + 2^{M-4}(4^3\Delta^2)$$
$$+ 2^{M-4}(4^4\Delta^2) + \ldots + (4^M\Delta^2)$$
$$= (1.5)2^{M+2}\Delta^2(1+2+\ldots+2^{M-1}) + (M-2.5)2^{M-1}K\Delta^2$$
$$+ 2^{M+3}\Delta^2 + 2^{M+4}(1+2+\ldots+2^{M-4})\Delta^2. \qquad (73)$$

Or

$$V(X_M) \approx (1.5)2^{2M+2}\Delta^2 + (M-2.5)2^{M-1}K\Delta^2 + 2^{M+3}\Delta^2 + 2^{2M+1}\Delta^2$$
$$\approx 2^{2M+3}\Delta^2 + (M-2.5)2^{M-1}K\Delta^2 + 2^{M+3}\Delta^2. \qquad (74)$$

K is the average of the square of the absolute values of the initial complex array. Hence, applying Parseval's theorem, Equation (55), the average of the square of the absolute values of the final array will be 2^MK. What is most meaningful in this case, however, is the mean square of the real numbers which is $2^MK/2$. Hence, we have:

$$\frac{V(X_M)}{2^MK/2} \approx \frac{2^{M+3}\Delta^2}{K/2} + \frac{(M-2.5)\Delta^2/2}{1/2} + \frac{2^3\Delta^2}{K/2}, \qquad (75)$$

and finally,

$$\frac{\text{R M S (error)}}{\text{R M S (result}} \approx \frac{2^{\frac{M+3}{2}}\Delta}{K/2} \approx \frac{2^{\frac{M+3}{2}}\,2^{-B}(0.3)}{\text{R M S (initial array)}}\ . \qquad (76)$$

Equation (76) gives an approximate upper bound for the ratio of the R M S of the error to R M.S. of the answer. Notice that this bound increases as the $\sqrt{N}$ or 1/2 bit per stage.

II.4.2.3 Lower Bound Analysis

We will now obtain a lower bound for the ratio of R M S of the error to the R M S of the answer. We obtain this lower bound by assuming that there are no overflows in the calculation and, hence, no shifts of the array. In this case,

$$V(X_0) = V(X_1) = V(X_2) = 0 . \tag{77}$$

In the third stage, half of the inner loops involve a multiplication and hence:

$$V(X_3) = (1/2)\ (2^2 K\Delta^2) + 1/2\,(\Delta^2) . \tag{78}$$

This can be seen by considering the first term of equations (69). The first term of (78) comes from the second term of the first of equations (69). The second term of (78) is caused by the rounding to B bits. Now, as before:

$$\begin{aligned} V(X_4) &= 2V(X_3) + 2^3 K\Delta^2 + \Delta^2 \\ &= 2^2 K\Delta^2 + 2^3 K\Delta^2 + \Delta^2 + \Delta^2 . \end{aligned} \tag{79}$$

Finally,

$$\begin{aligned} V(X_M) &= 2^{M-2} K\Delta^2 + (M-3)2^{M-1} K\Delta^2 + 2^{M-3}\Delta^2 + 2^{M-5}\Delta^2 \\ &\quad + 2^{M-b}\Delta^2 + . . . + \Delta^2 \\ &= (M-2.5)2^{M-1} K\Delta^2 + 2^{M-3}\Delta^2 + (1 + . . . + 2^{M-5})\ \Delta^2 \\ &\approx (M-2.5)2^{M-1} K\Delta^2 + 2^{M-3}\Delta^2 + 2^{M-4}\Delta^2 . \end{aligned} \tag{80}$$

As in subsection 2.2, the mean square of the array of resulting real numbers is $2^M \cdot K/2$. Hence, we have:

$$\begin{aligned} \frac{V(X_M)}{2^M K/2} &\approx (M-2.5)\,\Delta^2 + \frac{\Delta^2/8}{K/2} + \frac{\Delta^2/6}{K/2} \\ &\approx (M-2.5)\Delta^2 . \end{aligned} \tag{81}$$

and

$$\frac{\text{R M S (error)}}{\text{R M S (result)}} \approx (M-2.5)^{1/2} \tag{82}$$

The lower bound increases as $M = \log_2 N$. This is the rate of increase which has been observed for the floating point calculation.[11,12]

II.4.2.4 Some Experimental Results

An IBM 7094 program was written to perform a fixed point calculation using the fast Fourier transform algorithm, as described above. The program was capable of simulating a fixed point machine of any word size up to 35 bits plus a sign. Experiments were run with fixed point numbers of 17 bits plus a sign. This corresponds to $B = 17$ in the analysis of subsections II.4.2.2 and II.4.2.3.

The experiments were performed in the following way. Floating point input was fixed to 17 bits plus a sign. This fixed input was then transformed with the fixed point program. The fixed point output was then floated. Next, the fixed point 17-bit input was floated and a floating point transform taken. Since this floating point transform uses a floating point word with a 27-bit mantissa, it was considered the correct answer. Finally, the R M S of the difference between the fixed point and floating point answers was taken. We also obtained the maximum absolute error and average error.

Figure 27 contains the result oι transforming random numbers which lie between zero and one (placed in both the real and imaginary parts). In this and subsequent tests, three runs were made for every power of 2 from 8 to 2048. Since these random numbers have a DC component of one-half, the fixed point program must rescale at least every stage but one. Hence, one would expect the error to lie close to the theoretical upper bound as given by Equation (76). This theoretical upper bound is also plotted in Figure 27 and the results are seen to lie slightly above it. The R M S of the original array, K/2, is approximately 0.6.

Figure 28 contains the results of transforming three sine waves plus random numbers between zero and one-half in the real part and all zeros in the imaginary part. Specifically:

$$\mathrm{Re}\{X(j)\} = 1/2[Y(j) + (1/2)\sin(2\pi 8j/N)$$

$$+ (1/4)\sin(2\pi 4j/N) + (1/4)\sin(2\pi 8j/N)]$$

$$\mathrm{Im}\{X(j)\} = 0$$

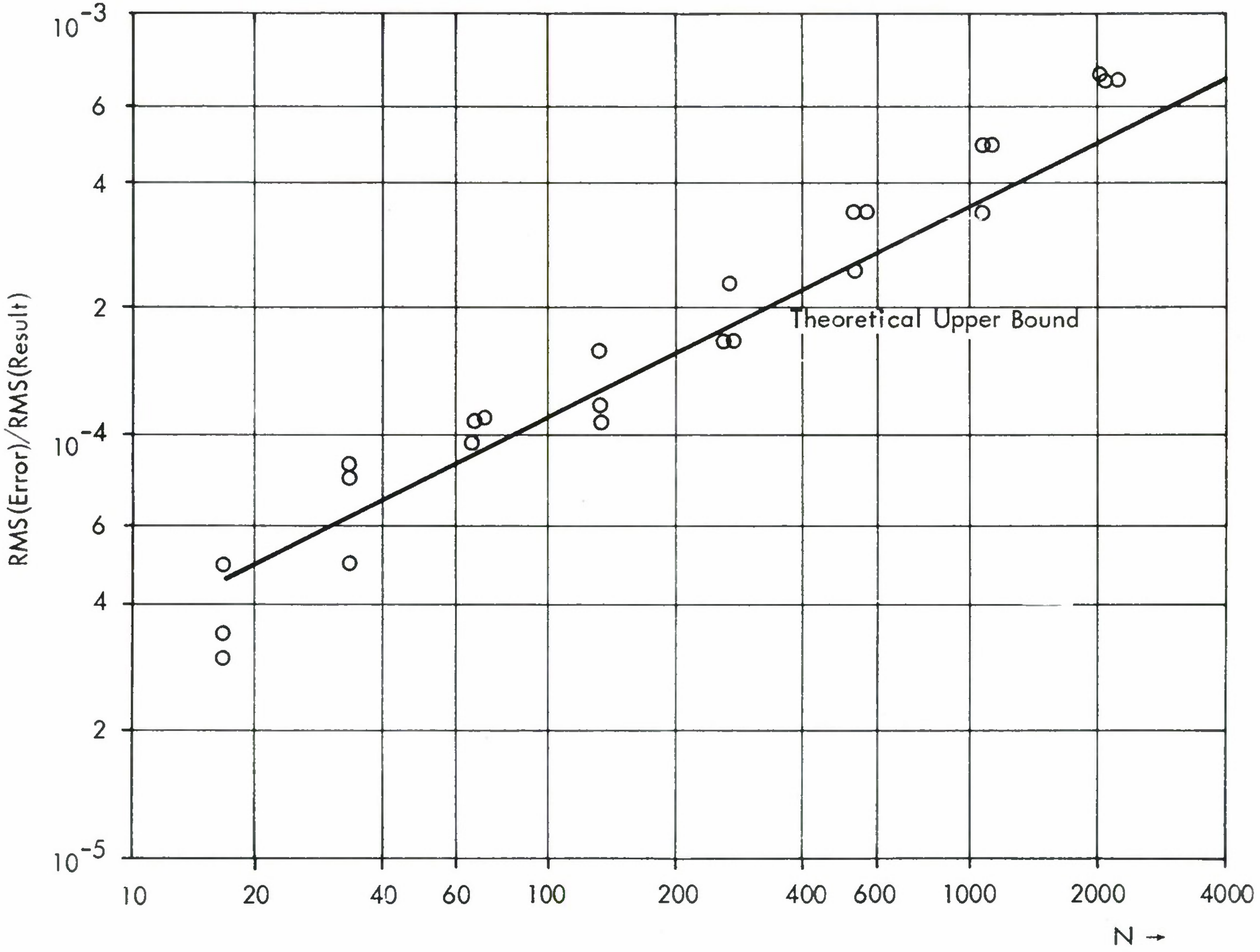


Figure 27. Experimental Error Results: Random Numbers Between 0 and 1, B=17

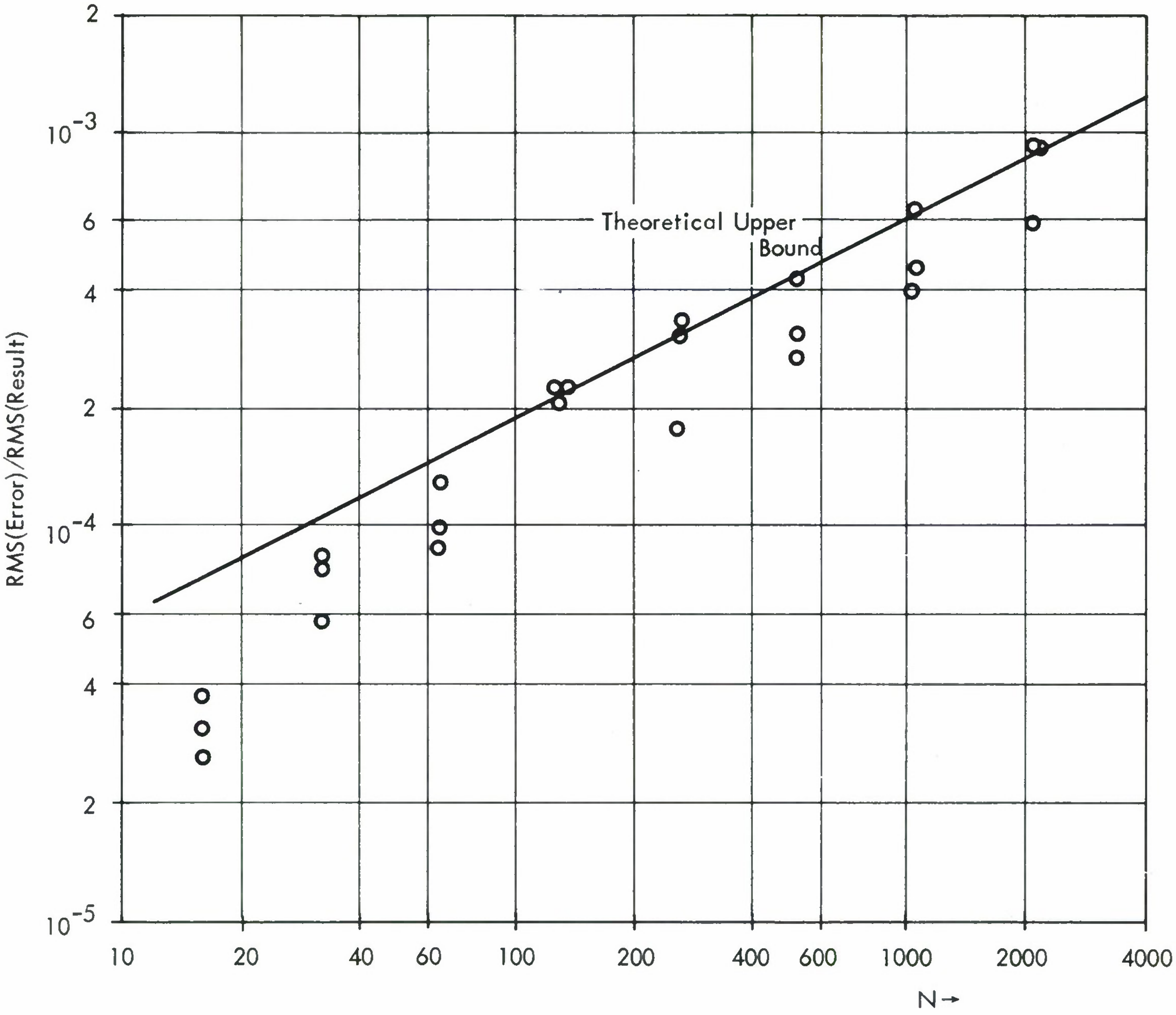


Figure 28. Experimental Error Results: Random Numbers Plus 3 Sine Waves
$0 <$ Random Numbers $< 1/2$, $B = 17$

where the Y(j) are random numbers between zero and one. Again, there is a DC component of magnitude one-fourth and the array must be rescaled at least at every stage but two. Thus, one would expect these results to be lower relative to the theoretical upper bound than the case depicted in Figure 27. From Figure 28 one can see that this is in fact the case. The R M S of the original array, K/2, is, in this case, approximately 0.35. This is the reason the upper bound curve is higher than that of Figure 27.

Figure 29 contains the results of transforming random numbers from minus one to one (in both real and imaginary parts). In this case, the DC component is zero and there is no other strong component. The number of shifts should be approximately $(\log_2 N)/2$ or one-half shift per stage. Hence, one would expect the error curve to lie well below the theoretical upper bound as is the case. In this case, K/2 = 0.6.

Figure 30 contains the results of an experiment identical to that corresponding to Figure 28, except the random numbers are between plus and minus one half. The results are as expected. In this case, K/2 = 0.35.

Finally, Figure 31 contains the results of transforming a sine wave in the real part and zero in the imaginary part. The sine wave was sin $(2\pi j/8)$. Although in this case the array must be rescaled in at least every stage but two, the error is well below the upper bound. Here, K/2 = 0.5.

These calculations were made with rounding rather than truncation. The error with truncation was somewhat larger. Rounding can be included in the microcode described in Section II.4.3 with a negligible decrease in speed. The bias, as reflected by the average error, was in general negligible compared with the R M S error. The maximum error was roughly what would be expected.

II.4.2.5 Conclusions and Additional Comments

Figure 32 below gives the upper bound on the ratio of R M S of the error to the R M S of the answer assuming B = 15 and assumming that the R M S of the initial array is three-tenths (i.e., K/2 = 0.3). This corresponds to the microcoding situation for the Model 40.

There are two main uses to which the fast Fourier transform would be put in seismic signal processing. The first would be in estimating the spectral matrix of a subarray of seismometer outputs as a first step in the calculation of the coefficients for optimum processing (see 13, 14, 2 or 8). The second would be in estimating a sequence of short term spectra of the LASA beam output as one of the first steps in discrimination processing. In both of these cases, an acceptable and computationally efficient procedure is to take transforms of short, overlapping segments of the time series; calculate periodograms or cross-periodograms and time average over these to increase the statistical accuracy. (See Section 8 of ref. 11, and ref. 13.)

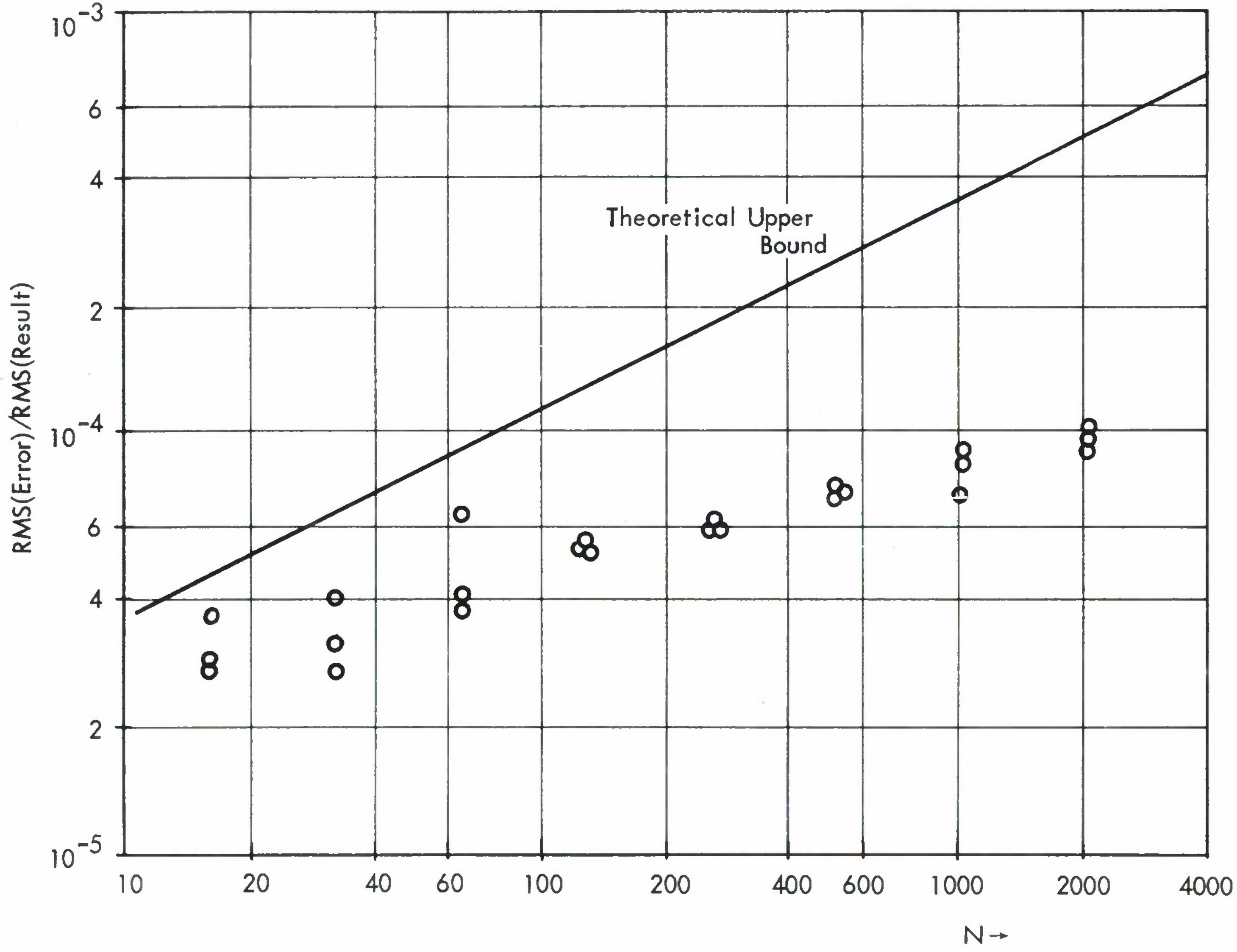


Figure 29. Experimental Error Results: Random Numbers Between -1 and 1, B=17

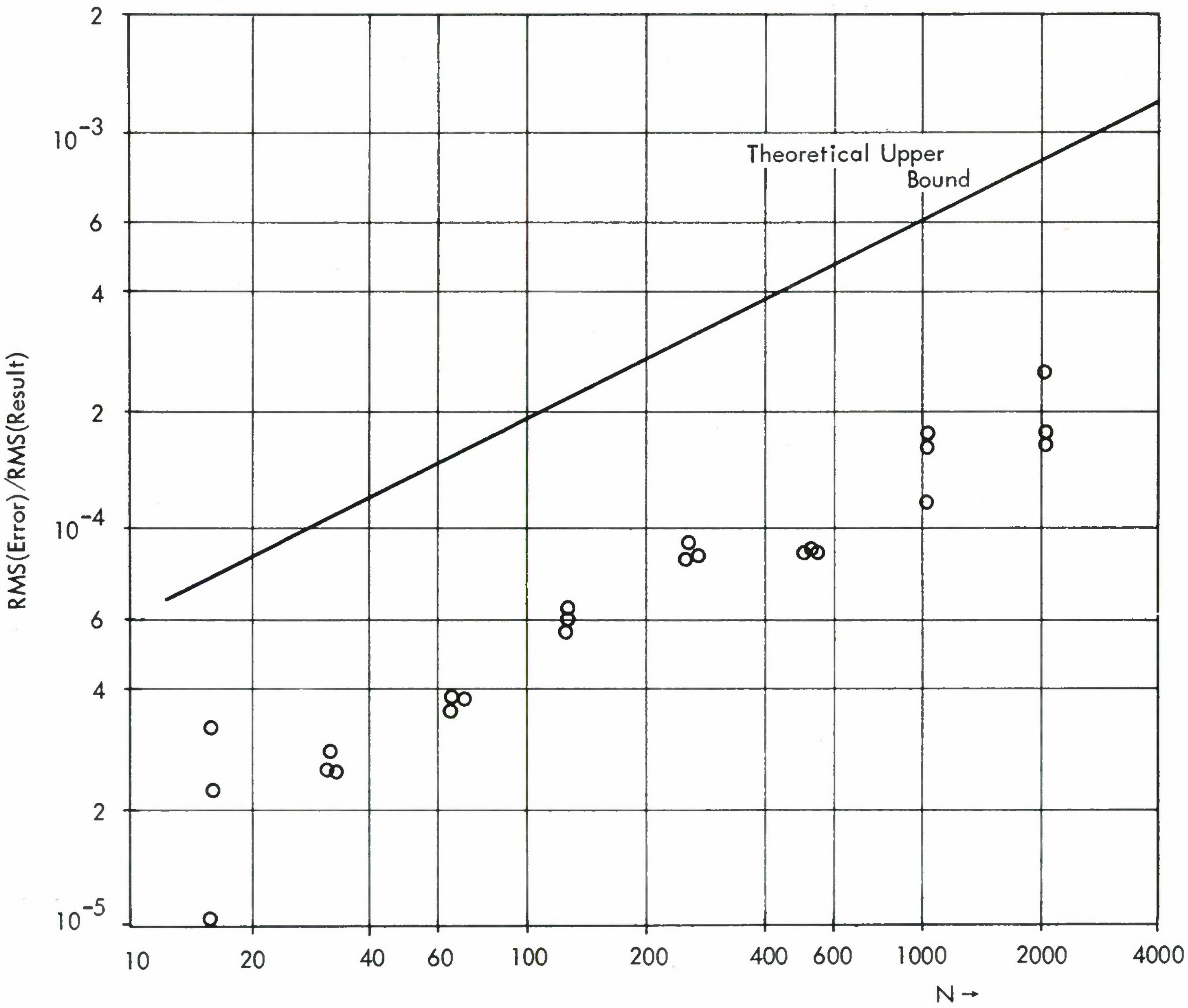


Figure 30. Experimental Error Results: Random Numbers Plus 3 Sine Waves - 1/2 < Random Numbers < 1/2, B = 17

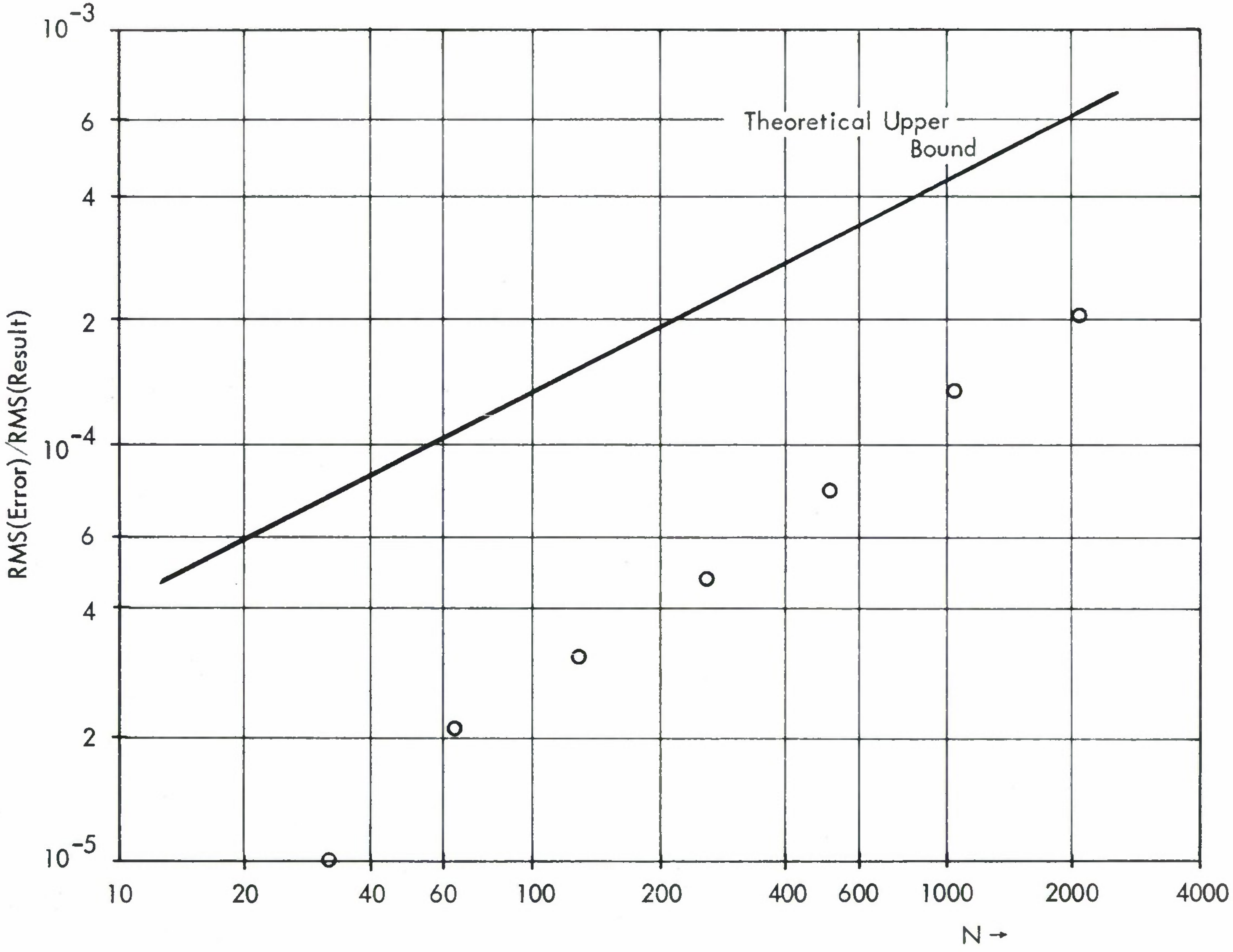


Figure 31. Experimental Error Results: Single Sine Wave, B = 17

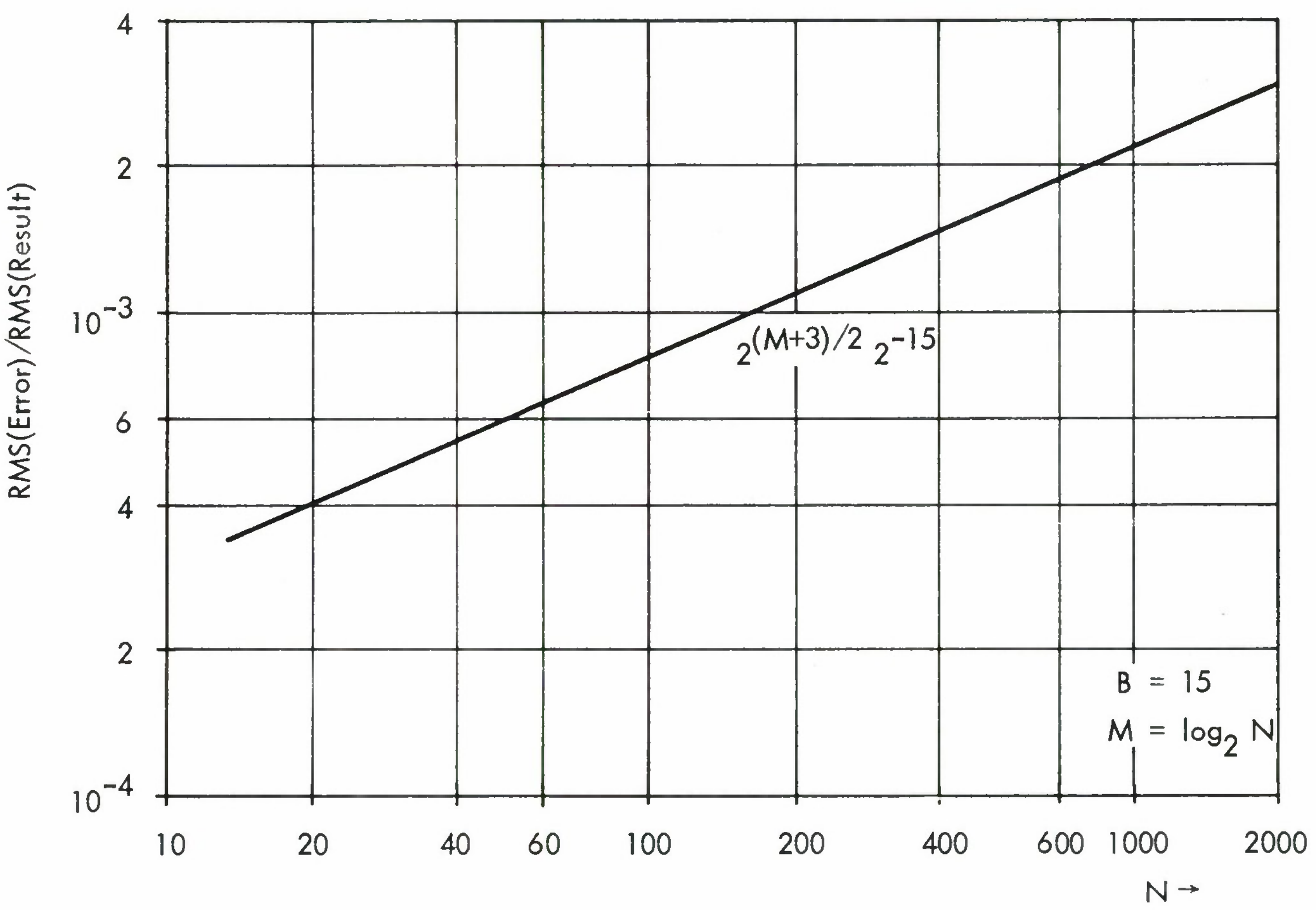


Figure 32. Error Upper Bound for 15-Bit Calculation

As concerns the fixed point accuracy, these remarks are important in two respects. First, since short segments are transformed, the fixed point error would be kept small. In the spectral matrix estimation, the size of the segments would be of the order of the number of coefficients in the optimum processing procedure. In the estimation of LASA beam spectra, it would be of the order of the reciprocal of the resolution desired (in terms of the Nyquist frequency as unity). In both of these cases, 256 would be a high figure for the length of the segments to be transformed. For segments of length 256, the fixed point error would be small compared to the statistical error in the spectral estimates. Its only effect might be a slight one on the available dynamic range. Second, the time averaging would reduce this fixed point calculation error as well as the statistical error. Hence, for the purposes of spectral estimation important in LASA signal processing, a fixed point calculation appears to be more than adequate.

As a final comment, since the magnitude of the error has a sensitive and direct relationship to the number of times the array is scaled, it would be best to remove any substantial DC component which happened to be present.

II.4.3 Speed Advantage of the Microcoded FFT

II.4.3.1 The Inner Loops

The FFT algorithm operates on an array of 2^M complex numbers, taking M stages on the array. Each stage comprises 2^{M-1} passes through the inner loop computations, where each pass processes two complex points of the array. In terms of these points, called OPerand X and OPerand Y, and the complex root of unity U, the inner loop is either a Type I or Type II:

Type 1:

Complex

$$\begin{aligned} \mathrm{OPX} &= \mathrm{OPX} + \mathrm{OPY} \\ \mathrm{OPY} &= \mathrm{OPX} - \mathrm{OPY} \end{aligned}$$

Real

$$\begin{aligned} \mathrm{ROPX} &= \mathrm{ROPX} + \mathrm{ROPY} \\ \mathrm{ROPY} &= \mathrm{ROPX} - \mathrm{ROPY} \end{aligned}$$

Imaginary

$$\begin{aligned} \mathrm{IOPX} &= \mathrm{IOPX} + \mathrm{IOPY} \\ \mathrm{IOPY} &= \mathrm{IOPX} - \mathrm{IOPY} \end{aligned}$$

Type II:

Complex

$$\begin{aligned} \mathrm{OPX} &= \mathrm{OPX} + \mathrm{OPY} \cdot \mathrm{U} \\ \mathrm{OPY} &= \mathrm{OPX} - \mathrm{OPY} \cdot \mathrm{U} \end{aligned}$$

Real

$$\begin{aligned} \mathrm{ROPX} &= \mathrm{ROPX} + (\mathrm{ROPY} \cdot \mathrm{RU} - \mathrm{IOPY} \cdot \mathrm{IU}) \\ \mathrm{ROPY} &= \mathrm{ROPX} - (\mathrm{ROPY} \cdot \mathrm{RU} - \mathrm{IOPY} \cdot \mathrm{IU}) \end{aligned}$$

Imaginary

$$\begin{aligned} \mathrm{IOPX} &= \mathrm{IOPX} + (\mathrm{ROPY} \cdot \mathrm{IU} + \mathrm{IOPY} \cdot \mathrm{RU}) \\ \mathrm{IOPY} &= \mathrm{IOPX} - (\mathrm{ROPY} \cdot \mathrm{IU} + \mathrm{IOPY} \cdot \mathrm{RU}) \end{aligned}$$

The Type I inner loop is used for all passes in the first and second stages through the array. Thereafter, its application is halved for each subsequent stage (see Table 2).

The number of Type II inner loop executions in the total FFT process does not exceed the Type I inner loops unless $M > 5$. Actually, the ratio Type II/ Type I is asymptotic to M/2, but this is not a good measure of the relative frequencies of the types unless M is large. Nevertheless, the Type II inner loop is the dominant factor in the FFT process, because it involves four real multiplications and six real additions. Type I involves no multiplications, and only four additions. For this reason, our estimate has centered around the Type II inner loop.

Table 2

INNER LOOP

Stage	Number of Type I Inner Loops	Number of Type II Inner Loops
1	2^{M-1}	0
2	2^{M-1}	0
3	2^{M-2}	2^{M-2}
. . .	. . .	. . .
M-1	2^2	$2^2 + 2^3 + \ldots + 2^{M-2}$
M	2	$2 + 2^2 + \ldots + 2^{M-2}$

II.4.3.2 Type II Inner Loop Process and Flow Chart

To maximize and preserve significance in the fixed point operands, the 2^M complex numbers are normalized in the sense that:

a. The high order bit (bit position 0) is a sign bit

b. The binary point is assumed to be between bit positions 0 and 1

c. At least one real or imaginary component of one of the complex points has the value 1 in bit position 1.

With this representation of the complex points, the execution of inner loop passes may give rise to an overflow of one or two bit positions. Actually, a two position overflow has very low probability; and a three position or more overflow is impossible.

Consider the 2^{M-1} inner loop passes in any stage. If an overflow of one bit position occurs first in pass K, this overflow divides the stage into two parts:

a. Previously executed passes; that is, passes 1 to K-1

b. Subsequently executed passes; that is, passes K+1 to 2^{M-1}

Similarly, previous and subsequent passes may be defined with respect to a two position overflow. This will divide the stage into three parts, if the two-overflow initially occurs in a pass subsequent to the one-overflow.

When an overflow of one or two positions occurs, the entire array must be shifted right one or two bit positions to preserve significance. This is most expeditiously done by shifting the results of each subsequently executed pass in the stage before these results are returned to main storage; and, after the stage is completed, by returning to the beginning and shifting the results of all previously executed passes. The latter process, of shifting previously executed results, will be called post shifting. An array is post shifted until an array element with a stop-address is detected. This is the address of an inner loop operand in the pass which first gave rise to the overflow. Saving these addresses is a bookkeeping operation which must be performed when an initial overflow of one or two places occurs.

From the above, we see that the nature of the inner loop process depends on whether an overflow condition has occurred in a previously executed pass, and on whether a new or additional shift condition has occurred in the currently executed inner loop pass. All of the possibilities are given in Table 3.

For each case in Table 3, we give a probability. These assigned probabilities are very rough estimates. They are not to be taken literally, except as an indication that, for purposes of estimating microprogram execution time, it suffices to consider only Cases A, D, and E. The effect of the low probability Cases B, C, F, and G is negligible. These conclusions are consistent with error analysis results.

Table 3

INNER LOOP OVERFLOW CONDITIONS

Case	Ovfl Positions Previous Pass	Ovfl Positions Current Pass	Probability	Total Shift Before Storing
A	0	0	0.500	0
B	0	1	0.010	1
C	0	2	0.001	2
D	1	0	0.230	1
E	1	1	0.230	1
F	1	2	0.001	2
G	2	0	0.028	2

If two 16-bit normalized binary numbers (binary points assumed between bit positions 0 and 1) are multiplied together, as with SUMP hardware, the high halfword of the result will have its binary point between bit positions 1 and 2. Hence, the high halfword of the product cannot be added to a normalized number without first either shifting the product left one position, or shifting the normalized number right one position, to line up the binary points. The alternative chosen depends on the part of the inner loop being processed (computation of RX and RY, or computation of IX and IY), and on whether or not there is a pending shift from a previous pass.

A flowchart of the inner loop process, showing the testing and handling of the various overflow conditions, is shown in figures 33, 34, and 35. The states Y2 and Y3 are used to signal overflow conditions from previously executed passes as illustrated in Table 4.

Table 4

OVERFLOW CONDITIONS

Y2	Y3	
0	0	No Pending Overflow
0	1	Pending Overflow of 1 Position
1	1	Pending Overflow of 2 Positions

In the currently executed pass, the occurrence of overflows in the computation of RX and IX is recorded in the B1 REG, and overflows in the computation of RY and IY are recorded in the B0 REG.

New overflows, conditions B, C, and F, can be recognized either in the calculation of RX and RY, or IX and IY. Thus, we have two exits each for these conditions in the flow charts.

A three-character notation near the lower right of the flow chart boxes correlates the flow chart with the microprogram given on figures 36, 37, 38, and 39. The first character gives the sheet number, and the second two give the block coordinates of the microinstruction on the sheet.

II.4.3.3 Execution Time Estimates

The microcoded execution time estimates for the Type II inner loop cases shown in figures 34 and 35 are given in Table 5 and assume both fixed point operands and the SUMP hardware for multiplication.

Figures 40 and 41 give the OS/360 program for the inner loops. The execution for these is as follows:

Fixed Point, assuming no pending shift:	485 μsecs
Fixed Point, assuming one pending shift:	480 μsecs
Floating Point:	560 μsecs

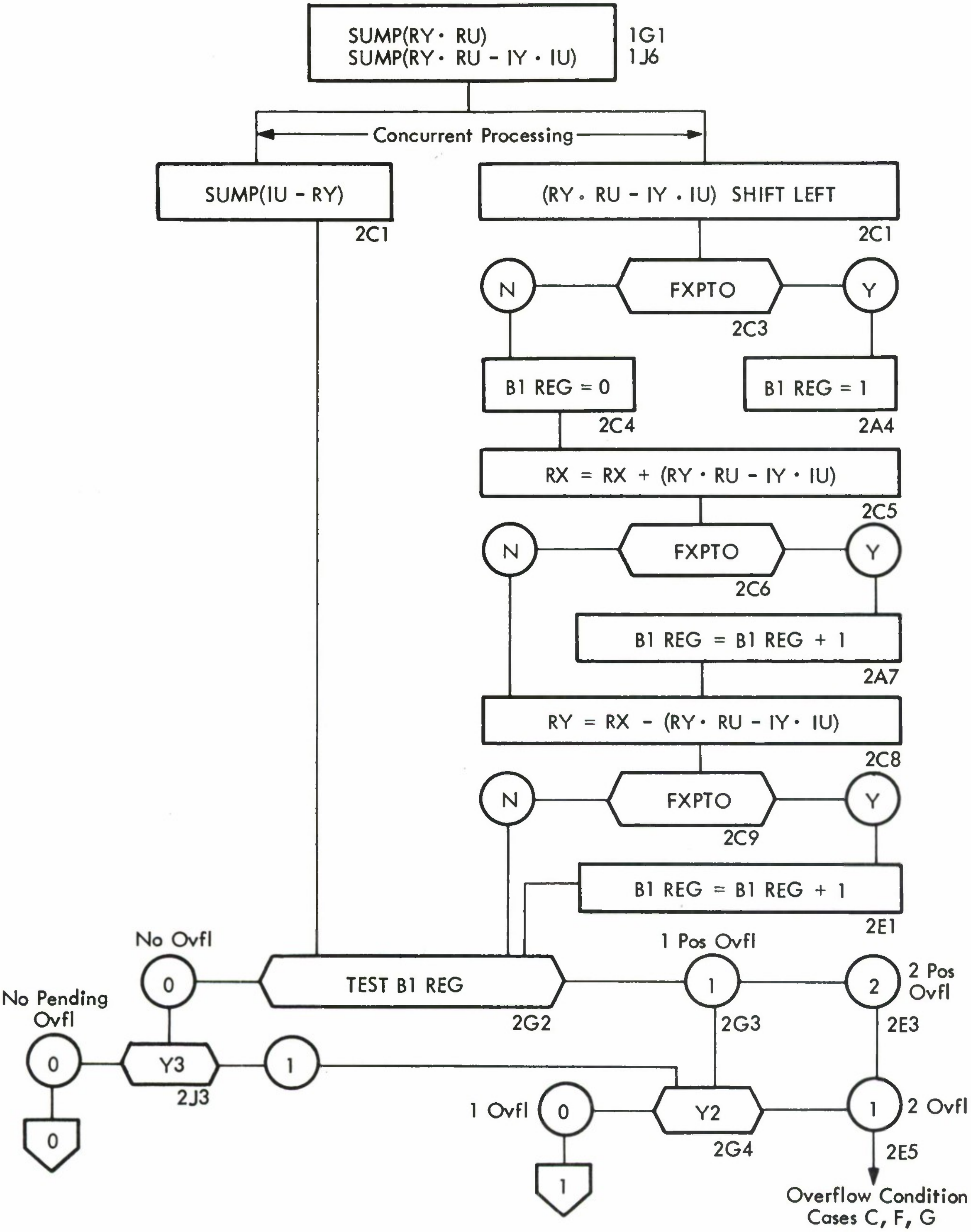


Figure 33. Flow Chart for Inner Loop of Fast Fourier Transform-I

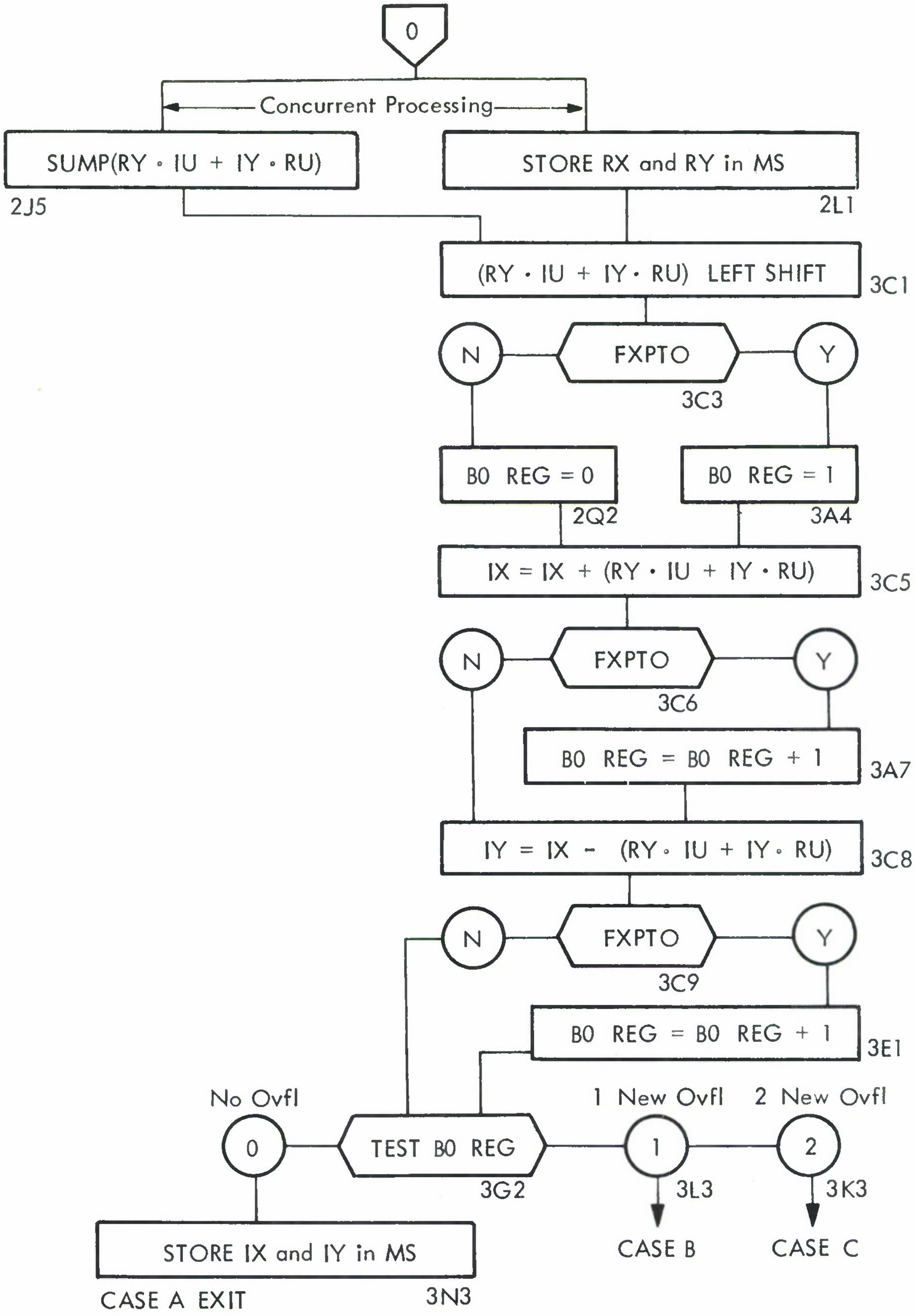


Figure 34. Flow Chart for Inner Loop of Fast Fourier Transform—II

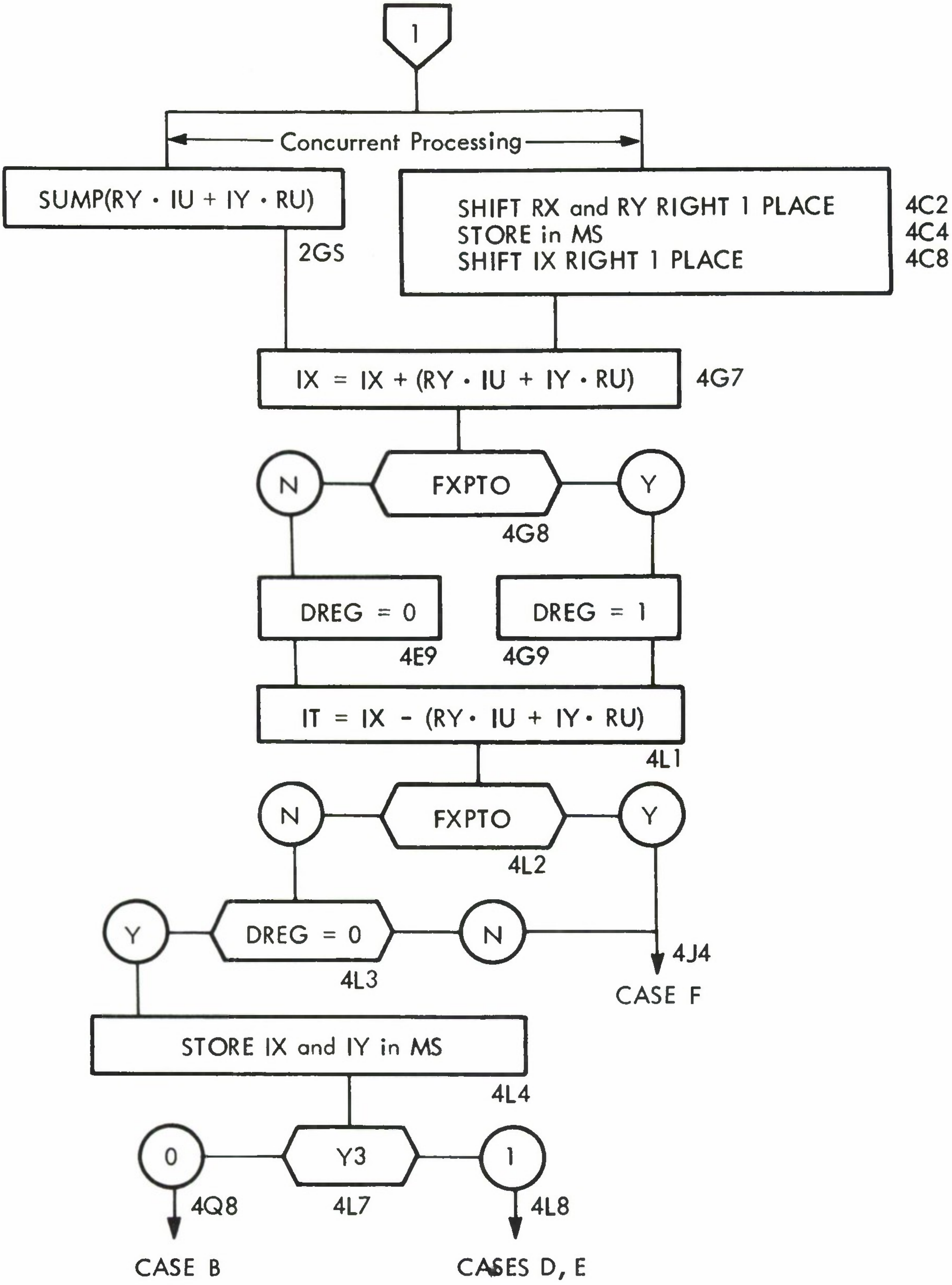


Figure 35. Flow Chart for Inner Loop of Fast Fourier Transform—III

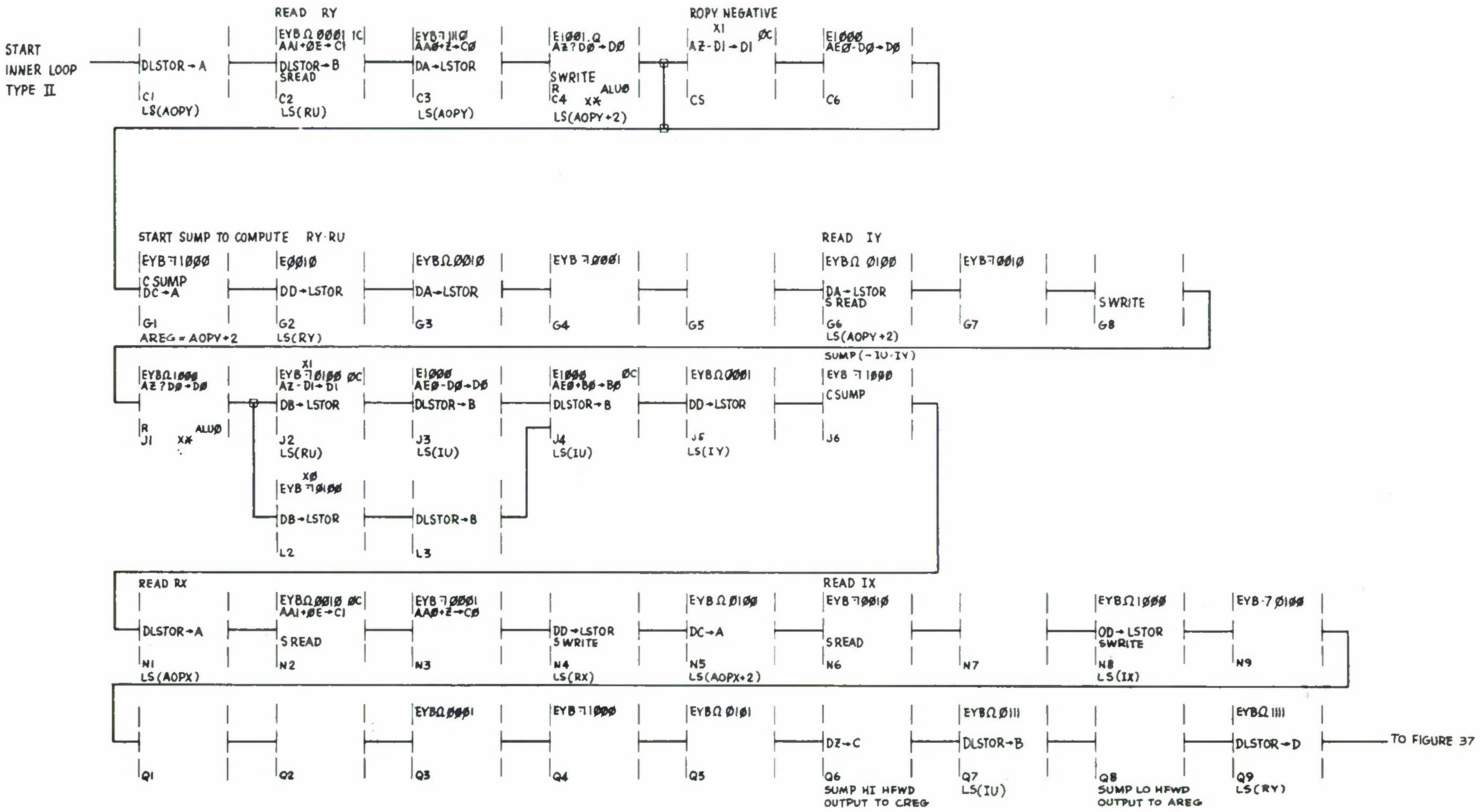


Figure 36. Inner Loop Microprogram I

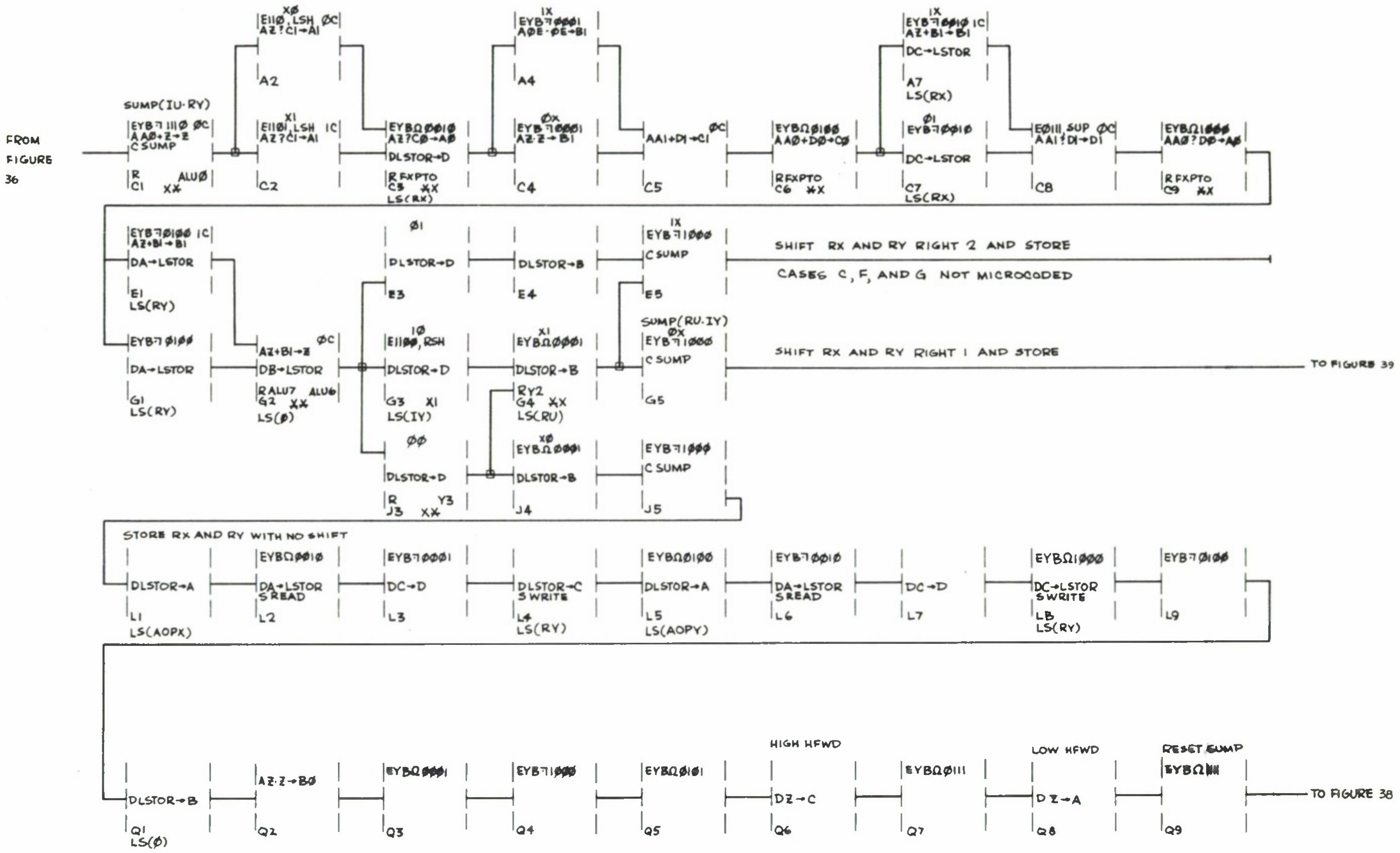


Figure 37. Inner Loop Microprogram II

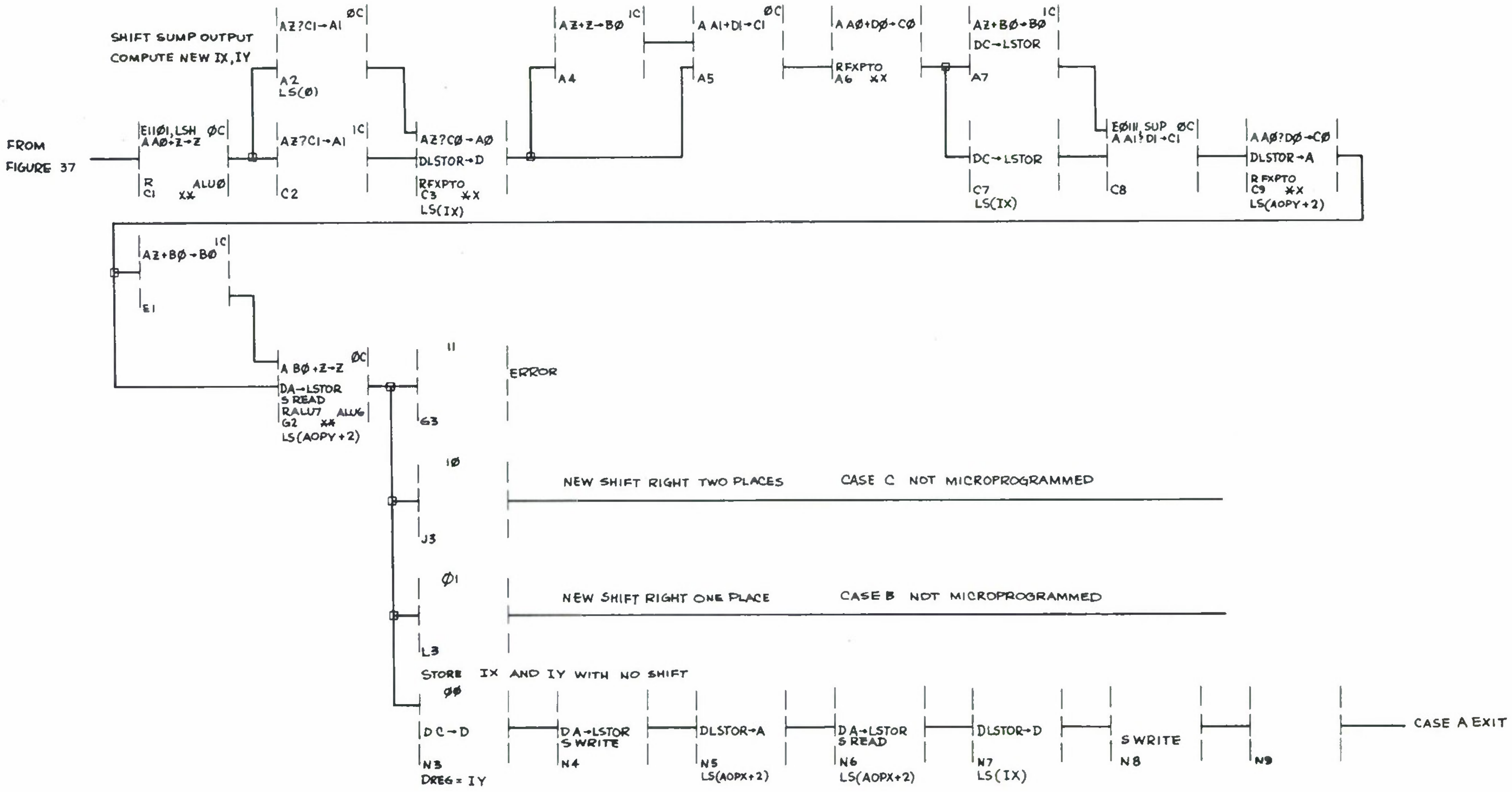


Figure 38. Inner Loop Microprogram III

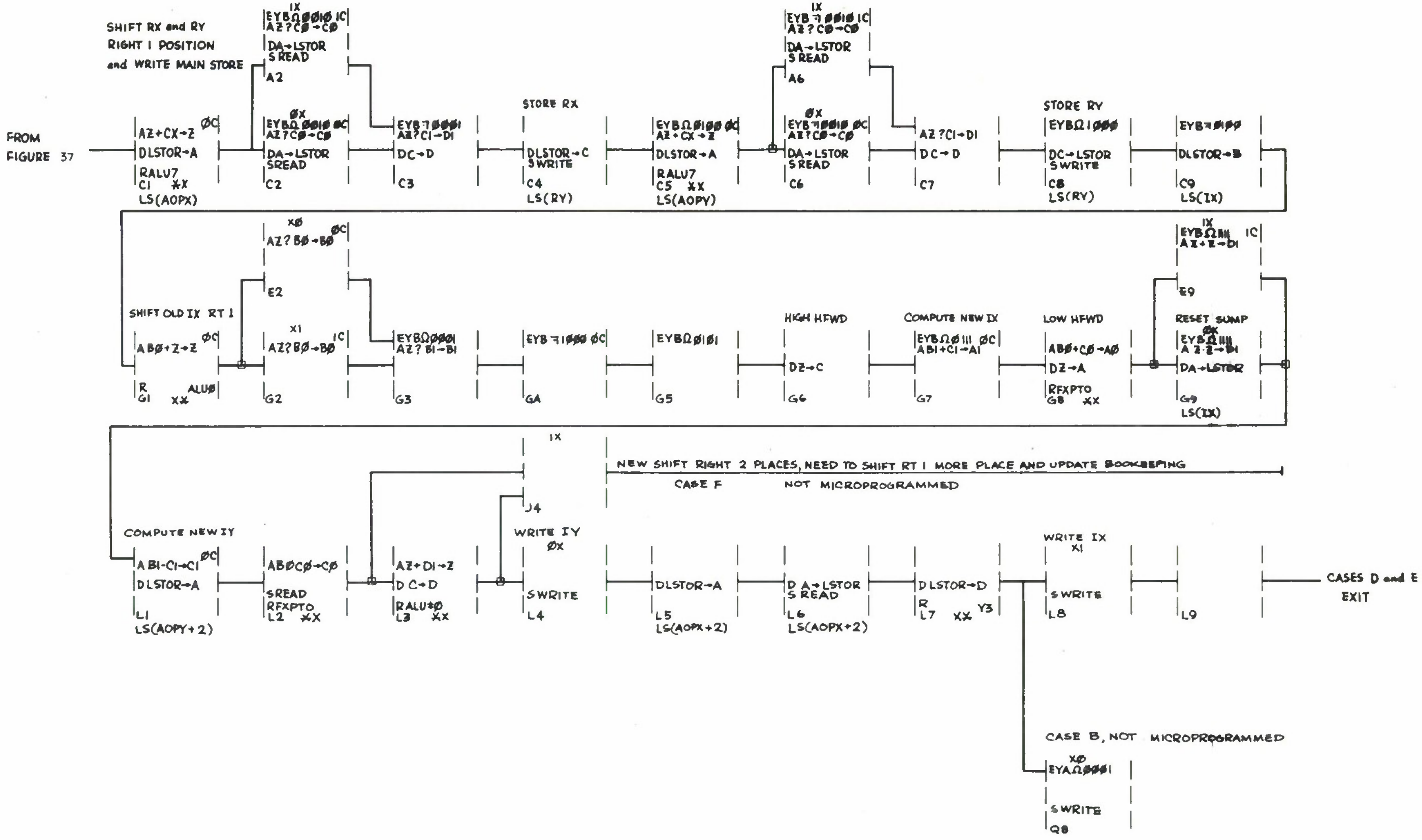


Figure 39. Inner Loop Microprogram IV

Table 5

TYPE II INNER LOOP EXECUTION ESTIMATES

Case	Probability	Microinstructions	Microseconds
A	0.5	86	54
B	0.01	78 + (5)*	52
C	0.001	81 + (20)*	63
D	0.23	78	49
E	0.23	78	49
F	0.001	73 + (15)*	55
G	0.028	(85)*	53
		Average	52

*In the cases not microcoded, the numbers in parentheses are estimates of the instructions to accomplish the following:

B: Set Y3, save array stop-addresses for post shift processing

C: Save stop addresses, shift operands 2 places, store

F: Save stop addresses, shift operands 1 place, store

G: Inner-loop process, shifting all results two places.

```
*                  REGISTER 1 CONTAINS ADDRESS OF OPERAND X, AOPX
*                  REGISTER 2 CONTAINS ADDRESS OF OPERAND Y, AOPY
*                  REGISTER 3 CONTAINS REAL PART OF W,RW, LEFT ADJUSTED
*                              IN LOW HALF WORD
*                  REGISTER 4 CONTAINS IW, LEFT ADJUSTED IN LOW HALF WORD
*
*                              PROGRAM 1   FIXED POINT DATA
*

 S1     LR         5,3         REG5=RW
        LR         6,4         REG6=IW
        LR         7,3
        LR         8,4
*
        MH         5,0(2)      REG5=RW*RY
        MH         6,2(2)      REG6=IW*IY
        MH         8,0(2)      REG8=IW*RY
*
        TM         PRIOR,X'01' IF BIT 7 OF PRIOR IS 1 CONDITION CODE IS 1
        BC         1,LET       BRANCH TO LET IF AT LEAST 1 PENDING SHIFT
*
*                              NO PRIOR PENDING SHIFT
*
        SR         5,6         RW*RY-IW*IY
        SLA        5,1
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        AR         7,8         RW*IY+IW*RY
        SLA        7,1
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        SRA        5,16
        SRA        7,16
        LH         9,0(1)      REG9=OPERAND X  REAL PART
        LR         10,9        REG10=OPERAND X
        SR         9,5      RY=RX-(RW*RY-IW*IY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        AR         10,5     RX=RX+(RW*RY-IW*IY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        STH        9,0(2)      STORE RY
        STH        10,0(1)     STORE RX
        LH         9,2(1)      REG9=IMAGINARY PART OF OPAND X
        LR         10,9
        SR         9,7      IY=IX-(RW*IY+IW*RY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        AR         10,7     IX=IX+(RW*IY+IW*RY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        STH        9,2(2)      STORE IY
        STH        10,2(1)     STORE IX
*
*                  END OF NO PRIOR PENDING SHIFT CASE
```

Figure 40. OS/360 Programs and Execution Estimates for FFT Inner Loop (I, II, and III)

```
*       AT LEAST ONE PRIOR SHIFT PENDING
*
 LET    SR         5,6         RW*RY-IW*IY
        AR         7,8         RW*IY+IW*RY
        TM         PRIOR,X'02' IF BIT 6 OF PRIOR IS 1 CONDITION CODE IS 1
        BC         1,ANTHER    2 PRIOR SHIFTS,CASE NOT PROGRAMMED
*                              ONE PRIOR SHIFT
        SRA        5,16
        SRA        7,16
*
        LH         9,0(1)      REG9=RX
        SRA        9,1
        LR         10,9        REG10=RX
        SR         9,5      RY=RX-(RW*RY-IW*IY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        AR         10,5     RX=RX+(RW*RY-IW*IY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
*
        STH        9,0(2)      STORE RY
        STH        10,0(1)     STORE RX
*
        LH         9,2(1)      REG9=IX
        SRA        9,1
        LR         10,9        REG10=IX
        SR         9,7      IY=IX-(RW*IY+IW*RY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
        AR         10,7     IX=IX+(RW*IY+IW*RY)
        BC         1,OVFL      OVERFLOW CASE NOT PROGRAMMED
*
        STH        9,2(2)      STORE IY
        STH        10,2(1)     STORE IX
*
*                  END OF ONE PRIOR SHIFT CASE
```

Figure 40. (continued)

```
∷
∷       TIME FOR INNER LOOP  FIXED POINT
∷       ASSUMING NO PENDING SHIFT
        INSTRUCTION NUMBER TIME TOTAL TIMES
            LR          6    7.5      45.0
            MH          4   45.0     180.0
            TM          2     9.38     18.76
            BC          6     9.38     56.28
            SR          3    7.5       22.5
            AR          3    7.5       22.5
            SRA 16      2   18.75      37.50
            SLA  1      4   10.        40.
            LH          2   18.13      36.26
            STH         2   10.63      21.26
∷       TOTALS         34            480.06 MICROSECONDS
```

```
∷

        TIME FOR INNER LOOP  FIXED POINT
∷       ASSUMING ONE PENDING SHIFT
∷
∷       INSTRUCTION NUMBER TIME  TOTAL TIMES
            LR          6    7.5      45.0
            MH          4   45.0     180.0
            TM          2     9.38     18.76
            BC          6     9.38     56.28
            SR          3    7.5       22.5
            AR          3    7.5       22.5
            SRA 16      2   18.75      37.50
            STH         4   10.        40.
            SRA 1       2   18.13      36.26
            LH          2   10.63      21.26
∷       TOTALS         34            480.06 MICROSECONDS
```

Figure 40. (concluded)

```
*
*
*
*                  REG1 CONTAINS AOPX,IMAGINARY PART AT AOPX+4
*                  REG2 CONTAINS AOPY,IMAGINARY PART AT AOPY+4
*                  FLREGO CONTAINS RW
*                  FLREG2 CONTAINS IW
*
         LER       4,0          FLREG 4 = RW
         LER       6,2          FLREG 6 = IW
         ME        4,0(2)       RW*RY
         ME        6,4(2)       IW*IY
         SER       4,6          RW*RY-IW*IY
         LE        6,0(1)       RX
         SER       6,4          RX-(RW*RY-IW*IY)
         STE       6,TEMP       TY TEMP STORE
         LE        6,0(1)       RX
         AER       6,4          RX+(RW*RY-IW*IY)
         STE       6,0(1)       RX STORED
         LER       4,0          RW
         LER       6,2          IW
         ME        4,4(2)       RW*IY
         ME        6,0(2)       IW*RY
         AER       4,6          RW*IY + IW*RY
         LE        6,4(1)       IX
         SER       6,4          IX-(RW*IY+IW*RY)
         STE       4,4(2)       STORE IY
         LE        6,4(1)       IX
         AER       6,4          IX + (RW*IY+IW*RY)
         STE       6,4(1)       STORE IX
         MVC       4(4,1),TEMP  STORE RY
*
*                  END OF FLOATING POINT PROGRAM
*        TIME FOR INNER LOOP         FLOATING POINT
*
*        INSTRUCTION NUMBER TIME       TOTAL TIMES
            LFR         4    7.5          30.0
            ME          4   80.63        322.52
            SER         3   14.3          42.9
            LE          4   11.88         45.76
            STE         4   12.5          50.0
            AER         3   14.3          42.9
            MVC         1   26.25         26.25
*        TOTALS        23                560.33 MICROSECONDS
```

Figure 41. Floating Point Data

These figures, divided by the 52 μsecs microcode estimate, give:

Advantage over OS/360 Fixed Point:	9.2
Advantage over OS/360 Floating Point:	10.8

Remarks on Above Results

a. Note that the inner loop with 0-pending and 0-found overflow is longer in S/360 execution than the case with 1-pending and 0-found. This is so because the normalized format of the data gives a natural right shift of one place in multiplication. This must be unshifted in the no-pending shift case; in the 1-pending shift case the result is correctly positioned after the multiply, without further processing. Taking advantage of this gives rise to the discrepancy between Case A, and cases D and E, making the pending shift cases D and E faster.

b. The floating point S/360 process does not appear unduly longer than the fixed point (560 to 480). Note that the floating point process involves no testing for overflow, and no special provisions for shifting.

c. In the microcoded cases a pending shift is processed by shifting the result of the innerloop computation. Alternatively, a pending shift may be executed on the input data before the inner loop processing.

If input data is shifted, there is an improvement in speed over the output result shifting because some ALU idle time during SUMP operation can be utilized for the process. But the preshift loses accuracy: Note that if σ denotes shift,

$$\sigma x + \sigma y \;=\; \sigma(x+y)$$

For $\quad \sigma(01) + \sigma(01) = 0 + 0 = 0$

but $\quad \sigma(01 + 01) = \sigma(10) = 1$

The amount of time saving in executing the pending shift prior to inner loop execution will be about five microinstructions, or about five percent of the inner loop.

d. The estimates given are for the power-of-two FFT process.

11.4.3.4 The Overall FFT Process

Besides the inner loop, the Fast Fourier Transform comprises:

a. Index Processing, for accessing operands for the inner loop

b. Initialization, instruction fetch and setting initial counter values and working local storage.

c. Post shifting, suppose an operand array has 256 data points. The inner loop is executed 128 times in each stage. If an overflow occurs first in the 64th execution, all inner loop outputs from the 64th to the 128 will be properly positioned in storage. But the stage will not be completed until the outputs from the first 63 inner loop executions are also right shifted, in the post shifting process.

d. Data scramble routine, for bit reversal of operand addresses. A FORTRAN program (Figure 42), flowchart (Figure 43) and microcode estimate (Figure 44) of the scramble routine are given. The microprogramming is illustrated by figures 45 and 46. It is interesting to note that the advantage for the scramble routine is appreciably higher than the usual seven or eight for a scientific application. In general, the advantage of microcode derives from the elimination of I-Fetches in a sequence of system instructions, and from the determinate operands. In the microcoded scramble routine, there is another factor: The "branch on condition" is virtually free in microcode, but costs a system instruction. The scramble routine's high advantage results from the highly branching nature of the routine.

It is well known that the number of inner loop executions in the total FFT processing of $2^M = N$ complex points is $M \cdot 2^{M-1}$ (M is the number of stages through the array, and 2 points of an array are processed with each inner loop execution). Experience indicates that the overall FFT execution time may be estimated as twice the inner loops execution time:

$$2 \cdot M \cdot 2^{M-1} = t \cdot N \cdot \log_2 N,$$

where the initial 2 provides for items a through d above, and where t is execution time per inner loop. Our estimate of t is 52 microseconds, so that:

$$52N \log_2 N, \ \mu\text{secs}$$

is the microcoded FFT process time for an array of N points. A few values are given in Table 6.

```
*
*
*
*
        J=1
        N1=N/2
        IMAX=N-1
        DO 3 I=1,IMAX
        IF(I.GE.J) GO TO 5
        XT=X(J)
        X(J)=X(I)
        XI=XT
        YT=Y(J)
        Y(J)=Y(I)
        Y(I)=YT
 5      K=NT
 4      IF(K.GE.J) GO TO 3
        J=J-K
        K=K/2
        GO TO 4
 3      J=J+K
        RETURN
*
*
*               OS/360 PROGRAM
 INIT   LH      2,ONE           REG2=J,INITIALLY = 1
        LH      3,N
        SRA     3,1             REG3=NT, INITIALLY = N/2
        LH      8,ONE           LOAD INCREMENT
        LH      1,ONE           REG1=I,  INITIALLY = 1
        LH      9,IMAX          LOAD COMPARAND
        LR      4,1
 LOOP1  SR      4,2        LOOP 2      I-J
        BC      4,EACH
 LOOPX  LR      5,3        LOOP 1      K=NT
 LOOP3  LR      6,5
        SR      6,2                    K-J
        BC      10,UPIJ
        SR      2,5                    J=J-K
        SRA     5,1
        BC      15,LOOP3
 UPIJ   AR      2,5                    J=J+K    LOOP 1
        BXLE    1,8,LOOP1              I=I+1
        EXIT
 EXCH   LR      11,1
        SLA     11,2
        AR      11,BASE
        LM      13,13,0(11)     LOOP 2
        LR      12,2
        SLA     12,2
        AR      12,BASE
        MVC     0(4,11),0(12)          SET NEW XT
        STM     13,13,0(12)            SET NEW YT
        BC      15,LOOPX
```

Figure 42. Jennrich Scramble Fortran Program

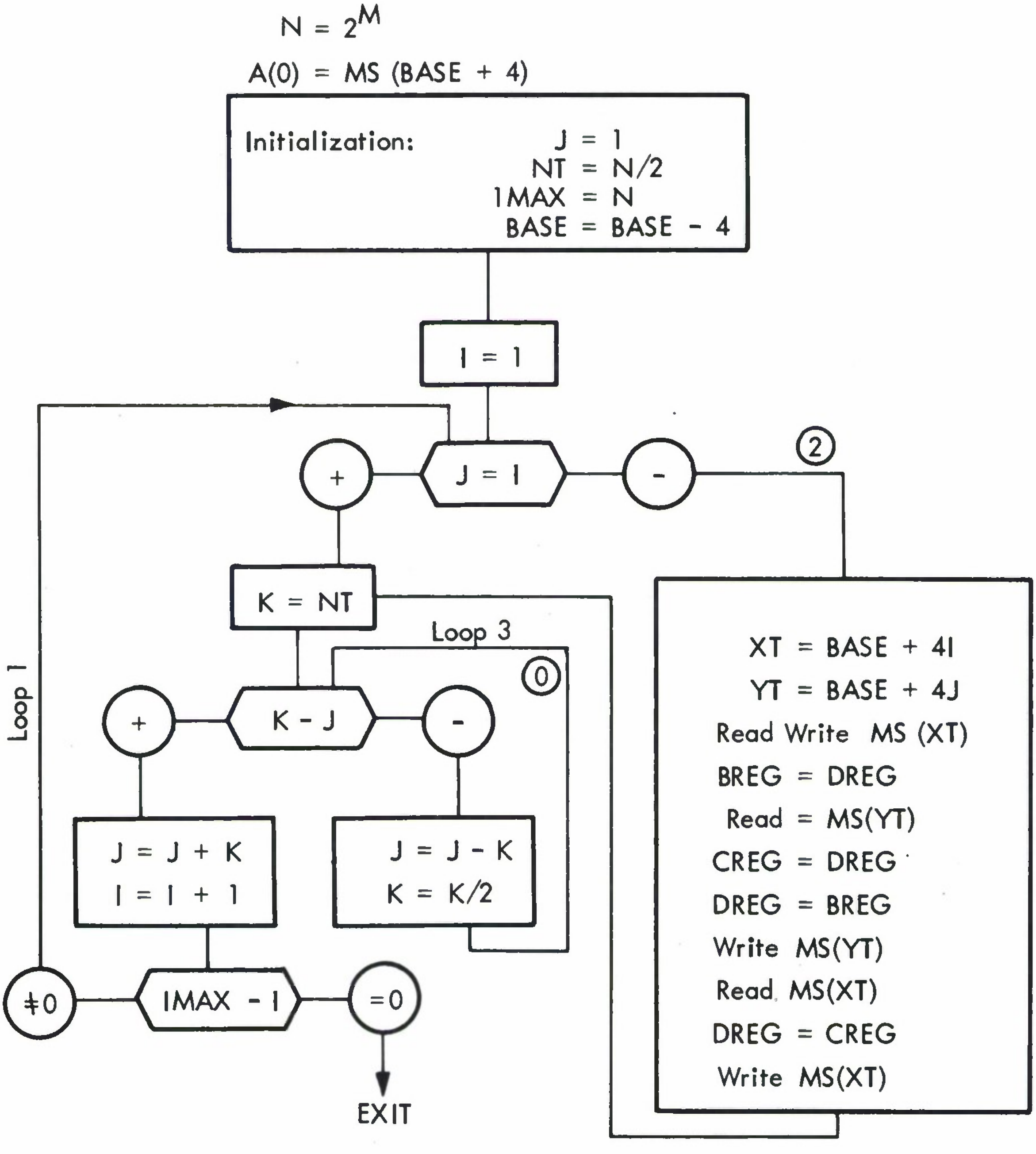


Figure 43. Jennrich Scramble Flow Chart

```
*

*
*       LOOP 1
*       I-J GREATER THAN OR EQUAL TO 0
*       K-J GREATER THAN OR EQUAL TO 0
*       INSTRUCTION NUMBER TIME   TOTAL TIMES
            SR,AR      3    7.5       22.5
            BC         2   9.38       18.76
            LR         2    7.5       15.0
            BXLE       1   16.26      16.26
        TOTALS         8              72.52 MICROSECONDS
*       LOOP 2
*       INSTRUCTION NUMBER TIME   TOTAL TIMES
            AR,SR      3    7.5          22.5
            BC         2   9.38          18.76
            LR         2    7.5          15.
            SLA        2   25.           50.
            LM         1    6.25+5       11.25
            MVC        1   16.25+2.5*4   26.25
            STM        1    6.87+5       11.87
        TOTALS        12                155.63 MICROSECONDS
        LOOP 3
        INSTRUCTION NUMBER TIME   TOTAL TIMES
            LR         1    7.5        7.5
            SR         2    7.5       15.
            BC         2   9.38       18.76
            SRA        1   18.13      18.13
        TOTALS         6              59.39 MICROSECONDS



*       SUMMARY
*       LOOP S/360 CODE     MICROCODE    MICROCODE TIME
              MICROSECS     INSTRUCTIONS MICROSECS            ADVATAGE
         1      72.52          10.5          6.56                 11
         2     155.63          22           13.75                 11
         3      59.39           6            3.75                 15
```

Figure 44. Speed Advantage Estimates for the Loops

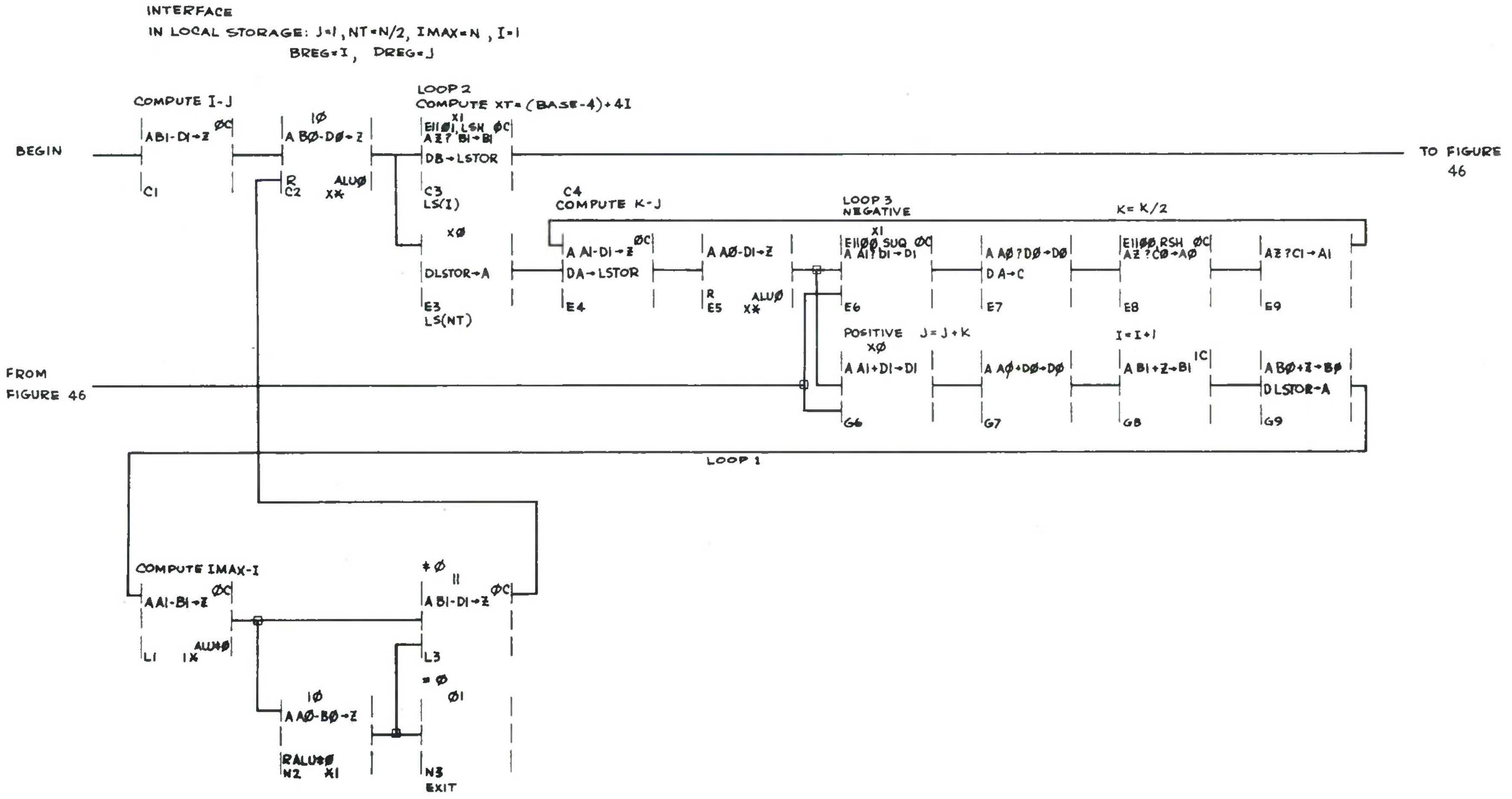


Figure 45. Jennrich Scramble Microprogram I

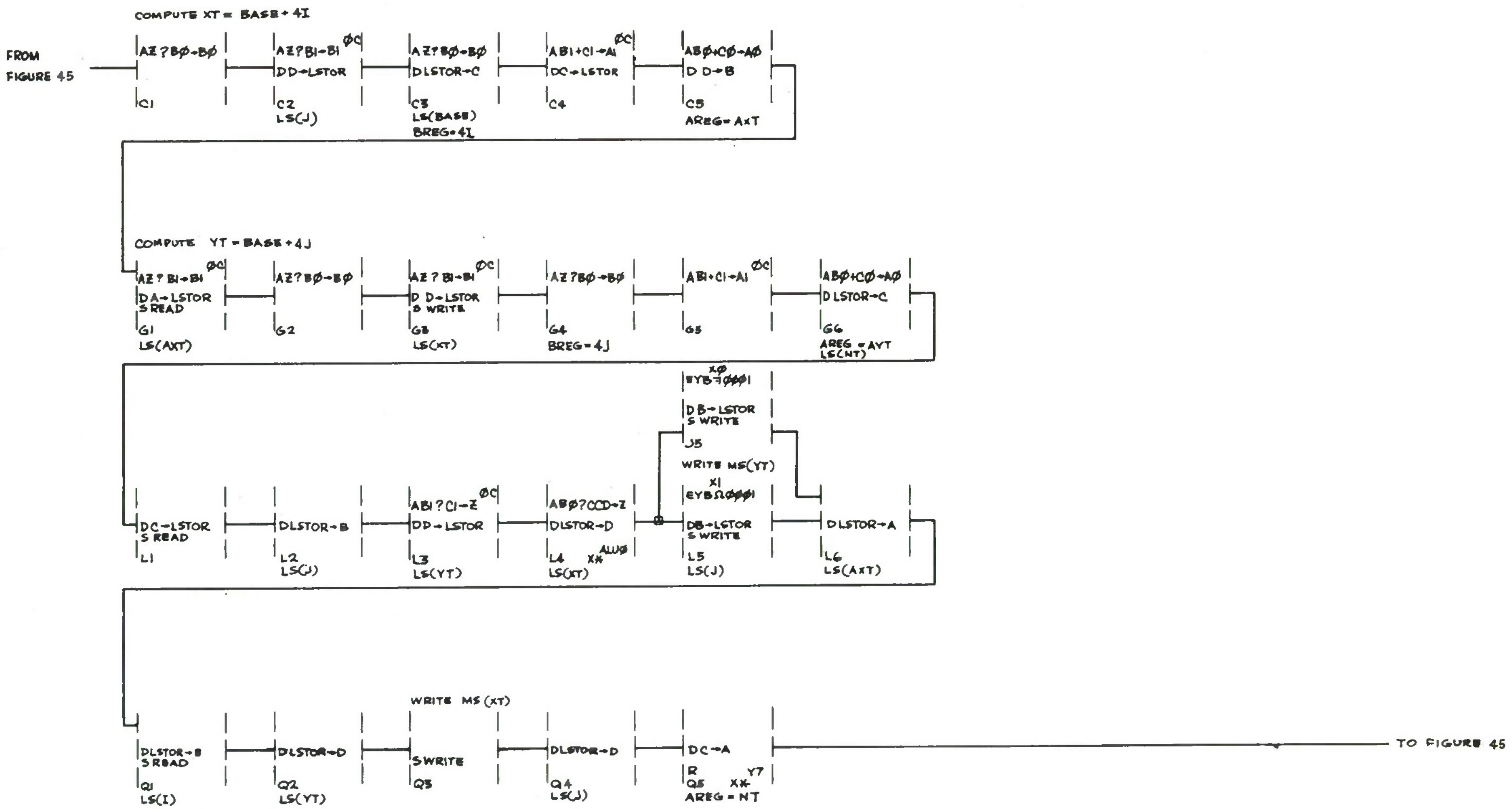


Figure 46. Jennrich Scramble Microprogram II

Table 6

FFT TOTAL TIME

Array Size		Total Time
M	N	
4	16	3 m secs
8	256	100 m secs
12	4K	2.3 sec
16	64K	52 sec

II.5 ARRAY OPTIMIZATION

II.5.1 Introduction

An aspect of array design based upon maximizing gain as a result of judicious choice of element placement in accordance with a model of the noise field is presented. The technique has utility in a relative sense since its practical application is dependent upon and limited to situations where noise data is available and a comparative analysis is desired. In Section 2, the formulation and procedure is outlined and some results are presented in Section 3.

II.5.2 Formulation

Assuming signal coherence and equal noise power at each element in an N element array, the conventional processing gain[6] can bc expressed as:

$$G = 10 \log_{10} \frac{N}{1 + (N-1)\,\overline{\rho}} \, dB, \tag{83}$$

where the average correlation coefficient is:

$$\overline{\rho} = \frac{1}{N(N-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \rho_{ij} \tag{84}$$

and where ρ_{ij} is the pair-by-pair element noise correlation coefficient.

By replacing ρ_{ij} with $\hat{\rho}(r_{ij})$, data derived models of the correlation coefficient as a function of distance between elements (r_{ij}) can be synthesized for application in Equation (83). By restricting the class of functions to prohibit maximums, the average value of the differential of $\bar{\rho}$ is:

$$\overline{d\bar{\rho}} = (1/N) \sum_{i=1}^{N} \vec{S}_i \cdot d\vec{r}_i \, , \tag{85}$$

where

$$\vec{S}_i = \frac{2}{N(N-1)} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[\frac{\partial \hat{\rho}(r_{ij})}{\partial r_{ij}} \right] \left[\frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} \right] . \tag{86}$$

The arrow notation is used to denote vector quantities.

To simplify computation $\hat{\rho}$ has been modified to be a function of r_{ij}^2 as follows:

$$\vec{S}_i = \frac{4}{N(N-1)} \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{\partial \hat{\rho}(r_{ij}^2)}{\partial (r_{ij}^2)} \vec{r}_{ij} \quad . \tag{87}$$

This eliminates the need for computing $|\vec{r}_{ij}|$. If $\vec{r}_{ij}$ equals zero when computing $\vec{S}_i$, instrument j is moved to reduce the magnitude of the $\vec{r}_j$ which is measured relative to the center of the array. This restricts two instruments from having the same position. The maximum change in $\bar{\rho}$ is produced when $d\vec{r}_i$ is in the same direction as $\vec{S}_i$; therefore, the sensitivity vector $\vec{S}_i$ computed for each instrument determines the direction of movement. The instrument movement is determined by:

$$\Delta\vec{r}_{ij} = \frac{-K \vec{S}_i}{\sqrt{\sum_{i=1}^{N} |\vec{S}_i|^2}} \, , \tag{88}$$

where K controls the incremental step size, and, thus, the magnitude of each instrument move is proportional to the ratio of its sensitivity vector to the system sensitivity.

II.5.2.1 Constraints

If an instrument is on the array constraint boundary, an instrument movement outside of the boundary is modified to be the projection of the $\overrightarrow{\Delta r_i}$ on the tangent to the constraint circle at the instrument position. When an instrument is moved outside of the constraint boundary, the instrument is returned to the constraint boundary along the instrument vector relative to the center of the array. A similar technique is used for other types of constraints. For a spoke configuration where all instruments must lay on spokes through the center of the array, in addition to being within a fixed diameter circle, the instrument position vector is rotated to correspond to the nearest spoke after each move.

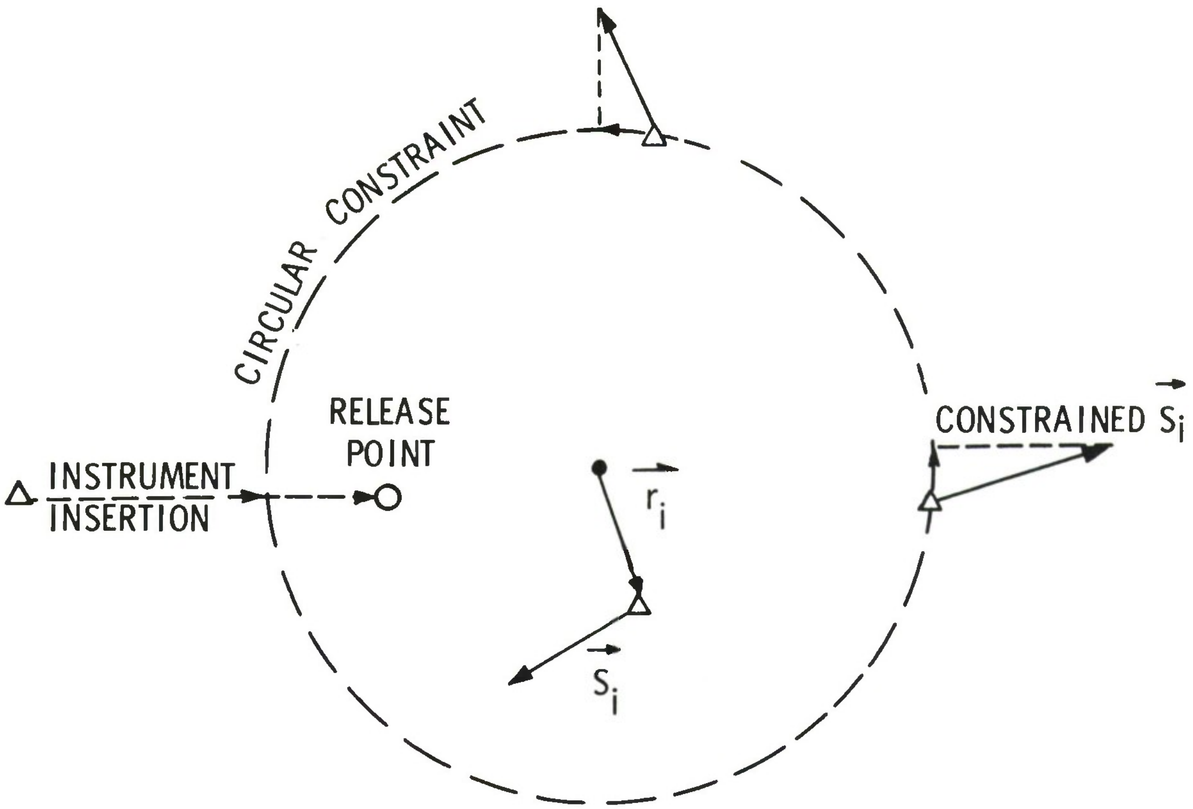
II.5.2.2 Program Operation

After computing the directions of movement, all instruments are moved at the same time. Then, a new set of sensitivity vectors is computed. This process continues until either the sensitivity vector or its projection on the constraint (when constrained) falls below a variable threshold. A three instrument configuration is shown in Figure 47 to demonstrate the vector relationship associated with the program. For sufficiently small values of K, this technique should produce convergence along the path of steepest descent toward an array configuration with a minimum value of $\overline{\rho}$.

II.5.2.3 Operator Control

To permit the array designer to dynamically control the threshold and increment of instrument position change, the array configuration is displayed on an IBM 2250 CRT Display. Controls for modifying the program parameters are provided.

The array configuration that satisfies the threshold setting is a function of the threshold values. A better instrument configuration can be achieved by reducing the magnitudes of both the threshold and instrument position change. The array configuration thus achieved satisfies a local minimum value of $\overline{\rho}$. The program does not have a test to determine that the minimum corresponds to a global minimum (the "optimum" configuration). Confidence in selecting an array configuration can only be achieved by repeated runs (preferably with different initial conditions) to test for better design configurations. This technique is intended only to provide insight into selecting an array geometry by providing a model to serve as a departure point. A final array design must be tempered with geographic constraints and data acquisition considerations.



SENSITIVITY VECTOR:

$$\vec{S}_i = \frac{2}{N(N-1)} \sum_{j \neq 1} \frac{\partial \hat{\rho} (r_{ij}}{\partial \bar{r}_{ij}} \frac{\vec{r}_{ij}}{|\bar{r}_{ij}|}$$

CORRELATION COEFFICIENT

$$\bar{\rho} = \frac{1}{N(N-1)} \sum_{\substack{i,\ j=1 \\ i \neq j}}^{N} \rho_{ij}$$

NOTE: Instruments exceeding constraint are radially retracted.
Gain Estimation:

$$G = 10 \log_{10} \frac{N}{1 + (N-1)\bar{\rho}} \text{ dB}$$

Figure 47. Array Optimization Program

II.5.3 Results

II.5.3.1 Array Performance

Array configurations generated using the optimization program are shown in Figure 48 for circular constraints with diameters of 7 and 20 kilometers using an exponential function for the estimated noise correlation model. Predicted array performance using exponential functions with different decay rates are shown in figures 49 and 50. The configurations generated by the array optimization programs are compared with the current LASA subarray configurations and configurations where all instruments are constrained to be on a circle. For the LASA seven-kilometer subarray, the inner rings are removed to demonstrate the change in performance. Using both noise correlation models, the LASA E3 subarray configuration approached the maximum gain obtained with the array optimization program. The LASA subarray configurations are shown in figures 51 and 52.

The 16 kilometer NORSAR subarray with a modified hexagonal geometry currently being installed is shown in Figure 53. Perturbations in the geometry resulted from practical instrument siting considerations. A modified circular geometry and the configuration generated by the optimization program with a 16 kilometer constraint circle, are presented in figures 54 and 55, respectively. The modified circular geometry also resulted from practical considerations. In Figure 56, the optimal gain geometry is superimposed on the NORSAR subarray to demonstrate the differences in the configurations. Table 7 summarizes the array performance for the three configurations using an exponential noise correlation model. The NORSAR subarray and the circular configuration are evaluated over a range of exponential decay rates. In addition, the minimum communication line length required in each of the configurations is tabulated.

II.5.3.2 Measured Noise Correlation Data

To verify the validity of the noise correlation models, measurements of LASA subarray performance are presented in figures 57, 58, 59, and 60. The subarray performance for both unfiltered and filtered cases is shown. These performance characteristics we obtained by measuring the average correlation coefficient for the subarray over an 80 second interval. The two filters used were third order Butterworth with bandpasses of 0.9 to 1.4 Hz and 0.7 to 3.0 Hz, respectively. Figure 61 shows the exponential noise correlation models used for evaluating array performance. These models can be compared with the measured correlation coefficient between instruments as a function of the distance between them for three noise records. An 80 second noise record was used for each case. The separate noise records are identified by the event name or location which they preceded. The correlation coefficients were fitted with a sixth order polynomial to generate a continuous function. This data is presented in figures 62 through 70.

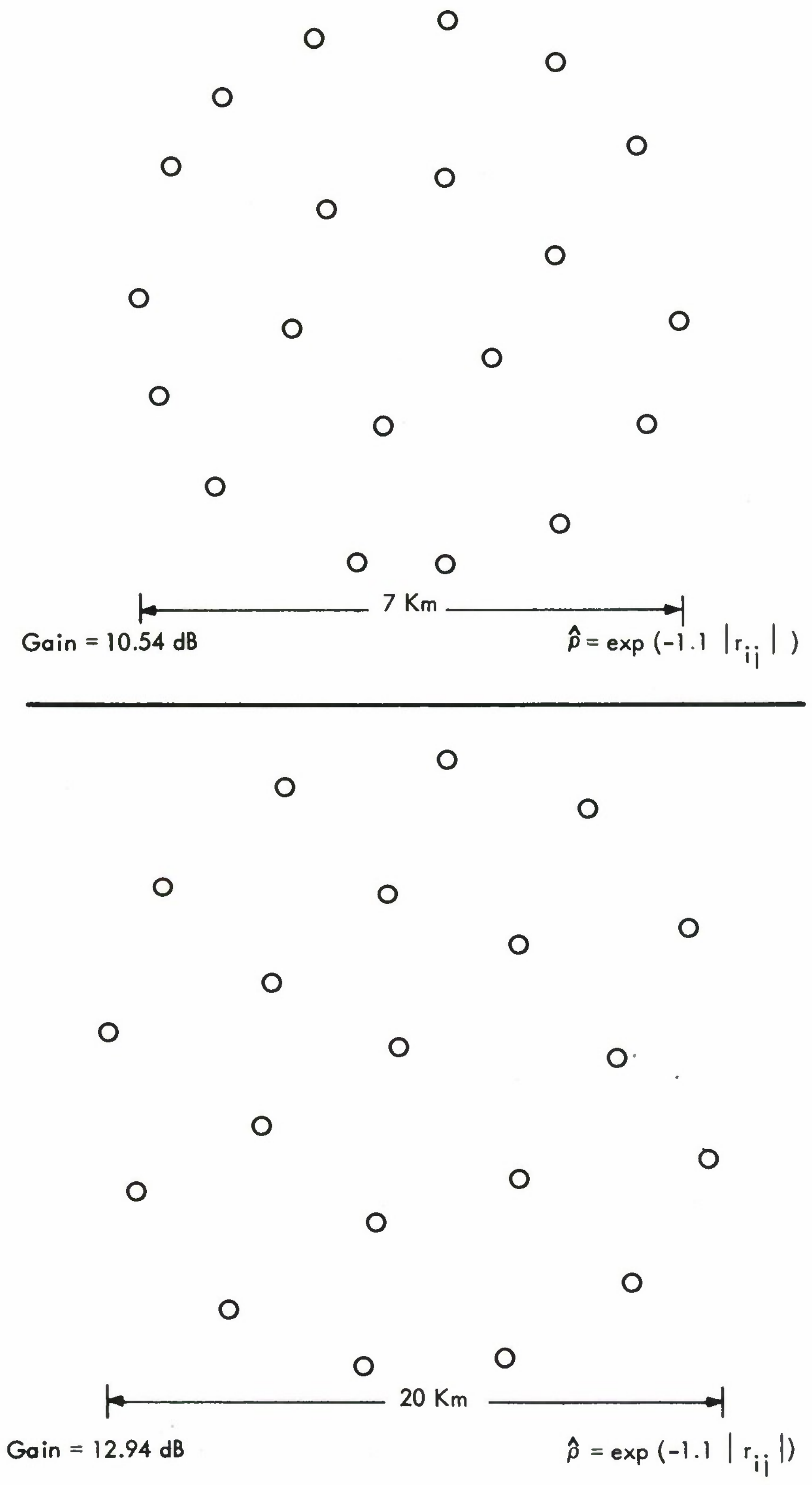

7 Km
Gain = 10.54 dB
$\hat{\rho} = \exp(-1.1 \; |r_{ij}|)$
20 Km
Gain = 12.94 dB
$\hat{\rho} = \exp(-1.1 \; |r_{ij}|)$

Figure 48. Array Configurations for 20 Instruments

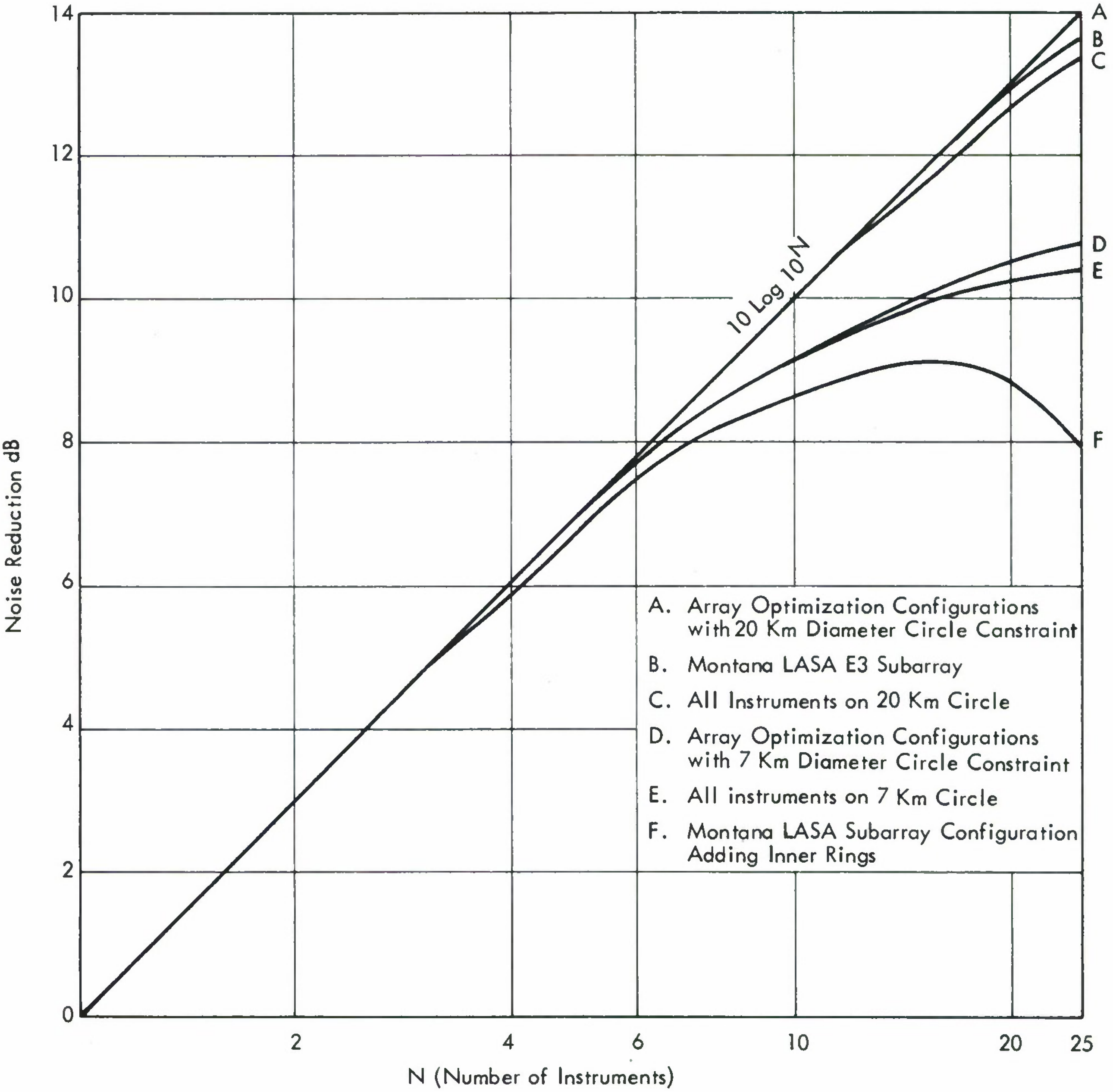


Figure 49. Predicted Array Performance I

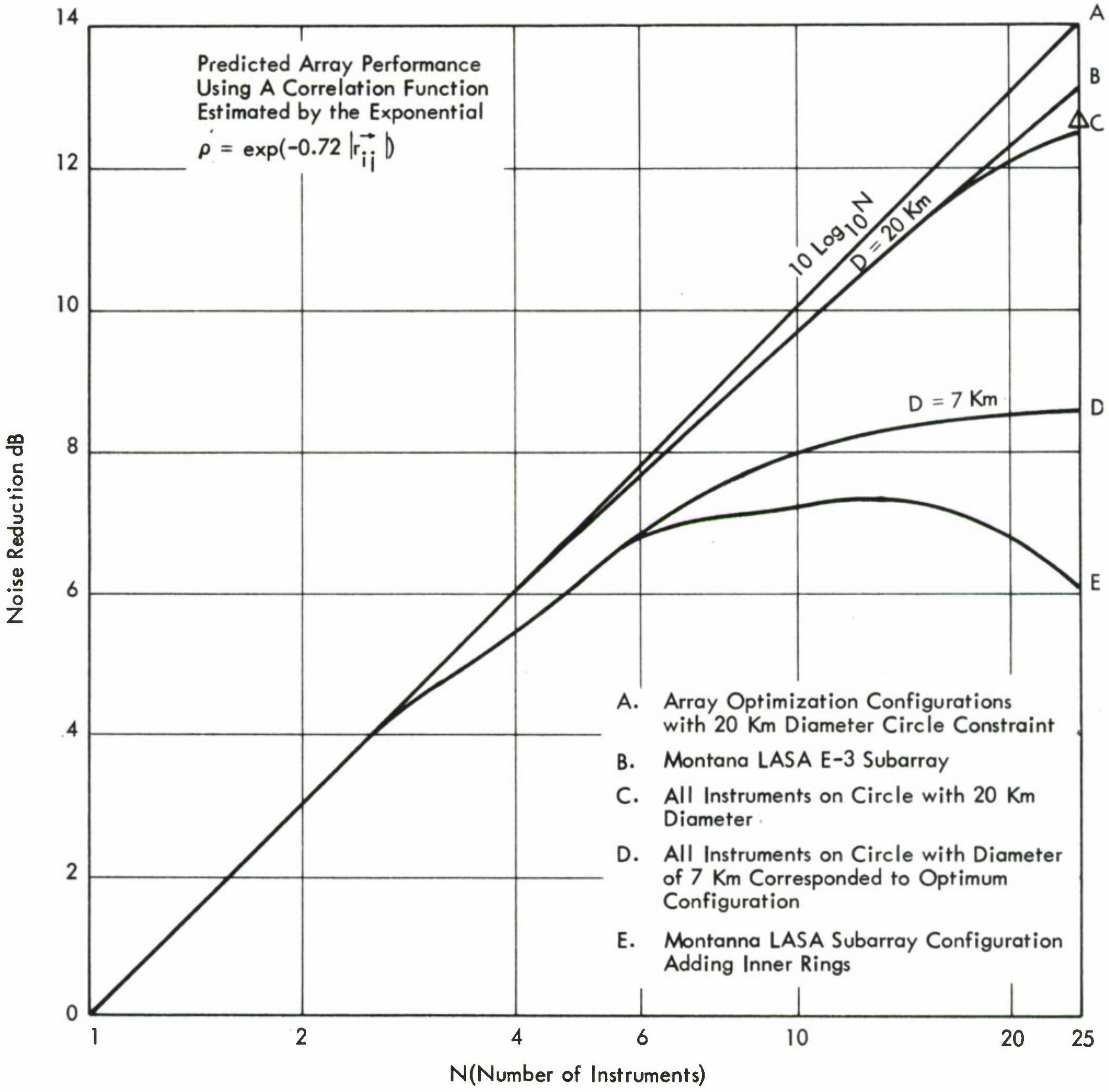
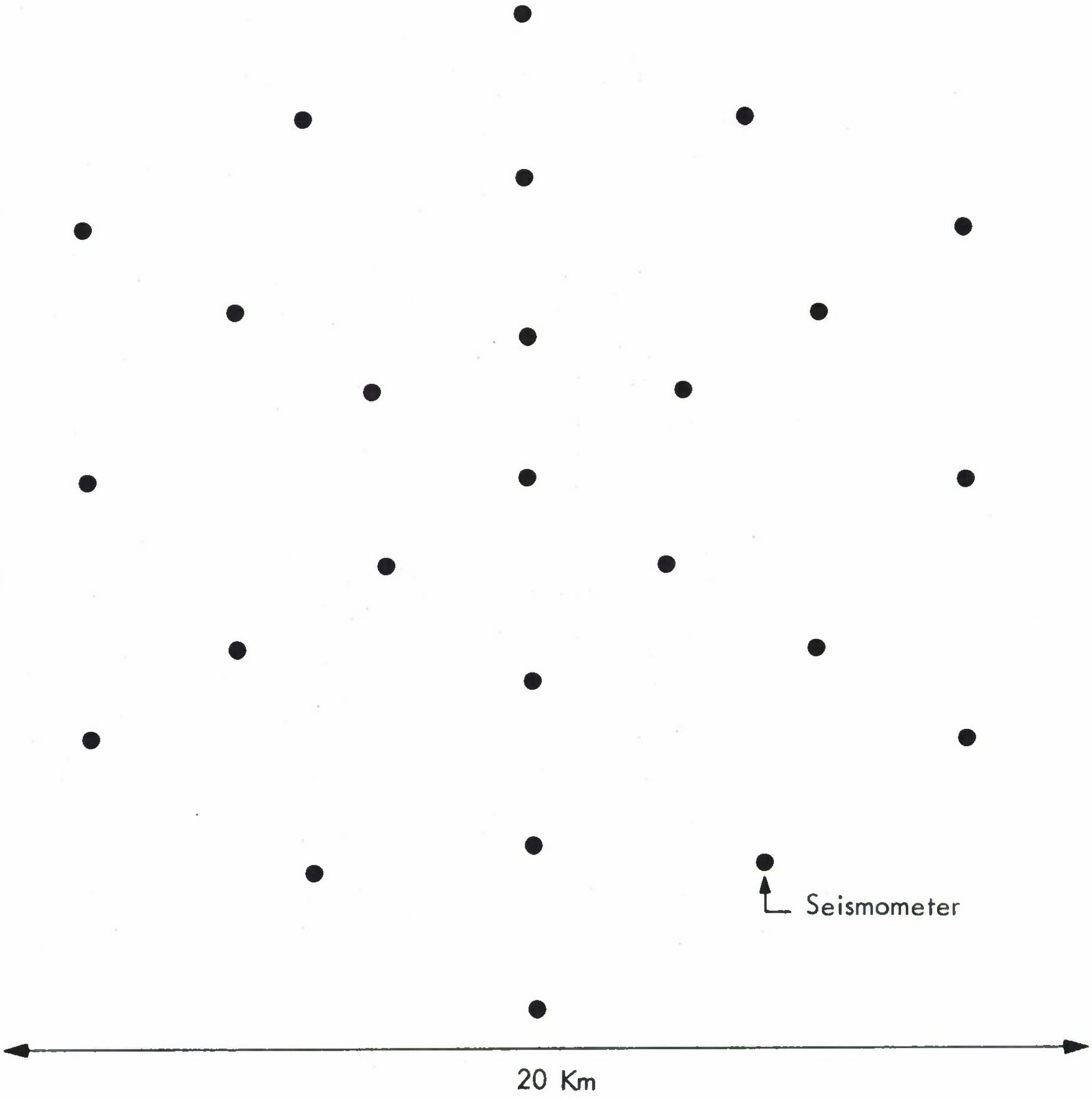


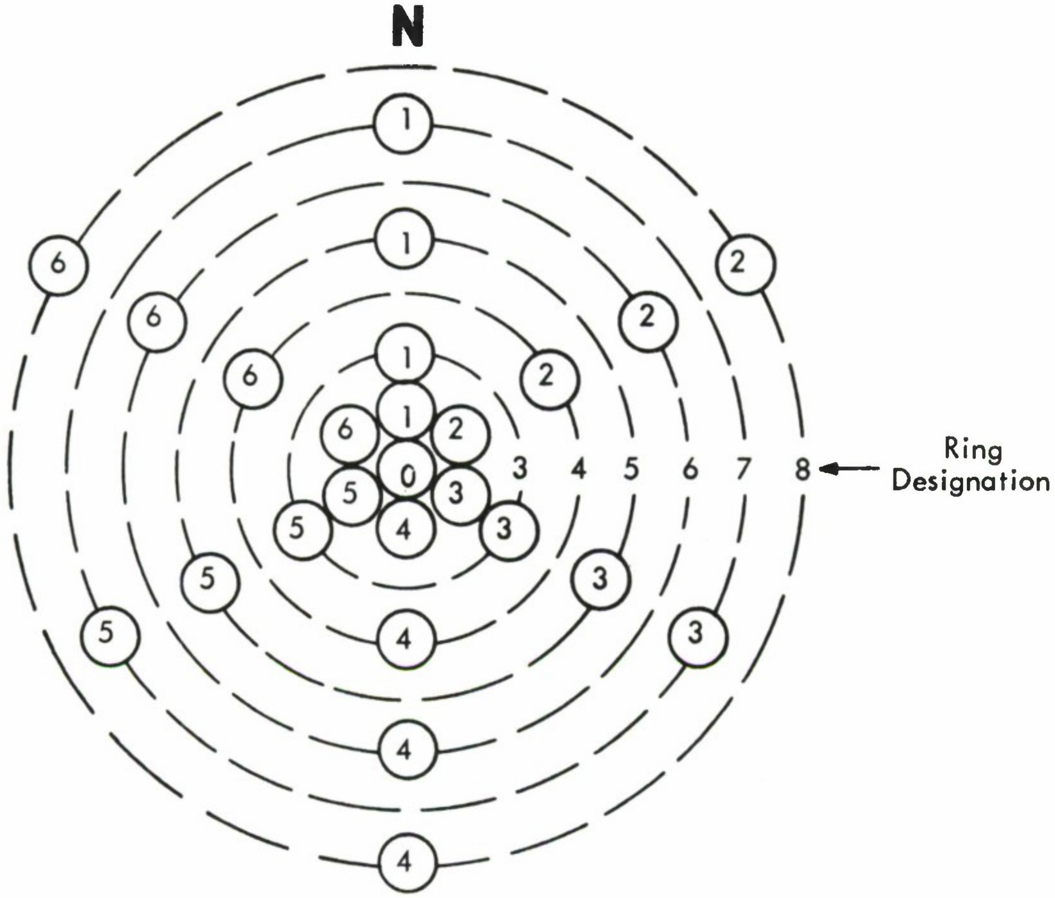
Figure 50. Predicted Array Performance II



Taken from:

MIT - Lincoln Lab.
Technical Note 1967-26
27 June 1967

Figure 51. 25 Seismometer E3 LASA Subarray



Taken From:

MIT - Lincoln Lab
Technical Note 1967-26
27 June 1967

Figure 52. LASA Subarray

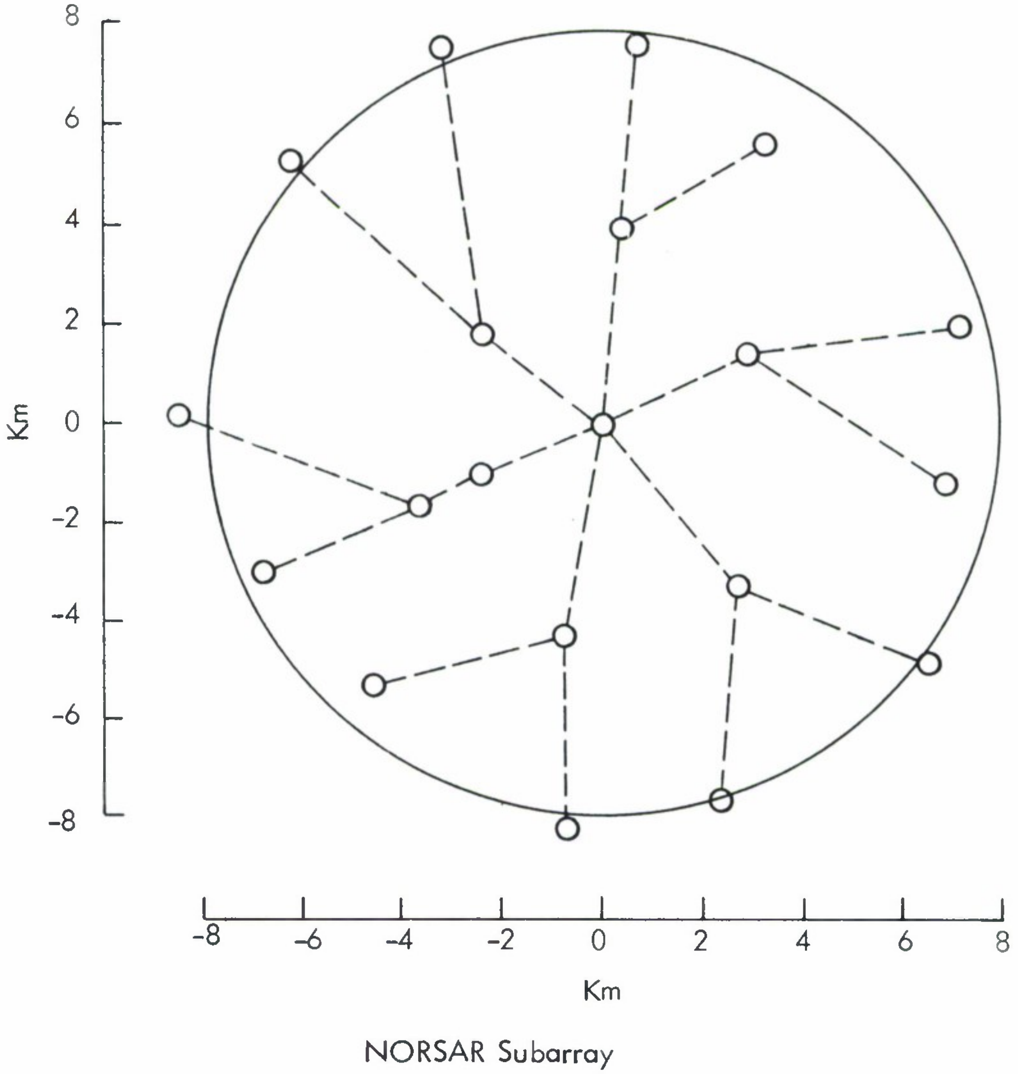


Figure 53. "Hexagonal" Geometry

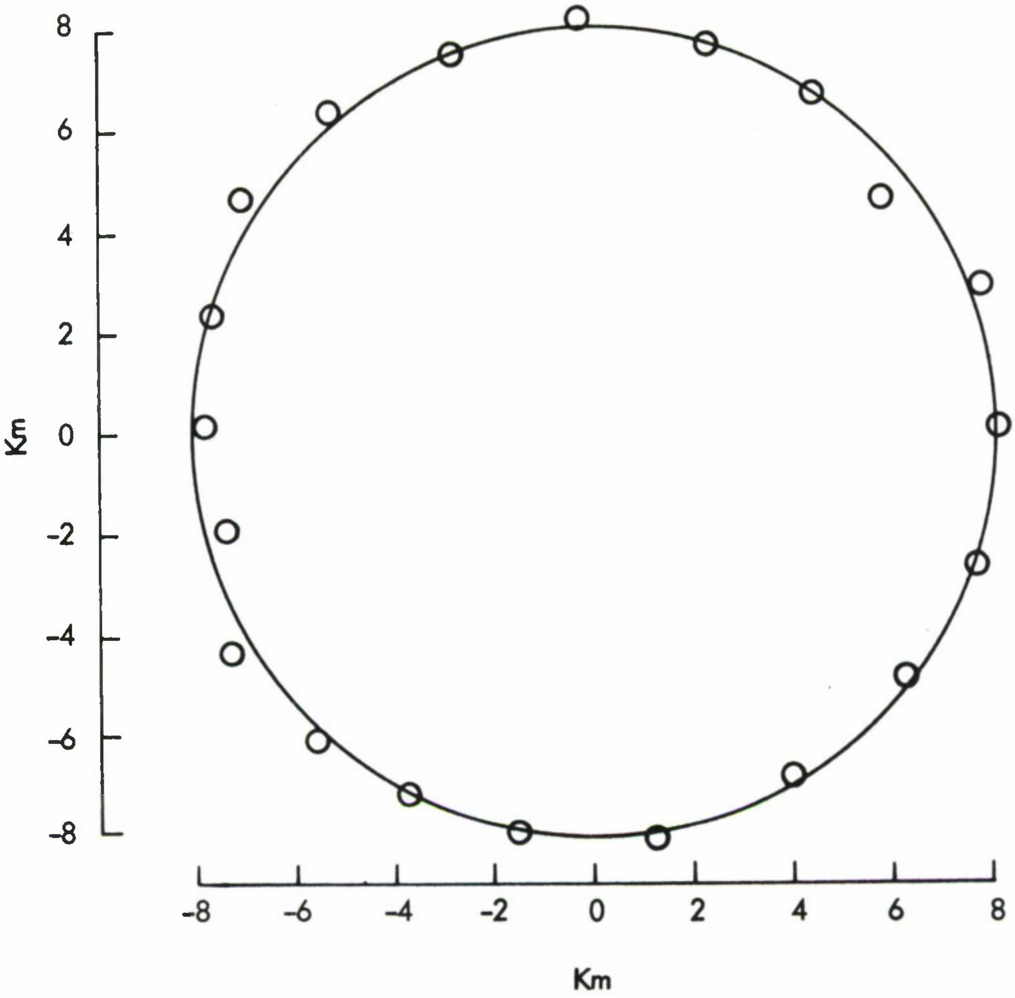


Figure 54. Modified Circular Geometry(D ≈ 16 Km)

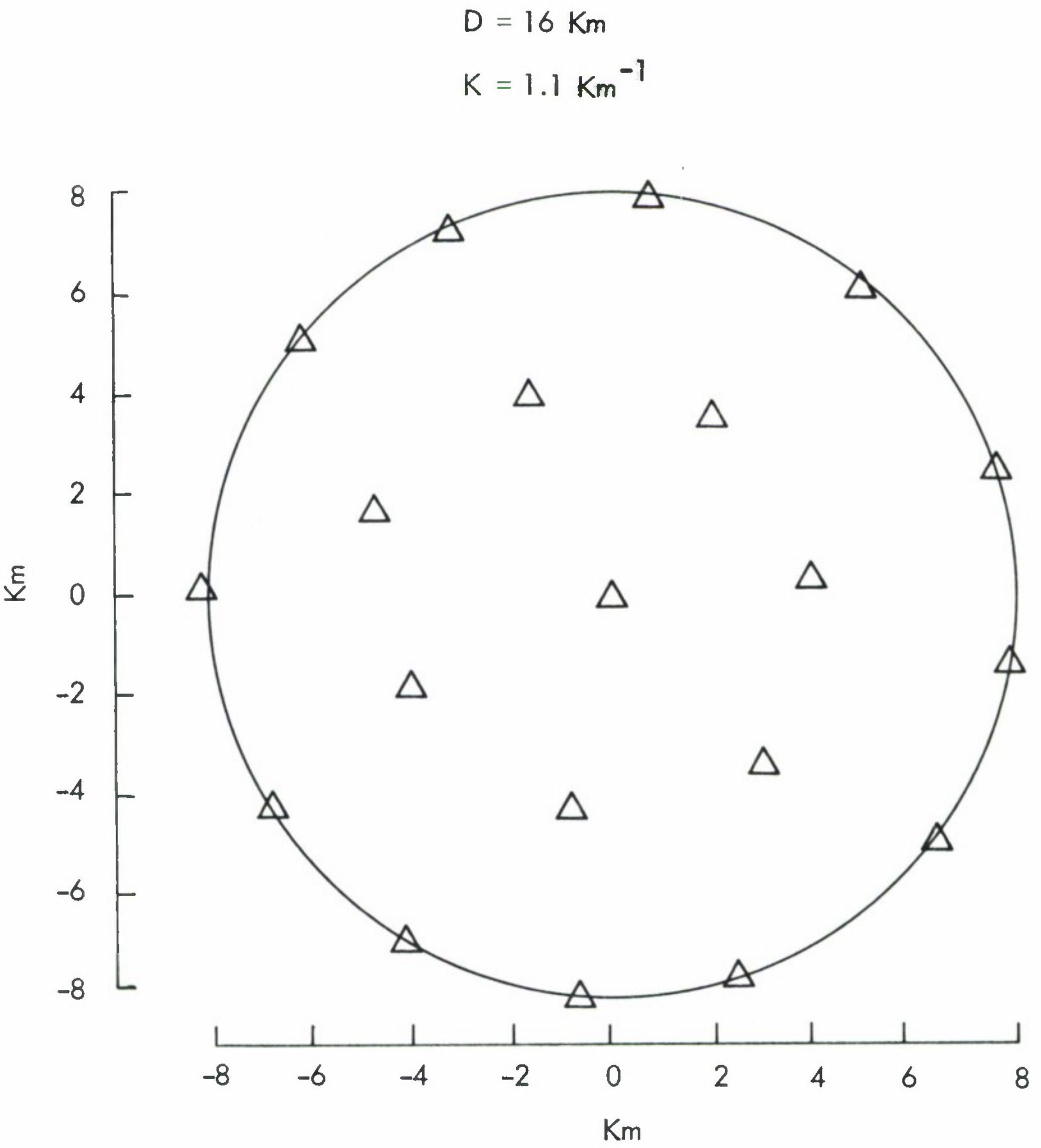


Figure 55. Optimum Gain Geometry

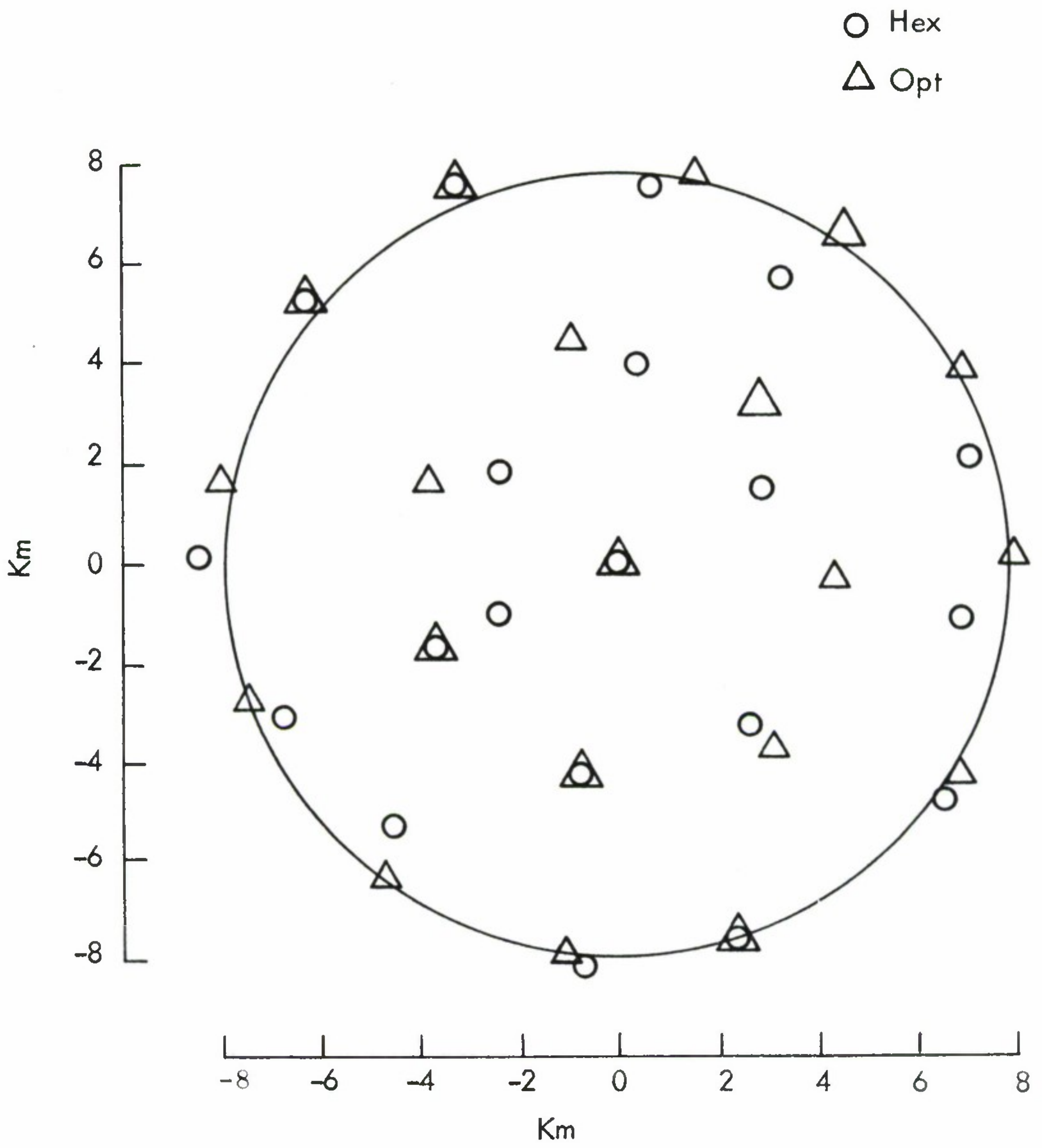


Figure 56. Superimposed Geometry (D ≈ 16 Km)

Table 7

SUMMARY OF NOISE REDUCTION PERFORMANCE
USING EXPONENTIAL NOISE CORRELATION MODELS FOR THE PRESENTED 16 Km,
20 INSTRUMENT ARRAY CONFIGURATION

	Km^{-1}	Optimum Gain Constraint dB	Norwegian Hexagonal* Subarray dB	Circular* D = 16 Km dB
Noise Reduction	0.54	---	10.55	10.66
	0.72	---	11.66	11.54
$\hat{\rho} = \exp - K\lvert\vec{r}_{ij}\rvert$	0.90	---	12.26	12.08
	1.10	12.80	12.61	12.45
	1.30	---	12.79	12.67
Subarray Communications (Kilometers)		77.0	74.3	48.2

*Geometry Perturbed To Place Instruments At Accessible Locations

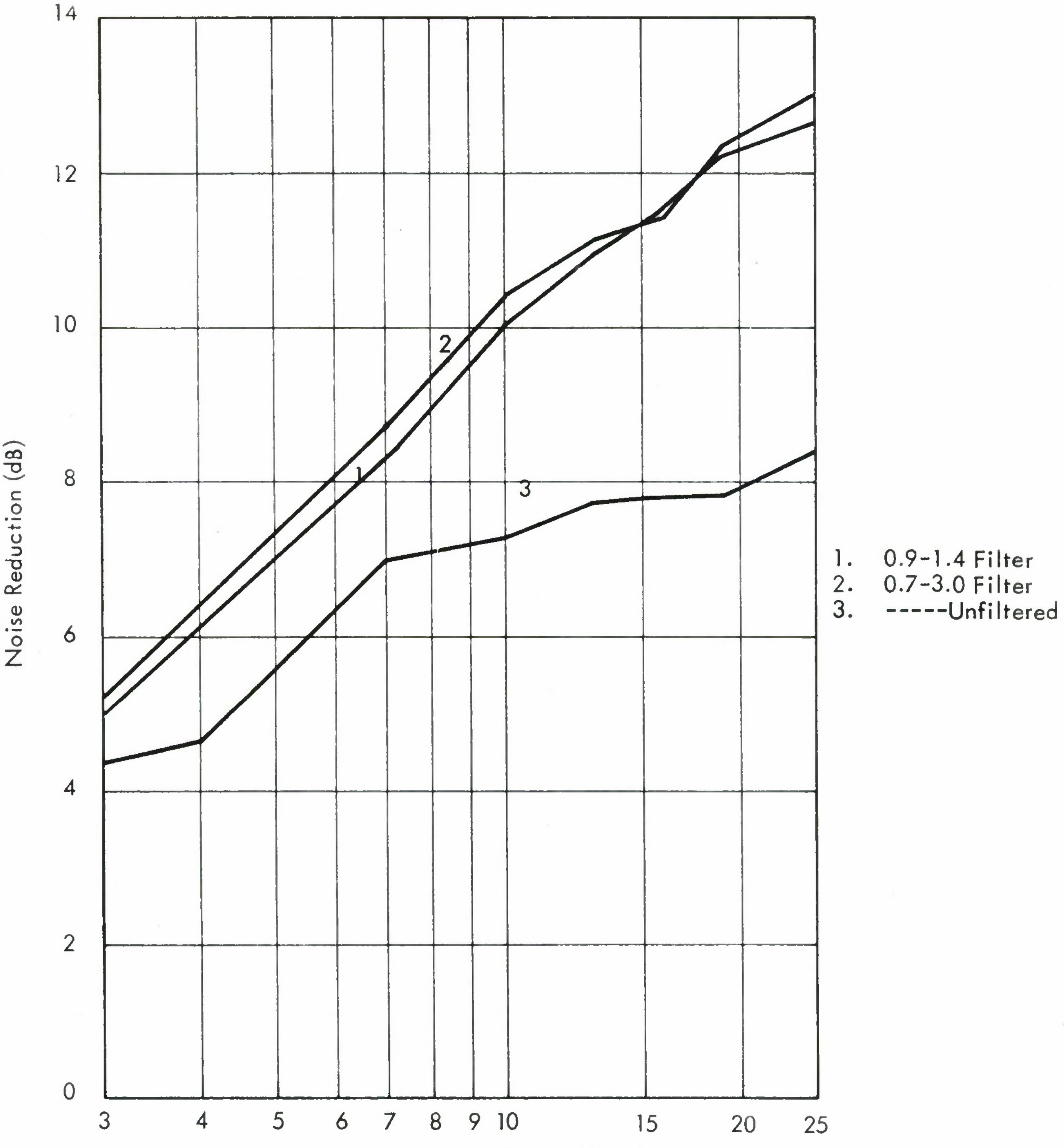


Figure 57. Observed Subarray Performance, Montana LASA E3

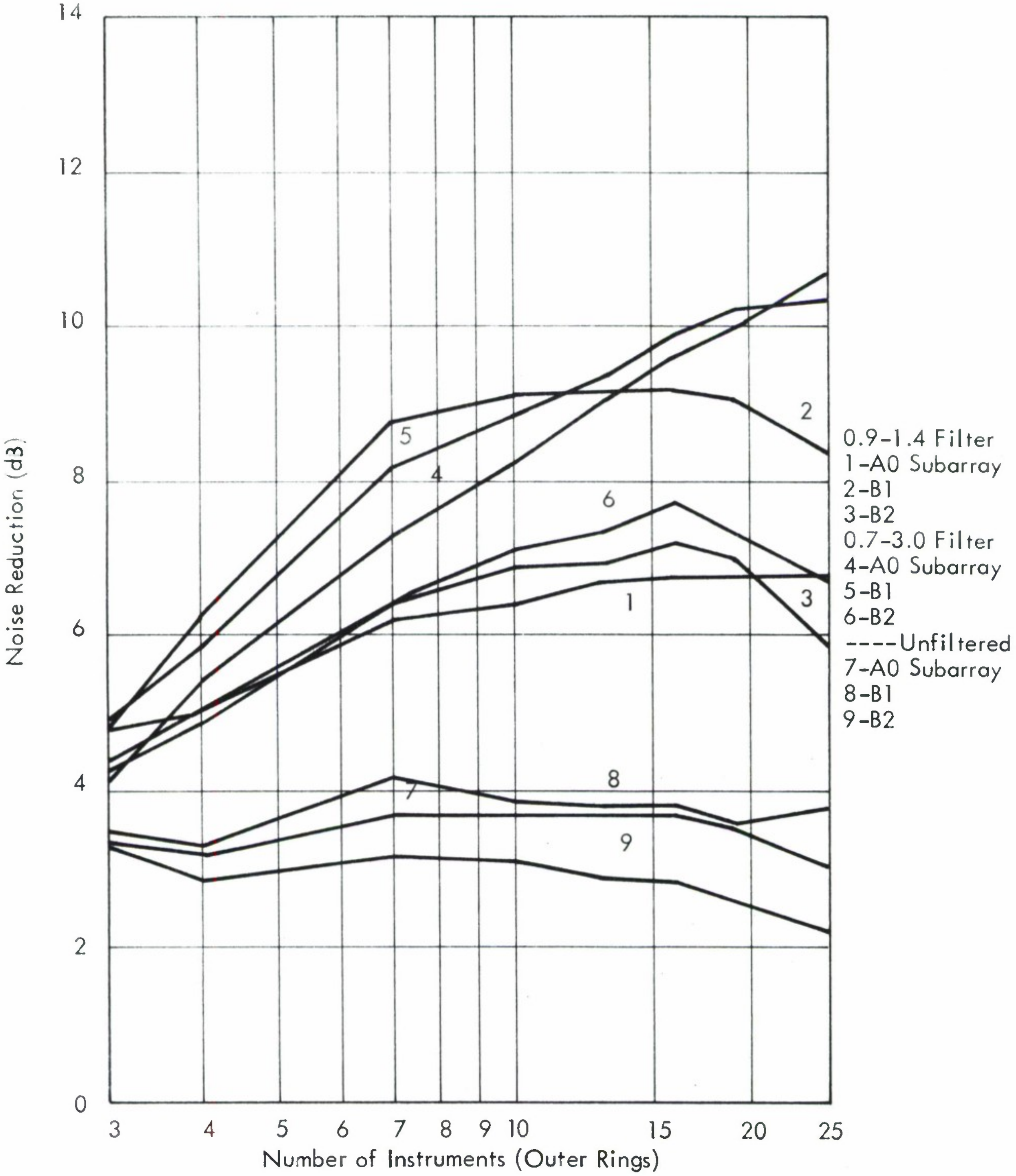


Figure 58. Observed Subarray Performance, Montana LASA I, 7 Km Subarray

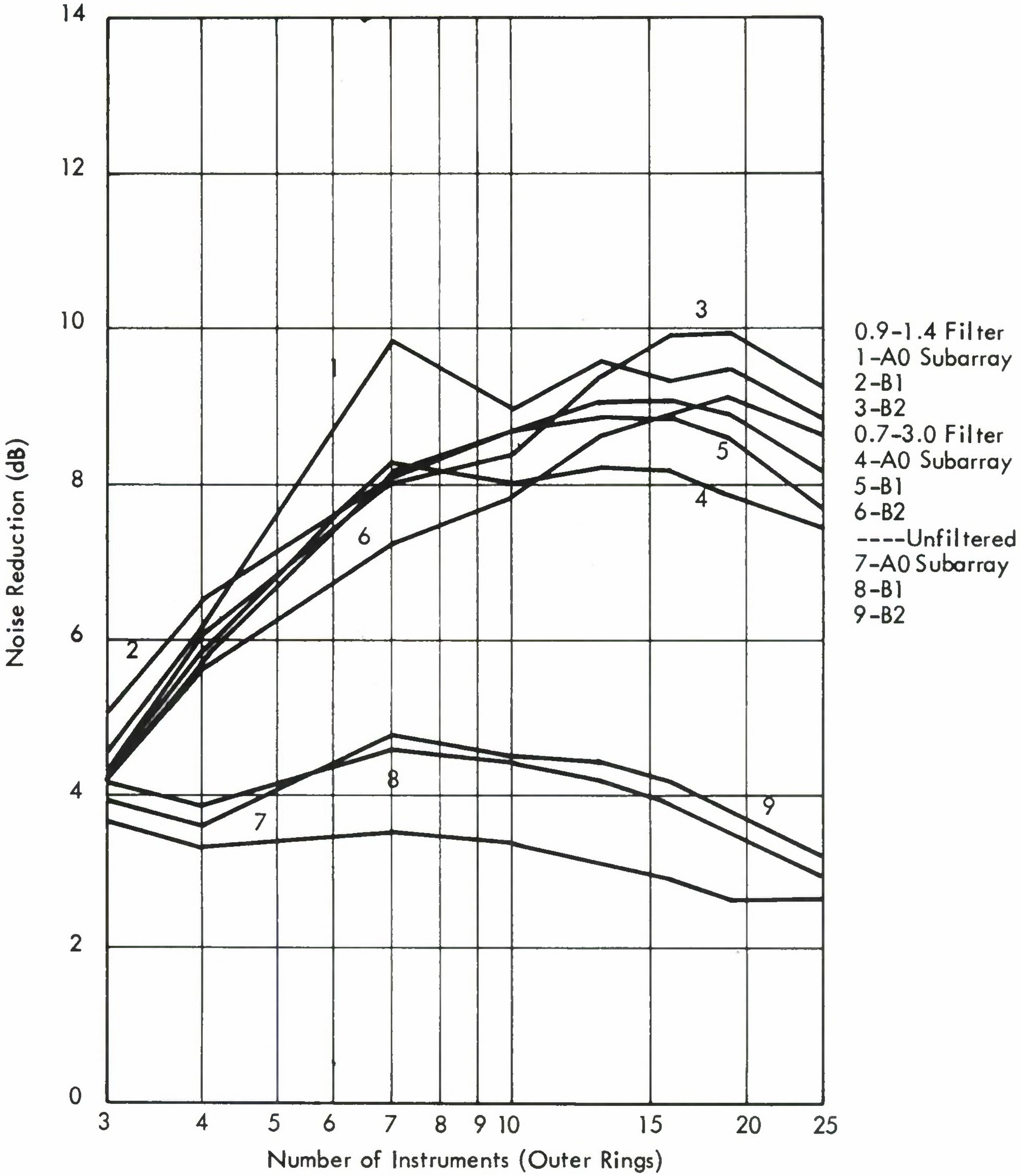


Figure 59. Observed Subarray Performance,
Montana LASA II, 7 Km Subarray

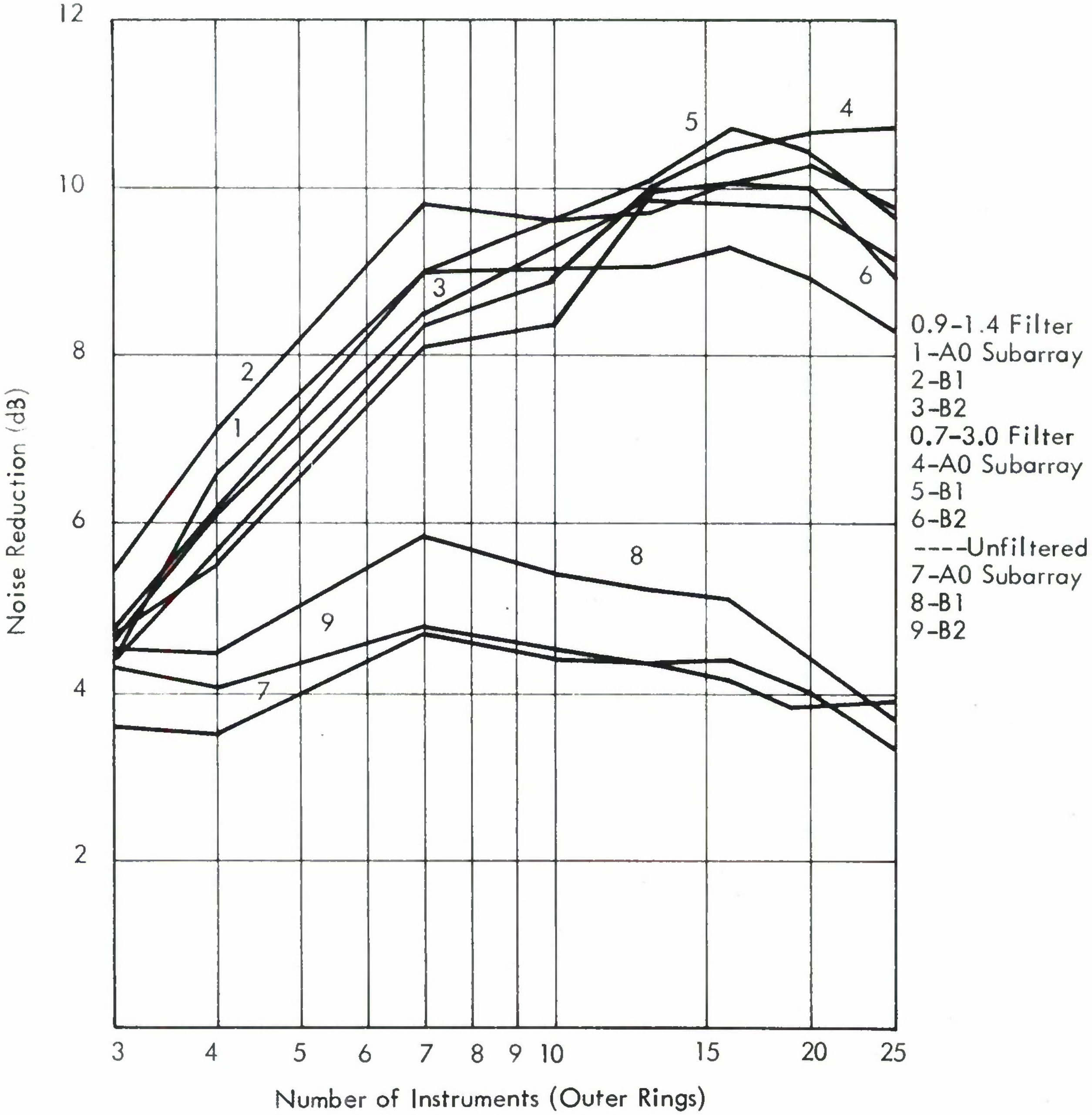


Figure 60. Observed Subarray Performance, Montana LASA III, 7 Km Subarray

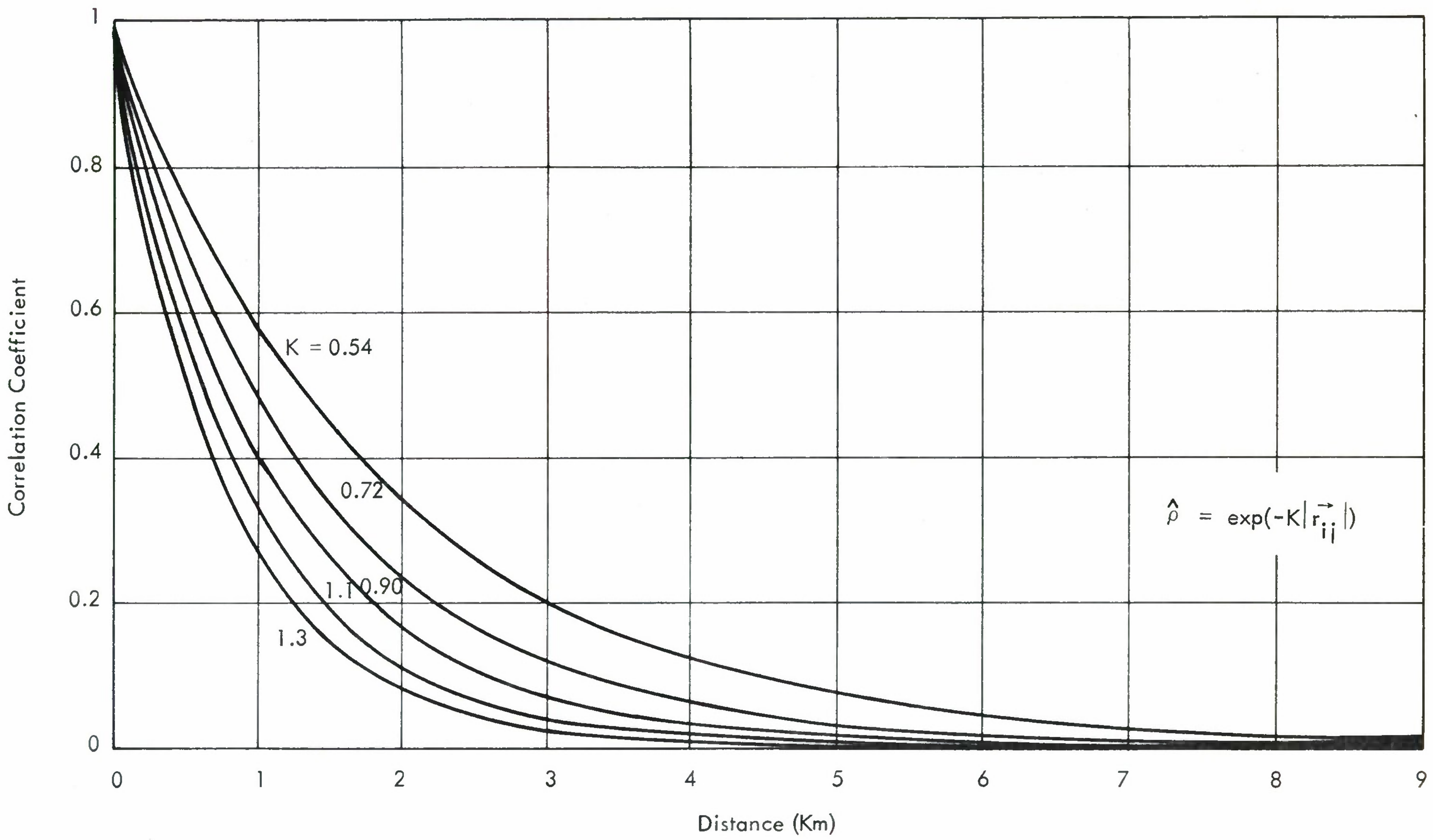


Figure 61. Noise Correlation Model

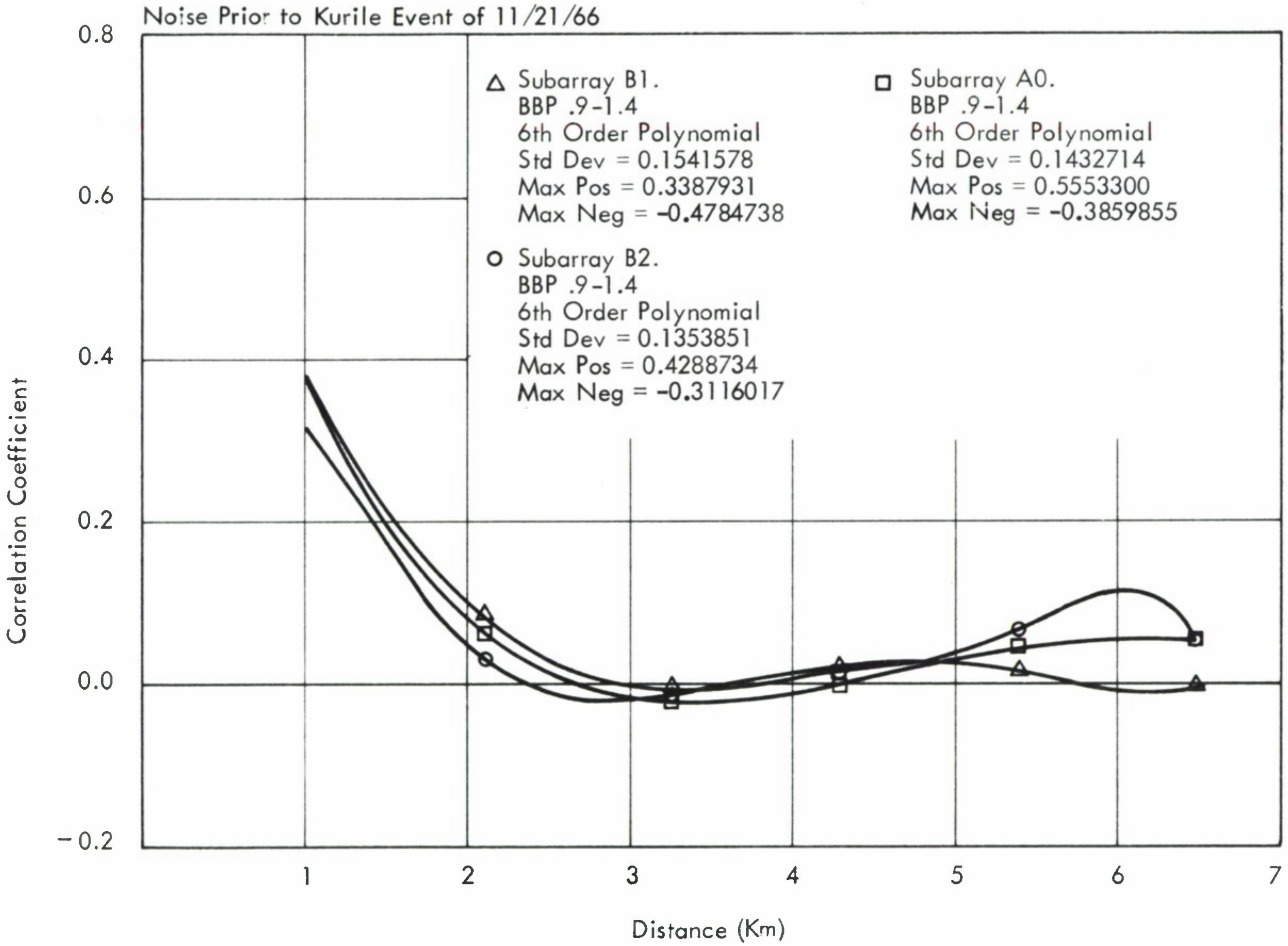


Figure 62. Correlation Coefficient vs Distance Between Instruments I

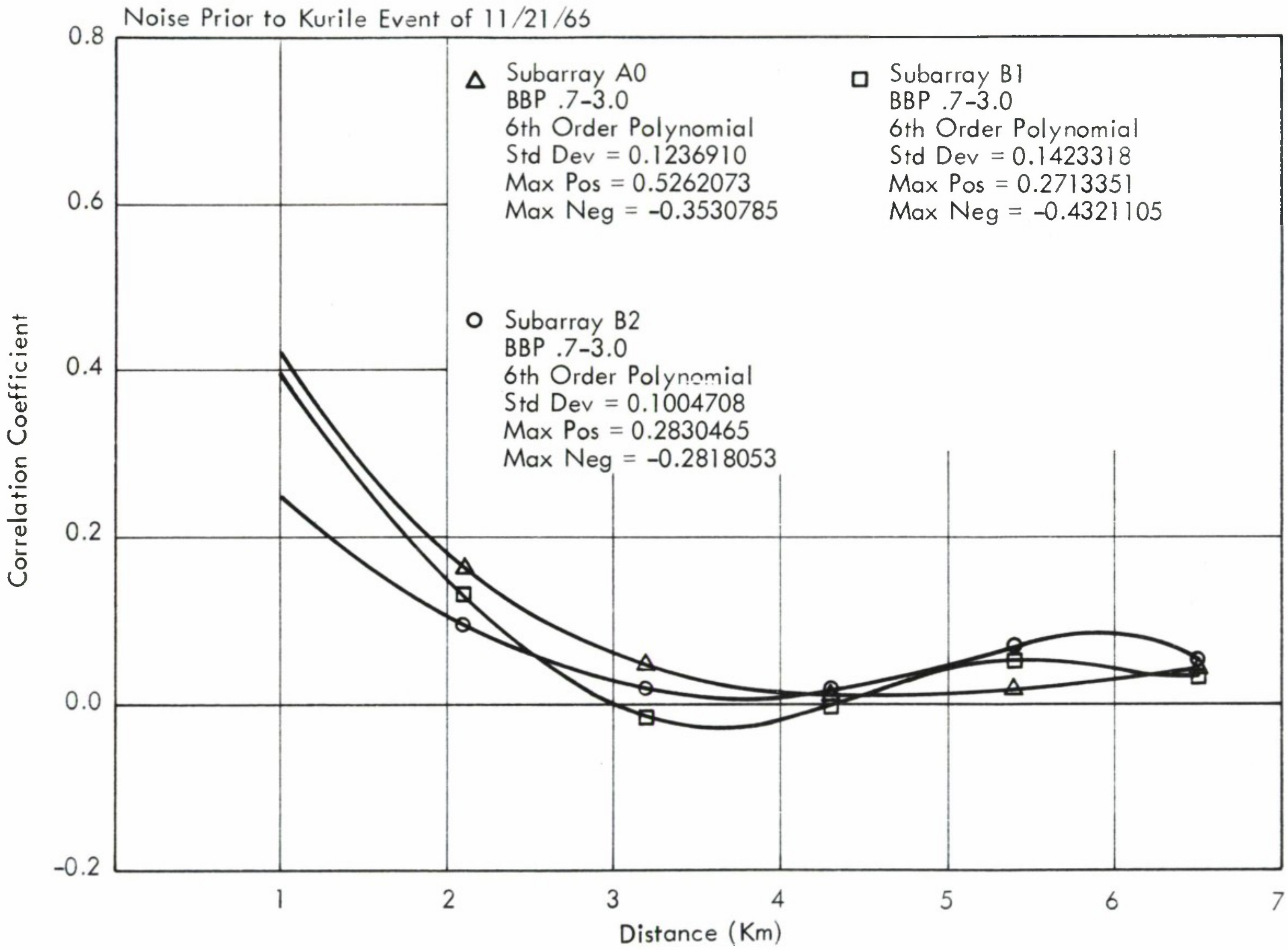


Figure 63. Correlation Coefficient vs Distance Between Instruments II

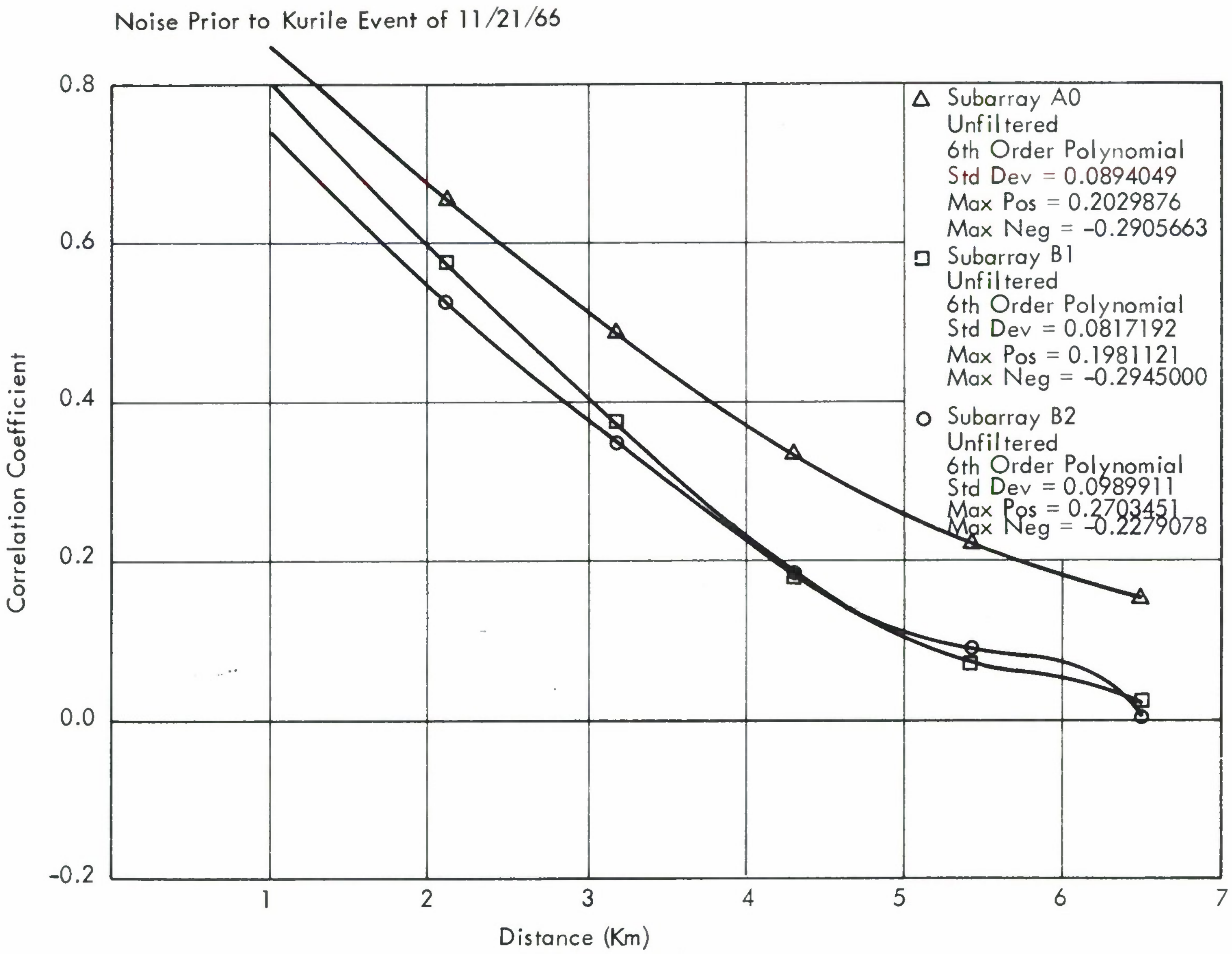


Figure 64. Correlation Coefficient vs Distance Between Instruments III

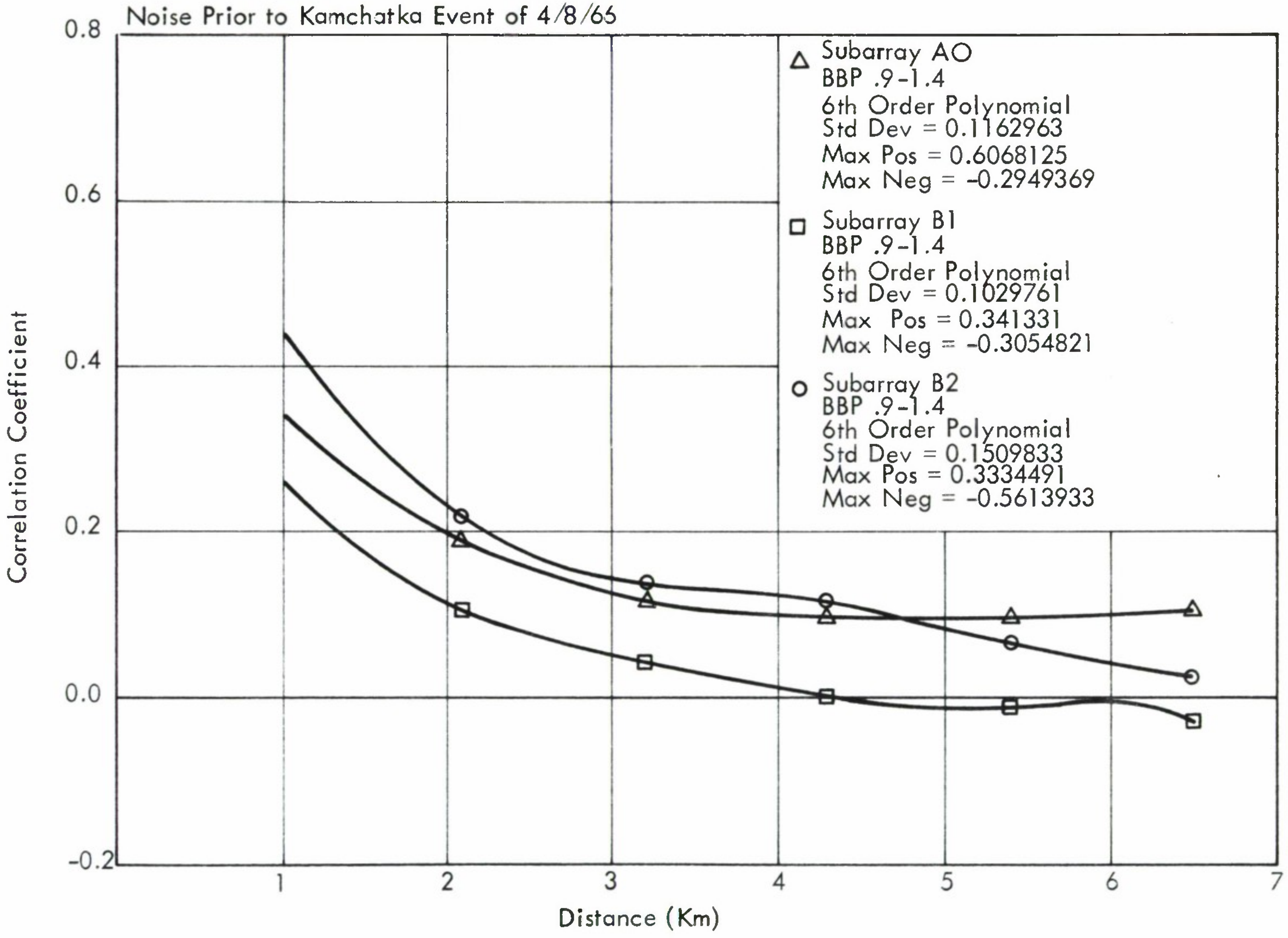


Figure 65. Correlation Coefficient vs Distance Between Instruments IV

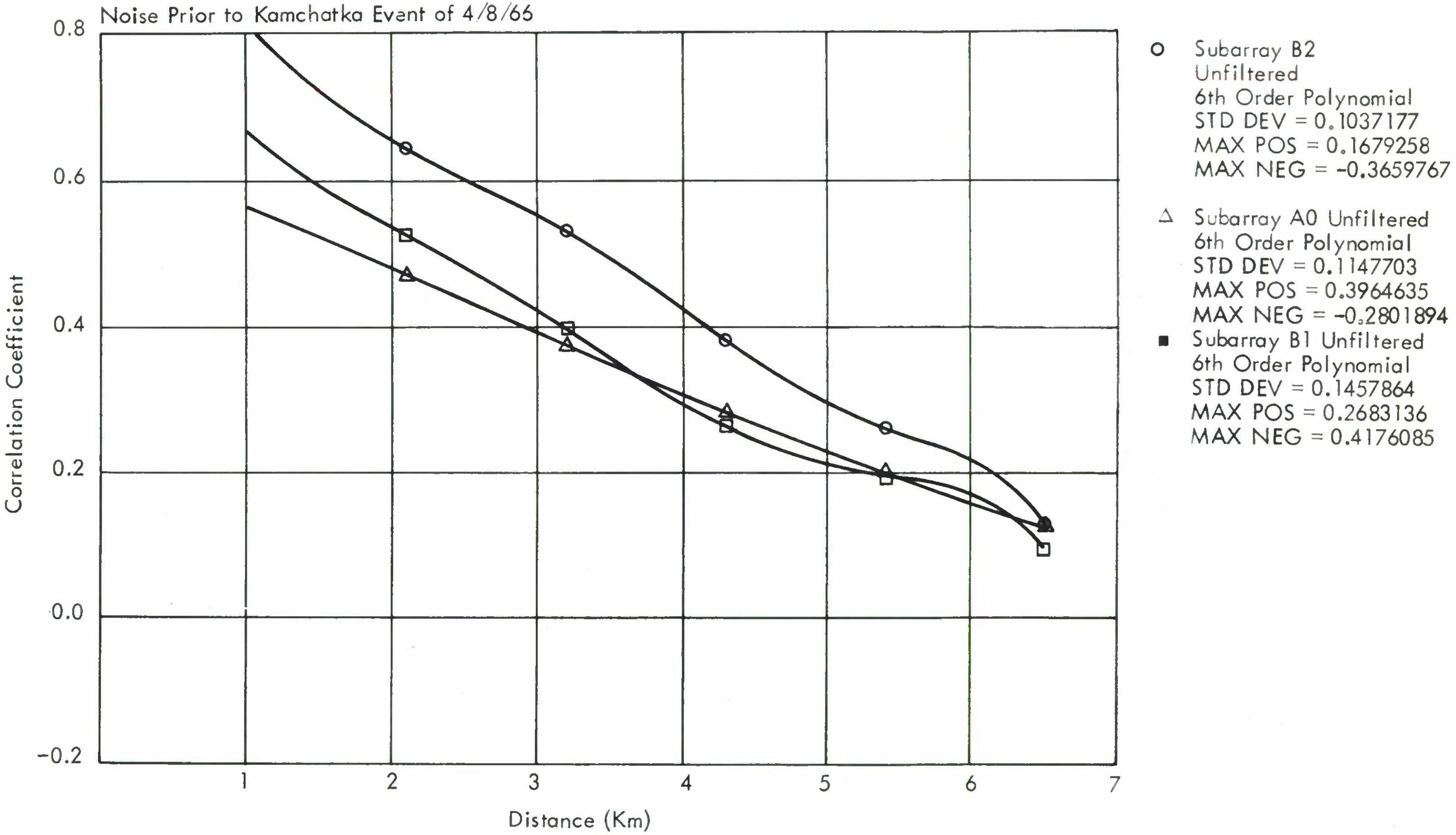


Figure 66. Correlation Coefficient vs Distance Between Instruments V

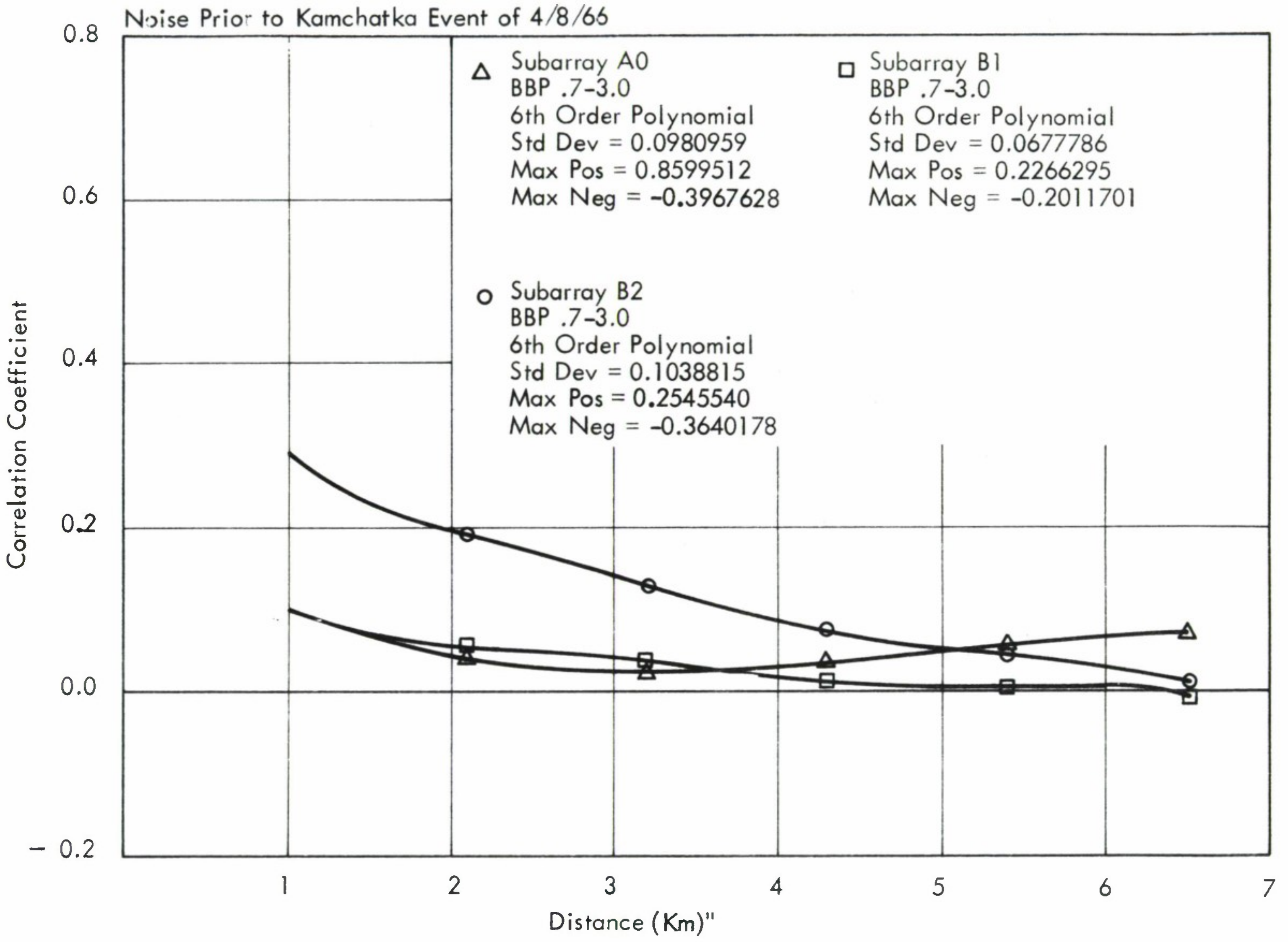


Figure 67. Correlation Coefficient vs Distance Between Instruments VI

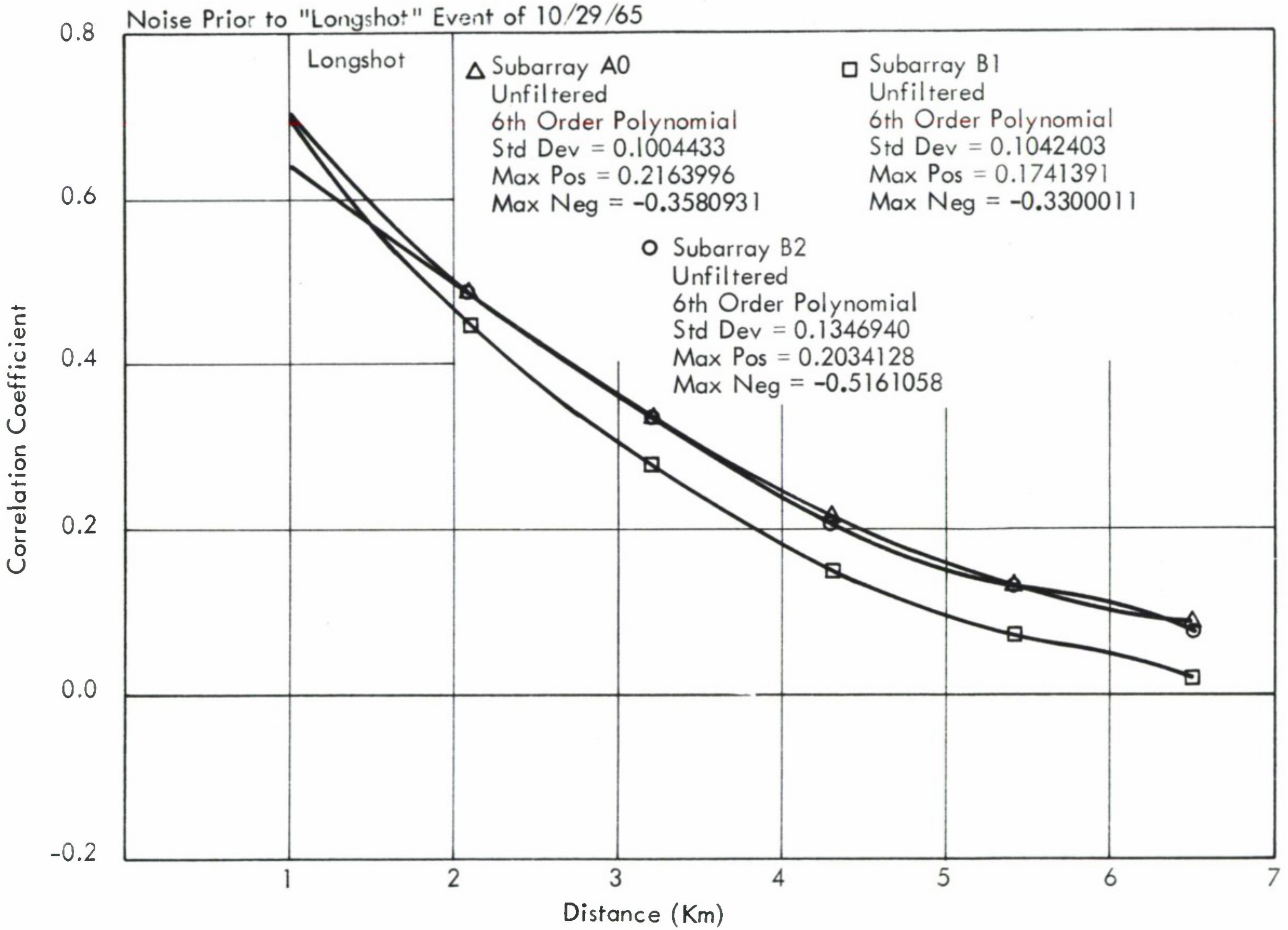


Figure 68. Correlation Coefficient vs Distance Between Instruments VII

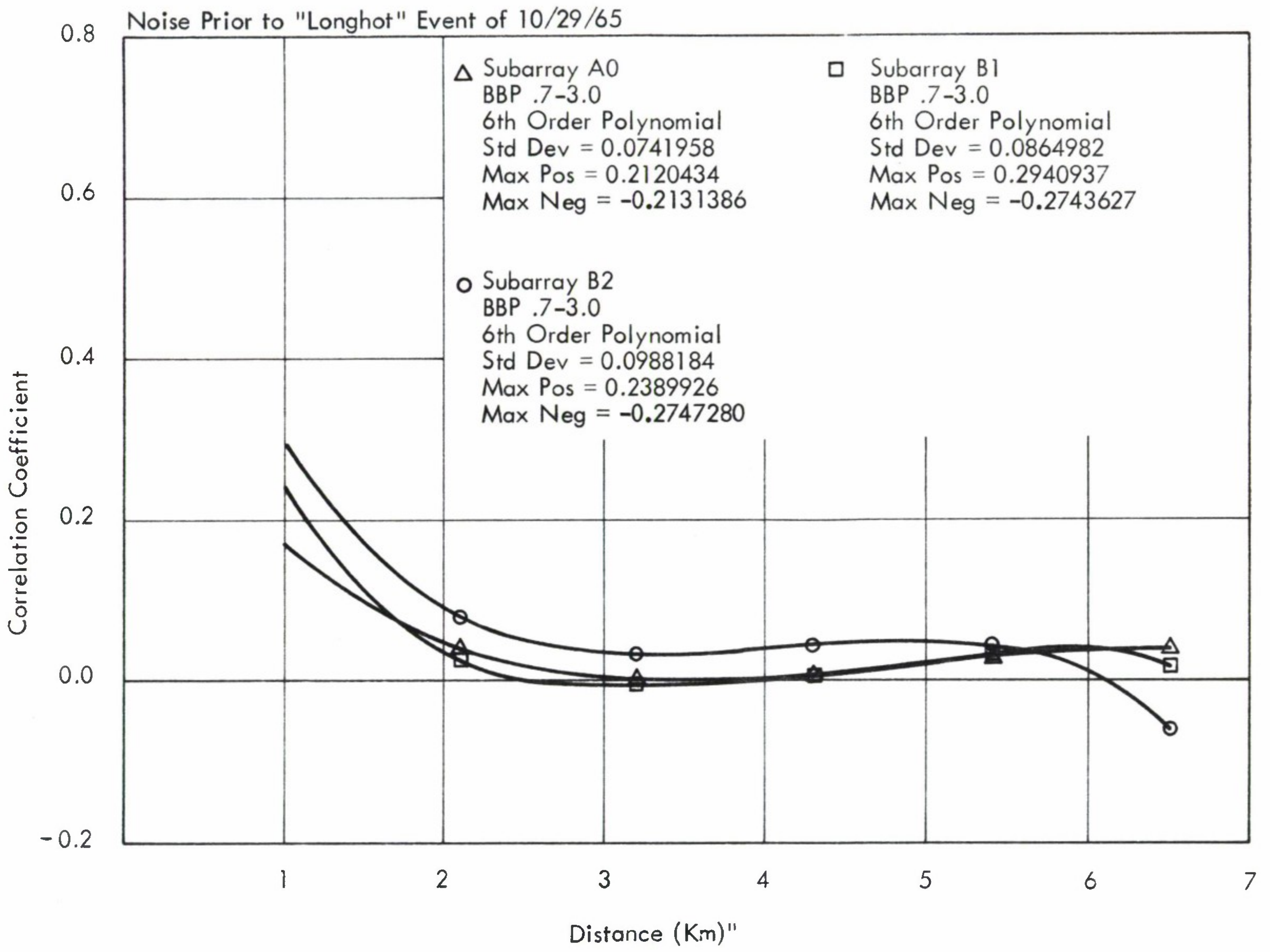


Figure 69. Correlation Coefficient vs Distance Between Instruments VIII

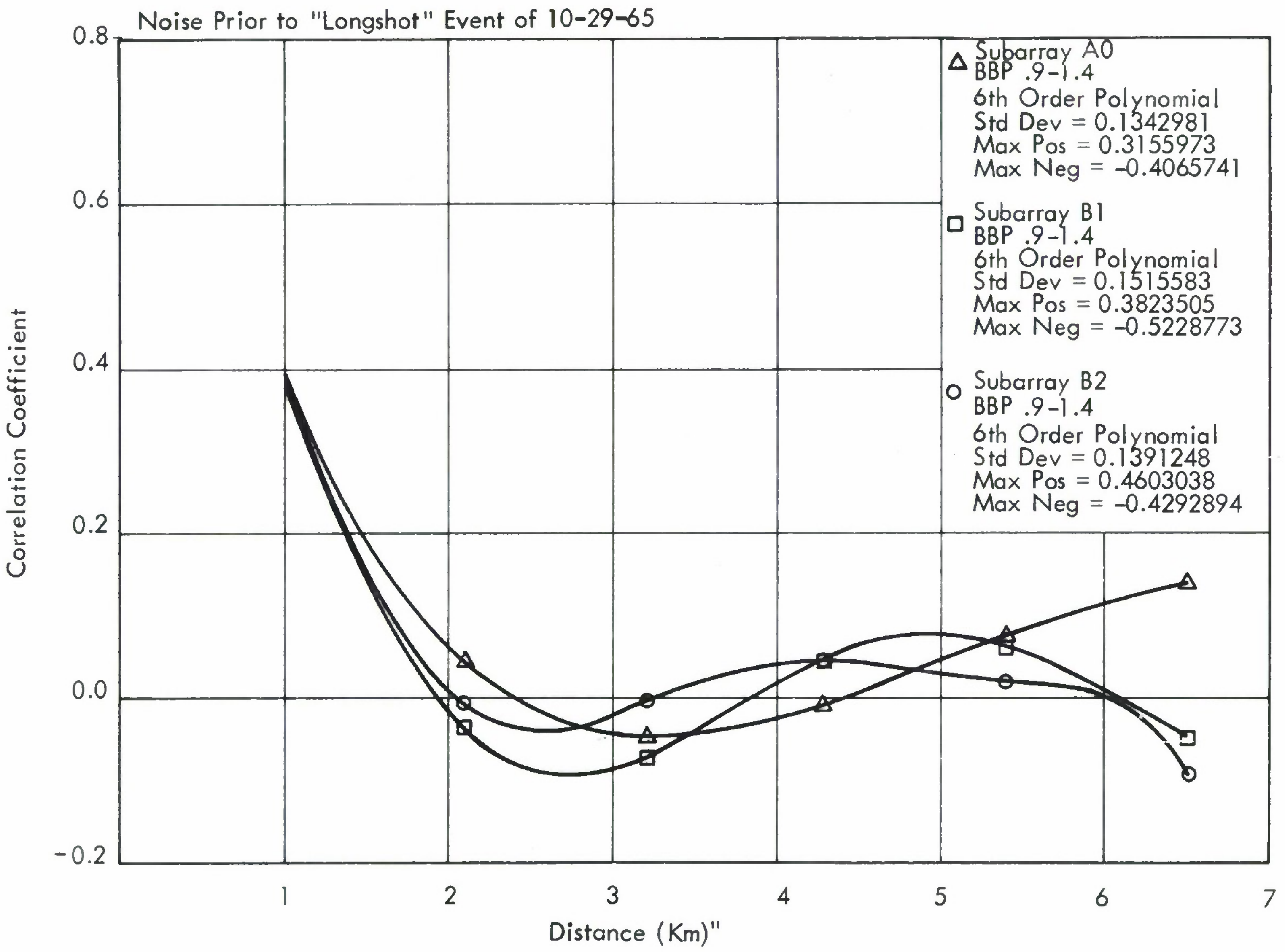


Figure 70. Correlation Coefficient vs Distance Between Instruments IX

Appendix III

PHASE DELAYS

Operation of the LASA beam former requires the use of 21 time delays—one for each subarray. Previous studies have demonstrated that plane wave steering of subarray beams must be augmented by correction factors to take account of travel time anomalies.

A key requirement for both Detection and Event Processing is the library organization of phase delays. The implications of using a library form of process structure are examined in this appendix. Procedures, based on correlating a LASA beam with each of the subarray beams used in forming it, have been developed for calculating improved steering delays, given an initial set. These procedures described in Appendix II-1 will be employed to maintain, correct, and enlarge the library of phase delay parameters. Regions and sectors in U-AZI-space are defined, and associated subarray correction factors are computed, reflecting the discrepancies between theoretical and actual travel times observed for a sample of several hundred events. A test performed to evaluate the effectiveness of the subarray correction factors in reducing mis-steering loss is also described.

The methodology to cover an arbitrary location, $\vec{U}$, in inverse velocity space by a judicious choice of preformed, presteered subarray beams is addressed in Section 3. Special attention is given to subarray beam assignment in order to minimize the expected loss.

III.1 PROCESS STRUCTURE

The Detection Processor forms LASA beams by summing in phase appropriate beams chosen from a set of preformed, pre-steered subarray beams. Since the delays required for steering each subarray are a function of the location of the LASA beam being formed, they will be obtained from a library of phase delays, organized by LASA beam coordinates. In Event Processing the library delays will be employed or the optimal delays will be obtained from correlations that can be generated for events of special interest.

III.1.1 Phase Delay Library Parameters

The entries for the phase delay library parameters will be a combination of several source factors among which are

a. Regular plane wave steering delays

b. Sets of subarray steering corrections each of which is valid over a specified region of U-AZI space

c. Inter-regional corrections calculated from the sets of subarray corrections

d. Optimal delays obtained as a result of an iterative process based on correlation methods.

A library structure demands that requirements be specified for establishing, maintaining, and updating library entries.

It is planned to employ plane wave delays for beam steering, particularly for steering to aseismic locations.

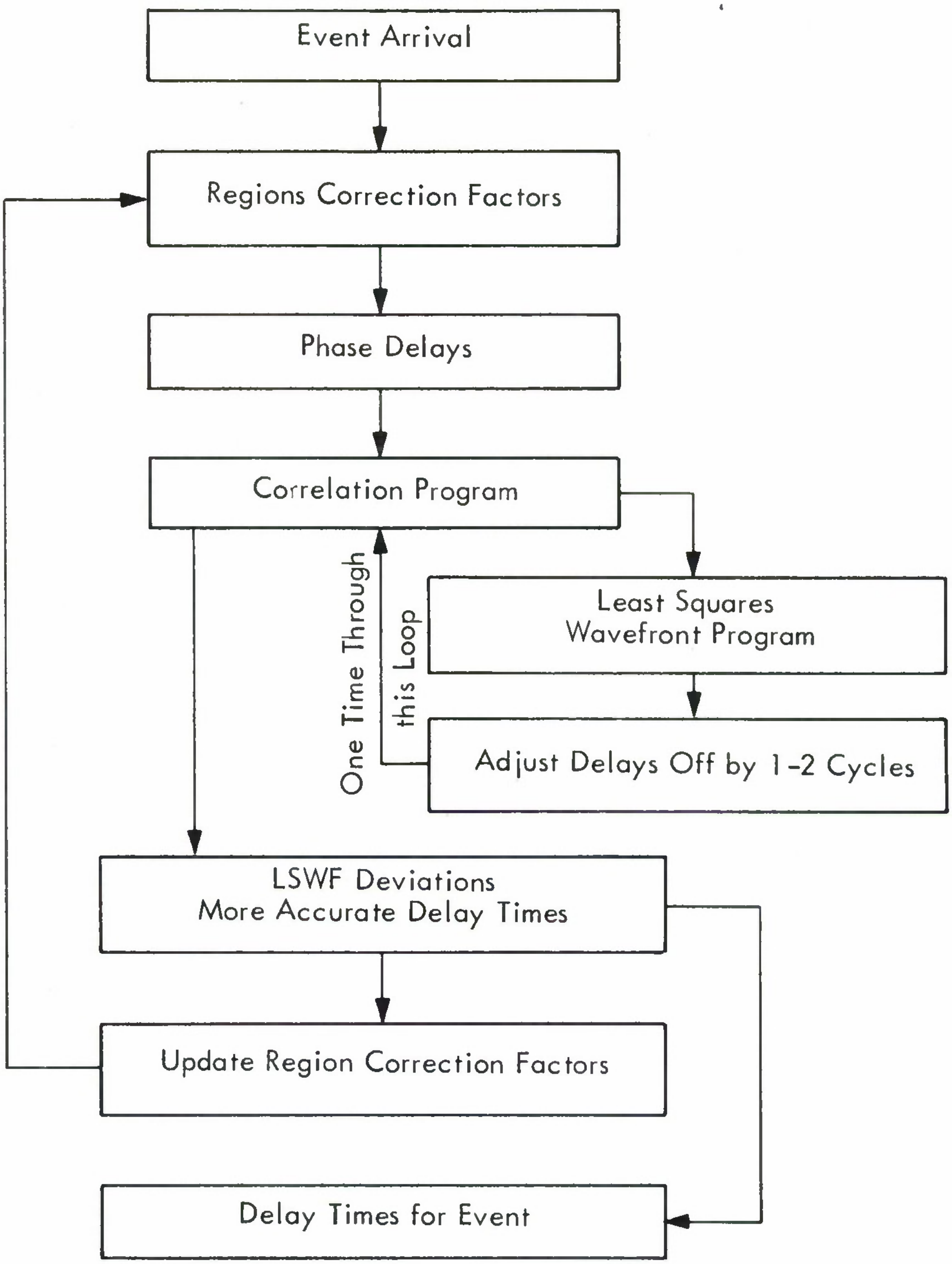
For seismic areas, it is planned to modify the plane wave delays by means of empirically derived subarray correction factors; each set is valid over a specified region in U-AZI space. The development of 35 sets of regional correction factors will be described in this appendix.

Formulas have been developed for interpolating between regions in order to obtain correction factors for events located outside regions. Plane wave delays will be replaced by optimal delays for certain regions and for weak events. These will be obtained by use of the correlation program described in Appendix II-1.

It will be routine practice to check the subarray regional corrections and the interpolation formulas on a regular basis. This will detect any systematic deviations which may reflect trends in the correction factors.

Analysis of strong events will be the chief instrument used to correct and enlarge library phase delays. Detection and identification of a strong event will be followed by the input of its delays to the correlation program, which will calculate optimal delays for the event. The optimal delays may be used to check existing region correction factors and associated interpolation formulas, or may serve as the initial set of correction factors for a new region.

Criteria must be established for determining when the correlation program is to be used for updating library entries, and for determining when an event may be considered so strong that its delays may be used to correct library parameters. The interplay between library parameters and the correlation program is shown in Figure 71. For routine events, appropriate phase delays will be obtained from



Event Arrival
Regions Correction Factors
Phase Delays
Correlation Program
One Time Through this Loop
Least Squares Wavefront Program
Adjust Delays Off by 1-2 Cycles
LSWF Deviations More Accurate Delay Times
Update Region Correction Factors
Delay Times for Event

Figure 71. Interaction between Phase Delays and the Correlation Program

the library. For special events, in particular, weak ones, the correlation program will be employed to calculate optimal delays, using as input either modified plane wave delays or delays for the beam on which the event was detected.

III.2 ANOMALY CHARACTERIZATION

III.2.1 Introduction

This section describes the development of steering correction factors resulting from an examination of several hundred events, each of which yielded a set of 21 time delay corrections. It is shown that these sets of correction factors can be grouped to produce average sets of corrections which are valid over judiciously selected geographic regions. Because of the irregular space and time distribution of earthquakes, these regions do not cover the entire earth. Consequently, it is necessary to interpolate to obtain correction factors for beams that are being steered to areas located between the nominally selected regions. The development of such interpolation formulas is described in this appendix.

III.2.2 Method of Correction

Theoretical steering delays must be amended by adding some correction factor. This is currently being accomplished by defining for each event region and receiver site a travel time anomaly. This anomaly is the difference between the actual travel time and a theoretical travel time, based on a worldwide travel time average for events at a given range, such as those found in the Jeffreys Bullen Seismological Tables.[15] The required event range can be obtained for each event from an event location based on the world seismic network. Time of origin is also determined by this network. The average anomaly for each receiver for a given geographic region is then calculated from a number of events occurring in that region, and is used to provide a correction to the theoretical travel time tables to give the required beam steering delays. The LASA system steering error for a given event would depend on how well the event arrival time differences fit the region averages. An interesting and informative study of time anomalies using this method on the LASA array has been completed by E. F. Chiburis.[16]

Another possible method of organizing the time delays is to establish a best fitting plane or second order wavefront for each event, and to use as anomalies the variabilities of the deviations from this best fitting wavefront. As in the currently used method, the delays are based on the anomaly averages for a given region.

A program to implement the wavefront method has been written. This program, called the Least Squares Plane and Quadratic Wavefront Program, accepts as inputs the positions, arrival times, and computation weights of up to 25 seismometers (or subarray beams), and calculates three different least squares wavefronts with the actual arrival time deviations from the best fits. The three fits are

a. A best-fit plane wavefront

b. A best-fit plane wavefront followed by an adjustment of arrival times to consider the expected effect of range upon wavefront curvature, with the above process repeated for any desired number of iterations

c. A best-fit quadratic wavefront plus the three derived second derivatives or curvatures, expressed in array coordinates, range coordinates, and the principal coordinates of the quadratic surface.

The input data for the wavefront program consisted of optically read arrival time data, made available by Seismic Data Laboratories (SDL) of Teledyne Industries, Inc., and the VELA Seismological Center (Air Force Technical Application Center), Alexandria, Virginia. A total of approximately four hundred event arrivals were made available in their original form by SDL.[17]

III.2.3 Procedure for Obtaining Steering Delay Corrections

Chiburis describes the grouping of several hundred events into 36 geographic regions. Table 8, which uses the regions so defined together with later data from SDL, lists the number of events in each region. As described in Reference 16, a region is formed by grouping teleseisms whose subarray travel time anomalies are consistent within the region. The 377 events and the 36 regions form the starting point of the development reported herein.

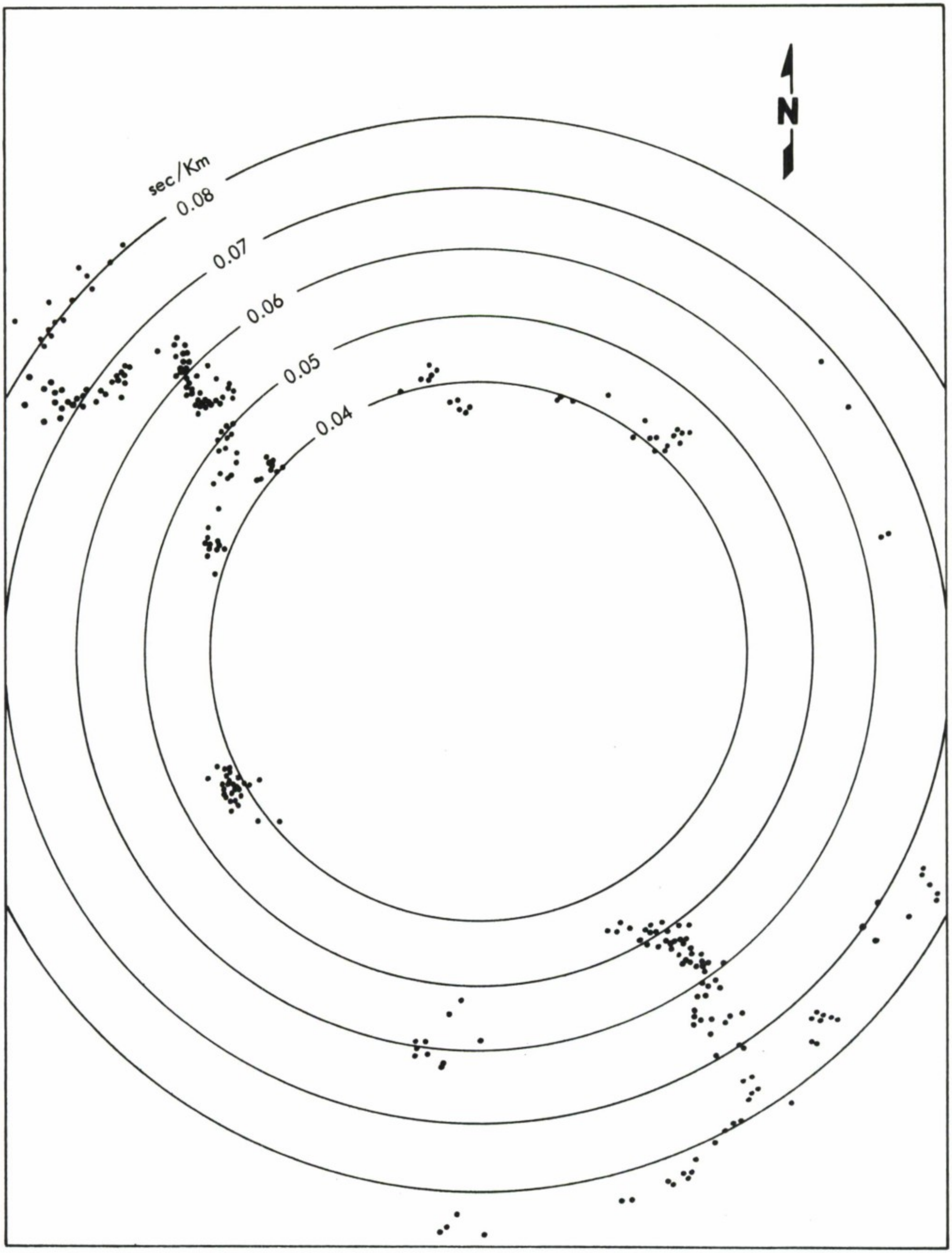
The 377 events have been processed through the Least Squares Wavefront Program and the resulting computed values of azimuth (AZI) and inverse phase speed (U) have been plotted for each event as shown in Figure 72. The other output of the wavefront program—the deviation in seconds of the actual arrival from the best fitting plane wavefront—serves as input to the Seismic Steering Delay Anomalies Program. Given a group of events, each with its set of 21 correction factors or deviations (one for each subarray), this program calculates the average deviation at each subarray. The difference between each event's deviation and the average deviation of the group at each subarray is computed, yielding a set of 21 differences, i.e., deviations of deviations. The average and standard deviation of this set of differences and the mis-steering loss in dB for the event's wavefront are calculated.

The division of the 377 events into 36 regions[16] furnished the initial groups used in the Steering Delay Anomalies Program. The program results indicated that some of the events yielded excessive dB losses and it was decided to regroup

Table 8

377 EVENTS GROUPED INTO 36 REGIONS

Region No.	Geographic Area	No. of Events In Region	Region No.	Geographic Area	No. of Events in Region
1	Central-So. Alaska	6	19	N. Atlantic	2
2	Kodiak Island	11	20	Azores	2
3	Alaska Pen.	3	21	No. West Indies	5
4	Unimak	4	22	Venezuela-So. West Indies	3
5	Fox-Andreanof	19	23	Eastern Mexico	9
6	Andreanof	4	24	N. Centr. Ameria	8
7	Andr-Rat	22	25	S. Centr. America	
8	Near Island	14	26	N. Columbia	12
9	Kamchatka	12	27	Peru-Brazil	8
10	Kurile	29	28	Peru-Bolivia	13
11	Sakhalin-Hokkaido	8	29	N. Chile-Bolivia	16
12	Honshu Marianas	20	30	N. Chile-Argentina	12
13	Marianas-Carolines	13	31	C. Chile-Argentina	13
14	Taiwan-Ryuku	10	32	West Mexico	4
15	Kazakh	9	33	Easter Island	3
16	Hindu Kush	7	34	Easter Island	7
17	East Caucasus	3	35	Samoa-Tonga	22
18	Greece-Turkey	14	36	Fiji Islands	23



Figgre 72. 377 Wavefronts, Calculated U and AZI as Observed at LASA

the events into new regions. There are 35 regions in the new grouping. They are plotted in Figure 73 and their coordinates are listed in Table 9. The revised regions are quite similar to the original 36. For example, revised regions 109 through 123 correspond exactly to original regions 12 through 26 respectively.

After regrouping the 377 events into 35 new regions, averages were again formed and the anomalies program was used to check the assignment of the events to the regions and to assign weights to the events. For those events which continued to show a loss of more than 1 dB at 1 Hz, a weight of zero was assigned while the others were weighted one. New averages were formed for each region, using only those events weighted one, i.e., using those events whose loss was less than 1 dB at 1 Hz. The anomalies program was used to provide a final check. This time there were no further changes—the events weighted one continued to show a loss of less than 1 dB at 1 Hz while those weighted zero showed even greater losses (as was to be expected since they had been excluded from the average forming proocess the second time). Thus, the phase delay averages for the revised regions are arrived at by a two step iterative process, based on stable, similar events each of which experiences a loss of less than 1 dB at 1 Hz due to steering to these regional averages. Table 10 lists the number of events with weight one for each region. As shown in the table, there is a total of 335 weight one events out of a total of 377 events and only 42 events were weighted zero.

The Anomalies Program yields the average loss (in dB) per region, i.e., the standard deviations of the events in a region are averaged, and converted to a loss.

For an event occurring in a region, this average loss figure represents the loss in dB which results from using the regional averages as steering correction factors rather than a set of corrections that would be optimum for the particular event. Table 11 lists the average loss per region for the 35 revised regions. As can be seen from the table, the average loss is less than 1 dB at 1 Hz for every region. These results indicate that the regional averages can be instrumental in providing correction factors for beams deployed within regions.

III.2.4 Definition of Sectors

To obtain correction factors for areas having no history of teleseismic activity, some form of interpolation between regions must be developed. Interpolation between regions is based on the assumption that correction factors are functions only of azimuth or inverse phase speed or both. This implication or assumption must be considered somewhat questionable, for examination of the regional corrections reveals no discernible pattern in the average deviations. They appear not to be simple functions of azimuth or of range, but they fluctuate in an erratic manner. Figures 74, 75, and 76 illustrate the deviations per region for three specific subarrays.

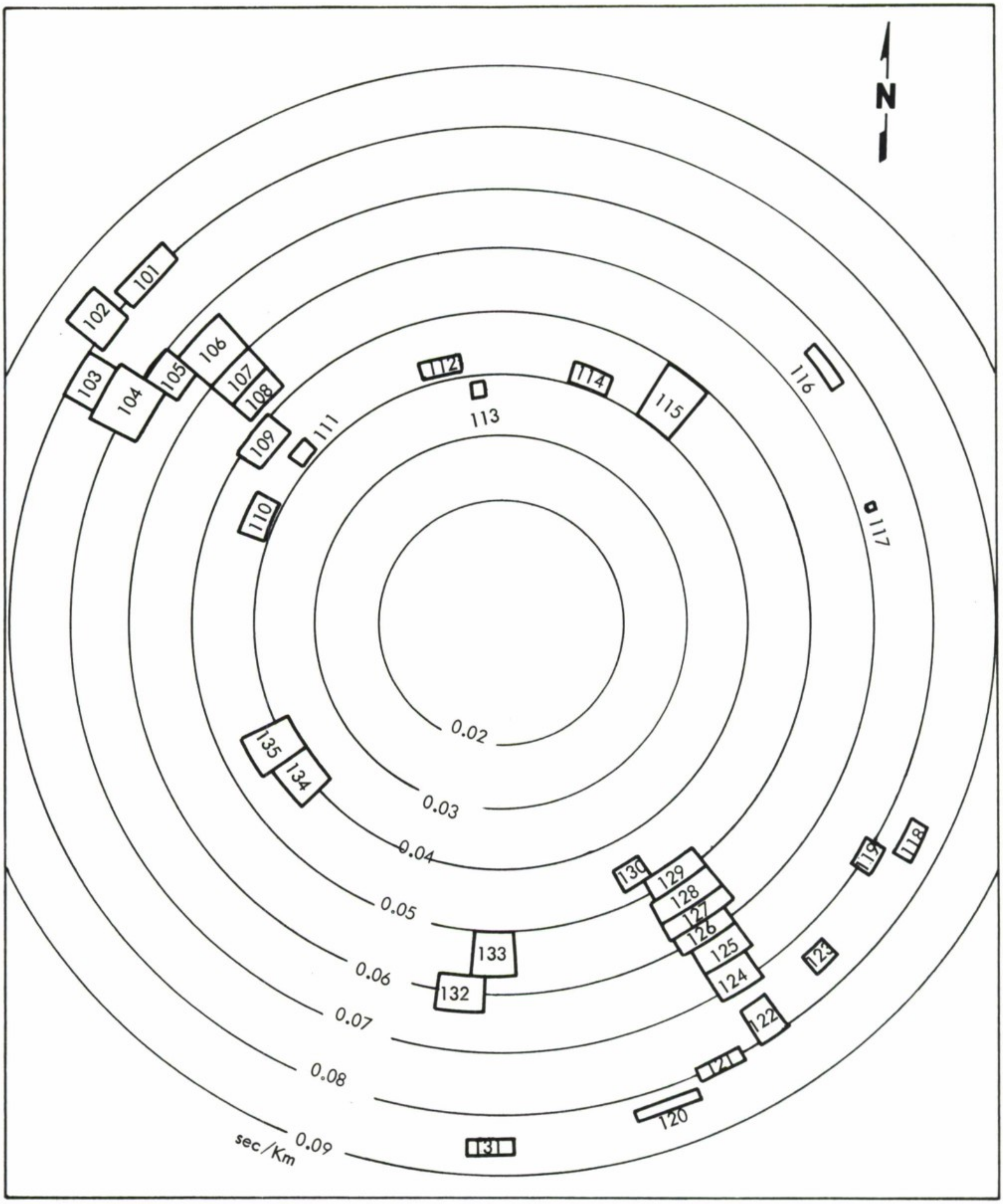


Figure 73. Revised Regions

Table 9

REGIONS

Region Number	Range Window (Sec/Km)	Geographic Azimuth Window (degrees)	Region Number	Range Window (Sec/Km)	Geographic Azimuth Window (degrees)
101	$0.079 < U < 0.083$	$311 < \theta < 319$	119	$0.069 < U < 0.073$	$121 < \theta < 125$
102	$0.078 < U < 0.086$	$305 < \theta < 310$	120	$0.083 < U < 0.084$	$157 < \theta < 165$
103	$0.075 < U < 0.080$	$298 < \theta < 304$	121	$0.079 < U < 0.082$	$151 < \theta < 157$
104	$0.066 < U < 0.075$	$297 < \theta < 305$	122	$0.074 < U < 0.081$	$145 < \theta < 149$
105	$0.064 < U < 0.070$	$305 < \theta < 310$	123	$0.073 < U < 0.077$	$135 < \theta < 139$
106	$0.060 < U < 0.068$	$310 < \theta < 318$	124	$0.067 < U < 0.072$	$144 < \theta < 150$
107	$0.056 < U < 0.060$	$310 < \theta < 318$	125	$0.062 < U < 0.067$	$142.5 < \theta < 150$
108	$0.052 < U < 0.056$	$310 < \theta < 318$	126	$0.059 < U < 0.062$	$142.5 < \theta < 152$
109	$0.046 < U < 0.051$	$303 < \theta < 313$	127	$0.056 < U < 0.059$	$140.5 < \theta < 152.5$
110	$0.041 < U < 0.045$	$290 < \theta < 299$	128	$0.052 < U < 0.056$	$140.5 < \theta < 152.5$
111	$0.041 < U < 0.044$	$309 < \theta < 314$	129	$0.048 < U < 0.052$	$140.5 < \theta < 151.5$
112	$0.041 < U < 0.044$	$344 < \theta < 352$	130	$0.044 < U < 0.048$	$150 < \theta < 155.5$
113	$0.037 < U < 0.039$	$354 < \theta < 357$	131	$0.084 < U < 0.087$	$179 < \theta < 184$
114	$0.040 < U < 0.044$	$17 < \theta < 26$	132	$0.0575 < U < 0.064$	$183 < \theta < 190$
115	$0.040 < U < 0.050$	$34 < \theta < 43$	133	$0.050 < U < 0.0575$	$178 < \theta < 185$
116	$0.066 < U < 0.068$	$49 < \theta < 56$	134	$0.037 < U < 0.044$	$228 < \theta < 238$
117	$0.063 < U < 0.064$	$73 < \theta < 74$	135	$0.037 < U < 0.046$	$238 < \theta < 247$
118	$0.074 < U < 0.077$	$116 < \theta < 121$			

Table 10

NUMBER OF CONTRIBUTING EVENTS

Region No.	No. of Events	Region No.	No. of Events	Region No.	No. of Events
101	6	113	5	125	8
102	10	114	2	126	6
103	7	115	12	127	8
104	37	116	2	128	13
105	13	117	2	129	10
106	16	118	5	130	7
107	13	119	3	131	4
108	16	120	9	132	5
109	19	121	8	133	2
110	11	122	7	134	4
111	8	123	10	135	34
112	9	124	4		

Table 11

AVERAGE LOSS PER REGION

Region	No. of Events Weight One/Total	Average Loss (in dB)
101	6/6	0.44
102	10/13	0.55
103	7/7	0.48
104	37/41	0.45
105	13/16	0.34
106	16/17	0.39
107	13/14	0.33
108	16/18	0.25
109	19/20	0.36
110	11/13	0.34
111	8/10	0.31
112	9/9	0.39
113	5/7	0.49
114	2/3	0.43
115	12/14	0.45
116	2/2	0.15
117	2/2	0.15
118	5/5	0.36
119	3/3	0.77
120	9/9	0.53

Table 11

(concluded)

Region	No. of Events Weight One/Total	Average Loss (in dB)
121	8/8	0.45
122	7/7	0.27
123	10/12	0.38
124	4/4	0.36
125	8/9	0.41
126	6/7	0.36
127	8/8	0.32
128	13/14	0.39
129	10/13	0.44
130	7/7	0.41
131	4/4	0.21
132	5/7	0.54
133	2/3	0.16
134	4/5	0.41
135	34/40	0.36
Total	335/337	

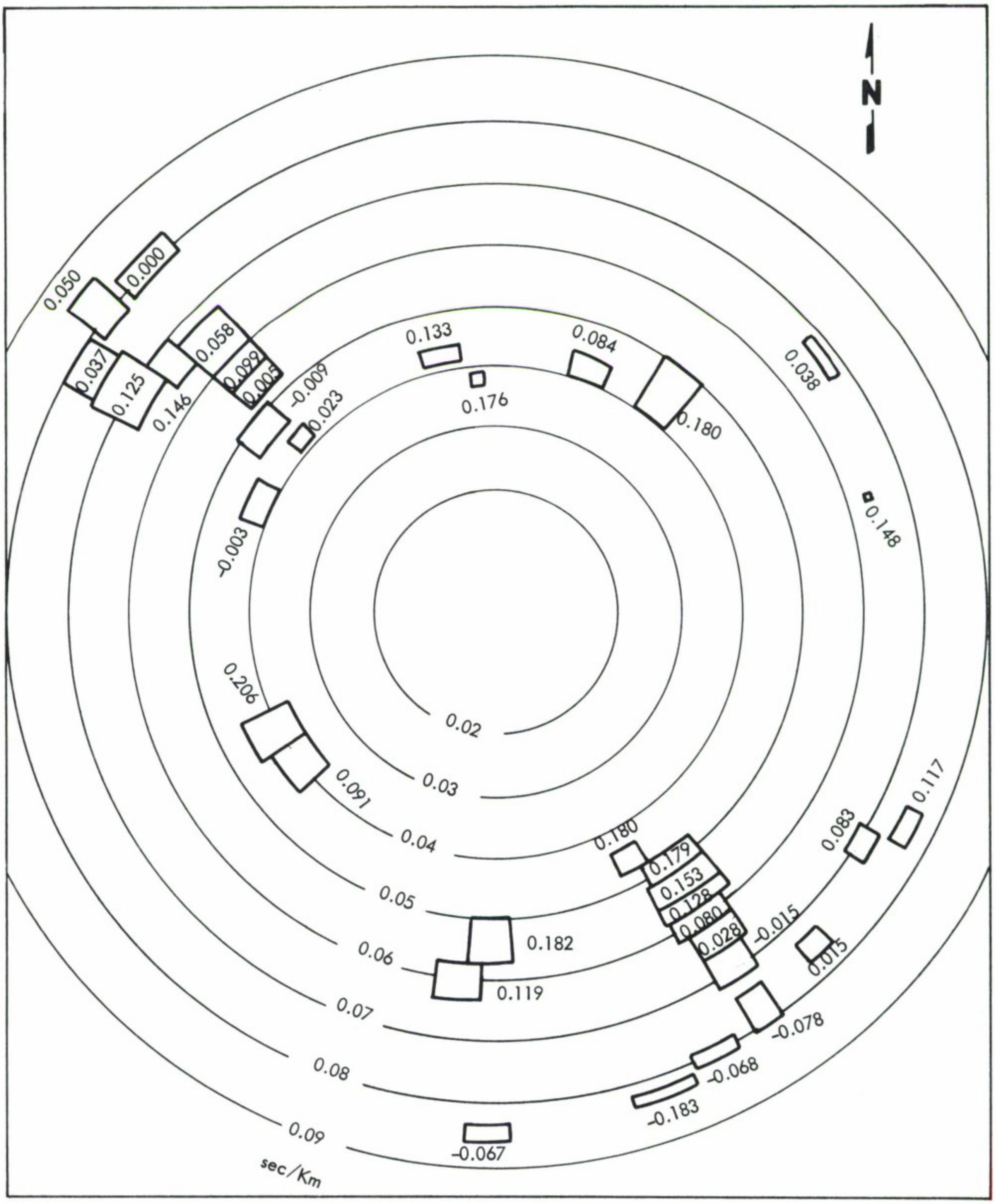


Figure 74. Regional Deviations (seconds) for Subarray C1

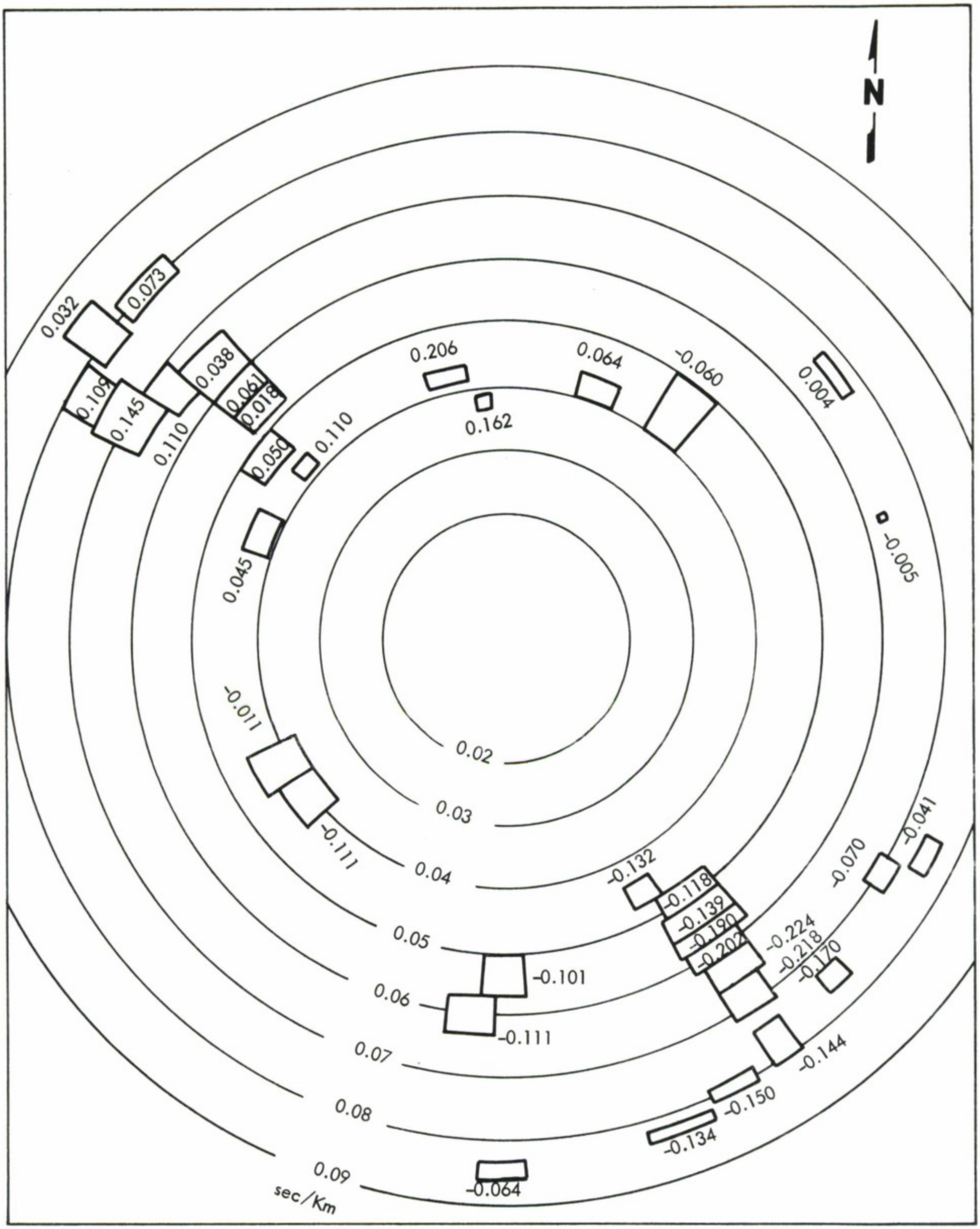


Figure 75. Regional Deviations (seconds) for Subarray C2

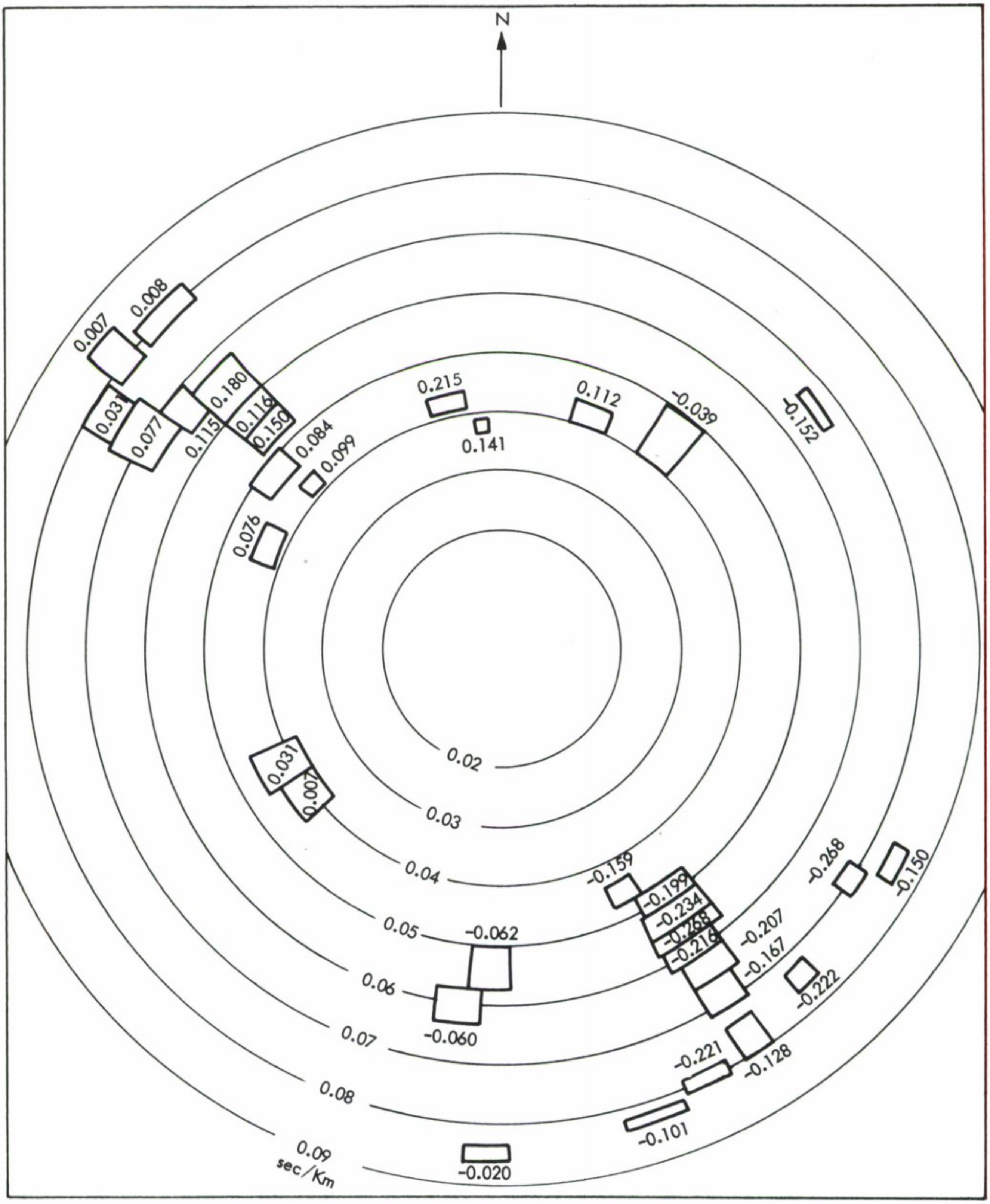


Figure 76. Regional Deviations (seconds) for Subarray C3

An interpolation procedure has been developed in an attempt to estimate correction factors for areas near regions. The interpolation method is based on the construction of areas called sectors which are essentially defined in terms of regions. A sector is an area containing two or more regions. The guideline used in establishing sectors has been to define their boundaries so that a sector is essentially constant either for azimuth or inverse phase speed. Figure 77 depicts the nine sectors that have been defined.

Table 12 gives the U-AZI coordinates for each sector and lists the regions contained in each sector. As shown in Figure 78 sectors do not extend significantly beyond the regions they contain.

III.2.5 Methods of Interpolation and Extrapolation

The interpolatory method used to obtain a deviation for each subarray and sector is simple linear interpolation in which either azimuth or speed is the independent variable. For example, if an event occurs in sector A, the deviation to be used for steering to that event is obtained by interpolating in azimuth between regions 114 and 115. The linear interpolation formula is stated in Table 13. Table 14 gives the (X, Y) values relative to subarray B1 for each region in each sector. Here, Y is the value of the deviation while X is either an azimuth value or a U value, depending on whether the independent variable is azimuth or inverse phase speed. The tables for the other subarrays are similar and therefore not presented.

Two or more regions are sometimes combined and treated as a single region. In particular, regions 122 and 124, which are contained in Sector C, are assigned the common azimuth value of 147° and the common deviation of 0.017 seconds; this pair of values is used for all 21 subarrays.

The interpolation procedure will be demonstrated by computing the correction factor for subarray B1 steered to event V whose location is estimated at U = 0.080 and AZI = 130°. Figure 78 shows that V is in sector C but not within any region in C. Table 14 shows that deviation is a function of azimuth in sector C; with respect to that variable, V is located between regions 119 and 123. Employing the notation of Table 13, let (X_1, Y_1) represent the coordinates of region 119 and (X_2, Y_2) represent the coordinates of region 123.

Thus, $(X_1, Y_1) = (123^{\circ}, -0.062)$,

$(X_2, Y_2) = (137^{\circ}, -0.188)$

and $(X, Y) = (130^{\circ}, Y)$, since AZI is the independent variable in this case. The deviation Y is computed as follows:

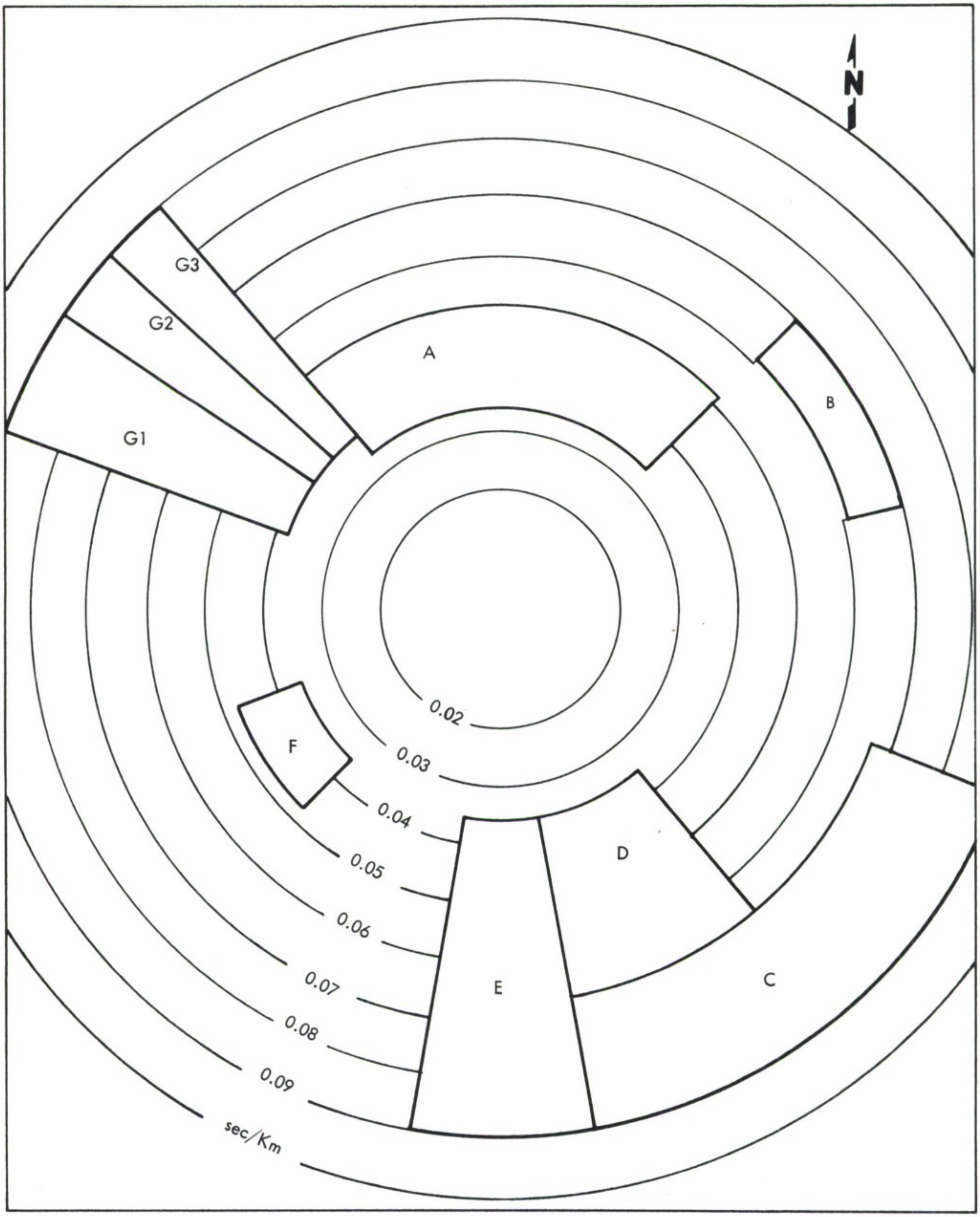


Figure 77. Sectors

Table 12

SECTORS

Sector	Range Window (sec/Km)	Geographic Azimuth Window (degrees)	Regions Contained in Sector
A	$0.035 < U < 0.052$	$320 < \theta < 45$ (Through 360)	112, 113, 114, 115
B	$0.060 < U < 0.070$	$45 < \theta < 75$	116, 117
C	$0.067 < U < 0.090$	$110 < \theta < 170$	118, 119, 120, 121 122, 123, 124
D	$0.036 < U < 0.067$	$140 < \theta < 170$	125, 126, 127, 128 129, 130
E	$0.036 < U < 0.090$	$170 < \theta < 190$	131, 132, 133
F	$0.036 < U < 0.048$	$225 < \theta < 250$	134, 135
G1	$0.039 < U < 0.090$	$290 < \theta < 304$	103, 104, 110
G2	$0.039 < U < 0.090$	$304 < \theta < 312$	102, 105, 109
G3	$0.039 < U < 0.090$	$312 < \theta < 320$	101, 106, 107, 108

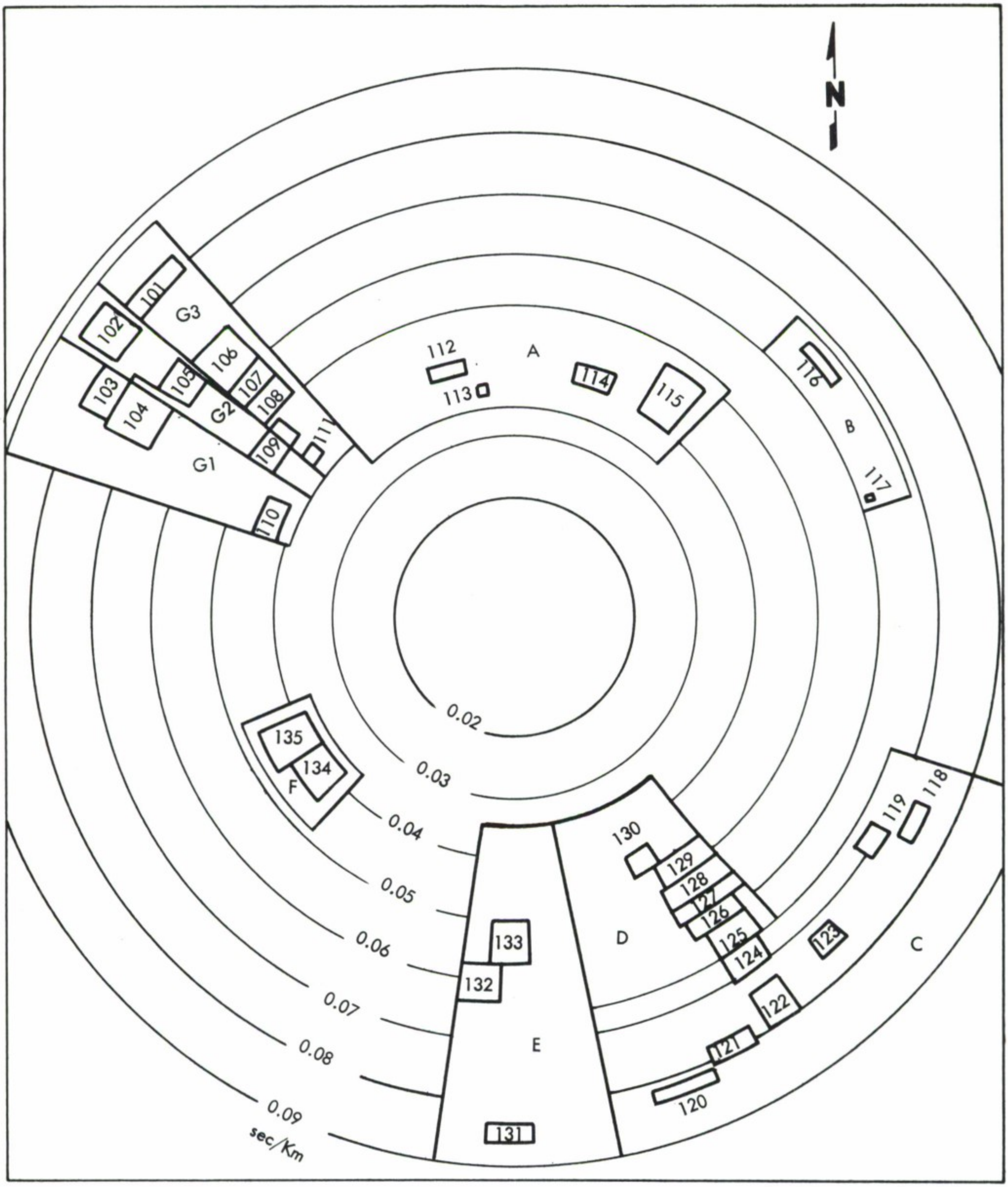


Figure 78. Relationship Between Regions and Sectors

Table 13

LINEAR INTERPOLATION FORMULA

Given: Points (X_1, Y_1) and (X_2, Y_2)

The straight line $Y = aX + b$ which passes through (X_1, Y_1) and (X_2, Y_2) may be expressed as:

$$Y = Y_1 + \left(\frac{X - X_1}{X_2 - X_1} \right) \cdot (Y_2 - Y_1)$$

Table 14

SUBARRAY FORMULAS FOR SECTORS

SUBARRAY B1

Sector	Region	U (Sec/Km)	Azimuth (degrees)	Deviation (seconds)
A	112	0.0435	348	0.144
	113	0.0435	355	0.147
	114	0.0435	22	0.210
	115	0.0435	39	0.147
B	116	0.065	52	0.090
	117	0.065	73	0.073
C	118	0.078	119	-0.011
	119	0.078	123	-0.062
	123	0.078	137	-0.188
	122-124	0.078	147	-0.209
	121	0.078	154	-0.253
	120	0.078	161	-0.273
D	125	0.064	145	0.238
	126	0.060	145	0.184
	127	0.057	145	0.131
	128	0.054	145	0.078
	129	0.050	145	0.024
	130	0.046	145	0.066
E	131	0.085	185	-0.159
	132	0.061	185	-0.075
	133	0.054	185	-0.012
F	134	0.042	224	0.151
	135	0.042	224	0.151
G1	103	0.077	297	0.054
	104	0.070	297	0.098
	110	0.043	297	0.044
G2	102	0.082	308	0.041
	105	0.067	308	0.031
	109	0.048	308	0.016
G3	101	0.081	316	0.100
	106	0.064	316	0.001
	107	0.058	316	-0.053
	108	0.054	316	0.011

$$Y = Y_1 + \frac{(X - X_1)}{(X_2 - X_1)} \cdot (Y_2 - Y_1)$$

$$Y = -0.062 + \frac{(130^o - 123^o)}{(137^o - 123^o)} \quad (-0.188 - (-0.062)) = -0.125 \text{ seconds.}$$

III.2.6 Uncovered Areas

It is also necessary to determine deviations for events which occur in areas not contained in any sector. Since such areas are not near areas in which numerous events are observed, little knowledge of the expected phase delays exists. One possible solution is to steer by means of plane waves with no subarray deviation correction. A second possible method is to interpolate or extrapolate using sets of regional averages from several regions. A third possible solution is to define more complex functions which will cover the entire teleseismic zone. This method appears most promising, but is contingent upon the results achieved by the presently defined nine sectors.

III.2.7 Testing The Regions And Sectors

This section discusses a test which was performed to evaluate steering delay correction factors based on the construction of regions and sectors. The results of the test indicate that regional correction factors can be employed successfully to obtain improved phase delays.

The test commences with a simulated steering of the array to an event located within a region, by means of plane wave delays modified by correction factors associated with the region. Next, a best fit plane wave is obtained for the event by using the time of arrival at the center seismometer of each subarray as input data for the Least Squares Wavefront Program. The difference between computed and actual arrival time at the center seismometer for each subarray is next calculated. Finally the Seismic Steering Delay Anomalies Program is used to evaluate the difference (if any) between the steering correction factor and the best fit plane wave deviation at each subarray. The Anomalies Program calculates from the 21 differences the loss (in dB) that can be attributed to using subarray steering correction factors. The magnitude of this loss measures the effectiveness of the set of steering corrections associated with the region containing the event. Since a region is characterized by its set of steering corrections, this loss also gauges the reliability of the region itself. If the loss is unacceptably large (where unacceptably large would have been previously specified), the conclusion is that one or more of the following factors is responsible.

a. The region is not well-defined, i.e., the correction factors are ineffective for steering the array.

b. The event was inaccurately timed at several subarrays.

c. A grossly inaccurate reading was made at one or more subarrays.

d. Some other parameter, such as depth, must be taken into account.

Factors b, c, and d can be eliminated by rechecking seismometer records and repeating the computations as required. If a recalculation continues to yield an unacceptably large loss, it must be concluded that the steering correction factors are of no benefit in reducing mis-steering loss. Such a test procedure will be more meaningful and reliable if there are several events per region which can be tested.

III.2.8 Test Results

The test conducted in this quarter was based on a set of 56 events, each with center seismometer arrival times, made available by Seismic Data Laboratory, Teledyne, Inc.

The events were processed through the Least Squares Wave Front Program which calculated inverse phase speed and direction for each event. As shown in Figure 79, it was found that 33 events fell inside regions and 23 fell outside.

The results of the test for the 33 events which fell inside regions are shown in Table 15. Columns 1 and 2 are self-explanatory. Column 3 indicates the number of center seismometers that were used in the curve fitting process. The fewer the number used, the less reliable is the fit. Inspection of the loss figures given in Column 4 shows that 26 of the 33 events suffered a loss of 1 dB or less at 1 Hz. Thus it may be concluded that regional correction factors would have helped steer to these events successfully. As for the remaining 7, it is seen from Column 3 that for 4 of them, the number of sensors used was 10 or less out of 21. This small number of sensors implies that the wave fitting is not to be regarded as valid or reliable because the deviations from the best-fit wave front are so large that they cannot be compared with regional averages. Consequently these events have been disregarded.

Interpolation procedures were not available for application to the 23 events falling outside the regions. However, some of these events lay rather close to regions and provided an opportunity to examine how far regional deviations could be extended. Accordingly, it was decided to simulate steering to eight of these events using for the simulation the correction factors for the regions closest to the events. These eight events are shown in Figure 80; the closest regions are shaded in the figure. The results of this attempt to stretch regional corrections are shown in Table 16. Columns 1 through 4 are similar to Columns 1 through 4 of Table 16. Column 5 contains relevant comments. As can be seen from Table 16, five of the eight could have been steered to successfully using regional averages. The three events for which regional average steering would have been unsuccessful have been labeled by a check (√) in Figure 80.

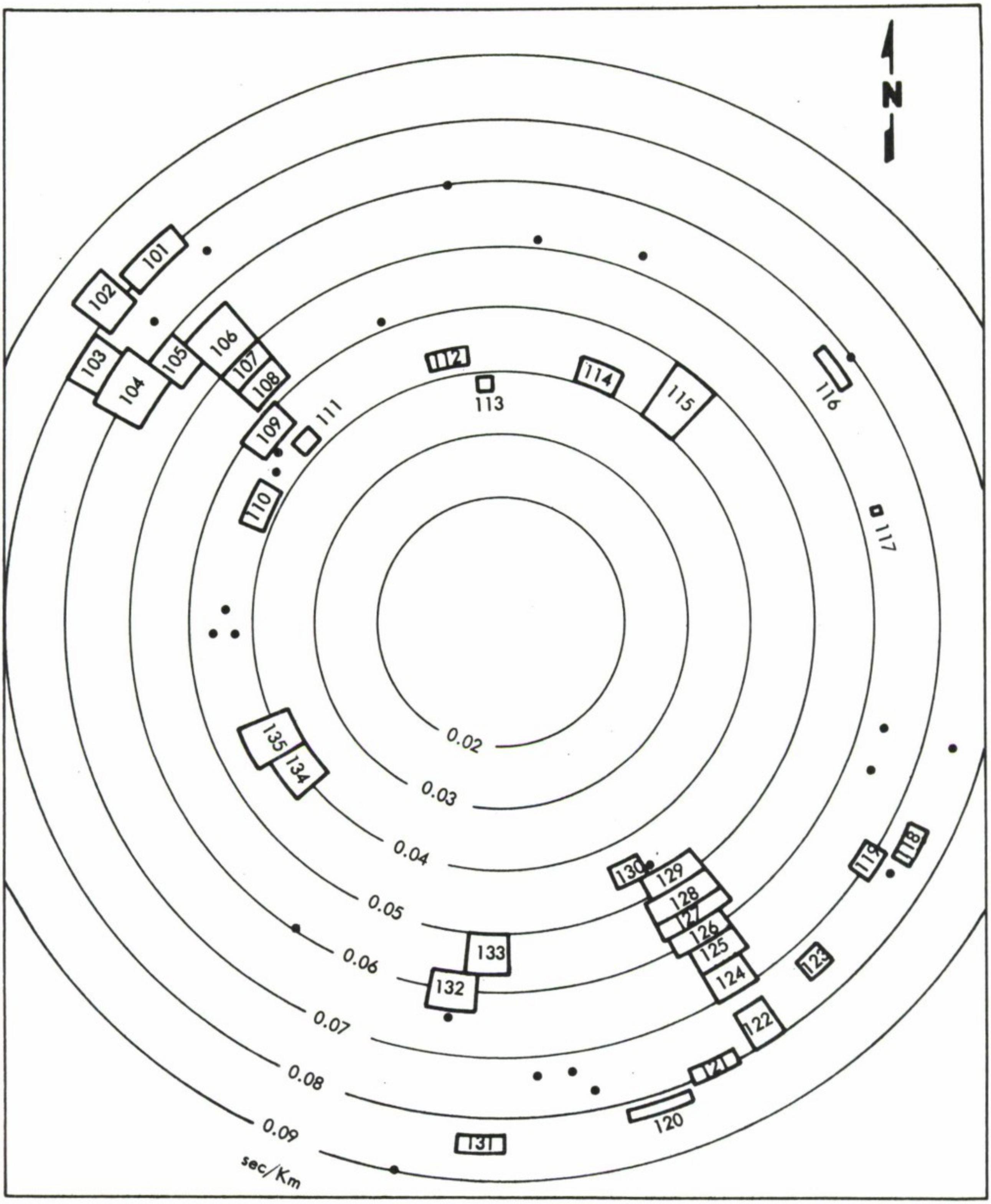


Figure 79. 23 Events Located Outside the 35 Regions

Table 15

TESTING OF THE REGIONAL AVERAGES USING
33 EVENTS LOCATED WITHIN REGIONS

Region Number	Event Identification Number	Number of Sensors	Loss (in dB)
101	4840	6	1.65
	4850	13	1.41
	4880	20	0.35
	5000	20	0.55
	5020	18	0.37
	5050	18	0.54
103	4830	17	0.33
	4980	17	0.60
	5100	20	0.57
104	4635	20	0.37
	4655	20	0.96
	4680	20	0.20
	5145	19	0.30
	5155	18	0.35
106	4625	20	0.51
108	4310	19	0.23
110	4639	9	1.01
114	4930	20	0.70
115	4780	16	1.01
	4810	14	0.43
	4860	17	0.57
	4940	21	0.50
116	4627	10	1.41
121	4700	11	0.74
	4790	19	0.81
	4870	10	1.31
124	4920	21	0.26
	4950	21	0.32

Table 15

(Concluded)

Region Number	Event Identification Number	Number of Sensors	Loss (in dB)
125	5060	21	0.37
128	4630	21	0.23
131	4800	19	0.34
	5010	12	1.45
135	4320	20	0.36

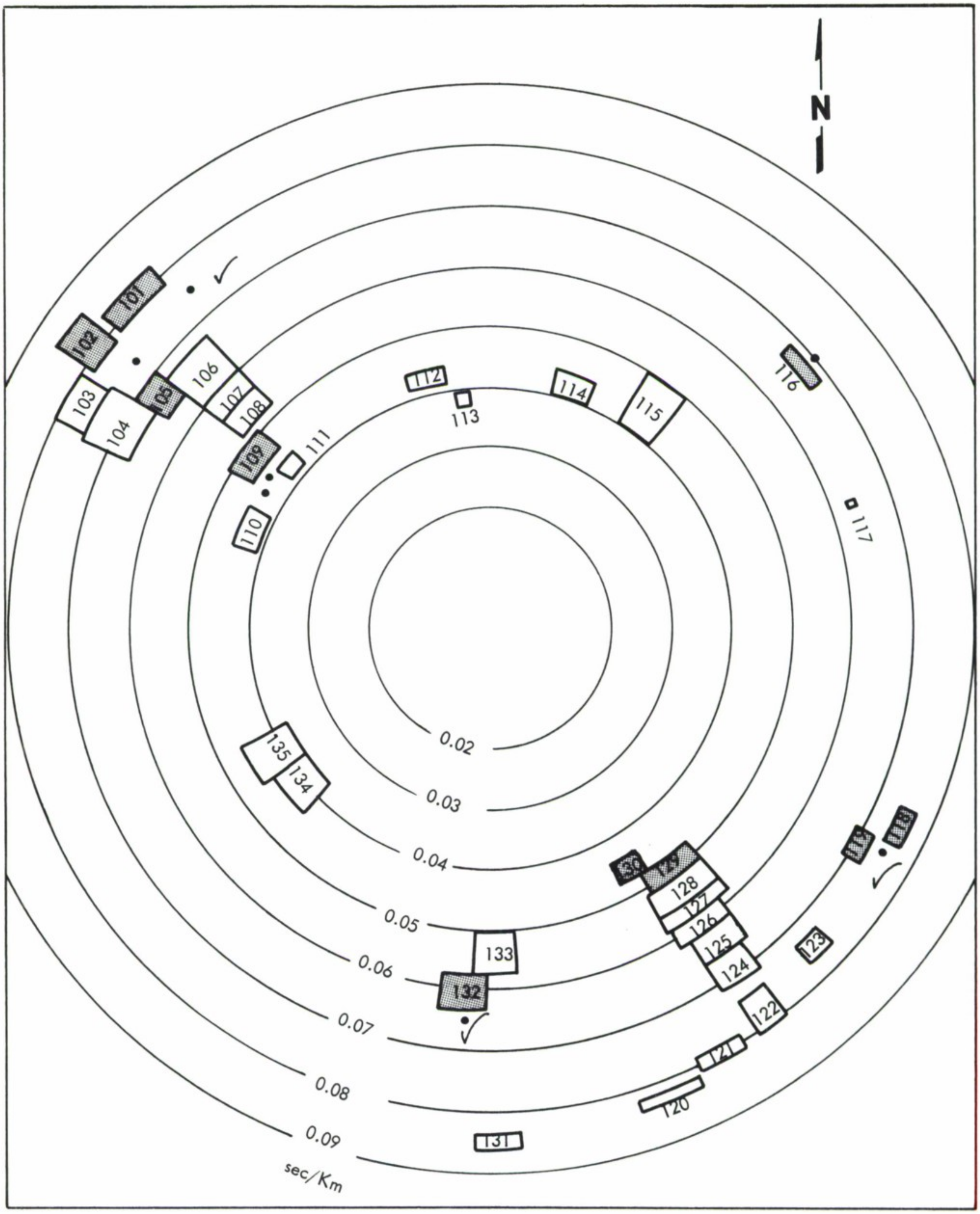


Figure 80. Steering to 8 Events Near Regions

Table 16

TEST OF STEERING TO EVENTS LOCATED NEAR REGIONS USING REGIONAL AVERAGES

Region Number	Event Identification Number	Number of Sensors	Loss (in dB)	Comment
101	4960	21	1.38	
102	4623*	11	1.61	This set unacceptable
105	4623*	11	0.71	This steering acceptable
109	4629	13	0.23	
	5115	20	0.23	
116	4631	17	0.52	
118	5080*	18	1.16	Neither set is acceptable
119	5080*	18	1.73	
129	5070*	18	1.01	Unacceptable
130	5070*	18	0.79	Acceptable
132	4637	15	3.24	

*Event is steered to using two or more sets of regional averages.

III.2.9 Conclusions

The test results verify that the present regions are well-defined. Correction factors associated with the regions have been used to steer successfully to nearly all events located within the regions. The fact that five events located near regions were also steered successfully by using the regional corrections lends credence to the planned method of interpolating between regions.

III.3 SUBARRAY ASSIGNMENT

III.3.1 Introduction

The task addressed here is to cover an arbitrary location, $\vec{U}_j$, in inverse velocity space by a judicious choice of pre-formed, pre-steered subarray beams. Ordinarily the problem is resolved by choosing that set of beams, one from each subarray, which is closest to the point $\vec{U}_j$. However, consider the events plotted in Figures 81, 82, and 83. The square (□) in each figure denotes the location of the event in question. Observed seismometer arrival times were used to best-fit a plane wave to each of the 21 subarrays. Then for each subarray, the best-fitting plane wave was used to calculate a value $\vec{U}_{\ell j}$ which may be interpreted as the event location as estimated by using only subarray ℓ. For each event, the 21 values $\vec{U}_{\ell j}$ have also been plotted in Figures 81, 82, and 83 where they are shown as dots (.). It is seen from the figures that in no case is $\vec{U}_{\ell j}$ equal to $\vec{U}_j$ for any pair (j,ℓ).

The situation depicted in these figures has several implications for beamforming and subarray steering. Since the dots denote the positions to which the subarrays should be steered to form the best LASA beam covering the corresponding square (□), it is possible that for one or more subarrays the preformed beam which is closest to the nominal location $\vec{U}_j$ is not the same as the beam which is closest to $\vec{U}_{\ell j}$. If this is true, the assignment of subarray beams is no longer an elementary task.

The fact that $\vec{U}_j$ and $\vec{U}_{\ell j}$ may not be the same point indicates that in some cases phase delays will be obtained by a chaining process. The location being steered to will not supply the delays, but it will point to several other locations, each of which will furnish its associated delay. Organization of the phase delay library must take cognizance of this possibility.

III.3.2 Notation and Definitions

Notation for indexes,as used in this section, is defined as follows:

NOTATION	DEFINITION	RANGE
i	instrument index	1, . . . , I
j	LASA array beam index	1, . . . , J
k	subarray beam index	1, . . . , K
ℓ	subarray index	1, . . . , L

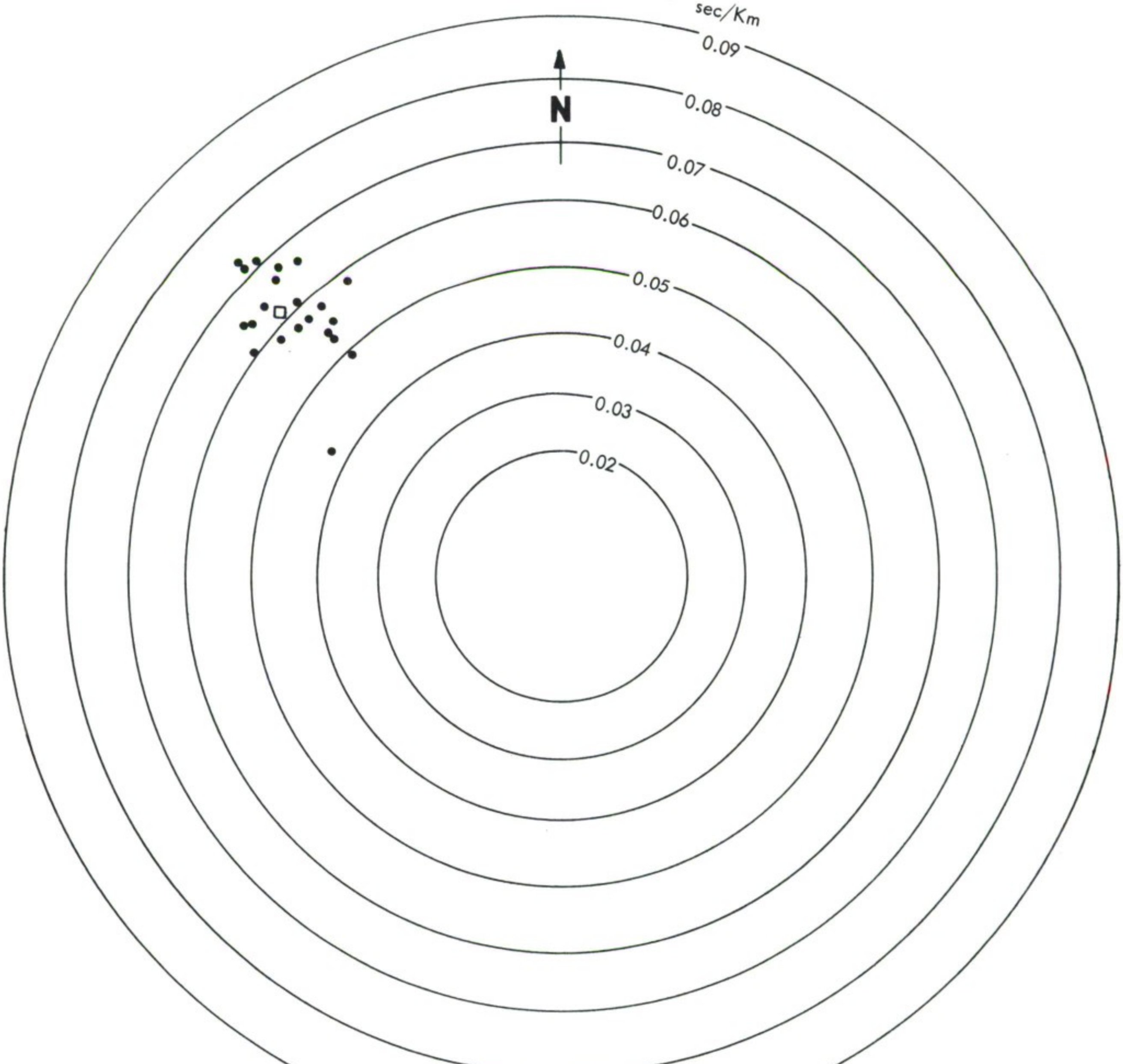


Figure 81. Kamchatka Event

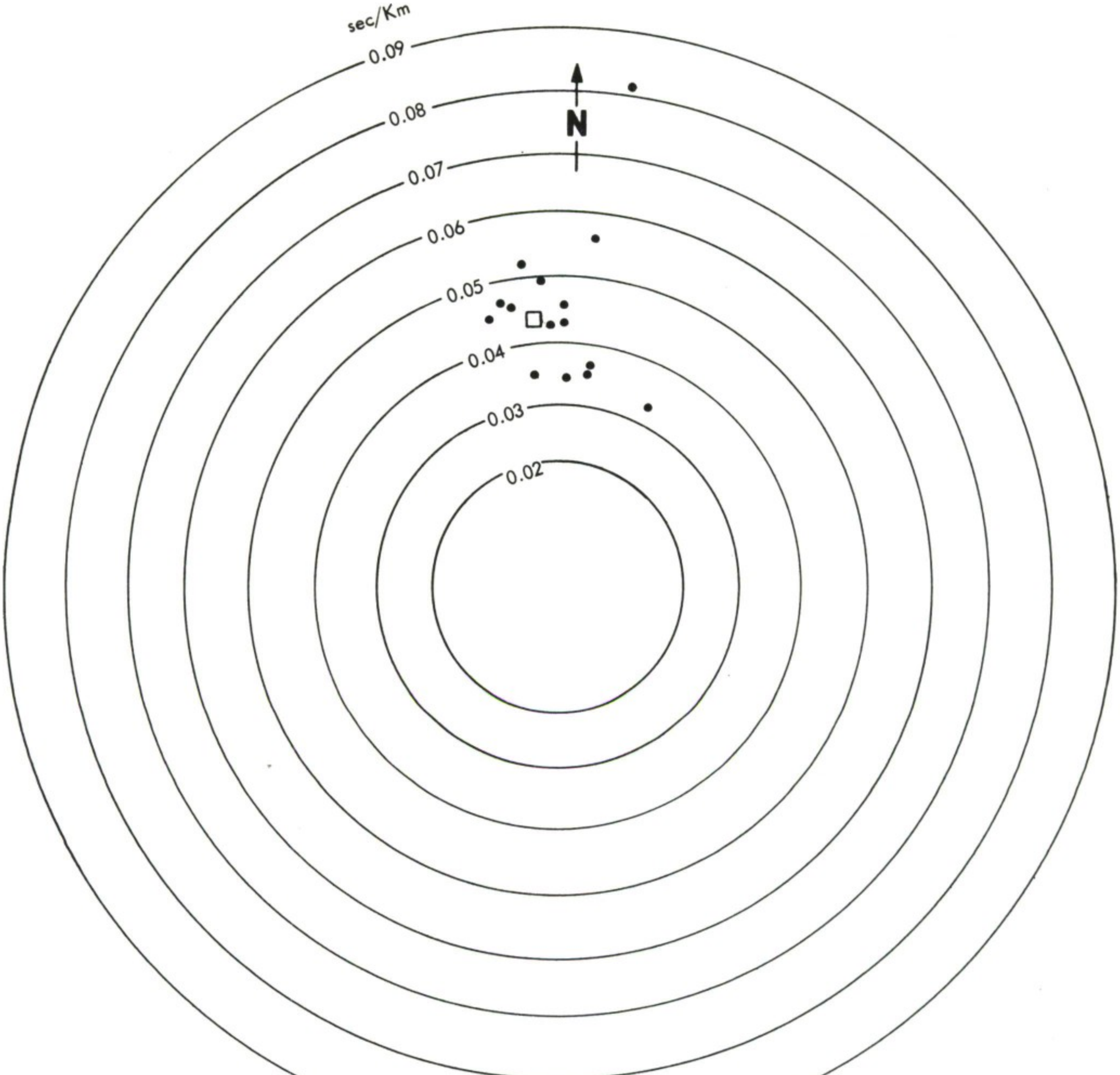


Figure 82. Kazakh Event

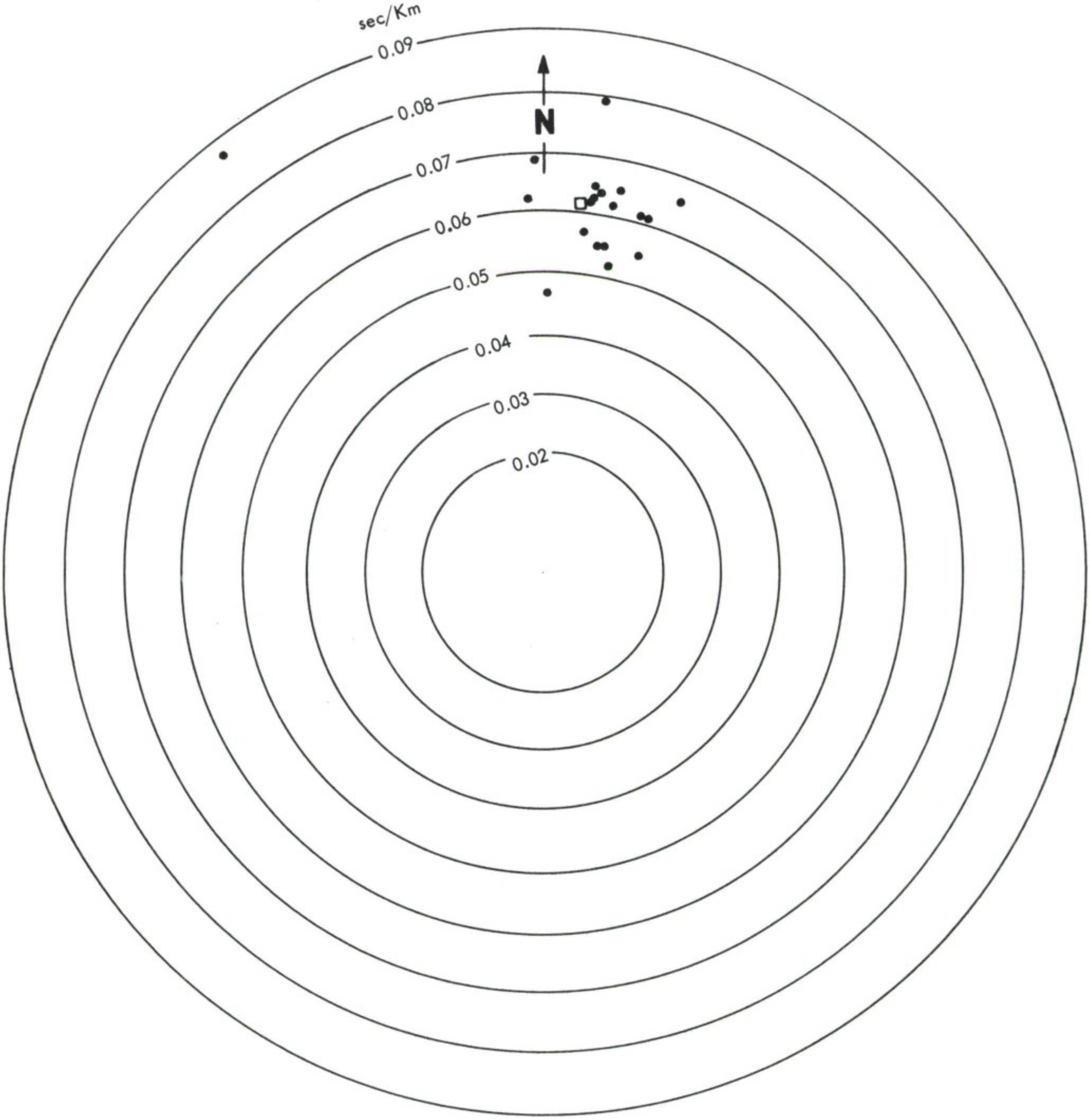


Figure 83. Novaya Event

The following bounds may be assumed for the above indexes: $I \leq 25$, $400 \leq J \leq 1024$, $K \leq 10$, and $L \leq 21$. The inequalities $I \leq 25$ and $L \leq 21$ imply that selective assignment of instruments and subarrays may be utilized for beamforming processes.

Predefined sets of points in inverse velocity space, assumed as inputs for the problem of subarray assignment, are as follows:

a. The arbitrary set $\{\vec{U}_j = (U_{xj}, U_{yj})\}$, assumed to be covered by a corresponding set of LASA beams.

b. The set $\{\vec{U}_{\ell j} = (U_{x\ell j}, U_{y\ell j})\}$, where $\vec{U}_{\ell j}$ is interpreted as an "aiming point" covered by that beam of subarray ℓ which best contributes to the formation of LASA beam j, under optimal processing. This set of points is referenced to the set$\{\vec{U}_j\}$, each point $\vec{U}_{\ell j}$ being an estimated location of $\vec{U}_j$, as viewed from subarray ℓ. It is noted that in the absence of anomalies, $\vec{U}_j = \vec{U}_{\ell j}$ for each pair ℓ, j.

c. The set $\{\vec{U}_{\ell k} = (U_{x\ell k}, U_{y\ell k})\}$, where $\vec{U}_{\ell k}$ is interpreted as an "aiming point" covered by the Kth beam of subarray ℓ. This set of points is assumed to have been selected on the basis of some previously defined optimization criteria and is subject to reassignment on the basis of these criteria.

III.3.3 Evaluation of Loss

Evaluating the maximum loss observed for any LASA beam is the first problem considered. This loss is associated with a specified steering pattern $\{\vec{U}_{\ell k}\}$ of preformed beams. A procedure for evaluating such loss is discussed below and presented diagrammatically in the flow chart of Figure 84.

First a specified LASA beam j is considered. For each subarray ℓ the subarray beam k is determined so that the vector magnitude $\Delta_{\ell jk} = |\vec{U}_{\ell j} - \vec{U}_{\ell k}|$ is minimal, and this minimal value is designated by the symbol

$$\Delta_{\ell j}^{k0} = \text{MIN}_k \{\Delta_{\ell jk}\} = |\vec{U}_{\ell k0} - \vec{U}_j| .$$

The minimization is accomplished by calculating the set of numbers $\{\Delta_{\ell jk}\ (k = 1, 1 \ldots, K)\}$ and selecting its smallest member $\Delta_{\ell j}^{k0}$.

The program now calculates the subarray loss $\lambda_{j\ell k_{o,\ell}}$ corresponding to $\Delta^{k}{}_{o\,\ell j}$, utilizing Formula 2.3[1] .

$$\lambda_{j\ell k_{0},\ell} = A_{\ell}(f\Delta^{k}{}_{o\,\ell j})^2 .$$

In the above formula, f is frequency and A_ℓ is a constant which depends on the subarray geometry.

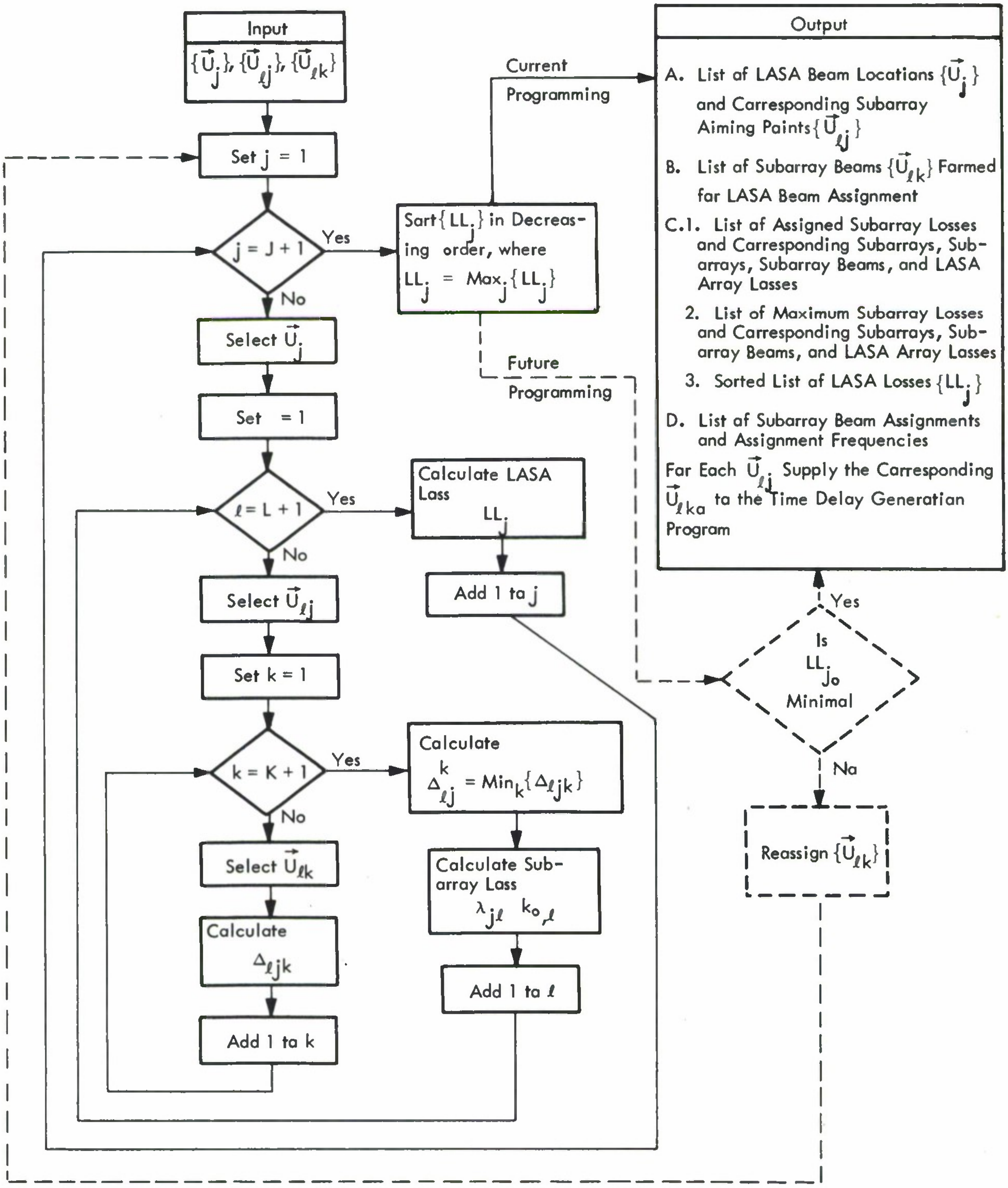

Input
$\{\vec{U}_j\}, \{\vec{U}_{\ell j}\}, \{\vec{U}_{\ell k}\}$
Set j = 1
j = J + 1
Yes
No
Select $\vec{U}_j$
Set = 1
ℓ = L + 1
Yes
No
Select $\vec{U}_{\ell j}$
Set k = 1
k = K + 1
Yes
No
Select $\vec{U}_{\ell k}$
Calculate $\Delta_{\ell jk}$
Add 1 to k
Calculate $\Delta^k_{\ell j} = \mathrm{Min}_k\{\Delta_{\ell jk}\}$
Calculate Sub-array Loss $\lambda_{j\ell}$ $k_{o,\ell}$
Add 1 to ℓ
Calculate LASA Loss LL_j
Add 1 to j
Sort $\{LL_j\}$ in Decreasing order, where LL_j = $\mathrm{Max}_j\{LL_j\}$
Current Programming
Future Programming
Output
A. List of LASA Beam Locations $\{\vec{U}_j\}$ and Corresponding Subarray Aiming Points $\{\vec{U}_{\ell j}\}$
B. List of Subarray Beams $\{\vec{U}_{\ell k}\}$ Formed for LASA Beam Assignment
C.1. List of Assigned Subarray Losses and Corresponding Subarrays, Subarrays, Subarray Beams, and LASA Array Losses
2. List of Maximum Subarray Losses and Corresponding Subarrays, Subarray Beams, and LASA Array Losses
3. Sorted List of LASA Losses $\{LL_j\}$
D. List of Subarray Beam Assignments and Assignment Frequencies
For Each $\vec{U}_{\ell j}$ Supply the Corresponding $\vec{U}_{\ell ka}$ to the Time Delay Generation Program
Yes
Is LL_{jo} Minimal
No
Reassign $\{\vec{U}_{\ell k}\}$

Figure 84. Subarray Assignment Program

There is a loss at the LASA level, represented by the symbol LL_j, which occurs because of the use of pre-formed beams. This loss is a function of the L subarray losses: $\{\lambda_{j\ell ko,\ell}\}$ ($\ell = 1, \ldots, L$). Loss LL_j is given by:

$$LL_j = -20 \log_{10} \left| \frac{1}{L} \sum_{\ell=1}^{L} 10^{-(\lambda_{j\ell ko,\ell}/20)} \right| .$$

Subarrays will be identified by number so that if one or more are deleted, their numbers will be passed over in forming LL_j. The purpose of such numbering is to permit identification of exactly which subarrays are being used in cases where the LASA beam is formed using fewer than the full number of subarrays.

The preceding calculations are iterated to obtain the sets of subarray losses $\{\lambda_{j\ell ko,\ell}\}$ ($j = 1, \ldots, J$; $\ell = 1, \ldots, L$) and the set of LASA losses $\{LL_j\}$ ($j = 1. \ldots, J$) associated with all LASA beams previously defined. The LASA losses $\{LL_j\}$ will be output in decreasing order. In addition, the computed subarray losses $\{\lambda_{j\ell k_o,\ell}\}$ will be listed. An independent output of the above program will be a list specifiying the subarray beams $\vec{U}_{\ell k_o}$ to be used in forming each LASA beam $\vec{U}_{\ell j}$. This list will be supplied to the Time Delay Generation Program.

III.3.4 Minimization of Loss

Remaining problems that must be formulated more precisely and for which solutions are needed are as follows:

a. Formulation of criteria for acceptable minimization of the maximum loss LL_{j_o} of the set $\{LL_j\}$ ($j = 1, \ldots, J$).

b. Formulation of criteria for iterated reassignment of the set $\{\vec{U}_{\ell \kappa}\}$ of subarray assignments. Such criteria should provide a method of feedback control maximizing the probability that iteration of the outer loop of Figure 84 will produce successively smaller values of LL_{j_o}.

Appendix IV

EVENT PROCESSING STUDIES

Event processing, as outlined in Reference 4, uses seismometer data along with detections supplied by the Detection Processor to estimate the location, depth, magnitude, and origin time of an event. Inherent to the system will be the necessary filtering, recording, and display processes with optional operator intervention.

This appendix contains two studies which support event processing. The first is concerned with an energy "density" method of calculating event magnitude. The method is related to the classical measure of magnitude for a sine wave ground displacement model. The second deals with the least squares orthogonal polynomial fitting method used to reduce voluminous discrete travel time data to a polynomial form. The polynomials can be used by the Event Processor to determine post P arrival windows and to estimate geographic locations of an event.

IV.1 MAGNITUDE ESTIMATION

An integral part of event processing will be the capability to automatically estimate the parameter magnitude which is related to the energy released at the source of the event. The current classical technique uses manual measurement to obtain waveform amplitude and dominant period. Computer extraction of these parameters appears difficult especially when the signal-to-noise ratio is small.

In this section an average energy estimation method is presented and is related to the calculation of event magnitude for a sinusoidal model of ground motion. Furthermore, the method is expected to produce reasonable estimates of magnitude for non-sinusoidal waveforms because, unlike the classical method, all spectral components contribute to the average energy estimate.

IV.1.2 Classical Determination of Magnitude

Gutenberg and Richter[18] have related the unified "magnitude" of a seismic event to the compressional body wave amplitude received at a seismic station as:

$$m_b = Q\,(\Delta,\, h) + \log_{10} \frac{10^6 A}{T} + S, \tag{89}$$

where $Q(\Delta,h)$ is the depth distance function for P wave using a vertical instrument, A is the maximum zero to peak amplitude of ground displacement in the first three cycles and measured in meters, T is the dominate period in seconds, S is the station correction, Δ is the epicentral distance in degrees between the event and the seismic station and h is the event depth in kilometers.

The above formula gives consistent results for manual measurements of A and T from seismograms. It is, however, not easily implemented on the digital computer, nor does it account for the spectral characteristics of the signal and difficulty is expected when the signal to noise ratio is low.

IV.1.3 Sine Wave Ground Motion Model and Magnitude Estimate

Richter[19] notes: "The quantity A/T has the advantage of being simply related, in theory at least, to the kinetic energy of its wave train." Using this, we assume that the ground displacement along the axis of a seismometer resulting from a distant event is of the form:

$$g(t) = A \sin \frac{2\pi}{T} t \text{ meters,} \tag{90}$$

where A is the maximum ground displacement in meters, T is the period in seconds and t is the time in seconds. The ground velocity is:

$$v(t) = \frac{2\pi A}{T} \cos \frac{2\pi}{T} t \text{ meters/sec.} \tag{91}$$

The kinetic energy of an incremental mass beneath the seismometer is:

$$E'(t) = 1/2\, \delta m\, v^2(t) \text{ joules,} \tag{92}$$

where δm is the incremental mass in kilograms. The average kinetic energy normalized with respect to incremental mass is:

$$< E(t) > = \pi^2 (\frac{A}{T})^2 \text{ joules/kg.}$$

The average energy for sinusoidal ground displacement can now be related to event magnitude.

To relate the average energy measurement of the sinusoidal model to magnitude, solve Equation (92) for $|\frac{A}{T}|$ and substitute in Equation (89) to give:

$$m_b = 0.5 \log_{10} |<E(t)>| + Q(\Delta, h) + K + S, \text{ and} \tag{93}$$

$$K = 5.502502 = -\log_{10}\pi + 6.0 \tag{94}$$

Equation 93 is valid for a sine wave input and should give a reasonable estimate of magnitude for other periodic waveforms because it includes the average power contribution of all spectral components. However, it does not include the effects of seismic noise.

Because low level seismic noise is always present, the average energy measurement indicated in Equation (92) is a sum of signal and noise energy. Assuming that the noise is stationary and random:

$$<E_{s+n}(t)> = <E_s(t)> + <E_n(t)>, \tag{95}$$

where $E_s(t)$ is the kinetic energy of the signal per unit mass and $E_n(t)$ is the kinetic energy of the noise per unit mass; so that

$$<E_s(t)> = <E_{s+n}(t)> - <E_n(t)> \tag{96}$$

Therefore, average signal energy can be found by subtracting the average noise energy from the measured average signal plus noise energy.

The magnitude estimation process is outlined in Figure 85. The signal is not truly periodic, but rather it is pulse-like in nature and has a large amplitude for just a few seconds of time. It is during these few seconds that the signal plus noise average energy must be estimated. The magnitude estimation process is accomplished by sliding a window along the waveform and calculating the average energy in the window. When the window covers the signal, its output is maximum. The peak value is taken as the estimate of the signal plus noise average energy. Equation (93) is rewritten as

$$m_b = 0.5 \log_{10} \left|<E_s(t)>\right|_p + Q(\Delta, h) + K + S \tag{97}$$

where $<E_s(t)>_p$ is calculated from Equation (96) at the peak value of $<E_{s+n}(t)>$.

The signal plus noise energy estimation window should be the same length as the significant signal. This is about one or two cycles depending on the signature of the event. Similarly, Equation (92) cannot be evaluated exactly for seismic noise because it is stationary for only a few seconds. Therefore, the average signal energy and consequently the event magnitude, is based on the two estimates of the average noise energy and the average signal plus noise energy.

IV.1.4 Sensor and Transfer Function

The sensor system transfer function is depicted in Figure 86. The velocity transfer function is much flatter than the displacement transfer function indicating that the seismometer is essentially a velocity measuring device. Using the instrument's velocity measuring capability, a direct approximation of the quantity A/T can be obtained and employed as indicated above.

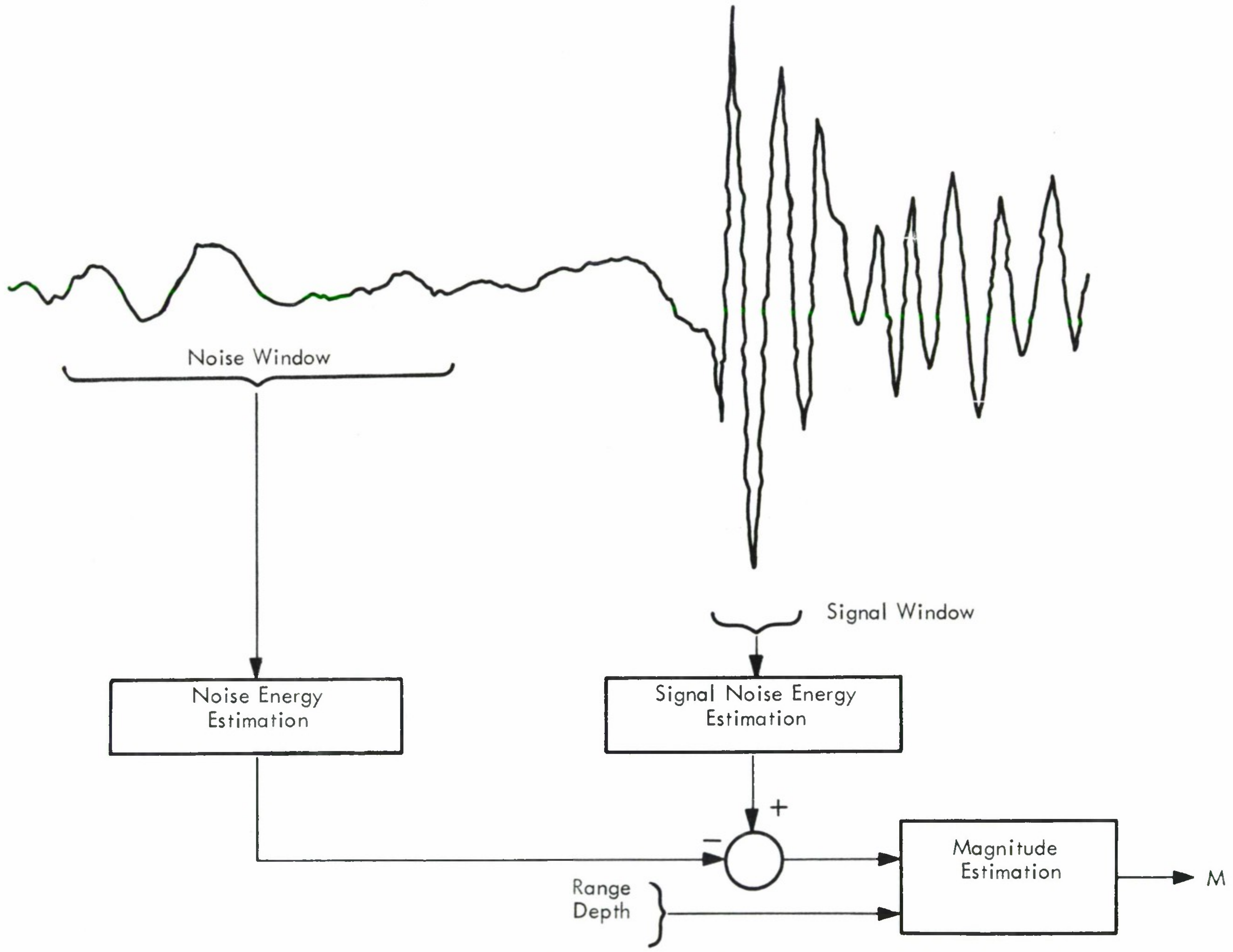
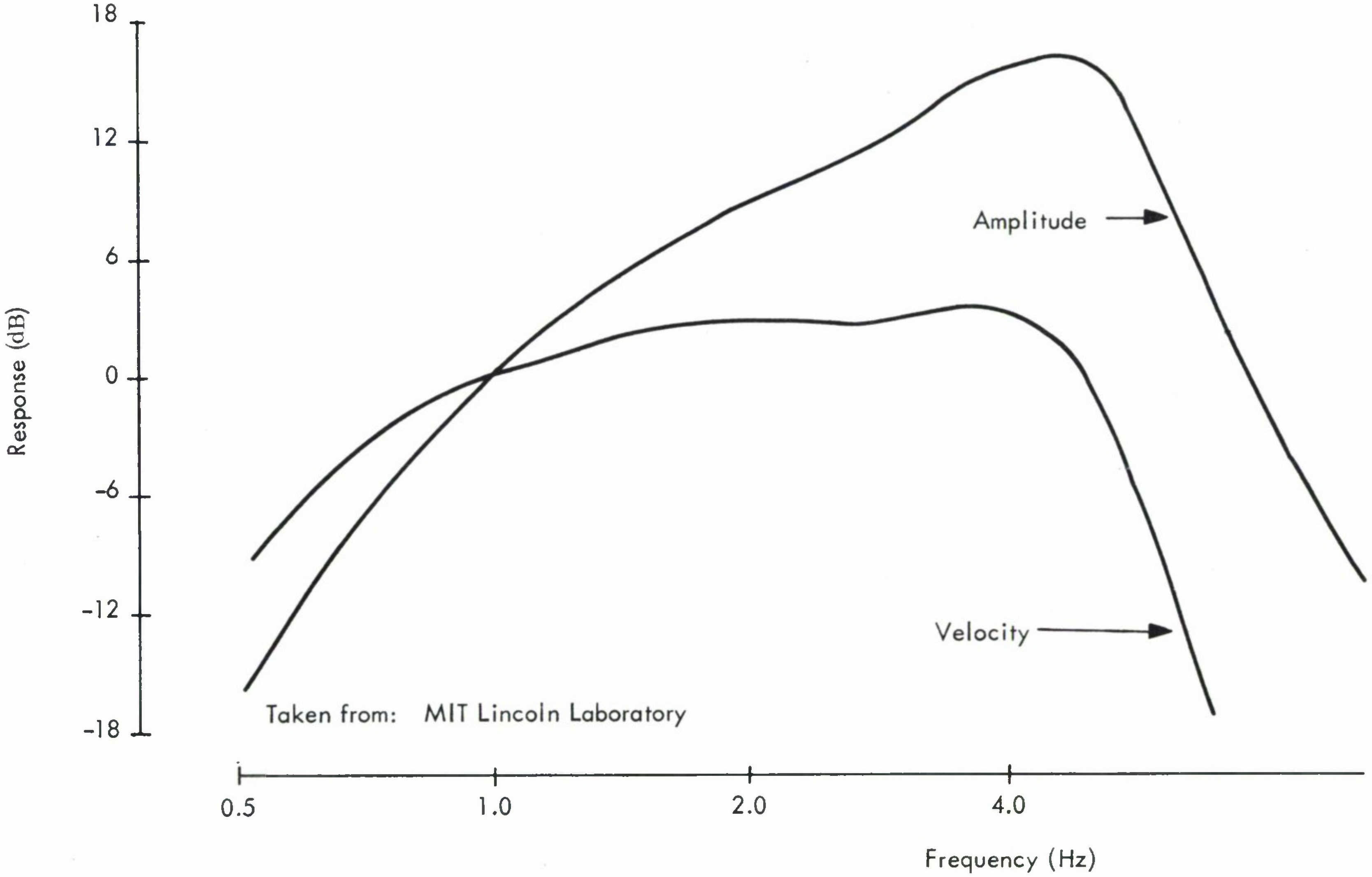


Figure 85. Automatic Magnitude Estimation Process



Taken from: MIT Lincoln Laboratory

Figure 86. Instrument Response

IV.2 TRAVEL TIME AND HORIZONTAL INVERSE PHASE VELOCITY CHARACTERIZATION

Seismic body wave travel time along with the related horizontal inverse phase velocity are featured in two areas of event processing.[4] First, the estimation of post P arrivals in both time after the P arrival and also location in the horizontal inverse phase velocity space (hereafter designated as $\vec{U}$ space). The second area of interest is the conversion of $\vec{U}$ space locations and arrival times for P and post P waves to geographic coordinates and an estimate of event depth.

The purpose of this section is to demonstrate the use of least squares orthogonal curve fitting in one dimension to reduce the necessary travel time and $\vec{U}$ space magnitude data to a minimum. Curve fitting also appears to automatically assist in interpolation and in obtaining the derivative of the discrete data.

IV.2.1 The Jeffreys-Bullen Travel Time Tables

The Seismological Tables by Jeffreys and Bullen[15], are used to determine event location and time of origin from event arrival times recorded at a set of seismic stations. They have been employed to aid in identifying a method for concise data representation and, quantitatively, how far such a process need be carried out.

The total travel time, T, for a seismic body wave can be found directly in the J-B tables as a two dimensional function of event range and event depth. The magnitude of the horizontal inverse phase velocity vector in seconds per kilometer is by definition the partial derivative of the travel time with respect to distance.

Finding each value of $|\vec{U}|$ from the J-B tables would require calculating the derivative of a low order smoothing function passed through several immediate data points or the physical storage of a corresponding table. Storage and manipulation of large tables (P alone has over 1400 entries) can be a burdensome process. Direct usage of the J-B tables to find total travel time, T, and the inverse horizontal phase velocity from an assumed range and depth is a case in point. While the tables are consistent in having 14 fixed depths, the epicentral range increments are not the same for all arrival waves. In particular, the change of range from one entry in the table to the next varies considerably in the depth allowance tables.

As an aid in identifying later arrivals of an event, there is a need to compute theoretical travel time differences between an assumed P wave and the later arrivals, and the utilization of approximating analytical functions must be considered.

IV.2.2 Table Approximation by Classical Least Squares

Since direct usage of the J-B tables would be computationally unwieldy in event processing, attention was given to the classical curve fitting of data points in the least squares sense. The fit to P wave data of surface depth deteriorated beyond fourth order despite the use of double precision (16 digit) arithmetic computer operations. This was not surprising, since the classical method requires the inversion of a matrix and involves the notorious principal minor of the well known Hilbert matrix[20].

Curve fitting, by the use of polynomials which are orthogonal under summation, was deemed the next logical step since their use does not require the inversion of a matrix. While tabulations of precalculated orthogonal polynomials are available[21], their application requires equal spacing of the independent variable increments.

Since the J-B tables have an unequally spaced variable of epicentral range, Forsythe's regression form of orthogonal polynomial curve fitting was adopted[20,22]. A brief description of this method follows.

Given data for (m) points (x_k, y_k), $k = 1, 2, \ldots, m$; express (y) as a function of (x) by the use of orthogonal polynomials of degree 0 through degree (n), where $(n < m - 1)$. Thus, find $b_0, b_1, b_2, \ldots, b_n$ so that:

$$y(x) = b_0 P_0(x) + b_1 P_1(x) + \ldots + b_n P_n(x), \tag{98}$$

where $P_i(x)$ is a polynomial of degree (i) and y(x) best fits the data in the sense of least squares:

$$\sum_{k=1}^{m} [y_k - y\,(x_k)]^2 \text{ is a minimum,}$$

where y_k is the given tabular value and $y(x_k)$ is the corresponding calculated value.

After y(x) has been expressed in the form 98, it can be converted to a polynomial in powers of x since each $P_i(x)$ is a polynomial in x:

$$y(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n . \tag{99}$$

The coefficients $b_0, b_1, b_2, \ldots, b_n$ may be found from the formula:

$$b_i = \frac{\sum_{k=1}^{m} y_k P_i(x_k)}{\sum_{k=1}^{m} (P_i(x_k))^2} . \qquad (100)$$

The orthogonal polynomials are generated by a recursive formula as follows:

$$P_{-1}(x) = 0$$

$$P_0(x) = 1$$

$$P_{i+1}(x) = x\, P_i(x) - \alpha_{i+1} P_i(x) - \beta_i\, P_{i-1}(x). \qquad (101)$$

The coefficients α_{i+1} and β_i are chosen so that the polynomials P_0, P_1, P_2, . . . , P_n are orthogonal under summation:

$$\sum_{k=1}^{m} P_i(x_k)\, P_j(x_k) = 0 \text{ for all } (i \neq j). \qquad (102)$$

The formulas for α_{i+1} and β_i are as follows:

$$\alpha_{i+1} = \frac{\sum_{k=1}^{m} x_k (P_i(x_k))^2}{\sum_{k=1}^{m} (P_i(x_k))^2} , \qquad (103)$$

$$\beta_i = \frac{\sum_{k=1}^{m} (P_i(x_k))^2}{\sum_{k=1}^{m} (P_{i-1}(x_k))^2} \quad . \qquad (104)$$

Forsythe suggests transforming the independent variable x to the range -2. to +2. to avoid possible overflow. This transformation also permits fewer digits to be used for each coefficient a_i for a specific accuracy of fit.

IV.2.3 Results of Least Squares Orthogonal Polynomial Fit to the Jeffreys-Bullen Tables

The Forsythe regression form of orthogonal polynominal curve fitting was used to reduce the tables to manageable polynomials. The errors of fitting the travel time, T, for the P wave, PcP wave, and pP-P wave difference are evident on figures 87 to 90.

Figure 87 is primarily a plot of the error of fit in seconds versus the order of the polynomial fit. Two indications of fitting error are given: the standard deviation (RMS error) and the maximum deviation. Both are plotted as a range covering the 14 depths. Using $|\vec{U}| = 0.04$ sec/Km at 30° distance as a worst case, a timing error of one second is equivalent to a 25 Km error in location and is called the "location error bound" on the graph. Q is the quantization level of the original J-B data and σ is the expected RMS error in the original data caused by quantization.

The ninth order fit for T seems best because the maximum location error is only 3 Km and the RMS error value of the fit was bounded by σ.

Figure 88 is very similar to Figure 87 except that the wave in question is PcP and for an event at the surface only. Using $|\vec{U}| = 0.023$ sec/Km at 30° distance as a worst case, a timing error of one second is equivalent to a 43.5 Km error in location. An eighth order fit looks best for T.

Figure 89 shows the error of fit for the 13 depth correction polynomials for PcP. The d/R_0 is a standard form for showing depth in a compact form and can be viewed as ratio of the event depth below the crust (the crust is about 33 Km thick) to the radius of the earth minus the crustal layer. Thus, a depth of 33 Km, as measured from the surface of the earth, becomes equal to zero in d/R_0 format. The worst case for an error in the estimation of depth due to a timing error occurs at 80° range and $d/R_0 = 0.11$. Here, only 5.1 seconds

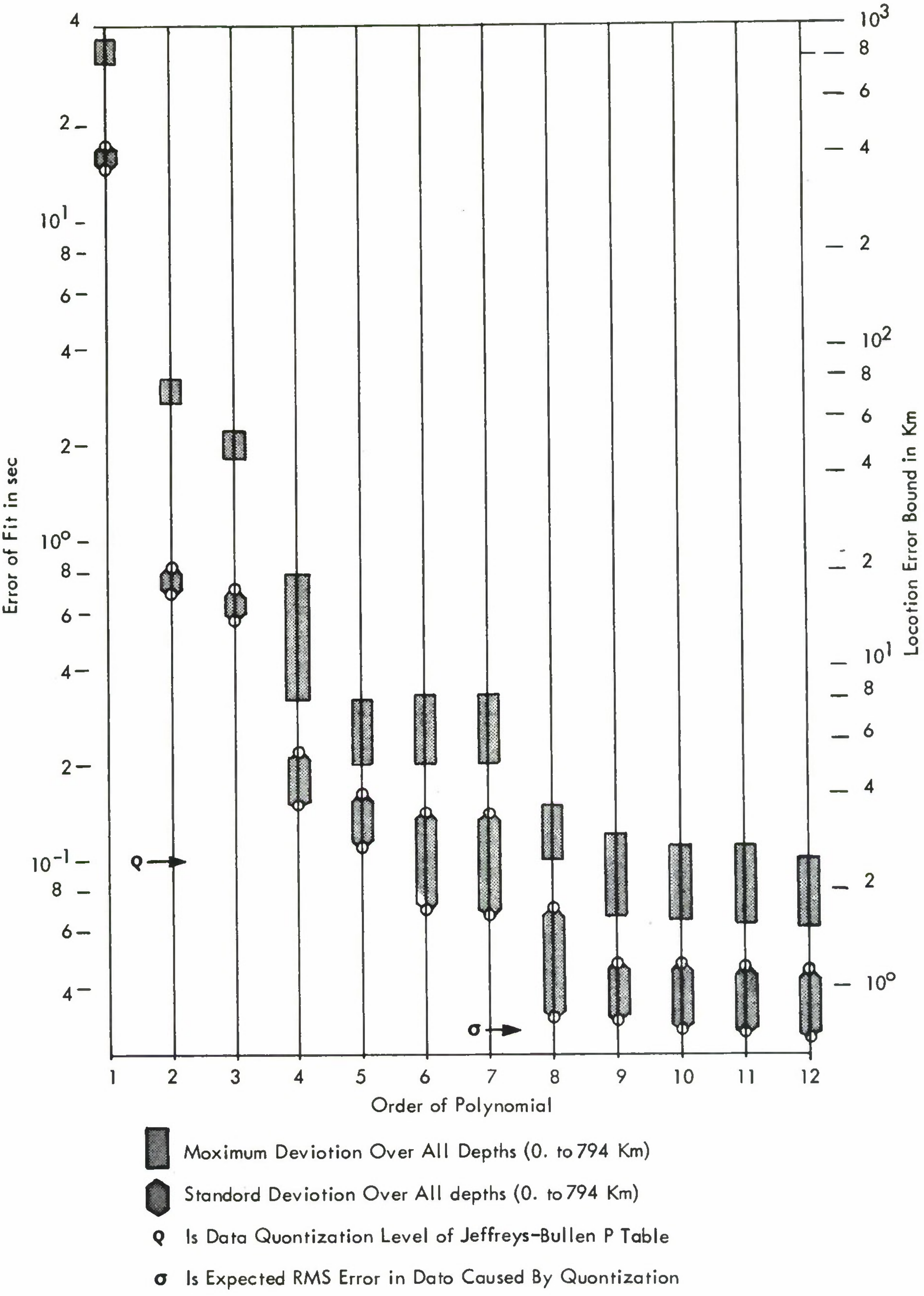
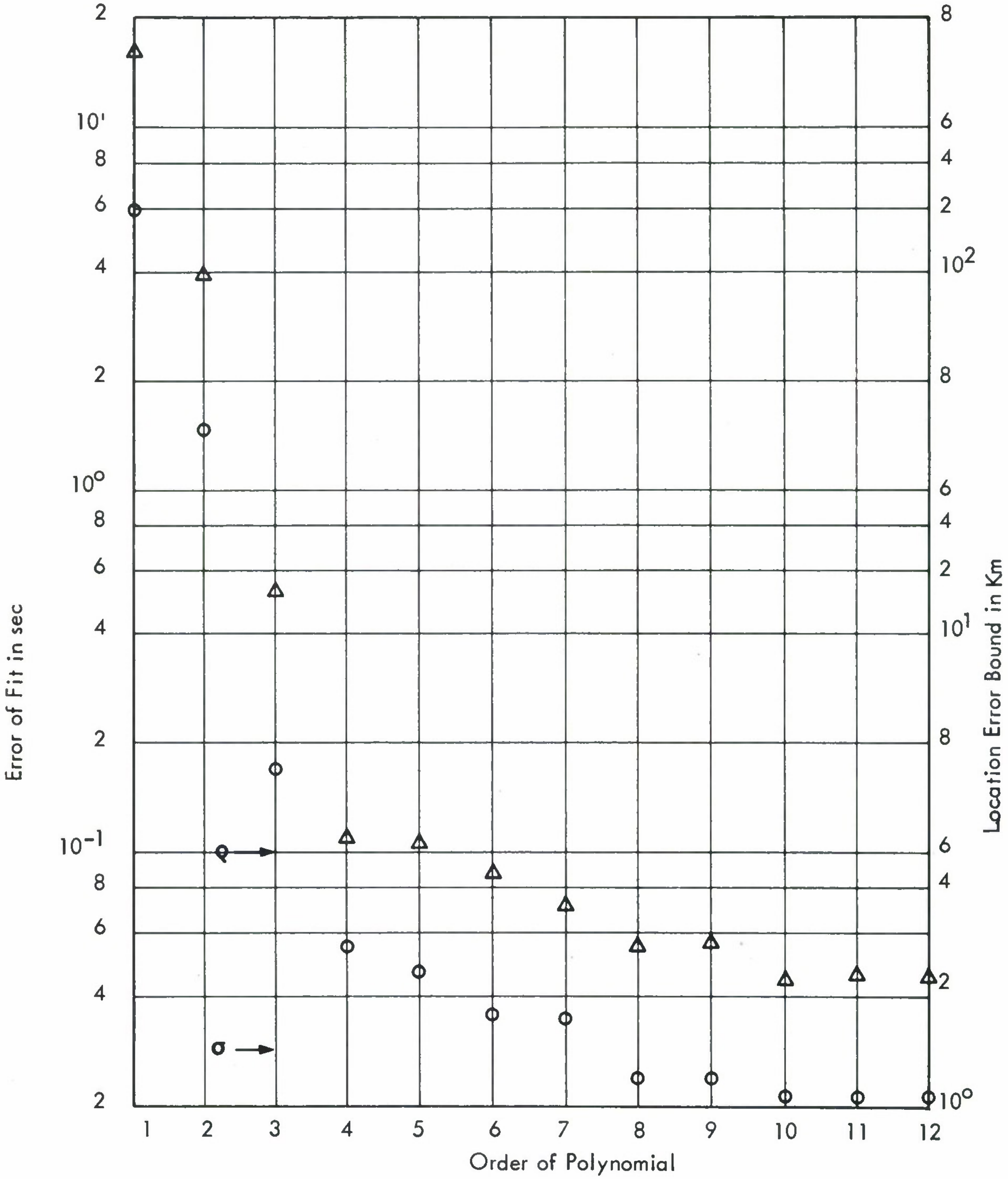


Figure 87. Least Squares Fit for P Wave at 14 Depths for 25° - 105° Range



△ Maximum Deviation

○ Standard Deviation

Q Is Data Quantization Level of Jeffreys-Bullen P_cP Table

σ Is Expected RMS Error in Data Caused by Quantization

Figure 88. Least Squares for PcP Wave at Surface Depths for 25° - 100° Range

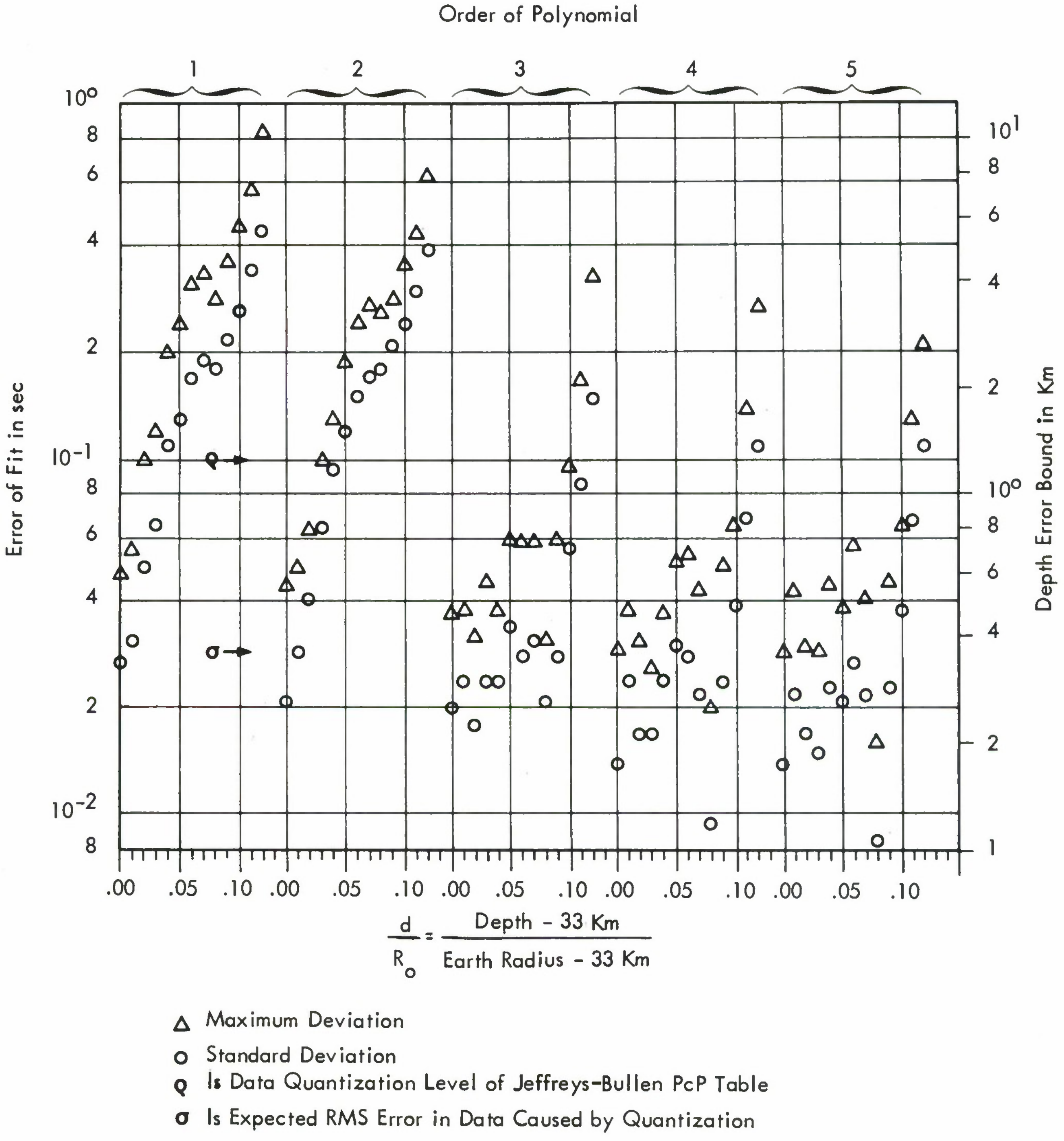


Figure 89. Least Squares Fit for PcP Wave at 13 Depths for 0° - 80° Range

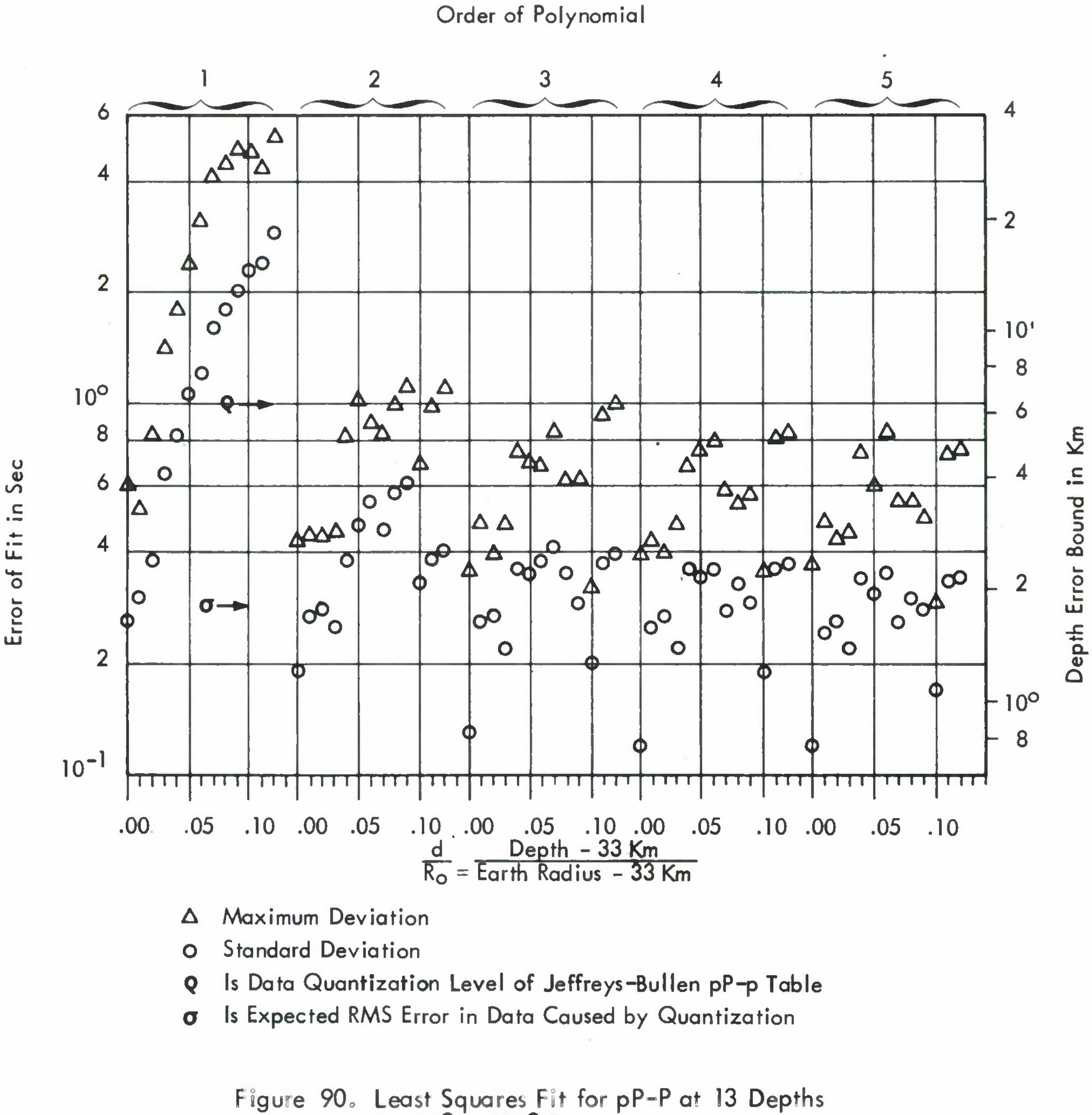


Figure 90. Least Squares Fit for pP-P at 13 Depths for 30° - 100° Range

separates the two depth columns (0.11 and 0.12) in the J-B tables. This results in an upper bound of 12.4 km error in depth for a one second timing error.

Figure 90 is similar in design to Figure 89 and shows the error of fit for the pP-P time difference. Since $|\vec{U}|$ is the derivative of travel time with respect to distance, the order of the $|\vec{U}|$ polynomial will be one order less than the proper fit for T.

IV.2.4 Conclusions and Recommendations

Forsythe's regression form of orthogonal polynomial curve fitting outperforms the classical least squares method and appears to be quite useful in fitting one dimensional travel time data. While taking the derivative of the polynomial fit to obtain $|\vec{U}|$ as a function of range seems better than the derivative of discrete data, it may offer only marginal accuracy in geographic conversion yet be perfectly acceptable for post P detection windows in $\vec{U}$ space. Fitting T and $|\vec{U}|$ data obtained at LASA should be no problem since they are direct measurements. It is recommended that a two dimensional surface fitting method be developed so that both range and depth are independent parameters.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* International Business Machines Corporation 18100 Frederick Pike Gaithersburg, Maryland 20760	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

THIRD QUARTERLY TECHNICAL REPORT
EXPERIMENTAL SIGNAL PROCESSING SYSTEM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
None

5. AUTHOR(S) *(First name, middle initial, last name)*

None

6. REPORT DATE August 1967	7a. TOTAL NO. OF PAGES 189	7b. NO. OF REFS 22

8a. CONTRACT OR GRANT NO. F19628-67-C-0198	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-68-149
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Directorate of Planning and Technology, Electronic Systems Division, AFSC, USAF, L G Hanscom Fld, Bedford, Mass. 01730

13. ABSTRACT

The Third Quarterly Technical Report for the LASA Experimental Signal Processing System identifies the effort expended to provide the hardware and software in support of research and development directed toward the study of seismic signal processing. It also delineates work tasks planned for the next quarter. This document presents detailed information related to machine configurations, time delay correlation, event location accuracy, optimum processing, fast Fourier transform, array design, steering delay library, magnitude estimation, and travel time characterization.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

Security Classification