A Geometric Interpretation of Diagnostic Data from a Digital Machine: Based on a Study of the Morris, Illinois Electronic Central Office

By J. B. KRUSKAL and R. E. HART

(Manuscript received June 7, 1966)

Using the diagnostic data collected for the Morris Central Control malfunction dictionary, we devise a natural concept of "distances" between malfunctions. Ten thousand malfunctions were placed as points in six-dimensional space in such a way that the Euclidean interpoint distances approximately equaled the diagnostic "distances". The remarkable fact that this is possible has many implications.

By finding circuit characteristics common to a cluster of neighboring malfunctions, we are able to associate these characteristics with the region of the six-dimensional space which holds these malfunctions. By this means, we characterize various regions of space according to functional troubles. This suggests a technique for locating malfunctions and also suggests some longer-range possibilities.

Table of Contents

I. INTRODUCTION

To identify a malfunction is always difficult. The great size, complexity, and speed of modern digital machines render this difficulty severe. An age-old method is to observe the symptoms and deduce their cause. For larger machines, this is exceedingly impractical.

A familiar aid is the use of tests. The record of tests passed and tests failed provides many more clues to the trouble. However, even the use

of tests does not avoid time consuming analysis by a highly trained expert, which is slow and expensive.

One very successful and ingenious approach to alleviating this difficulty (proposed by Werner Ulrich) is to make a large dictionary listing many malfunctions with corresponding test results. For each of many known malfunctions, we obtain a pattern of 0's and 1's which indicate the test results. For example:

Test Number	1	2	3	4	5	6	7	8	···
Result	0	0	1	0	1	1	0	0	···.

A 0 indicates a correct result and a 1 an incorrect result. These patterns are then arranged in some systematic order. Together with each pattern we include identification of the malfunction. When we wish to find a malfunction, we simply locate the pattern of test results in the dictionary. If a sufficiently comprehensive set of tests is used and a sufficiently comprehensive set of known malfunctions is included, such a dictionary can achieve a high degree of success. We note that to collect the test patterns for the dictionary, the only practical procedure may be actually to insert the malfunctions in a real model of the machine.

Not all malfunctions can be found by using such a dictionary. Some conceivable malfunctions will not be listed in the dictionary, and other malfunctions produce different test results on different occasions (inconsistent results). However, it is not necessary to use a dictionary only for exact pattern matching. If a malfunction produces different patterns on different occasions, we may expect these different patterns to be "similar" to each other and, indeed, this has been found to be true of the data from the Morris, Illinois electronic central office. Broadly speaking, we feel that patterns are similar if they differ in only a few places. We call the number of places in which two patterns differ the "Hamming distance" between the patterns. This is a rough measure of dissimilarity between patterns.

However, some tests are more important than others. Therefore, we have refined the idea of Hamming distance by weighting the various tests and using weighted Hamming distance (WHD). We have found that the WHD between two patterns is a good and meaningful measure of dissimilarity between the malfunctions which yield those patterns.

"Distances" suggest a geometric model. Is it possible, for example, to represent each malfunction by a point in a plane, in such a way that the (ordinary Euclidean) distance between two malfunctions is approximately equal to the WHD between the corresponding patterns? If true, it would be tremendously significant. For it would mean that the patterns and hence the malfunctions somehow form a two-dimensional

set, that each malfunction can be represented by two coordinates in a way that contains the information in the patterns.

It would be equally significant if we could represent the malfunctions by points, not in the two-dimensional plane, but rather in three-dimensional space, or even by points in n-dimensional space, as long as n is reasonably small.

In fact, the Morris data can be represented by such a geometric model. For these data, six dimensions were found to be appropriate. In six dimensions the typical deviation for our data between WHD and Euclidean distance (ED) is reasonably small (about 7 percent).

We emphasize the fact that the small number of dimensions is not something that would happen with just any data, nor could it happen by chance. Random data might have fitted into 100 dimensions by chance. But the smaller the number of dimensions needed, the more significant the result. Six dimensions are remarkably few to represent 10,000 patterns of 657 bits each.

It should also be understood that the number 6 is approximate, and that 5 or 7 are also reasonable values. Fewer dimensions can be used at the cost of larger deviations between WHD and ED, while more dimensions can be used to further reduce these deviations. However, the value 6 was chosen by following the principle of parsimony, which recommends that data be represented by as few numbers as are needed to fit the data satisfactorily.

Not only the malfunctions have a geometrical interpretation (as points in six-dimensional space): it appears that a test can be represented as a "hyperplane" (that is, a flat cut of all space) that separates the malfunctions that fail from the malfunctions that pass the test.

A convincing demonstration of the meaningfulness of the geometric interpretation of malfunctions as points in space lies in the way malfunctions with some common characteristic cluster together in space. For example, malfunctions internal to a single register may cluster together in a small region; malfunctions which often affect the logical state of a single lead may cluster together; and malfunctions which affect the common function of a group of related operations may cluster together. In this paper, we discuss several such clusters of malfunctions in the Morris data. It is not easy to predict in advance how the malfunctions will cluster together, but by examining the geometric model we may observe which characteristics describe clusters of malfunctions. It should be obvious that since closeness in the geometric model is based on WHD, malfunctions which cluster together are those which affect the functioning of the machine in similar ways.

It should be noted that to predict how a particular malfunction affects

the functioning of the machine is exceedingly difficult. Thus, while two malfunctions in the same circuit might be thought to have similar effects, and in many cases do, it also happens that two malfunctions which might be expected to react similarly turn out to be quite different. Detailed analysis of such cases reveals unexpected facts about how the machine reacts to malfunctions. Such analysis has given us new insights into the nature of malfunctions.

The WHD has a definite utility in locating a malfunction if we are not able to locate it by exact pattern match in the dictionary. For given the pattern of the unknown malfunction, and some pattern in the dictionary, we may judge the likelihood of the unknown malfunction being the dictionary malfunction by the WHD between the two patterns. The smaller the WHD, the greater the likelihood of the two malfunctions being the same.

The utility of the geometric model lies partly in the fact that it gives a more compact (or parsimonious) way of representing much of the information which is contained in the patterns and WHD's. For example, to provide a way of locating those patterns which lie within some small WHD of the pattern of the unknown fault is very difficult without the geometric model. But with the geometric model we can simply cut (six-dimensional) space into cells, and list the faults within each cell; this serves the same purpose.

Another potential utility of the geometric model is the possibility that it may reveal some underlying truths about malfunctions. Since each malfunction can be represented by six coordinates, it is natural to ask whether each malfunction is characterized by the degree to which it possesses each of six hypothetical underlying characteristics. If we could find such underlying characteristics, there would surely be many benefits. However, we have not isolated such characteristics.

II. BACKGROUND

2.1 *The Electronic Central Office at Morris, Illinois*

All the data and illustrations in this paper are associated with the electronic central office which was in commercial use for over a year at Morris, Illinois, between 1960 and 1962. A duplicate system was built and tested during the same period at Whippany, New Jersey, and the dictionary data we shall describe were actually collected on the Whippany laboratory model. We shall give an extremely brief description of these systems. We hope that readers already familiar with the system will excuse the omissions and extreme simplification necessary in such a

brief account. For a good general description of the system, see Ref. 1. For more detail, see Refs. 2 and 3. The final report (Ref. 4) gives a good account of the results of the whole experiment.

The electronic central office (ECO) contains the central control (CC), the flying spot store (FSS), the barrier grid store (BGS), as well as the subscriber lines, trunk circuits, the switching network itself, and other important units. Our attention is focused on the CC, which controls all the other units (see Fig. 1). The CC is a stored program machine. One memory device for it is the FSS, which provides semipermanent memory for the stored program and for large tables of "translation" information. The other memory device is the BGS, which provides changeable memory in which the CC records calls in progress, numbers being dialed, etc. The CC communicates directly with the switching network.

To assure continuous operation of the central office, certain subsystems are provided in duplicate. At any given moment, one unit of each duplicate pair of units has "active" (controlling) status, while the other unit has either "standby" (ready to take control) or "out-of-service" (malfunctioning) status. The system is so organized that either unit of a duplicate pair can be made active, independently of the status of the other pairs. To insure that the standby units will be ready to assume active status when needed, they are continuously exercised. Even in the absence of any malfunction, the roles of active and standby are exchanged periodically.

To prevent machine malfunctions from propagating large amounts of wrong information in the changeable memory, malfunctions must be detected quickly and the processing of telephone traffic interrupted until the faulty unit is taken out of service and replaced by its standby. To achieve very rapid detection of machine malfunctions, the outputs

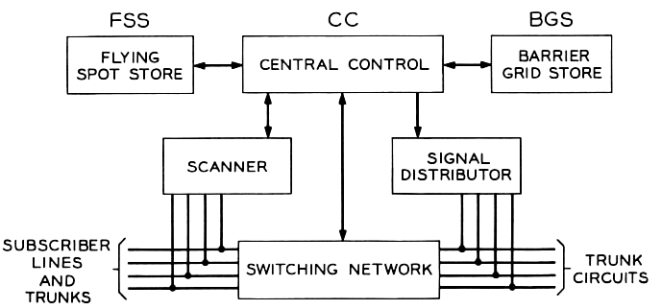

Fig. 1 — Simplified block diagram of the electronic switching system at Morris, Illinois.

of the active and standby units of each duplicate pair are continually compared. In the case of units other than the CC, it is the CC which does the comparing and which takes the proper action. This includes checking to determine whether the trouble symptoms will recur, and if so which unit is responsible (for example, one particular BGS or one particular FSS). The CC then performs diagnostic tests on the unit and reports the results to the maintenance craftsman for corrective action.

The treatment of malfunctions occurring in a CC is necessarily different. Since the CC is the "doctor" who decides which unit has the malfunction, trouble in the CC leads to a situation in which the "doctor" must diagnose himself. This is a complex situation and one that requires special precautions and handling. Briefly, the procedure is this. Under normal circumstances, the two CC's are performing identical operations at the same time. Through a limited number of interconnections between them, each CC receives the results of certain operations in the other CC. These are compared, by means of matching circuits in each CC, with its own results. In the event of a disagreement, the active CC proceeds to check itself by programmed tests. These same tests are run in the standby CC at the same time. If the active CC decides that it is functioning correctly, then the match circuits are interrogated to determine whether the standby CC had the same test results. Ordinarily, when the active CC decides it is sick, it will turn control over to the other CC. To guard against a CC which is so sick that it cannot do this, a timer is provided in each CC which must be reset periodically. If it is not reset, it will turn control of the system over to the other CC regardless of any CC operations that may be going on at the time.

Once the malfunctioning CC has "out of service" status, the active CC requests corrective action by the maintenance craftsman on the system teletypewriter. The active CC then proceeds (in the free intervals between taking care of normal telephone traffic) to perform a diagnostic program which provides information to help the maintenance craftsman locate the trouble.

2.2 *The Diagnostic Program for the Central Control*

Each CC consists of several thousand small pluggable circuit cards. Each card contains an assembly of diodes, transistors, resistors, etc. to form a digital "building block". One card may contain up to 5 AND gates, or 5 OR gates, or one flip-flop, or up to two amplifiers, etc. Practically every malfunction which occurs during normal use is due either to failure of a component on a card, or to a bad connection between a

card and its connector. Thus, the primary problem of maintenance is to isolate a malfunction to the card involved.

To aid the maintenance craftsman in this very difficult task, the CC is provided with a large diagnostic program containing nearly a thousand tests. A typical test consists of a short sequence of operations (performed in parallel by both CC's). The matching circuits are turned on only at certain critical points during the sequence. Since the active CC has previously been found to be good, it is the standard by which we test the standby CC. Therefore, any (unintended) mismatch of information between CC's is assumed to be due to a malfunction in the standby CC. Any test whose result in the standby CC matches that of the active CC is recorded as passed, otherwise it is recorded as failed. A 0 denotes a test passed, and a 1 denotes a test failed.

The diagnostic tests are divided into eight logical phases (*A* through *H*). Within each phase the tests are numbered (in octal) from 0 to a maximum (in some phases) of 177. Phases *D* and *H* each require more than 177 tests and therefore, each is divided into two physical phases. The additional phases in *D* and *H* are denoted *DP* and *HP* (for *D* Partial and *H* Partial).

The tests of the diagnosis were organized as much as possible so that phase *A* tests the most basic equipment and phase *B* tests the next most basic equipment. The remaining phases test the remainder of the CC. In normal operation, test failures during phases *A* or *B* cause all the remaining phases to be omitted. However, it should be realized that many malfunctions in the basic equipment exercised by phases *A* and *B* do not cause test failures during phases *A* and *B*. Also many malfunctions outside the basic equipment do cause test failures during phases *A* and *B*.

Although the test results were actually presented on the system teletypewriter in a quite different manner, it is best for theoretical purposes to visualize the test results from a single pass of the diagnostic program as a long row of 0's and 1's. The first digit represents the result of the first test in phase *A*; the remaining digits represent the rest of the phase *A* tests, followed by the phase *B* tests, etc. In this paper, we shall sometimes refer to the results of a single run-through of the diagnostic program as "the test pattern" or "the pattern of test results".

2.3 *The Central Control Maintenance Dictionary*

The dictionary described here was constructed under the supervision of S. H. Tsiang, and was completed prior to our use of the data in it.

The whole dictionary project is very well described by Tsiang and Ulrich (Ref. 5).

The large size and complexity of the CC made it necessary to find a better technique than direct reasoning for using the diagnostic test results to locate the malfunctioning circuit card. It was decided to make a dictionary which shows the actual test results for a large number of possible malfunctions that may occur in the system during normal use. The only practical method of compiling such a dictionary was actually to insert the malfunctions in an operating machine and perform the diagnostic tests.

The dictionary was prepared on the Whippany Laboratories model, which was a duplicate of the system at Morris, Illinois. Approximately 50,000 malfunctions were inserted in the CC including such troubles as a shorted diode, an open diode, an open resistor, a flip-flop permanently set or reset, etc. These malfunctions were introduced into every card in the CC.

For each malfunction introduced, the results of the diagnostic tests together with the identification of the malfunction were punched on paper tape by the system. This information, which represents the raw data from which the dictionary was constructed, was transferred to a magnetic tape and sorted on an IBM 704 computer.

The data that we have just described can be visualized as a large matrix with about 50,000 rows and nearly 1,000 columns. Each row corresponds to a malfunction from which the test pattern was obtained. Each column corresponds to a particular diagnostic test. An entry of 0, for example, indicates that the malfunction on that row passed the diagnostic test for that column, while an entry of 1 indicates the malfunction failed for that test.

To facilitate looking up a particular pattern of test results, it is necessary that the rows of the matrix be sorted into some systematic order. Basically, the dictionary consists of the sorted matrix (each row represented in a condensed form) with the malfunction identification pertaining to each row. In many cases the same test pattern appeared in more than one row, that is, different malfunctions produced the same test pattern. In such cases the test pattern was listed only once for all the rows, but with all the corresponding malfunctions, so that each test pattern appeared only once in the dictionary. The dictionary, listing about 30,000 malfunctions, required 1300 pages.

The dictionary just described is a good measure of the effectiveness of the diagnostic tests. The diagnostic program was designed with two basic objectives: (*i*) to contain tests sensitive to virtually every mal-

function that might occur spontaneously in the CC and (*ii*) to produce distinct test failure patterns that identify each malfunction and distinguish it from all others. The data collected yielded the startling fact that the diagnostic programs fell quite short of the first objective. Of the 50,000 malfunctions inserted, approximately 20,000 of them resulted in the all 0's test pattern (all tests passing).

There are a variety of causes for these undetected malfunctions. Some are due to components which serve only a protective purpose, so that their malfunction is observable only in the presence of the trouble being protected against. Others were due to auxiliary equipment which was installed but was neither used, tested, nor covered by the diagnostic programs. Other undetected malfunctions in the CC could have been detected with more diagnostic program or more hardware. No doubt other causes operated as well.

However, we view the problem of undetected malfunctions as a solvable problem which is outside our scope of interest. We presume that in other digital machines to which our ideas might apply, this problem will have been solved. We restrict our attention to the 30,000 detected malfunctions.

The second objective was met fairly satisfactorily (for the detected malfunctions). The extent to which it was met is measured by how many circuit cards are listed in the dictionary for each pattern. The average number was less than 3, which is quite satisfactory but a few patterns had hundreds of associated circuit cards, which is unfortunate.

2.4 *An Evaluation of the CC Dictionary*

The CC dictionary was intended to be used by finding in the dictionary the exact test failure pattern obtained in the field. The dictionary entry indicates the list of circuit packages to replace. When this works, it is an easy method of locating malfunctions. Unfortunately, two facts complicate this technique. Many malfunctions yield different test patterns on different occasions. Other malfunctions, though always yielding the same test pattern in the field, yield a pattern that does not appear in the dictionary.

The major reason that test patterns differ from one test run to the next is that the test runs start with the machine in different configurations. Most notably, various flip-flops may have different states. Although the test program attempts to place the machine in a uniform initial state before each test sequence, the malfunction may prevent this being done. One reason that field test patterns may consistently

differ from the dictionary pattern are the intermachine differences, both in electrical parameter values due to manufacturing variability and in logic due to the inevitable program and hardware modification required for dictionary construction.

After the dictionary was prepared, many informal experiments were performed to test its efficiency. Much practical knowledge was gained as to the detailed manner in which test results might differ from the dictionary test results for the same malfunction. This information was partly formalized by S. H. Tsiang (in an unpublished memorandum) who developed "empirical rules" for use with the dictionary. When the test pattern was found in the dictionary but replacement of the listed circuit cards failed to correct the malfunction, or when the pattern was not found in the dictionary at all, these rules were used to alter the pattern to a likely candidate. Use of these rules significantly improved the use of the dictionary.

One formal experiment to evaluate the CC dictionary was performed on the Morris, Illinois model by R. N. Breed and described in an unpublished memorandum. Approximately 600 faults were selected for this evaluation. Malfunctions of various types were chosen in proportion to their frequency of occurrence in certain failure records, and so as to represent all parts of the CC. However, malfunctions known to produce no test failures were avoided.

Of the 600 malfunctions, 30 were eliminated (for unstated reasons) at the time the data was collected, 47 more were eliminated from the summary figures in the memorandum because they "probably could have been found by diagnosis of some unit other than the central control". These 47 malfunctions would have belonged to the last two categories below. The remaining 523 malfunctions were divided into categories as shown:

47 % findable with perfect match to dictionary results,
13 % findable using the empirical rules,
21 % not findable because all the tests were passed,
19 % not findable, even with the aid of the empirical rules.

The third category is of interest to logic circuit designers and diagnostic programmers. Our interest is primarily with the fourth and second categories. Our methods offer real possibilities for identifying the malfunctions responsible for otherwise mysterious test patterns, and for more easily identifying malfunctions which would otherwise require the use of complex empirical rules.

III. DATA

3.1 *The Data Used in our Study*

Of the 30,000 malfunctions with test patterns not all 0, about 10,000 malfunctions failed one or more tests in phase A and/or phase B. When the dictionary was originally prepared, it was observed that malfunctions of this sort, besides consistently failing tests in phase A and/or B, generally failed a great many tests in the other phases, and did so in an inconsistent way. For this reason the CC dictionary suppresses the test results of the other phases for this group of test patterns.

As we wished to reduce the scope of our study (to cut down computation time and cut down the bulk of printed results), we removed these 10,000 malfunctions (failing phase A and/or B) from the data.

However, we were eager to reduce the bulk of the data still further. We realized that for test patterns with few test failures, our methods offer less potential advantage than for patterns with many failures. This is true for several reasons. The direct deductive method tends to work well for test patterns with few test failures. Furthermore, so very many other malfunctions yield test patterns within a very small WHD of the observed pattern that it may not be practical to use the dictionary for nonexact matching in the way we shall discuss.

We found that the patterns with 3 or less test failures constituted about half of the remaining 20,000 malfunctions. Certain of our calculations would have been distorted by 2800 malfunctions which all gave exactly the same pattern of 3 test failures (namely, phase H tests 100, 101, and 102). (These malfunctions caused the standby CC to "lock up", that is, stopped its master-clock; the tests are part of a special set of tests comprising the "CC lockup diagnosis.") To avoid this distortion and to reduce the data to a reasonable quantity, it seemed natural to restrict ourselves to patterns with 4 or more test failures.

Thus, we finally used a matrix with 10,937 rows (malfunctions). As we have eliminated test failures in phases A and B, the only columns with 1's in them can be those for phases C through H. Of these, just 657 columns actually contained 1's.

A few facts about this matrix may be of interest. Fig. 2 shows graphically the number of rows with exactly k 1's in them, as a function of k. The few rows with the greatest number of 1's in them have, respectively, more than 511, exactly 466, 441, 325, 310, 252, and 247 1's in them. Fig. 3 shows the number of columns with k 1's in them as a function of k. The few columns with the most 1's in them have, respectively, 4335,

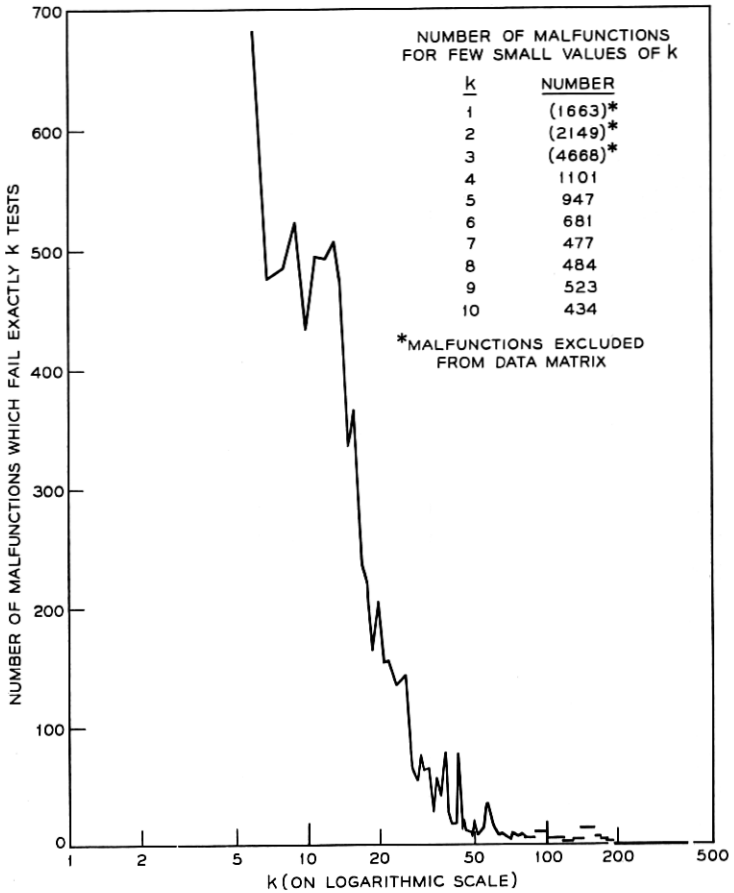

Fig. 2 — Number of malfunctions which fail exactly k tests, that is, the number of rows of the matrix with exactly k 1's. For larger k, the data have been grouped.

2978, 2711, 2653, 2383, 1933, and 1744 1's in them. The column with 4335 1's in it corresponds to test HP 61; thus, this test was failed by about 40 percent of the malfunctions in the matrix.

3.2 *The Test Weights*

We now restrict attention to the data we used in our study. It consists of a matrix with 10,937 rows and 657 columns, whose entries are 0 or 1. In the introduction we referred to weights w_i which we associated with the diagnostic tests, or in other words, with the columns of the matrix. These weights are all positive, and the largest possible value is 1.

Before describing the meaning of these weights and the formulas used to obtain them, we shall mention a few facts about the weights actually obtained from the data. Just one weight is greater than 0.999. There are 35 weights greater than 0.95. The weights in the interval from 0.25 up to 0.95 are very sparse, while below 0.25 the weights are densely but erratically distributed. A graph of the density of weights versus w is shown in Fig. 4. The brief table below summarizes the same information.

Dividing points	0.05	0.10	0.15	0.20	0.25	0.95
Number of weights	272	73	119	72	34	52	35

The smallest weight is 0.0097.

The weights are intended to reflect the extent to which the information given by the entries in one column of the matrix is independent of the information given by other columns. Thus, a column which does not at all resemble any of the other columns would have a full weight of 1, while a column which is almost the same as a great many other columns would have a very small weight.

For example, suppose 10 columns are identical. Then they all contain the same information, so it is natural to give each one a weight of 1/10

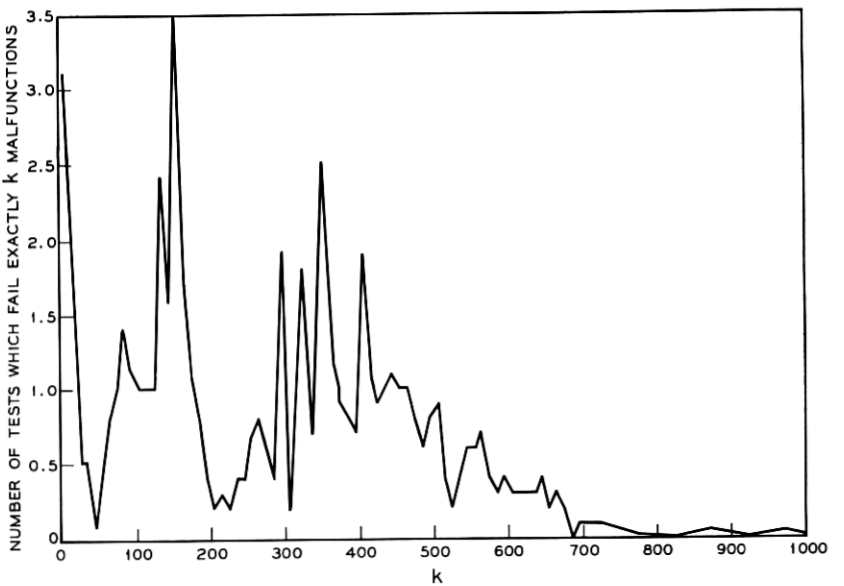

Fig. 3 — The number of tests which fail exactly k malfunctions, that is, the number of columns of the matrix with exactly k 1's in them. Curve has been smoothed by grouping 10 or 50 values of k.

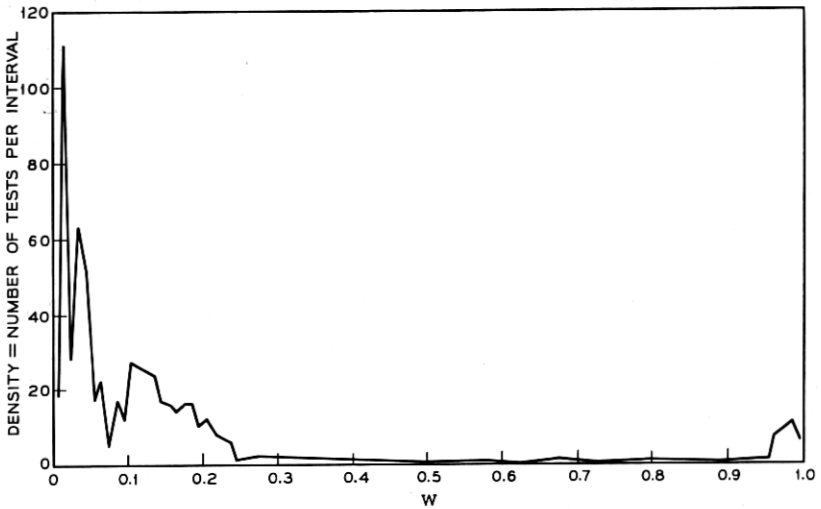

Fig. 4 — The density of test weights as a function of weight w. The vertical scale is the number of tests in an interval of 0.01 on the w axis In some cases, the curve has been smoothed over longer intervals.

to reduce the importance of that information to its proper value. However, it is not enough to consider columns which are exactly identical; in our data there are a great many columns which are almost the same and which must be taken into account.

Suppose we have a way to measure how much alike two columns are. In particular, suppose L_{ij} is the amount of likeness or similarity between columns i and j. We suppose that L_{ij} lies between 0 and 1, with $L_{ij} = 1$ if the two columns are identical, and $L_{ij} = 0$ if the two columns are not at all alike. Of course $L_{ij} = L_{ji}$. Then a natural formula for the weights is

$$w_i = \frac{1}{L_{i1} + L_{i2} + \cdots + L_{i,\,657}} = \frac{1}{\sum_j L_{ij}}.$$

For example, if columns 1 through 10 are identical to each other but do not resemble the other columns at all, then the weight of each of these columns is 1/10. (In more detail, consider say, column 7. Then $L_{7j} = 1$ for 10 values of j, and $L_{7j} = 0$ for all other values of j, so the denominator is 10.)

As another example, suppose there are just three columns (instead of 657) and that the values of L_{ij} are those given below:

		j = 1	2	3
i	1	1.0	0.5	0.1
	2	0.5	1.0	0.7
	3	0.1	0.7	1.0

Then

$$w_1 = \frac{1}{1.0 + 0.5 + 0.1} = \frac{1}{1.6} = 0.625,$$

$$w_2 = \frac{1}{0.5 + 1.0 + 0.7} = \frac{1}{2.2} = 0.455,$$

$$w_3 = \frac{1}{0.1 + 0.7 + 1.0} = \frac{1}{1.8} = 0.555.$$

How shall we measure the likeness of columns? One practical answer, which we used, is the square of the correlation coefficient between the columns. For readers who are not familiar with this widely used statistical quantity, we tell a little about it. The correlation coefficient itself lies between -1 and $+1$. It is $+1$ for two identical columns, -1 for two columns which are exact opposites, and 0 if the 1's are arranged as if they had been sprinkled randomly and independently in the two columns. The correlation coefficient has intermediate values in intermediate situations. To be very concrete, suppose the matrix has 12 rows (so each column has 12 entries). Suppose one column has three 1's and the other column has four 1's. Then the correlation coefficient depends on the number of rows in which both columns have 1's at the same time. The following table gives the actual values.

Number of 1's in common	0	1	2	3
Correlation coefficient	-0.408	0	$+0.408$	$+0.816$
Likeness value	$+0.167$	0	$+0.167$	$+0.667$

The formula for the correlation coefficient which applies to the present circumstances involves

N = the number of rows in the matrix,
n_i = the number of 1's in column i,
n_j = the number of 1's in column j,
n_{ij} = the number of 1's in common to columns i and j.

Then we have

$$\text{correlation coefficient between columns } i \text{ and } j = \frac{Nn_{ij} - n_i n_j}{\sqrt{n_i(N - n_i)} \cdot \sqrt{n_j(N - n_j)}}.$$

The suitability of this way of measuring likeness is established by its successful application to this data. However, we may justify it in part by appealing to intuition and numerical experimentation.

The correlation coefficient is $+1$ if and only if the two columns are identical. It is clearly appropriate that the likeness should be 1 in this case.

The correlation coefficient is -1 if and only if two columns are complementary, that is, if one column has 1's precisely where the other column has 0's. In this case, we may say that the two columns carry the same information even though their entries are opposite. For this situation means that one test fails just when the other passes. Clearly if we know this, it is enough to perform just one of the tests; we can predict the result of the other. Thus, it is reasonable to assign a likeness of $+1$ to this situation, as our likeness measure does.

If the 1's in the two columns appear as if they are independently located (in the statistical sense), so that knowledge of the entry in one column has no predictive power at all for the entry in the other column, then it is reasonable to assign likeness 0. This situation occurs if and only if

$$\frac{n_{ij}}{N} = \frac{n_i}{N}\frac{n_j}{N},$$

also if and only if the correlation coefficient is 0. Thus, we assign likeness 0 in this case.

This pins down the value of our likeness value for two extreme situations and one intermediate situation. If we are considering the correlation coefficient as the basis for measuring likeness, this still leaves many possibilities. For example, the likeness can be the absolute value of the correlation coefficient, the square of the correlation coefficient, any positive power of the absolute value of the correlation coefficient, to mention only a few possibilities. We experimented with various possibilities including the absolute value, the square and three functions of the correlation coefficient whose graphs are made up of straight-line segments.

Some tests are so individualistic that they surely deserve weights of

almost 1. On the other hand, there is a large cluster of perhaps 100 tests (that is, columns) which are so nearly alike that at least the most typical of them deserve weights as small as 0.01. As the absolute value formula yielded weights ranging from about 0.1 to 0.01, it was clearly inappropriate. The reason was equally clear; the effect of statistical fluctuations on the many small likenesses is cumulative and noncancelling and leads to unduly large denominators. Since there appears to be no way to arrange for cancellation, it is desirable to reduce the effect of small likenesses.

Two of the segmented-straight-line functions did this very successfully, and yielded weights ranging from virtually 1 to slightly under 0.01. Then Colin Mallows pointed out that the squared correlation coefficient lies very neatly between these two functions. When tried, the square yielded very similar weights, with the smallest one even a trifle smaller.

IV. GEOMETRY

4.1 *Weighted Hamming Distance*

Ordinary Hamming distance between two test patterns (two rows in the matrix) is just the number of places in which they differ, in other words, the number of tests for which they have different results. To calculate this, we accumulate 1 for each position in which the test patterns differ.

Weighted Hamming distance is similar except that instead of accumulating 1's we accumulate the weight associated with that position. For example, if two test patterns differ only in the results of tests 3, 5, and 17, then the WHD (weighted Hamming distance) between them is given by

$$\text{WHD} = w_3 + w_5 + w_{17} .$$

We note that this a true distance in the mathematical sense of the word. In particular, WHD satisfies the triangle inequality: the WHD between patterns 1 and 2, plus the WHD between patterns 2 and 3, is always greater than or equal to the WHD between patterns 1 and 3.

We measure the dissimilarity between malfunctions by the WHD between their test patterns. If two malfunctions yield test patterns between which the WHD is small, we consider the malfunctions similar, but if the WHD is large we consider them dissimilar. With this in mind, let us consider the intuitive meaning of the test weights. Suppose that

tests 1 through 100 are all very much like each other; that is, these tests generally fail together or pass together. This means that a test pattern will generally fail almost all or pass almost all these tests: it is unlikely that a test pattern will fail approximately half these tests. Thus, in comparing the dissimilarity of two test patterns with regard to this group of tests, the main information we get is whether they are the same or opposite. If we used ordinary Hamming distance, then test patterns which are opposite would have a distance of at least 100 just from these tests alone. Yet "same" or "opposite" on this group of tests may be no more significant than same or opposite on a single test which is an "individualist". By down-weighting like tests and using WHD, we prevent large groups of like tests from swamping the information contained in tests which are very "individualistic".

Now it is true that for test patterns which are the same for most of the tests in this large group, the few tests in the group which yield different results may be very significant. We view this as fine-grain information, however, in contrast to the broadbrush information contained in the group as a whole. We do not know of any practical way to have the WHD based on a single set of weights reflect both kinds of information.

Nevertheless, there is a way within our general scheme to make use of this fine-grain information, though it is not an idea which we have actually attempted. The technique is this. We would collect together some group of test patterns in our matrix which are fairly similar to each other; for example, we might arbitrarily pick some test pattern as the "center", then form the group of all test patterns in the matrix which are within some fixed WHD of the center. Presumably we would arrange things so as to get a group of several hundred test patterns. Using this group of test patterns we would have a new data matrix (actually a submatrix of the original, with all the columns but only a selected set of rows). Using this submatrix we would calculate new weights, using the same formulas but applying them to this new smaller matrix. We could call these "local" weights as they apply only to this one local group of test patterns *when compared with each other*.

These local weights could differ very greatly from the original "global" test weights. The global weight of the test could be high or low, independently of the local weight. Furthermore, the local weights for this local group of test patterns might be entirely different from the local weights we would derive from some other local group of test patterns.

Using the local WHD (based on local weights) to measure dissimilarity between test patterns in some group is probably a good way to make use of the fine-grain information.

4.2 *The Geometric Model*

Once a meaningful concept of distance between test patterns exists (such as WHD), it is natural to ask whether these distances can be realized in a geometric model. For example, can we represent each pattern by a single point in the plane, in such a way that the ordinary Euclidean distance (ED) between the points is equal to the WHD between the corresponding patterns?

First, we remark that there is nothing inherent in the concept of distance which will force this to happen. Thus, if this happens it tells us something about the data. It tells us that in some sense or other the test patterns form a two-dimensional set. What this means is not clear. But, that it means something important is indicated by the tremendous information compression which is achieved.

To understand this, let us suppose that we have 10,000 test patterns. Between these test patterns there are

$$(10{,}000)(9{,}999)/2 \doteq 50{,}000{,}000$$

WHD's. If we can represent each test pattern in the plane, that requires two coordinates per pattern, so that we require 20,000 numbers to represent the patterns. Since the ED's are of course computable from these 20,000 coordinates (by the usual formula learned in high school), and since the ED's equal the WHD's, we have compressed the information from 50,000,000 numbers into 20,000 numbers. In other words, from the 20,000 numbers required to represent the patterns, we can recover by simple arithmetic all the 50,000,000 WHD's.

Any model which achieves such compression is bound to be useful, for it permits us to handle information in a much more concentrated manner. Beyond its direct utility, however, any model which achieves such compression is trying to tell us something about the data.

(The classic example of this are the 20 years worth of extremely accurate astronomical observations made by Tycho Brahe in the sixteenth century. Kepler found a model consisting of his three famous laws from which it was possible to explain these observations. Basically, his model represented each planet's motion by an ellipse. Thus, using his model it was possible to explain all of Tycho's observations of one planet from 12 numbers — 6 for the planet's motion and 6 for the earth's motion. It is clear that the enormous compression of information in itself was useful in this situation. It is also clear that the model was trying to say something, however, even if it took Newton to hear it.)

Without trying to compare ourselves to Kepler, we feel that the in-

formation compression of our model is a striking phenomenon which demands investigation, and must produce something of value.

We do not get a representation by points in the plane, nor by points in three-dimensional space, but only by points in six-dimensional space. We can represent each pattern by six coordinates in such a way that the ED's are approximately equal to the WHD's. This applies to not quite all the 10,937 patterns in our matrix — there were three exceptions that did not fit. (These three exceptions probably result from malfunctions which in fact cause A or B phase test failures, but were not excluded from our data due to some recording failure which dropped the A and B phase results.)

We notice first that the compression of information goes down as the number of dimensions goes up. For 10,000 patterns represented in 6 dimensions, the same 50,000,000 WHD's are recoverable not from 20,000 coordinates but from 60,000 coordinates. The compression is slightly less.

We notice second that the value of the compression depends on how good the approximation is. The more accurately the ED's represent the WHD's, the more valuable the compression is. In our case the typical difference between matching ED and WHD is about 7 percent. More exactly,

$$\sqrt{\sum (\text{ED} - \text{WHD})^2 / \sum \text{WHD}^2}$$

is in the neighborhood of 7 percent. Though not striking, it seems entirely adequate when matched with the compression we have.

A scatter diagram of WHD's against ED's is shown in Fig. 5. Each point displays the WHD and ED between one pair of malfunctions. The figure contains almost 5000 points, corresponding to all possible pairs from among the list of 100 malfunctions referred to in the next section, and is impressive testimony to how well the ED's match the WHD's.

4.3 *How to Compute the Geometric Model*

In this section, we describe the necessary computation very briefly, just enough to take the mystery out of it. Suppose then that we wish to place 10,000 points in some space — we will use the plane to make it easier to visualize, though exactly the same procedure works in three-dimensional space or six-dimensional space. The information we have consists of the approximately 50,000,000 WHD's between these points.

We start by placing the 10,000 points in the plane in any arbitrary

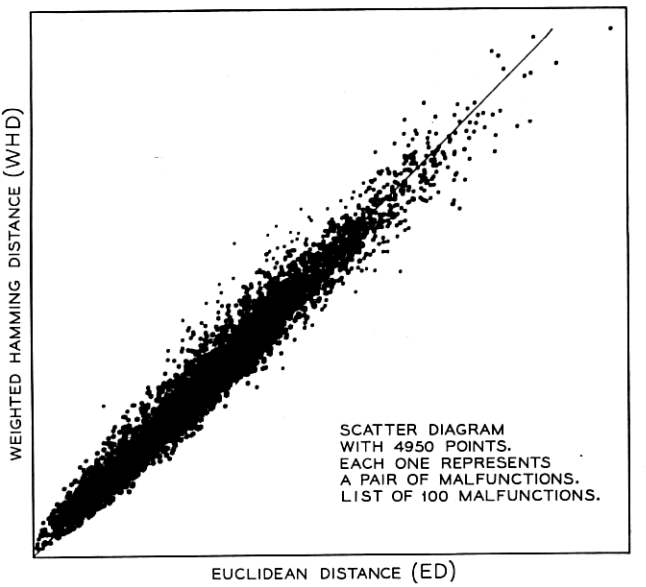

Fig. 5 — Scatter diagram of weighted Hamming distance (WHD) against Euclidean distance (ED). Each point represents one pair of malfunctions. All pairs from among 100 malfunctions are displayed.

configuration, and pinning them down so that they cannot slide about. Next, we buy 50,000,000 ideal springs. These springs are massless and all have the same restoring force ratio (Young's modulus) — let us suppose the common value is 1. However, the springs are all of different lengths. In fact, each spring has a length equal to one of the WHD's. Now we fasten each spring between the two appropriate points. Thus, between points i and j we attach the spring whose unstretched length is the WHD between patterns i and j. Of course it is necessary to stretch or compress the springs, and we do so as required. Naturally these ideal springs do not buckle when compressed, and furthermore several of these springs can occupy the same space at the same time, so that we do not need to worry about how they cross each other.

After all the springs are attached, we suddenly pull out all 10,000 pins, permitting the points to slide about (but only on the plane — we do not permit them to fly up in the third dimension). If there is some dissipative force, such as air resistance or friction, the springs and points will eventually come to rest. By the laws of physics, they will come to rest at a minimum energy configuration, that is, a configuration at which the potential energy stored in the springs is a minimum.

What can we say about this configuration? The potential energy in each spring, according to our assumptions, is $(ED - WHD)^2$. The total energy is the sum of all these. Thus, the final resting configuration is one which minimizes this sum, or almost the same which minimizes

$$\sqrt{\sum (ED - WHD)^2 / \sum WHD^2}.$$

Our computation is basically an imitation of the spring motion. We also start with an arbitrary configuration. Then we figure the net force on each point exerted by the springs, and move all the points where they would be a short time later. We again figure the net forces, and again move the points. After enough repetitions, the net forces reach 0, and we know that we have reached the minimum energy configuration.

This is a good intuitive description of what we do, but we would not like to leave the impression that our computation is in any way unrigorous. In the language of numerical computation, we are seeking to minimize the expression given above. To do so, we perform an iterative process known as the method of gradients (or the method of steepest descent). Thus, we start with an arbitrary configuration, and compute the (negative) gradient, which is just the same thing in this case as the net forces on all the points. We move a little in the direction of the (negative) gradient — that is, just the motion along the force vectors. Then we again calculate the gradient and again move. When the gradient is zero, we have reached a minimum.

Of course, we cannot really perform this computation as described on all 10,000 points at once. To perform one single movement would require nearly 30 hours (on the IBM 7090), even if we could manage to keep all the numbers required in the internal memory.

To get around this difficulty we hoisted ourselves by our own boot straps. We started with 33 points, and performed the computation exactly as described. Then we "pinned down" these 33 points, and introduced 67 more points, and 33×67 "springs." During this computation only the 67 new points were allowed to move. Thus, we had 100 points located, though not quite perfectly. We then performed the original computation on these 100 points, starting with the configuration we had already achieved. This just moved the 100 points a little — it was a "polishing" operation. We then picked a set of 20 from these 100 points, in such a way that these 20 are well spaced over the region of space covered by the 100 points, with no pair of the 20 points too close together. We "pinned down" these 20 points very firmly, and introduced $20 \times 10{,}000$ "springs". During this computation only the 10,000

new points were allowed to move. This located all 10,000 points, though not quite perfectly. We simply tolerate this imperfection (though there are practical ways to reduce it if it should seem intolerable).

The reason this computational scheme is practical is that when we pin down the 20 points and introduce 10,000 new points, we can handle them one by one. Thus, at any time we only need to deal with 20 fixed points, one movable point, and 20 springs from the movable point to the fixed points. After the one movable point comes to rest, we remove it before introducing the next.

4.4 *Why Six Dimensions?*

As we have noted, we do not achieve perfect equality between the ED's and WHD's. The typical difference is about 7 percent. Obviously in seven dimensions we can reduce this figure while in five dimensions it would be larger. The more dimensions, the better we can make the ED's match the WHD's.

It would be possible to draw a curve of the typical error versus the dimension. (We would put dimension on the horizontal scale and error on the vertical scale.) We would then get a descending curve. On this basis, the more dimensions the better. On the other hand, from the point of view of information compression, the more dimensions the worse. Thus, we wish to strike a balance.

The principle of parsimony advocates obtaining the highest compression possible while retaining "satisfactory fit". In other words, use as few dimensions as possible with the typical error satisfactorily small.

Actually, we did not draw such a curve with the complete data. We did draw such a curve, however, with a small sample of the data (using a similar but more complicated model than the one we have described). For this sample, we computed the typical error for 2, 4, 6, and 8 dimensions. The typical error in 4 dimensions seemed too large, while in 6 dimensions it was tolerable. Going to 8 dimensions produced little reduction. Thus, we decided to use 6 dimensions.

Another reason for using 6 dimensions was the fact that when W. Thomis used a different scheme for coordinatizing faults, he needed 6 coordinates, which seemed to point to 6 dimensionality also. It is clear from this discussion, however, that 5 or 7 dimensions would also be satisfactory, but 4 or 8 would probably not be. Though it is difficult to say which is best, we see that 6 dimensions represents a reasonable compromise value for these data.

4.5 *Tests and Hyperplanes*

The geometric model may have considerably more meaning than we have indicated so far. It may be possible to represent tests as well as malfunctions. We do not represent tests by points, however, but by "hyperplanes".

In general, a hyperplane is an infinite flat cut which divides space into two parts. In three-dimensional space a hyperplane is an ordinary flat plane.

In two-dimensional space, that is, in the plane, a hyperplane is a straight line. In one-dimensional space, that is, in the line, a hyperplane consists of a single point. In n-dimensional space, a hyperplane is an $(n - 1)$-dimensional flat subspace. (Hyperspace is an old-fashioned name for a higher dimensional space, and the hyperplane is the analogue in these spaces of the plane in three dimensions.)

In the following geometric discussion it would be well to have a mental picture of either two or three-dimensional space. Each hyperplane is then visualized as a line or a plane.

Thus, suppose we have space (actually six dimensional but visualized as two or three-dimensional). In it are 10,000 points representing malfunctions. Now pick some particular test. Every malfunction which fails this test we color red; there are relatively few of them. Every malfunction which passes this test we color black; these are the majority of malfunctions. How are the red points situated? Are they scattered among the black ones?

There is reason to believe that in most cases the red points and the black points may be separated by a hyperplane. That is, the red points and the black points are not all mixed up. If space is two-dimensional, this means that a straight line can be drawn with the red points on one side and the black points on the other. If space is three dimensional, then a plane exists with the red points all on one side and the black all on the other.

In a sample from the Morris data involving 27 malfunctions and about 200 tests placed in 6 dimensions, we found this to be true. In some cases, the hyperplane did not quite perfectly separate the two kinds of points; a few points would be slightly on the wrong side, but the amount by which the points were on the wrong side was extremely small.

We believe that most of the tests would be representable as hyperplanes in the main body of data analyzed. If a few tests are not representable, that would probably say something interesting about these tests. Among other things, it might suggest reducing their weight.

If we suppose that the tests can, in fact, be represented by hyperplanes, then we can calculate the information compression of the model in a different way. The original data consists of about

$$10{,}000 \times 650 \doteq 6{,}500{,}000$$

bits. To represent both the malfunctions and the tests in 6 dimensions requires about

$$(10{,}000 + 650) \times 6 \doteq 64{,}000$$

numbers. From these numbers we can reconstruct the original data (though not perfectly), for to find whether a particular bit is 0 or 1 we merely need to check which side of some hyperplane some point lies on. The imperfections result from the fact that the hyperplanes from some tests do not perfectly separate the malfunctions which pass from those which fail.

From this viewpoint, the information compression consists of representing 6,500,000 bits by 64,000 numbers. This viewpoint probably provides a more meaningful measure than the simpler one presented before.

4.6 *Utility of the Geometric Model*

There are several kinds of utility for the geometric model. One kind is theoretical and long range. By examining the data in the model, we hope to learn something about the structure of the data. It is basic procedure in data analysis to look at the data with one's common sense on the alert. Where the data can be represented in compact form, this is much more useful.

Another utility of the geometric model is very immediate. To have the malfunctions represented by coordinates simplifies the process of finding nearby malfunctions. To illustrate this most clearly, let us suppose for the moment that the malfunctions could be represented in two dimensions instead of six. Imagine the malfunctions placed on a "map". This would resemble a photograph of the starry night-sky. Now suppose a new malfunction is to be identified. We would calculate its coordinates, plot its position on the map, and look for the nearest few points. Suppose on the other hand that we wished to find the nearest few points without the aid of the representation in two dimensions. In principle this is easy enough. It is only necessary to run through every malfunction in the dictionary one by one, and compute its WHD from the unknown malfunction, and finally pick out the smallest few WHD's. Computationally,

however, this is very much more difficult than use of the map. We see from this that the map very much simplifies the computation necessary to pick out the nearest few malfunctions.

Unfortunately we cannot use the map in six dimensions. Other techniques relying on the coordinates, however, are available. For example, we can cut space up into small cells, and list the malfunctions which occur in each cell. Then to find the nearest malfunctions to an unknown one, we look in the same cell and the neighboring cells.

We may summarize this value of the geometric model as reducing the computation required to select nearby malfunctions.

V. CLUSTERS

5.1 *The Region Containing all Malfunctions*

It will be helpful to know something about the "galaxy" of malfunctions, that is, the region of 6-dimensional space in which the 10,937 malfunctions lie. The "healthy machine" (no malfunction, or a malfunction which yields the test pattern of no failures) corresponds to a point with coordinates approximately

$$1.0, \quad 0.0, \quad 0.7, \quad 0.1, \quad -0.8, \quad 1.3.$$

Rounded off to the nearest integer, these are

$$1 \quad 0 \quad 1 \quad 0 \quad -1 \quad 1.$$

This appears to be fairly near the edge of the "galaxy". The center of the galaxy is approximately at

$$2 \quad 0 \quad 2 \quad 0 \quad -1 \quad 1.$$

(By center, we mean the center of gravity, or average position.) If we exclude seven outlying malfunctions from consideration, then the extreme values of each coordinate are

minima	−3	−9	−4	−8	−8	−4;
maxima	11	5	10	9	5	8.

Thus, the range of each coordinate is roughly 14. Of course, the malfunctions are not at all evenly scattered. A tremendously heavy concentration exists near the "healthy machine". Other dense spots also exist.

To further study the distribution of the malfunctions in six-dimensional space, we split space up into cells of uniform size and shape. (We avoided cubic cells because in six dimensions the corners of the cube

"stick out" rather far. Instead, we used the so-called "Voronoi regions" associated with the "body-centered cubic lattice". Each cell can be thought of as a six-dimensional cube with the corners chopped off.) There are 417 cells which contain malfunctions. Thus, the average "populated" cell contains about 24 malfunctions. The cell containing the healthy machine, however, contains 1883 malfunctions. Altogether three cells contain more than a 1000 malfunctions each, while 162 cells contain only a single malfunction each. The median number of malfunctions per populated cell is 2.

The Euclidean distance of a malfunction from the healthy machine is a measure of how severe the malfunction is. It is interesting to compare this measure with the more primitive measure consisting simply of the number of tests failed. In Fig. 6 there is a little circle for each cell. The horizontal coordinate is the Euclidean distance of the cell center from the healthy machine. The vertical coordinate is the average number of tests failed for the malfunctions in the cell. The dense region displays a definite relationship, even though the great scatter shows that it is a loose one.

5.2 *Clusters of Malfunctions*

Suppose you had a map of a city, and a list of the people who live there, together with various census information — address, income, national origin, age, education, etc. Suppose you wished to understand the social structure of the city. One obvious approach would be to identify different neighborhoods. You might find that one neighborhood had mostly high income residents, another might contain people mostly of one national origin, near a university you might find many people with higher education, and so forth. You would be seeking to identify clusters of people who live near each other and who share some common characteristic.

We face exactly this situation. We have a six-dimensional "map", and we have a list of malfunctions. About each malfunction we know whether it is a diode, amplifier, or flip-flop trouble, etc. Also, we know the specific nature of the trouble: what circuit it is in, how it operates there, and so forth. It is natural to look for clusters of malfunctions which are near each other in space (that is, which have similar test patterns) and which share some circuit characteristic.

One reason for seeking clusters of malfunctions is to learn how malfunctions affect the operation of the system. We may find that apparently similar malfunctions (for example, flip-flops in a single register which are stuck in the 1 position) do not form a cluster, that they produce

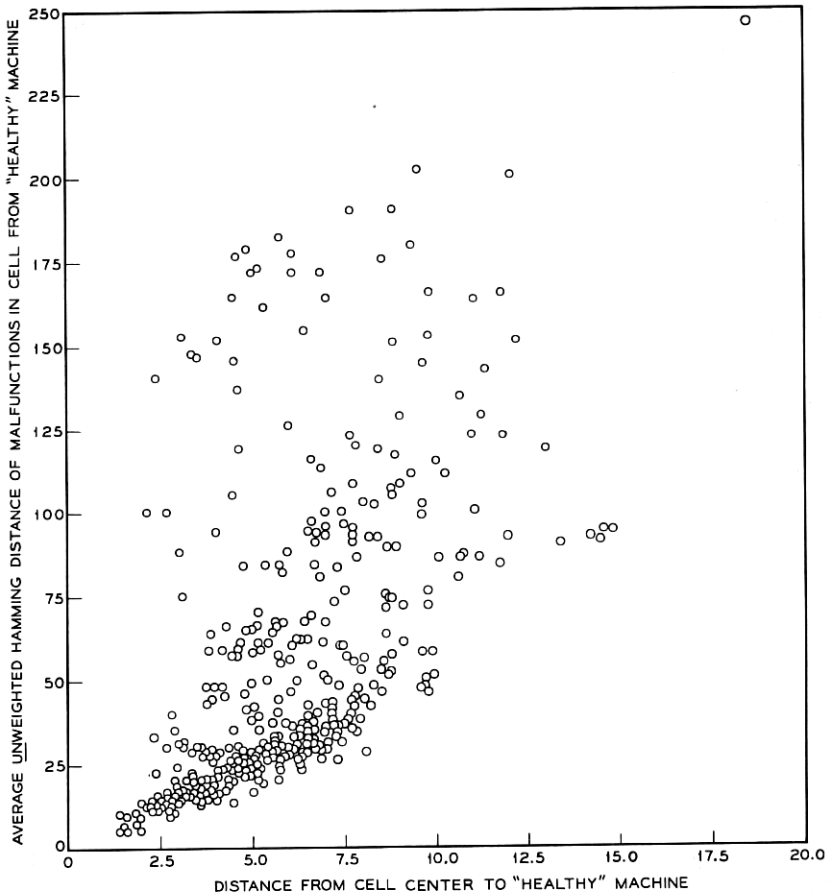

Fig. 6 — Each circle represents one of the 417 populated cells in 6-dimensional space.

very different test patterns. When we inquire into why this happens, we learn something about the nature of the system (and of the diagnostic test program). On the other hand, we may find that apparently rather different malfunctions produce very similar results. For example, a flip-flop stuck in the 1 position may yield very much the same test pattern that the same flip-flop stuck in the 0 position does (or it may not — we have observed both situations frequently). If these two malfunctions are near each other, then we learn that the significant aspect of these malfunctions is merely that this particular flip-flop is out of order, and that the precise nature of its failure is not important.

Of course, whether a malfunction is in a diode or in a resistor (say) has no direct bearing on its position in space, because its position results solely from the diagnostic test data, which, in turn, reflect its effect on the overall operation of the machine.

A cluster generally occupies a region of space without sharp boundaries. The malfunctions of the cluster are heavily concentrated at the center of the region, more lightly spread further out, and may sprinkle themselves out to a considerable distance. Thus, the boundaries of the region cannot be precisely located. It often makes sense to talk about the center of the cluster, however, which means the point of high concentration.

In a city, different clusters of people may overlap. For this reason, one neighborhood may contain, say, two nationality groups together with a sprinkling of artists. In the same way, different clusters of malfunctions often overlap, so the one small region may contain a mixture of malfunctions from several clusters. Occasionally, a cluster may totally dominate the region it occupies, so that practically every malfunction in its region belongs to it.

In the most common case, when clusters overlap, there should be some explainable reason. We discuss some cases of this sort.

Sometimes a very narrowly defined cluster may be a subcluster to a more broadly defined one. For example, we might have a small cluster of malfunctions whose common circuit characteristic is that they hold a particular wire down to a low voltage (prevent the lead from carrying the digit 1) under certain logical conditions (not necessarily identical among the malfunctions of the cluster). If this wire is one of a related group of wires, we may be dealing with a subcluster of a larger cluster of similar malfunctions involving any wire in the group.

5.3 *How Clusters are Found*

We wish to emphasize that clusters are not to be found by following preconceived notions as to which malfunctions resemble which other malfunctions. Instead one must try to look at the data with an open mind, or listen to what the data is trying to say.

Concretely, the procedure used was to examine a small region of space which contains not too many malfunctions. (When the data was still unfamiliar, we chose to examine regions with only 5 or 10 malfunctions, though later when our procedures improved we could handle many more.) We analyzed in detail the effects on the circuitry of every malfunction in this region. We then asked ourselves what common element there was to all or most of the malfunctions involved. If we found what

appeared to be a common element, we then traced out by means of the circuit diagrams all of the malfunctions which shared this common element, and noted their locations. If we found these to lie in a single compact region, we considered that we had indeed identified a cluster of malfunctions. Of course the region involved would include the smaller region from which we started. On the other hand, if we found the malfunctions with these characteristics to lie in several distinct regions, only one of which contained the original region, we knew that the malfunctions formed not one but several clusters. In this case, it was necessary to ask what characteristics differentiated the malfunctions in different regions.

Having identified a cluster as above, we did not always rest content with its description. We examined other malfunctions which lie in its region and asked whether a broader definition of the common characteristics would include some of these (without including malfunctions in other regions). Thus, by a process of referring back and forth between spatial locations and circuit effects, we arrived at brief meaningful descriptions for clusters.

5.4 *The Clusters We Found*

We have identified and described 23 clusters. It would require too much space to discuss them all, so we just illustrate our results briefly by a few examples. (Although specific circuits are named to avoid vagueness, readers unacquainted with the circuits of the CC should have no difficulty following the discussion.) A few more clusters are described in the appendix to illustrate some other aspects.

The 23 clusters each contain from 6 up to about 350 malfunctions. The median number of malfunctions is 65, and the quartiles are 27 and 220.

One cluster of 58 malfunctions is associated with two intertwined circuits which are called "Add 1 C" and "Add 1 D". (These circuits are used to add 1 to the C and D addresses in an instruction.) In Fig. 7 we show this cluster geometrically. As coordinates 3 and 6 vary most within this particular cluster, we use them to display the 57 points.

In six dimensions, the center of this cluster is approximately at $(1\frac{1}{2}, -1, 2, \frac{1}{2}, -1, 2)$. Extracting coordinates 3 and 6, we see that the center of the displayed set of points should be at (2, 2), which indeed it is. In the figure, we see that the points lie along a straight line of slope about $\frac{1}{3}$. (Experienced statisticians will cover up the 3 or 4 most deviant points to strengthen the visual impression, knowing that this aids the eye in forming a more valid impression of the goodness of fit. This is because the eye weights isolated points more heavily than points in a dense re-

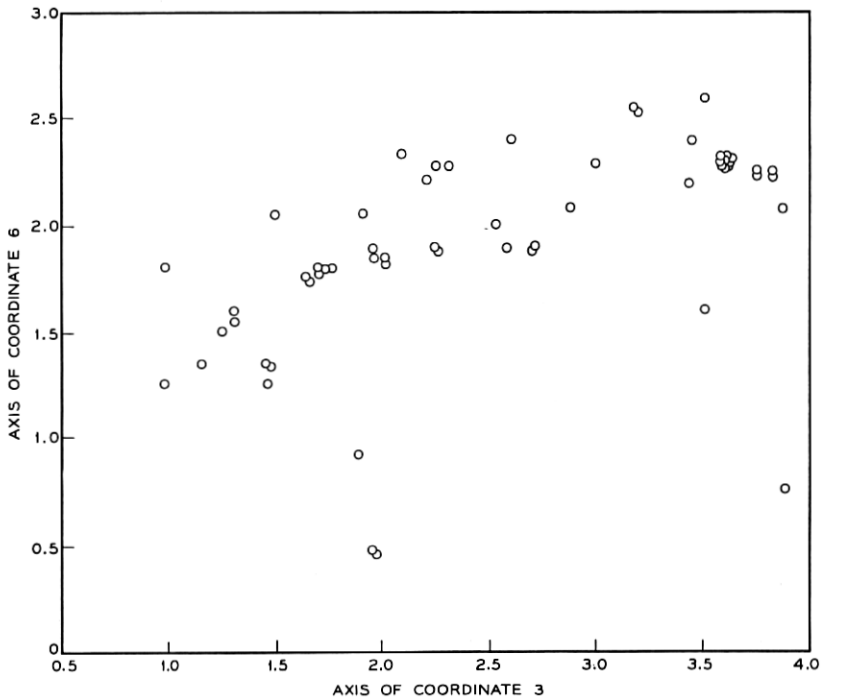

Fig. 7 — The 57 malfunctions of the "Add 1C Add 1D" cluster, displayed in the plane of coordinates 3 and 6.

gion, whereas equal weight should be attached to all points.) In fact, in six dimensions, the points lie more or less along a straight line. While we do not know the significance of this, it tends to indicate that the cluster has some internal structure.

The clusters are generally associated with "actions" rather than with circuits. By this we mean that the malfunctions in a cluster are often spread over many circuits which are considered as quite separate functional components by the circuit designers. The malfunctions in the cluster, however, always have their major disruptive effect on what circuit designers would consider as a specific action. The cluster above is unusual in this respect because it can be interpreted either way. The following examples are more typical.

One action consists of reading one or two bits from the BGS (outside the CC) and placing them in one or two flip-flops of the access register (in the CC). A cluster of about 230 malfunctions is associated with this action. The most typical malfunctions in this cluster cause the bits to be

placed in extra flip-flops of the register, or to be placed in the wrong flip-flops, or not to be placed in any flip-flops at all, under various conditions. The malfunctions in this cluster are often in one circuit (the CD memory), but occur in several other circuits as well.

Another action consists of reading one or two bits from the BGS and making a decision which depends on their values. A different cluster of about 220 malfunctions is associated with this action. The malfunctions in this cluster are scattered over many circuits and are of diverse types. Within this cluster we could pick out three subclusters, associated with much more specific actions. One such action consists of reading a 0 from the physical BGS tube numbered 0; the associated cluster has about 33 malfunctions. Another action consists of placing a bit from the BGS into a flip-flop (in the CC) called BG0, which holds it temporarily; the associated cluster has about 27 malfunctions. Another action consists of pulsing a lead called D06 from the D translator; the associated cluster has about 23 malfunctions. (The reason that this subcluster belongs in this cluster is too complex to explain here.)

VI. BY-PRODUCTS

6.1 *The Main Reason for Inconsistent Diagnostic Patterns*

During the cluster analysis we discovered several interesting by-products. The most significant one is the main reason for inconsistent diagnostic results.

It was discovered very early by those making the dictionary that the same malfunction could produce quite different diagnostic results on different occasions, that is, the diagnostic results are inconsistent from one occasion to another. There is great variability among malfunctions in this respect. Some are very badly inconsistent, and were never observed to produce precisely the same diagnostic pattern twice. Other malfunctions are very stable, and are never known to produce any inconsistencies at all. Also, there is great variability among the diagnostic tests. Some participate in many inconsistencies, others in none.

There has been much speculation as to the cause of these inconsistencies, and many possible explanations have been offered. However, it has been difficult to decide which explanations actually are correct.

We analyzed large numbers of recorded inconsistencies in detail. We believe that we have established the dominant reason for the actual observed inconsistencies. It is the differences in the state of the CC at the time the diagnostic test is performed. One source of such differences is externally controlled flip-flops which signal the state of external

circuits (for example, whether the ringing signal is on or off). Still another source consists of flip-flops which the CC attempts to initialize to a certain state before diagnosis but which are not actually initialized due to the malfunction. Since diagnosis is interspersed with the normal processing of telephone traffic, failure to initialize a flip-flop means that its value at the start of the diagnostic sequence will vary in an unpredictable manner from one diagnostic run-through to another.

During construction of the dictionary, the CC had no telephone traffic to process. Moreover, the dictionary making program was present. These two factors, operating through the sources of inconsistency just mentioned, caused some fairly regular differences between the dictionary patterns and the field test patterns. For example, during dictionary construction many externally controlled flip-flops did not ever change state because the corresponding circuits were not used.

These observations do not solve the program of how to handle inconsistent diagnostic patterns, but they do perhaps provide a framework within which it is easier to attack the problem.

6.2 *Three Incidental Discoveries*

We have suggested that browsing through the dictionary data can reveal unexpected conclusions, *if* the browsing is facilitated by methods which permit this enormous body of data to be examined incisively. Our geometric model is one such method. We briefly mention three easily describable discoveries by way of example.

One incidental discovery was that the relay point which simulated a particular shorted diode in a particular AND gate (malfunction 24 in package F52618) developed a high resistance during much of the time that the dictionary data was being taken. Thus, the malfunction closely resembled an open diode, rather than a closed diode.

A second discovery was an error in the test program by which the data was developed. In particular, the three instructions which constitute test HP11 were misarranged.

A third discovery was that one of the (hardware) malfunction simulators stepped through the malfunctions in the wrong order. This particular simulator could be substituted for any one of the current-supplying OR gate cards. It had the capability of acting as a properly functioning card, an unplugged card, or a card with either a shorted diode, an open diode, or an open resistor on any one input lead.

For example, for card type F52626, which consists of 3 two-input gates and has 6 input leads, Table I shows both the intended order and the actual order in which the 19 malfunctions were stepped through.

TABLE I

Input lead	Nature of malfunction	Should have been malfunction number	But was actually malfunction number
	unplugged card	1	1
1	shorted diode	2	4
	open diode	3	2 or 3
	open resistor	4	2 or 3
2	shorted diode	5	6
	open diode	6	5 or 7
	open resistor	7	5 or 7
3	shorted diode	8	10
	open diode	9	8 or 9
	open resistor	10	8 or 9
4	shorted diode	11	14
	open diode	12	11 or 12
	open resistor	13	11 or 12
5	shorted diode	14	18
	open diode	15	13 or 15
	open resistor	16	13 or 15
6	shorted diode	17	19
	open diode	18	16 or 17
	open resistor	19	16 or 17

6.3 *"Forward" and "Backward" Acting Malfunctions*

It seems worthwhile here to emphasize the important difference between "forward" and "backward" acting malfunctions. While this distinction is not original with us, its importance was made abundantly clear by the great complexity of test results for backward acting malfunctions versus the relative simplicity for forward acting malfunctions.

Suppose information flows between circuits A, B, C, and X as shown:

$$A \rightarrow \begin{matrix} \nearrow X \rightarrow B. \\ \searrow C \end{matrix}$$

Suppose circuit X has a malfunction. If this malfunction causes B to misoperate, we say that the malfunction acts "forward"; if it causes A or C to misoperate, we say that it acts "backward". For example, in the Morris CC circuitry, an open diode in an AND gate was forward-acting, as it only altered the output of the AND gate. However, a shorted diode in an AND gate was often backward-acting (as well as forward-acting) depending on the circuit configuration, as it could prevent the

input lead voltage from rising, thereby preventing other branches of the input lead from performing their intended functions.

It hardly seems possible to design a circuit which avoids forward-acting malfunctions. For if the information which flows along normal paths is wrong, the recipient circuit cannot be expected to act as intended.

On the other hand, one might hope to build a circuit in which backward-acting malfunctions are kept to a minimum. (Though, of course, this feature might have to be balanced against other desirable features.) Avoidance of backward-acting malfunctions would surely simplify the diagnostic problem greatly, not only during normal maintenance, but also during the process of debugging the first model of the machine.

VII. CONCLUSIONS

We list several conclusions which are surely true for the data described in this paper, and which might well hold for similar diagnostic data from other digital machines.

(*i*) If different diagnostic tests are weighted suitably, then the weighted Hamming distance between test patterns is a meaningful measure of dissimilarity between malfunctions.

(*ii*) It is possible to represent the malfunctions geometrically as points in a space of low dimensions in such a way that the Euclidean distances between the points approximate the weighted Hamming distances between the corresponding patterns.

(*iii*) There may also be a geometric representation of the diagnostic tests as hyperplanes (flat cuts) in the same low dimensional space, such that each hyperplane separates most of the malfunctions which fail from the malfunctions which pass the corresponding test.

(*iv*) Representation of diagnostic patterns as points in low dimensional space offers immediate possibilities as a tool for locating malfunctions.

(*v*) This representation and the concurrent representation of tests as hyperplanes offer longer range possibilities for selecting good diagnostic tests, for eliminating redundant or useless tests, for improving diagnostic procedures, and for generally studying the relationship of malfunctions to diagnostic tests. The possible value of these representations results both from the data compression they yield and from their possible validity as models of nature.

(*vi*) Studying the diagnostic results in detail, which is made much easier by the techniques discussed in this paper, can reveal weaknesses

in the diagnostic programs and in the malfunction-simulation hardware. Such study also leads to insight and understanding which is not easily acquired by other means.

APPENDIX

A.1 *"Repeat Order" Cluster*

There is a special circuit for repeating certain orders up to a maximum of 32 times, with the address in the order being incremented by 1 each time. In some cases, a second part of the instruction which specifies a flip-flop in one of the access registers is also incremented by 1 each time. The major use of this repeat facility is in writing or reading a whole word between the BGS (which has single bit readout) and an access register.

One cluster consists of about 200 malfunctions in the repeat counter, which counts the repetitions. Stuck flip-flops in the counter, bad carries, and bad input are typical.

The center of gravity of this cluster is about at

$$2, \quad -1, \quad 2\tfrac{1}{2}, \quad 1, \quad -1, \quad 1.$$

As this center is quite close to the center of the first cluster in Section 5.4, a great deal of overlapping might be expected. On examination this turns out to be correct.

The intimate connection between these two clusters of malfunctions is natural because both circuits involved are used only during repeat orders. In fact, it might be more natural to treat the two clusters as a single cluster of malfunctions whose common element is that they disturb the functioning of repeat orders.

A.2 *"Zero Flip-Flop Reading" Cluster*

There are about 80 "miscellaneous flip-flops" which can be explicitly read by the "read flip-flop" order. This cluster consists of 311 malfunctions whose common characteristic is that when any flip-flop whose value is 0 is read in this way, the answer is frequently 1.

The "read flip-flop" order operates through a large "flip-flop reading" circuit which is shown in Fig. 8. The action of this circuit is to transfer the value of the selected miscellaneous flip-flop into a control flip-flop known as FF, and to use the value from there. There are about 230 so-called "isolation" diodes through which the various miscellaneous flip-flop values are funnelled into FF. The circuit is so arranged that

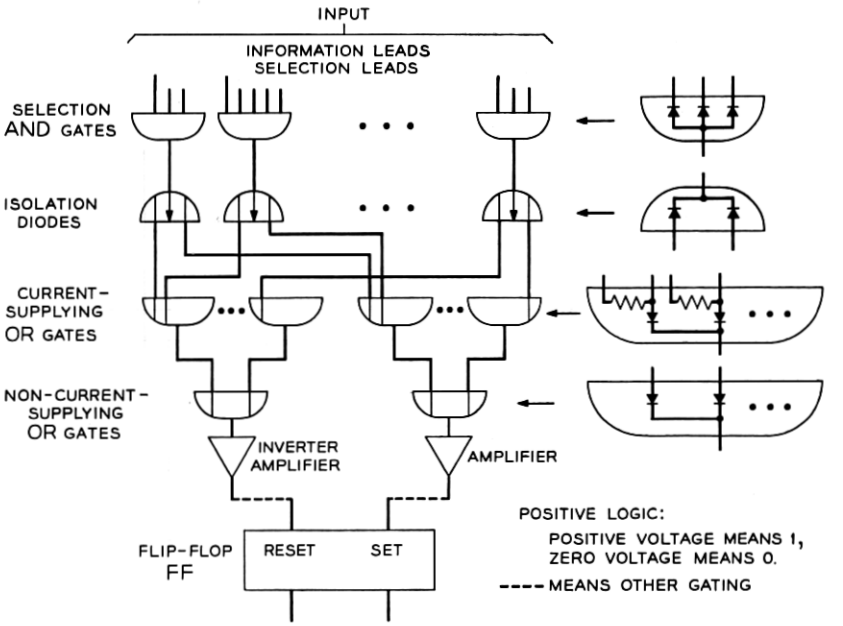

Fig. 8 — Simplified block diagram for flip-flop reading circuit.

open-circuiting *any* one of these diodes causes trouble when reading a 0 value from *any* miscellaneous flip-flop (though a 1 value is read correctly). For half these diodes, an open circuit causes the value read to be the value left in FF from before. For the other half of these diodes, an open circuit causes both the "set" and "reset" leads of FF to be pulsed simultaneously; we do not know what effect this has, but we believe that it usually leaves FF unchanged.

Besides open-circuited isolation diodes, removal of a package which contains several of these diodes, or removal of a gate package which feeds into these diodes, has the effect of causing a 0 value *always* to be read as a 1. Malfunctions of this sort also belong to this cluster.

There are a variety of other malfunctions which belong to this cluster. For example, there is an amplifier which feeds the "set" input to FF during the "read flip-flop" operation. If its output is stuck in the high voltage state, or if it is removed (which has almost identical circuit results), or if the diode through which it feeds is open-circuited, we should and do obtain malfunctions in the cluster. If the amplifier which feeds the "reset" input to FF during "read flip-flop" operation is stuck at low voltage output, we obtain a malfunction in the cluster. If the

gating lead (called RFFT), which gates the "set" and "reset" impulses into FF during "read flip-flop" operation, is prevented from operating by a short-circuited diode just on those occasions when a 0 value is being read, we again obtain a malfunction in the cluster.

A few of the malfunctions which belong to the cluster require deeper explanations. For example, if the RFFT lead (referred to above) *never* operates, because the amplifier feeding it is stuck at low voltage output, we again obtain a malfunction in the cluster. The effect of this malfunction is that FF *always* retains its previous value during a "read flip-flop" operation, regardless of whether the value being read is 0 or 1. It is not immediately clear why this malfunction should give test patterns closely resembling others in the cluster. However, by analyzing the diagnostic test program, we find that FF (which is also used by other operations) is most often left with a 1 in it upon entering the critical diagnostic test operations. Thus, this malfunction most often causes errors in reading the value 0.

This cluster has relatively sharp boundaries. Also, it is a "pure" cluster, that is, all the malfunctions in the region of space it occupies belong to it; other clusters do not overlap. The center of gravity of the cluster is approximately at

$$2\tfrac{1}{2},\quad 1,\quad 1,\quad 0,\quad 2,\quad 0.$$

The extreme values of the various coordinates are as shown:

minima	1,	0,	0,	−2,	0,	−1,
restricted minima	1,	0,	0,	−1,	1,	0,
restricted maxima	4,	2,	2,	1,	2,	1,
maxima	6,	4,	2,	2,	3,	4.

By "restricted maxima" we mean the maximum values for the 291 most centrally located of the 311 malfunctions; the other 20 are rather thinly sprinkled.

No two malfunctions in this cluster have identical test patterns. This may seem strange, for those package removals which cause a 0 value always to be read as a 1 have identical circuit effects. Also, those isolation-diode open-circuits that cause a 0 value to be read as whatever was left in FF from before have identical circuit effects. Why should the malfunctions within one of these groups produce diverse test patterns?

In the case of the package removals, almost all the variations in the test patterns result from reading flip-flops whose value is controlled from outside the CC and varies from time to time. Some examples are flip-flops RTA ("ringing tone active"), BSYT ("busy tone"), RNGS

("ringing scan") and 10MSCK ("10 millisecond clock"). When the value happens to be 1 the value is read correctly and the corresponding diagnostic tests are passed; when the value is 0, it is read incorrectly, and the contrary happens. Several of these flip-flops are read (in effect) four times during the diagnostic program, and we are able to follow any changes which take place. The slower changing ones like RTA are indeed observed to change either not at all or only once during the course of a single diagnostic run-through, while a faster changing one like RNGS is sometimes observed to change more often.

In the case of the isolation-diode open-circuits whose effect is to leave in FF its previous value, the test results are subject to the same source of variability. However, they are also subject to the additional variability of depending on the previous contents of FF. While this is more often 1 than 0, it is 0 significantly often. Thus, a test which the previous group fails may be passed by this group, and vice versa. All the tests ever failed by this group, however, include all the tests ever failed by the previous group, and more as well.

An interesting sidelight concerns the difference between even-numbered and odd-numbered tests. The diagnostic tests are conducted in pairs, with nondiagnostic work intervening between pairs. For this reason, each even-numbered test is entered from within the diagnostic program; analysis reveals that in this case FF contains a 1 when the individual test sequence is entered. As the relevant program was not available, we were unable to determine the situation for odd-numbered tests, but we infer indirectly that FF could have either value depending on unknown circumstances. As a consequence, certain tests on different flip-flops which are exactly similar to each other (including the fact that the flip-flop normally has value 0 at the time) fail or pass depending on whether the test is even or odd-numbered. There are several examples of this sort.

The reader may suspect that our circuit analysis is incomplete, and may suspect that the pattern differences for different malfunctions actually reflect different circuit effects. Fortunately we have at least three cases in which the same identical malfunction in this cluster was diagnosed twice. The differences between the test patterns for the self-same malfunction were quite as great as between test patterns for different malfunctions of one type in the cluster.

A.3 *Two Clusters Affecting the BGS Address Register*

These two clusters give a useful insight into the process of cluster analysis. It is probably true that they should be merged into one larger cluster which includes them both.

One consists of about 60 malfunctions affecting the EPO (execute program order) gating pulse which comes through a transformer into the BGS address circuit. This transformer serves only the BGS address circuit and an associated circuit. Almost all the malfunctions in this cluster are shorted diodes which short circuit this EPO lead. One or two are malfunctions in the transformer which prevent the pulse from appearing.

The second cluster consists of about 65 malfunctions which cause one or both of the gating leads BSBGX or BSBGY to operate or fail to operate. These leads are the leads which enable input to the X and Y halves of the BGS address register.

The centers of these two clusters are approximately at

$$3\tfrac{1}{2}, \quad -1, \quad 2\tfrac{1}{2}, \quad -\tfrac{1}{2}, \quad -\tfrac{1}{2}, \quad 1\tfrac{1}{2}$$

and

$$4, \quad -1, \quad 2, \quad -\tfrac{1}{2}, \quad 0, \quad 2.$$

Thus, they are fairly close together by comparison with the sizes of the clusters, which suggests that the clusters may overlap a good deal. Closer examination reveals that they do overlap a great deal.

This suggests that we do not have two distinct clusters but two types of malfunctions in one cluster. The next natural step would be to speculate that the common element to both clusters is a serious input difficulty to the BGS address register. To carry the analysis further we would then examine the other malfunctions in the region of space which these two clusters occupy, and see if many of them fit this new description. If so, we would systematically trace out (from the circuit diagrams) all like malfunctions. If essentially all of them should lie in this same region, then we would consider that we had arrived at a satisfactory cluster.

REFERENCES

1. Joel, A. E., Quirk, W. B., et al, An Experimental Electronic Telephone Switching System: A conference at Morris, Illinois, October 12, 1960. (Part of the 1960 General Fall Meeting of the AIEE.) (Unpublished document.)
2. Joel, A. E., Jr., An Experimental Switching System Using New Electronic Techniques, B.S.T.J., *37*, 1958, pp. 1091–1124.
3. Seckler, H. N. and Yostpille, J. J., Functional Design of a Stored-Program Electronic Switching System. B.S.T.J., *37*, 1958, pp. 1327–1382.
4. Haugk, G., Greenwood, T. S., and Yostpille, J. J., Morris Electronic Switching System — Final Report. (Unpublished document, March, 1963.)
5. Tsiang, S. H. and Ulrich, Werner, Automatic Trouble Diagnosis of Complex Logic Circuits, B.S.T.J., *41*, 1962, pp. 1177–1200.