November 12, 1959

Harlan E. Anderson
E.J.C.C. Publication Committee
Digital Equipment Corporation
Maynard, Massachusetts

Dear Mr. Anderson:

Enclosed are four copies of a paper which has been accepted
for publication in the Proceedings of the 1959 E.J.C.C.

Respectfully,

W. F. Miller

W. F. Miller, Director
Applied Mathematics Division

WFM:ed
enc.

ARGONNE NATIONAL LABORATORY

NORMALIZED FLOATING POINT ARITHMETIC WITH

AN INDEX OF SIGNIFICANCE[*]

by

Herbert L. Gray

Charles Harrison, Jr.

NORMALIZED FLOATING POINT ARITHMETIC WITH

AN INDEX OF SIGNIFICANCE

by

Herbert L. Gray

Charles Harrison, Jr.

SUMMARY

Two methods, to be used on machines now under construction, for automatically determining a bound for the propagated and generated error in a calculation can be called the "normalized" and "unnormalized" significance methods. In the first method, only normalized floating point numbers are used, thus retaining as many digits of the number as possible, and an index of significance indicates their degree of accuracy. In the second method, the numbers are not normalized, and only the digits considered significant are retained.

Since the error in any one step of a calculation is not usually a factor of the base, B, and both systems can only allow adjustment to within a factor of B, any set of arithmetic rules for determining significance will hold only "on the average". The design of such rules, therefore, rests upon general assumptions concerning the statistics of computation.

To test the accuracy of these rules, the normalized significance method was simulated on an existing computer, and the significanses indicated were compared with those established by other means. As yet, we have not found any calculation which causes the index to be more than one decimal digit in error from the true value.

NORMALIZED FLOATING POINT ARITHMETIC WITH

AN INDEX OF SIGNIFICANCE

Herbert L. Gray
Charles Harrison, Jr.

It has been frequently pointed out that the task of determining an error bound for the results of a problem is usually a long and difficult calculation which is avoided as much as possible by the programmer. The introduction of floating point arithmetic in modern computers and the ever growing use of compilers makes the task of error analysis even more difficult and its computation even less probable. Clearly a machine method is needed to automatically calculate a bound for the propagated and generated error, given the initial error in the input and the residual error due to approximating functions.

Two methods for doing this herein are called the "normalized significance" and the "unnormalized significance" methods. The "normalized significance" method always keeps the floating point number normalized and provides an index of significance. The "unnormalized significance" method does not normalize floating point numbers and uses the count of digits remaining after leading zeros as in indication of their significance.

Before discussing these methods, we should like to define what is meant by significance. If one says α digits of a number are significant, one means the error in the number is less than $B^{-(\alpha+n)}$ where B is the base and n is the number of leading zeros. In the normalized system n would be zero. In the unnormalized system, n would be the total number of digits minus α, as there are no insignificant digits.

Thus, in the normalized method as many digits as possible of a number are retained and its index determines the number of digits which are significant; in the unnormalized method only the digits considered significant are retained.

Since the error in any one step of a calculation is not usually a factor of the base, B, and α in either system only allows adjustment to within a factor of B, any set of arithmetic rules can have the desired property only "on the average". The design of such rules must rest upon general assumptions concerning the statistics of computation; these rules may, therefore, not be valid in special situations.

In the normalized method to be used with Argonne's arithmetic unit FLIP, each number is represented with base 2 by a triplet (x_f, x_p, x_i), where x_f is the fractional part of the number with $1/2 \leq |x_f| < 1$, x_p is the associated power, and x_i is the index of significance. Thus $X = x_f \cdot 2^{x_p}$ to x_i significant figures. In addition and subtraction the result has an index equal to the smaller of the two indices of the operands. Whenever a fraction is down-scaled, its index and power are increased; whenever a fraction is up-scaled its index and power are reduced. In multiplication and division, the index of the result is the smaller of the two indices and even if the result needs to be scaled to bring it into the normal range, the index is not changed.

The "unnormalized method" is very similar to the one to be used by the University of Chicago in their new computer.[1] Each number is represented by a couplet (x_f, x_p), where $0 \leq |x_f| < 1$ and the number of digits of x_f minus the number of leading zeros equals x_i of the normalized scheme. Insignificant digits are shifted off the register. In addition and subtraction, no scaling of the result is carried out unless, of course, its fraction is greater than or equal to unity. In multiplication and division, the resultant fraction is scaled so as to have the same number of leading zeros as the operand with the fewer number of significant figures.

1.   Ashenhurst, R. L. and Metropolis, N., Unnormalized Floating Point Arithmetic, Journal of the ACM, July, 1959, 415-429.

In his error analysis of floating point procedures in the Communications of the Association for Computing Machinery[2], John W. Carr III points out that a normalized floating point procedure will always give a better result than an unnormalized procedure. While the normalized method will create less error, the loss of possible significant digits due to the shorter register length remaining when an index of significance is used may outweigh this gain.

The logic for addition and subtraction, due to the normalization which a result may require and the handling of the index, are more complicated in the normalized method. However, division and multiplication are much more complicated in the unnormalized system because of the shifting required to obtain the correct number of leading zeros. Since the number of additions and subtractions in a calculation is usually greater than the number of multiplications and divisions, the unnormalized system might be a little faster. Following are some examples. Those numbered with (a) use the Argonne method, while those numbered with (b) use the second method. The length of the fractions is adjusted so that the register in each scheme contains the same number of digits.

2. Carr, John W. III, Error Analysis in Floating Point Arithmetic, Communications of the ACM, May, 1959, 10-15.

1a)   (0.5782, 6, 3) + (0.1485, 4, 3)   Using Argonne scheme.

      0.57820, 6, 3   Arithmetic register holds 1 more
                               digit than storage.
+ 0.00148, 6, 5   Power and index adjusted.
_____

      0.5797 , 6, 3   Result rounded, smaller index is
                               used.

1b)   (0.00578, 8) + (0.00149, 6)

      0.005780, 8   Register again holds one extra
                                 digit.
+ 0.000014, 8   Power adjusted.
_____

      0.00579, 8

2a)   (0.1397, 5, 4) - (0.9143, 4, 2)

      0.13970, 5, 4

- 0.09143, 5, 3   Power and index adjusted.
_____

      0.04827, 5, 3   Smaller index used.

      0.4827, 4, 2   Number normalized.

2b)   (0.01397, 6) - (0.00091, 7)

      0.001397, 7   Power adjusted.

- 0.000910, 7
_____

      0.00049, 7   Number rounded to 2 significant
                                 digits.

3a)   (0.9143, 4, 2) + (0.1875, 4, 4)

      0.9143, 4, 2   No power adjustment needed.

      0.1875, 4, 4
_____

      1.1018, 4, 2   Fraction exceeds one.

      0.1102, 5, 3   Normalization and roundoff.

3b) (0.00091, 7) + (0.01875, 5)

    0.000910, 7

    0.000187, 7
    ——————————

    0.00110, 7   Rounded to 3 figures.

4a) (0.5782, 6, 3) x (0.1485, 4, 4)

    (0.5782, 6, 3) x (0.1485, 4, 4) = ~~.08719920~~ .08586270, 10, 3
                          [Double register product]

                = ~~.8720~~ .8586, 9, 3
            [Normalized and rounded] ,

index not changed. The digit which was shifted into register
was assumed to be good (optimistic approach).

4b) (0.00578, 8) x (0.01485, 5)

    (0.00578, 8) x (0.01485, 5) = .0000858330, 13

                      = .00858, 11

Rule: result always contains same number of significant
        digits as the least significant operand.

5a) (0.5782, 6, 3) x (0.2485, 4, 4)

    (0.5782, 6, 3) x (0.2485, 4, 4) = .14368270, 10, 3

                      = .1437, 10, 3
                  [Rounded]

5b) (0.00578, 8) x (0.02485, 5)

    (0.00578, 8) x (0.02485, 5) = .0001436330, 13

                    = .00144, 12
                [Adjusted and rounded]

6a) $(0.5782, 6, 3) \div (0.2485, 4, 4)$

$(0.5782, 6, 3) \div (0.2485, 4, 4) = 2.327605.., 2, 3$

$= 0.2328, 3, 3$
[Normalized and rounded]

Index does not change after normalization. Extra digit assumed to be no good (pessimistic approach).

6b) $(0.00578, 8) \div (0.02485, 5)$

$(0.00578, 8) \div (0.02485, 5) = 0.2325955.., 3$

$= 0.00233, 5$
[Adjusted and rounded]

Same rule holds for division as well as multiplication.

7a) $(0.2485, 4, 4) \div (0.5782, 6, 3)$

$(0.2485, 4, 4) \div (0.5782, 6, 3) = 0.429782.., - 2, 3$

$= 0.4298, - 2, 3$
[Rounded]

7b) $(0.02485, 5) \div (0.00578, 8)$

$(0.02485, 5) \div (0.00578, 8) = 4.29930.., - 3$

$= 0.00430, 0$
[Adjusted and rounded]

In test cases we have run so far using the Argonne scheme the number of multiplications far exceeds the number of divisions which gives rise to optimistic indices, though none of the indices have been off by the equivalent of more than one decimal place. In both systems a special number exists which does not obey the above rules. This number is

zero and it follows the following rules in both systems.

$$X \pm 0 = X$$

$$X \times 0 = 0$$

$$0 \div X = 0$$

$$X \div 0 = \qquad \text{is not attempted.}$$

If during the normalization process in the Argonne scheme, the index of significance is about to become less than zero, then the shifting is stopped so that the index remains at zero. Thus at the end of this operation the fractional part of the number may be less than one half. This number, known as a relative zero, obeys the same rules as any normalized number during addition, subtraction and multiplication, but division by this number as with true zero is not attempted. Whenever the normalization is not completed the Argonne machine will jump to a fixed location in the memory and let the program decide what is to be done about the number.

Relative zero may also be introduced as an input number. In this case, the fraction is unnormalized, but the index need not be zero. This number is used in special operations such as finding the integral part of another number as follows:

$$
\begin{array}{l}
\ 0.4597,\ 2,\ 4 \\
+\ 0.0000,\ 4,\ 4 \\
\hline
\ 0.0045,\ 4,\ 4 \qquad \text{Adjust power (index at max)} \\
\ 0.4500,\ 2,\ 2 \qquad \text{Normalization.}
\end{array}
$$

This type of relative zero is also used in operations which convert

from floating to fixed point. A relative zero with an index not equal
to zero can never be generated by the floating point unit.

The only other special number which may arise in the second scheme
is a number with zero significant figures and thus a zero fraction. All
operations with this number except division follow the normal rules.

If a number in the Argonne scheme does not have any associated
error, it is said to be "totally represented". Such numbers are given
a special index and operated on in a slightly different way. As long
as a calculation uses only totally represented numbers, the index of
the result does not change even if normalization occurs. If, however,
during a calculation any non-zero digits of a number are shifted off
or if one of the operands is not a totally represented number, then
computation of the index of the result reverts to its normal form.
For example:

$$
\begin{array}{ll}
(.4583,\ 2,\ \text{TR}) & \text{(Totally represented)} \\
-\ (.4329,\ 2,\ \text{TR}) & \\
\hline
.0254,\ 2,\ \text{TR} & \\
(.2540,\ 1,\ \text{TR} & \text{Normalization.} \\
\end{array}
$$

$$
\begin{array}{ll}
(.4583,\ 2,\ \text{TR}) & \\
+\ (.1497,\ 1,\ \text{TR}) & \\
\hline
.4583,\ 2,\ \text{TR} & \\
.01497,\ 2,\ \text{TR} & \text{Adjust power} \\
\hline
.4733,\ 2,\ 4 & \text{Results rounded to 4 figures.} \\
\end{array}
$$

.4583, 2, TR

- .4329, 2, 4   Not a totally represented number.
_____

.0254, 2, 4   Index reverts to its normal form.

.2540, 1, 3

Such a scheme helps to simplify floating point integer arithmetic.

~~The second part of this seminar describes the results of tests of the Argonne scheme~~.

To test the accuracy of the index in the floating point scheme we simulated FLIP with a GEORGE program. We combined this program with some GEORGE subroutines and compared the results obtained in this manner with those computed by other means.

First, let us consider the routine involving the method of Leverrier for determinining the coefficients of the characteristic polynomial of a real valued square matrix.

Let λ_i = roots of the characteristic polynomial of our n x n matrix A.

Define S_k = trace (A^k) k = 1, 2, ..., n

$$S_k = \sum_{i=1}^{n} a_{ii}^{(k)}$$

The characteristic equation is

$$\lambda^n + C_1 \lambda^{n-1} + C_2 \lambda^{n-2} + \cdots + C_{n-1} \lambda + C_n = 0$$

where

$$- C_1 = S_1$$

$$- 2C_2 = S_1 C_1 + S_2$$

$$- 3C_3 = S_1 C_2 + S_2 C_1 + S_3$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$- nC_n = S_1 C_{n-1} + S_2 C_{n-2} + \cdots + S_{n-1} C_1 + S_n$$

The method requires the computation of A^2, A^3, ..., A^n and the corresponding S_1, S_2, S_3, ..., S_n. There are $n^3 (n-1)$ multiplications and $n^3 (n-1) + n (n-1)$ additions involved in these calculations. Obviously, inaccuracies can be caused if n is very large or the elements of A are too large.

We ran tests with our combined program using four different matrices. In the first two cases tested, the A's were of the tenth order and of the forms:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2 & \cdots & 2 \\ \cdot & \cdot & \cdots & \cdot \\ 10 & 10 & \cdots & 10 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 2 & 2 & \cdots & 2 \\ 4 & 4 & \cdots & 4 \\ \cdot & \cdot & \cdots & \cdot \\ 20 & 20 & \cdots & 20 \end{pmatrix}$$

In both cases $C_1 = - S_1$ and $C_2 = C_3 = \cdots = C_{10} = 0$. We obtained exact results and the index indicated this for each coefficient.

Next we tried a twelfth order matrix of the form

$$\begin{pmatrix}
4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4
\end{pmatrix}$$

Once again we obtained exact results with the corresponding correct indices. It was with this case that we noticed the vast difference a change in method made in the final results. We computed the coefficients from the equation

$$C_k = (S_1 C_{k-1} + S_2 C_{k-2} + \cdots + S_{k-1} C_1 + S_k) \div (-k)$$

If, however, we use the slightly different equation

$$C_k = -\frac{1}{k} (S_1 C_{k-1} + S_2 C_{k-2} + \cdots + S_{k-1} C_1 + S_k)$$

we get different results. In fact, C_{12} is completely in error and the index associated with it is zero. This zero index means the number is worthless as a result.

The initial error was introduced in the computation of $\frac{1}{k}$ since non terminating decimals like $\frac{1}{3}$, $\frac{1}{6}$, etc. can not be exactly represented. The use of these inexact numbers produced inexact coefficients. The errors in both of these were propagated in the calculations of subsequent coefficients. As a result C_{12} was completely in error.

The next matrix we used was a fifteenth order matrix having all its elements equal to 1. Even though the elements are not large the size of n leads to some very large intermediate results. For example, $S_{15} = 15^{15} = 437, 893, 890, 380, 859, 375$. The coefficients, however, were more reasonable. In fact $C_1 = -S_1$ and $C_2 = C_3 = \cdots = C_{15} = 0$. In this test we also found complete agreement with the indexed results and the known values for the coefficients.

Next we applied our program to Jordon's method for the inversion of

Hilbert matrices of various orders. Hilber matrices are of the form

$$
\begin{pmatrix}
1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\[2ex]
\frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\[2ex]
\frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\[2ex]
\cdot & \cdot & \cdot & \cdots & \cdots \\[2ex]
\frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1}
\end{pmatrix}
$$

Hilbert matrices are almost singular and as the size of the order increases the value of the determinant rapidly approaches zero. Several matrices were inverted including a 12 X 12 which had a determinant approximately equal to 10^{-75}. We compared the results we obtained with those computed by an inversion formula.[3]

The elements in our inverted matrices compared favorably with those in the table of inverses for all the orders tested. We were able to get 10 or 11 digit agreement in the elements of an inverted seventh order Hilbert matrix and 6 or 7 digit agreement for a 10 X 10. The indices associated with these inverse elements were either correct or one-digit optimistic. This is about what we expected since we employ the optimistic approach in our floating point scheme.

It may be interesting to note that Jordan's method on a floating point machine produced inverses above the 6th order that were completely in error. Also, our GEORGE interpretive floating point subroutine was useless beyond a 7th order Hilbert matrix. The primary difference in

3.   I. Savage and E. Lukacs, "Tables of Inverses of Finite Segments of the Hilbert Matrix" in Contributions to the Solution of Systems of Linear Equations and the Determination of Eigenvalues, NBS Applied Mathematics Series 39, 1954, p. 105-108.

these cases is the number of digits the machine can hold. However, the other two schemes give no indication of the approximate accuracy of the results.

A third test was performed with a subroutine which evaluates a polynomial with real or complex coefficients having real or complex roots. We inserted known roots for certain 4th and 8th degree polynomials with real coefficients and noted the computed value of the polynomial.

When the roots were real and exact the computed value of the polynomial was exactly zero. The same thing was true when the roots were complex but with exact real and imaginary parts. However, when the roots were complex and at most one of the parts was exact we get an evaluation of the polynomial which is not exactly zero and has a zero index. Here, the zero index meant the digits in the number were meaningless but the zeros preceding the first of these digits were correct.

Finally, we applied our program to a trapazoidal integration subroutine. We evaluated some simple integrals between exact limits. If the results could be represented exactly, we obtained exact results. When the results could not be exactly represented we obtained numbers whose error was less than 5×10^{-18}. For these cases the index on the partial sums was sometimes less than the maximum but continued summations caused it to increase up to the maximum allowed for non-totally represented numbers.

The tests we ran were considered fairly representative of the calculations involved in many problems. The fact that the index of significance was never more than one digit from its true value lends merit to the Argonne scheme. If some of the multiplications could have been replaced by divisions without increasing the errors the index would have been even closer to the true value.

November 12, 1959

MR. HARLAN E. ANDERSON
EJCC Publication Committee
Digital Equipment Corporation
Maynard, Massachusetts

Dear Mr. Anderson:

Attached are the required four copies of my paper. Due to the time limitation imposed by the deadline, I have inserted a "page 3a" instead of having the entire paper retyped.

Also included is the requested "100 word" biography of myself. I have attempted to prepare it in the style of the I.R.E. Proceedings.

I will require facilities for projecting the standard size 3-1/4 by 4 inch slides.

Yours truly,

James S. Bomba

JAMES S. BOMBA

MH-2213-JSB-CB

Att.
4 copies of paper —
biography —
originals of figures —

ALPHA-NUMERIC CHARACTER RECOGNITION USING LOCAL OPERATIONS

by

J. S. Bomba
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

ABSTRACT

A demonstration of the recognition of an alphabet of thirty-four distinguishable handprinted, block capital, alpha-numeric characters is described. The basis of the recognition method is the use of "local operations" to extract significant features from the characters. The recognition procedure is: (1) reduce the noise in the field by local averaging, (2) standardize the character line width, (3) extract the "essential" features from the character (typical features are - straight lines of varying slopes and line intersections), (4) derive the relative locations and sizes of these features, (5) identify the characters from items (3) and (4) with combinational logic. The method was successfully evaluated with a simulation program on an IBM 704.

ALPHA-NUMERIC CHARACTER RECOGNITION USING LOCAL OPERATIONS

by

J. S. Bomba
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

1.0 Introduction

This paper describes a demonstration of the recognition of thirty-four alpha-numeric characters. The IBM 704 EDPM was used as a tool to study the method which led to this demonstration. The Generalized Scanner[1] was used as an input transducer for this study.

The method of character recognition which was used here is to extract from the character its "essential" features and then recognize it from these features. For this study such features as horizontal, vertical, and slant straight lines, and intersections of lines have been used.

These features have been extracted by means of local operations. A local operation is a transformation which produces a small section of a new pattern from data in a corresponding small section of the original pattern (Figure 1). An entire pattern is transformed by considering all of the small sections, where each section is considered independently. Here, the pattern area was divided into a sixty by ninety array of bits (i.e. either black or white spots) in order to quantize the visual impression of a character. A typical local operation section would consist of fifteen bits.

In order to extract a specified feature, the whole
pattern is processed and the resulting pattern then is blank
unless the desired feature was present in the original pattern.

The procedure for recognition is as follows. <u>First</u>,
the characters are processed by a program which reduces the
noise in the field by the method of local averaging[2]. In this
method, the value of the majority of spots in a three by three
rectangular local area dictates the value of the center spot
in the new pattern. This process is more effective if it is
repeated twice. <u>Second</u>, the line width is standardized.
Since the matrix granularity is such that a typical line
trace as written is always greater than four matrix elements
thick, a line-width standardizing operation can extract the
middle four matrix locations. <u>Third</u>, the features as shown
in Figure 2 are extracted. These are:

    a) straight lines which are horizontal and vertical,

       and slant straight lines which are at ±45 degrees,

       ±30 degrees, and ±60 degrees from vertical:

    b) all four orientations of T- and L-intersections, and

    c) selected orientations of V-intersections.

The local areas (as defined in Figure 1) which are used to extract
these features, have the same shape as the desired feature.
There are seventeen different features for which the original
pattern is examined. In effect, a complete pass is made
through the pattern for each feature and whenever a local

area in the pattern matches the interrogation local area,
a mark is made in a corresponding output matrix. This is
done for each of the seventeen different features, thus
generating seventeen "new pattern fields". Fourth, these
fields are divided into nine equal rectangular areas
(shown in Figure 3). The prevalence of spots in certain
areas is used to indicate the general position of the
feature (i.e. top or bottom) and its significance (i.e.
whether a straight line is long or short). Fifth, the
recognition is done from the detection of the presence of
features. In most cases, it was only necessary to ask
whether a given feature was extracted. However, with the
limited variety of feature types which were used in evaluating
the method, it was necessary to ask further questions about a
few features such as horizontal and vertical straight lines.
The identification was done with combinational logic. The
logic was extended to allow some variation in character styles.

Hand printed, block capital letters and numbers of
the style shown in Figure 4 can be recognized by this method.
These are the letters "A" through "Z" and the numberics "1"
through "0"; a "one" and "I", and "zero" and "0", are not
separately distinguished. The characters may vary from full
to one-half size vertically and horizontally. The line widths

may vary horizontally and vertically, from a minimum average of four matrix elements to a maximum average of ten matrix elements. The characters must roughly be centered when they are less than two-thirds full size, and must be reasonably free from tilt.

2.0 Input Methods

The visual impressions have been transferred into the 704 by a manual and a machine method, both of which are described below (see 2.2 and 2.3).

The visual image is first quantized by dividing the pattern field into square elements and giving each element a value of "0" or "1" depending on whether the field is light or dark at that point. In this study the character lines were dark on a light backbround, and these character lines are represented by the bit locations which are ones.

The image, which then consists of "0"s and "1"s, is stored in the computer in this binary matrix form.

Each 60 element row of the matrix is split in half and each half is stored in the least 30 significant bits of a 704 core word. The rows are in 90 consecutive pairs of storage locations.

2.01 Matrix Size

A 60 x 90, 5400 element matrix was chosen because:
(1) It is large enough so that quantizing errors should be negligible.
(2) It is large enough so that the edges of the pattern need not be covered by the local operation if the character is more or less centered (thus, it was not necessary to program the special cases which occur when the local area is only partially present as would happen at the field edge. As you will see later in the discussion of feature extraction (see Fig. 10), since the largest local area used extends 7 spots from the local area center, the effective size of the matrix is thus 46 x 76.)

(3) Its width will just fit the printer which we use with the 704, (119 available type wheels are used to print the pattern with every other one printing a blank. The blanks are printed to limit the distortion of the field which the printer introduces because (a) in the original quantization the matrix elements are square whereas (b) on the printer the elements are rectangles whose height to width ratio is 6 to 5 when the blanks are printed and 6 to 10 when they are not).

2.2 Manual Procedure

In order to test the programs and to try sample characters before the Generalized Scanner[1] was finished, test patterns and characters were made up on IBM cards.

For test characters, first the true size characters were drawn on translucent paper. They were then made into viewgraphs which were enlarged by a slide projector. The enlarged image was focused on a 60 x 90 rectilinear grid 1.50 x 2.25 feet. Thus, each 1/4" x 1/4" element could be marked if more than half its area were black. IBM cards were then punched from this grid. One card per row was used with "ones" punched to correspond to black elements and nothing punched for white elements. Special program test patterns were made by marking the large grid.

2.3 <u>Machine Methods</u>

With the advent of the Generalized Scanner it became possible to write the characters on opaque paper and transcribe the scan results onto magnetic tape which can be fed to the 704. This machine method makes it possible to quickly and easily process a large number of characters.

3.0 <u>Character Preparation by Local Operations</u>

3.1 <u>Noise Reduction</u>

In order to allow as inputs to the feature extraction program, characters which have missing spots within the line of the character, extra spots in the field outside the line of the character, and fluctuations along the edges of the character lines, the character can be processed by a program which performs local averaging[2]. This program uses a local area, Figure 5, which is 3 x 3 rectangle. The criteria which are applied for this local area are; (1) if there are five or more "1"s in the nine possible spots and if the center spot was a zero, then the center spot is changed to a one. (2) If there are five or more "zeroes" and if the center spot was a one then the center spot is changed to a zero. (3) In all other cases, the center spot is retained as it was in the original pattern. This process works most satisfactorily if it is repeated two or three times. That is, the original pattern is processed, then the result of the first processing is used as an input to the second process, etc. An example of the

results of reducing the noise by this method can be seen by comparing a typical original pattern, Figure 6, and the resulting pattern (processed twice), Figure 7.

Other criteria for the 3 x 3 local area and a 5 x 5 local area with a variety of criteria were tested. The 3 x 3 local area and the criteria of the previous paragraph were found to be the most satisfactory of the possibilities which were studied.

3.2 Line Width Standardization

The feature extraction program works best when the characters have a uniform line width. Therefore, a program was written to process characters with line widths which vary from 4 to 10 spots and produce characters with standard line widths. This program will change the character lines to a uniform average width of approximately 4 spots. The local area which is used by this program is shown in Figure 8. The criteria for this operation are that (1) a spot will be a one only if the original spot were a one and the difference between areas A and B and between areas C and D in the local area was less than 4, or (2) the difference between the number of spots in the areas E and F and the areas G and H are less than 4. An example of "thinning" the previously "denoised" characters is shown in Figure 9.

This program might be improved if we made the actual radial distances in the local area all equal. (It is obvious, Figure 8, that five elements at 45 degrees extend further from the center than do five elements in a rectilinear direction).

4.0 Feature Extraction

The crux of this method of character recognition is the local operation which is performed by a program which is called Feastract. This program extracts the features which are shown in Figure 2. Because this program was designed to study the method, it actually extracts more features than are used by the recognition procedure. The most useful features are the long straight lines: horizontal, vertical, and slanted. The Figure indicates which of the features are used and which are not.

4.1 Local Area

The local area which is used for the feature extraction is shown in Figure 10. This local area is actually a combination of a number of local areas, each of which might be used for a different local operation to extract a different feature. However, for purposes of study, it was convenient to arrange a local area in the manner shown, a radial pattern which enabled me to vary both the criterion for extraction and the shapes of the features which are found to be most suitable.

It should be noted here that this local area, as applied in this program, is used for local operations; however, if the number of points at which the local area is used is limited to fewer than all points in the overall pattern, then this local area falls into the class of feature extraction methods which can be called radial or polar scan methods[3]. There is one significant difference, however, between the polar scans and this scan. That is, each radius in my local area is required to contain "a" spots (where "a" = the number of spots in the entire radius, or nearly all spots to allow for some noise) whereas in the polar scans each radius is only considered from the viewpoint of crossing a line. That is if any radius contains any spot, it is considered to have crossed the line for the polar scans.

4.2 Effective Operation of Feastract Program

In order to clearly present the feature extraction process, the details of the Feastract program will be omitted. Thus, we will discuss the feature extraction process.

Let us consider a specific feature as an example, namely an L-intersection. The local area for an L-intersection consists of radii 1 and 13 and the spot "p" as shown in Figure 10. The spot p with its associated spots x is moved over the pattern field in a scanning manner. Whenever all spots in this local area are found to be "ones", a one is written in the buffer image for the L-feature at the same coordinates which "p" has. Since the original pattern is not changed during the above process, the entire operation could, of course, be done in parallel.

The same process can now be done for each feature which we wish to extract.

The result is a group of buffer images, one for each feature. Figure 11 illustrates schematically some buffer images which would be produced from the original pattern. Each buffer image contains one feature.

After all of the spots in the original pattern have been examined and the results of the feature extraction process stored in various pattern buffer images, the program proceeds to examine each of these buffer images, and it determines whether or not there were any spots corresponding to the given feature in each image. If a spot is found in a buffer image, the program determines in which of nine areas the spot lies, as shown in Figure 3. A series of counters are used to remember how many spots lie in each area of the field for the feature under question at the time. When the buffer image for a given feature has been completely scanned and the number of spots in each area of the field have been recorded in the counters, then the result is transferred to the recognition program and also printed out. The basic program was designed so that it would handle up to 50 features.

Note that this feature extraction program does not consider the feature "closed loops" (as encountered in a well formed B). Although closed loops can be extracted with local operations[4], they are not a necessary feature for recognition of the symbols which were used here.

4.3 Output from Feastract

Typical information which is transferred from Feastract to the recognition program is shown in the following list for one feature. This list consists of - the identifying feature number and - the number of spots for the given feature which were found in each area of the field (Figure 3).

	Register Contents
Feature Number	2
Top Right	27
Middle Right	78
Bottom Right	24
Top Middle	0
Middle Middle	0
Bottom Middle	0
Top Left	30
Middle Left	59
Bottom Left	16

These numbers resulted from the extraction of vertical lines (Feature Number = 2) from an H.

5.0 Recognition From Features

Recognition is accomplished by combinational logic with the features as the input variables. Since there is a two-way choice corresponding to the presence or absence of

each variable, the logic takes the form of a logical tree
(Figure 12). However, before the actual recognition takes
place, certain preprocessing can be done on the features
and their locations.

5.1 Preprocessing

A series of subroutines was written to take the
information from Feastract and to partially interpret this
information to answer such questions as, "how long is a
horizontal line"? The detailed specification of said
question would be, "does a horizontal line cross three
areas in a horizontal row in the field"?

We can now consider more details of the answer
to the above question. A general statement of the criteria
for answering the question is: "Horizontal lines are con-
sidered to be long, if all three areas of the field in a
horizontal row each contain more than two marks which were
identified from horizontal feature sections". In other words,
first, all those areas with only two marks indicated in the
new or transformed pattern are eliminated. Second, a test
is made to see if there are horizontal features in the three
top areas. The same thing is repeated for the three middle
areas and then for the three bottom areas. The results are
stored in registers called TØP, MIDDL and BØTØM. Third, it
is then possible to say that if the TØP register has the
number three in it, there is a long horizontal line at the top.

When there is a number one in it, then there is a short
horizontal segment in the top. Thus, the number and length
of horizontal lines can be determined for characters which
have the restrictions as indicated in section 1.0.

The same procedure which was used for horizontal
lines can be applied to vertical lines. In this case, the
information is stored in registers called LEFT, MIDLE and
RIGHT.

There is also a subroutine to determine if there
is, for instance, only one horizontal line in the field.
The same procedure could be applied to slant lines. How-
ever, for this particular recognition tree, it was not
found to be necessary to apply this type of preprocessing
to other than horizontal and vertical lines.

5.2 Recognition

As indicated in Section 5.0, the recognition
proceeds as a series of binary choices. The majority of
tests which were used to make these choices are the follow-
ing two:

(1) The presence of a feature is determined by testing the
computer memory location of the feature number for non-zero.
(See Section 4.3)

(2) The presence of a feature in one of the nine field areas
is likewise determined, except that, in some cases, a minimum
number of spots is required in the test. The other tests
which were used are discussed in Section 5.1.

When a terminal branch of the recognition tree is reached, the identified character is printed.

It is interesting to note in Figure 12 that a great many more characters result from all of the tree branches than just the 34 which we initially started with.

We thus have evidence from this tested recognition tree that, even with the restricted input patterns which we have used, redundancy has to be included in the recognition logic. Some subtle variations in characters have only become apparent when they were encountered in testing this recognition logic.

6.0 Extensions of This Character Recognition Approach

This character recognition approach, which I have described, has been used to actually recognize characters in order to show that it works. Here, in this feasibility demonstration, only a few of the features which can be extracted with local operations have been used. It seems clear that if the local area of Feastract were used for other features which it can extract (ends, curved segments), many of the restrictions which have been here imposed on the characters could be removed. Along this same line, if other local operations, such as concavities as described by Unger[4], were included, the field would not have to be divided and thus larger variations in the characters could be admitted.

Unfortunately, there does not seem to be any good orderly way to specify the most desirable features or to specify the local area or local operation to extract a given feature. Fortunately we can use the 704 to test possibilities without the long process of building hardware. Nevertheless this is still a trial and error process.

Also, the logic for the recognition for this demonstration allows only limited variations in the characters. Although to the human being, characters of a wide variety appear to be the same, very often these characters are actually quite different from their feature viewpoint. Seriphs, slanted characters, and decorations tend to make the character into a different symbol. Thus, in an identification procedure which utilizes features, whether they be the normal features as specified by a human being or some special parameters which are not normally recognized by human beings, these various forms must be treated as separate symbols. The identification procedure could be lengthened to allow other forms of characters to be recognized.

Finally, unless self-organizing systems rapidly become a lot more sophisticated than they are now, any machine which we build to recognize large alphabets of characters of unspecified style will probably use a combination of several recognition methods, along with feedback to make more and more complicated tests until there is a high enough probability that there is a correct identification or a rejection.

7.0 Conclusions

It has been explained how features may be detected
to recognize certain alpha-numeric characters by using the
local operations mentioned above.

Three sample alphabets were tested on an IBM 704,
and the characters in these alphabets were separately
recognized. With this limited amount of test data, it was
felt that any attempt to specify allowable signal-to-noise
ratio, character distortion (such as tilt) or a recognition
error-rate would not be meaningful.

Others[4,5] have shown the power of using other
local operations for this same purpose. Only a few of the
many possible local operations have been investigated, and
it seems likely that this approach will be useful not only
for character detection but also in other areas of pattern
recognition.

8.0 <u>References</u>

1. W. H. Highleyman and L. A. Kamentsky "A Generalized Scanner for Pattern and Character Recognition Studies", PROC. W. J. C. C., pp 291-294, March 1959.

2. G. P. Dineen "Programming Pattern Recognition", PROC. W. J. C. C., pp 94-100, 1955.

3. T. L. Dimond "Devices for Reading Handwritten Characters", PROC. E. J. C. C., pp 232-237, 1957.

4. S. H. Unger "Pattern Detection and Recognition", PROC. I. R. E., vol. 47, pp 1737-1752, October 1959.

5. R. L. Grimsdale, et al, "A System for the Automatic Recognition of Patterns", PROC. I. E. E., vol. 106, Part B., No. 26, pp 210-221, March 1959.

List of Figures and Captions

Fig 1

1.　2　3　4　7　18　6　5

12　13　14　15　16　17

19　20　21

LINE
ENDS

8　9　10　11　22-38

FIG. 2

Fig. 3

A B C D E F G

H I J K L M N

O P Q R S T U

V W X Y Z

1 2 3 4 5 6 7

8 9 0

Fig 4

Fig 5

Fig 6

Fig. 7

Fig. 8

Fig. 9

Fig. 10

O

F1 (—)

F2 (l)

ETC

FIG.11

FIG. 12

Mr. Harlan E. Anderson, Chairman
1959 EJCC Publication Committee
Digital Equipment Corporation
Maynard, Massachusetts

Dear Mr. Anderson:

In accord with instructions contained in various letters from you
and Mr. Felker, I am enclosing with this four complete copies of
the manuscript of the paper "The Automatic Digital Computer as
an Aid in Medical Diagnosis" by Dr. C. E. Rupe and myself. Also
enclosed are four copies each of very brief biographies of the
authors.

Table 3 is included to give a general idea of the magnitude of a
typical table of correlation constants. It is not intended to be
actually read.

For the presentation of this paper, I shall have on hand two sets
of thirteen slides of the familiar 35 mm. type, two inches by
two inches outside dimension. These are the six figures included
with the manuscript, plus one "List of Steps" figure and six
whimsical sketches to illustrate points in the presentation.

Yours truly,

C. B. Crumb, Jr.

CBC/hmr

Enclosures

for Systems

Bendix

BIOGRAPHY OF DR. C. E. RUPE

Dr. Rupe was born in Topeka, Kansas, in 1922. His undergraduate
work was completed at Baker University in 1941 with the awarding
of his A. B. degree. He attended Washington University School of
Medicine and earned the M. D. degree in 1945. He served as a
medical officer in the United States Air Force from 1946 to 1948,
and then joined the staff of Henry Ford Hospital in Detroit as a
Resident in Internal Medicine. Since 1951 he has been Physician-
in-Charge of the Fourth Medical Clinic of Ford Hospital. He is an
Associate of the American College of Physicians.

BIOGRAPHY OF C. B. CRUMB, JR.

Mr. Crumb was born in Cincinnati, Ohio, in 1919 and attended primary and secondary schools in that area. He received his A. B. in 1941 and B. S. in Mechanical Engineering in 1942, both at Columbia University.

His principal professional work has been in electrical engineering at Los Alamos Scientific Laboratories; control systems and analog computers at Boeing Airplane Company; digital computers at Tally Register Corporation; and systems engineering and management at Bendix Aviation Corporation, Research Laboratories Division and Bendix Systems Division, Ann Arbor, Michigan.

With his wife and two daughters, he is now living near Seattle, Washington.

THE AUTOMATIC DIGITAL COMPUTER

AS AN AID IN MEDICAL DIAGNOSIS

by

C. B. Crumb, Jr.
C. E. Rupe, M. D.

I. INTRODUCTION

The idea to be presented here is that an automatic digital computer can
provide great assistance to the medical diagnostician by rapid calculation,
based upon symptoms and physical findings, of the relative probability
that a certain disorder fits that set of findings. In a given diagnostic
problem, the physician is practically always able to select the correct
category of disorders which will include the disorder which constitutes
the correct diagnosis. But it is also true that in many such categories
the distinction between the different disorders included is quite difficult.

This paper proposes that the incidence of correct diagnosis in such difficult
cases may be very greatly increased by the use of systematic statistical
correlation technique. Basically, the idea is that for each category of dis-
orders, the computer remembers a large number of cases with the correct
diagnoses and the complete pattern of associated symptoms and physical findings.
From these, it forms correlation constants between symptoms and

disorders, and is able when confronted with a new set of symptoms to bring to bear all of the case history evidence in selection of the disorder most likely to be the correct diagnosis for the particular case.

The need for the development of this technique is demonstrated by three conditions which apply to present day diagnostic procedures. The most important of these is the fact that up to this time medical diagnosis has remained, to a marked degree, an art. Although some scientific methods are utilized, the necessary practical limitations of laboratory and clinical procedures, plus the great complexity of many diagnostic problems have thus far prevented the full development of a science of medical diagnosis. This plan of statistical diagnosis with calculating machine aid should enable the medical doctors to move much closer to the "science" objective, and achieve a markedly increased incidence of correct diagnosis.

The other two justifications hinge on the cost of present day medical services. The computer technique should greatly increase the speed of reaching a decision in diagnosis, and thus should very significantly reduce the total diagnosis cost. Furthermore, in many instances, it may be that a clear cut differentiation reported by the digital calculator would make unnecessary further expensive and time consuming clinical and laboratory procedures which might now be employed to verify an initial diagnosis. The second area of cost saving would be avoidance of expensive hospitalization which might be rendered unnecessary by quick and accurate diagnosis.

There is a by-product virtue in this technique which should not be overlooked. This is that in order to use the technique, it is necessary first to record all symptoms and manifestations concerning the particular disorder in a systematic and uniform fashion. This is fed into the machine for the calculation of appropriate probabilities. That same information also becomes a part of the machine memory and of the foundation for correlation constants. Thus, the record on a particular case is not only made fully and recorded permanently, but quickly added to the body of statistical data upon which future diagnoses of cases in that category will be based. This represents a very marked improvement over present day case history recording, filing and utilization.

II. TECHNIQUE PROPOSED

Here we will use the term "symptom" in the broad sense to include all pertinent forms of evidence, -- patient's reports of symptoms, doctor's examination, laboratory test results, etc.

Assume a table of disorders D_a, D_b, D_c etc., and symptoms S_1, S_2, S_3, etc.; let's say we can derive a correlation constant C expressing the probability of association of each symptom with each disorder. Then, we will have a table like Table 1.

Now the simplest imaginable approach to the task is to note a set of observed and reported symptoms such as S_2, S_3, S_5, and S_8 applying to the case at hand; and for each disorder in the table obtain the total of the corresponding correlation constants. This process is illustrated in Table 2, which is Table 1 with the not-observed symptoms and corresponding constants deleted, and the totals indicated. The largest total N should be the correct diagnosis, or the one with highest correctness probability. Table 3 is the actual set of correlation constants used in our experimental work.

In the very limited testing of the idea which has been done thus far, this straightforward simple summation scheme has been tried and yielded the rather inconclusive results shown in Figures 1 and 2. It seems likely that some more sophisticated approach may have to be used for fullest effectiveness of the technique. For example, it appears that in some instances the correlation constant C must contain not only the occurrence correlation but also a factor representing the importance of the symptom in indicating the particular disorder, that is, whether or not it is a primary symptom having a clear cut relation to the disorder. It also appears that the intensity of certain symptoms should be noted in the preparation of the summation equations yielding the relative probability figures. For example, a mild abdominal pain might be considered a generally inconclusive evidence; while an intense abdominal pain localized in a certain area would be much more telling.

There may also be some more decidedly non-linear factors in this technique such as the existence of a symptom which positively eliminates one or more of the disorders in the appropriate category. Another possibility is that a certain combination of two or more symptoms may positively eliminate or positively identify a certain disorder within the appropriate category. A further example is the importance of history of symptoms in the preparation of the input to the computer. Thus the time rate of change of certain conditions occurring among the symptoms may be highly significant in the determination of the diagnosis.

In spite of all of the above possible changes, we feel strongly that the fundamental technique of summation of an appropriate set of correlation constants, modified or unmodified, will be very effective in indicating the most likely correct diagnosis.

III. MEDICAL FOUNDATIONS

The history of medicine shows a well established parallel of the strides in therapeutics and diagnostic methods. When the practice of medicine was confined largely to the relief of symptoms it often mattered little, except academically, whether or not the precise diagnosis was made so long as the resulting symptoms could be managed. In the current era, on the other hand, remarkable strides in specific treatments for diseases have greatly increased the need for improved precision in diagnosis.

Present day diagnosis is largely an intuitive and subjective process which is not easily described. Essentially it consists of the collection of data and interpretation leading to the synthesis of a diagnosis. The data collected by the patient's recounting of symptoms and the doctor's examination are interpreted in the light of the physican's training, experience, interest, and knowledge of published statistics. While the data collection process may be fairly uniform among various doctors, the interpretation in some areas of medicine is highly variable. These areas are the rare and unusual disorders.

Past efforts to attack that problem of variability in interpretation of symptoms have lead to the evolution of the practice called differential diagnosis. This is the technique of listing all the possible disease processes which might account

for the patient's symptoms and physical findings. Each of these possibilities

is individually related to the pattern of symptoms, and the diagnosis selected by

an elimination process. The big weakness of this technique lies in the fact

that the disease list must contain the correct diagnosis. If it is not there,

the further study of the patient is not oriented toward the discovery of the

proper diagnosis unless the various laboratory examinations happen to over-

lap the areas between the differential diagnosis and the correct diagnosis.

From this standpoint, then, the physician as a diagnostician, which he must be if

he is to choose precise therapy, is the repository of data from his training, his

reading of the current literature, and his day-to-day experience. This body of

knowledge to be applied at the bed side in deriving a diagnosis from symptoms

and signs suffers a few obvious disadvantages. First, the individual physician's

practice may or may not have contained experience with the unknown disease at

hand. Second, the individual physician's memory, not only of past experience

but of past training and current efforts to keep abreast of his field, may not be

adequate to the task. Finally, and much less frequently, the individual physician's

own interest in one area of disease may focus attention in this direction effect-

ively raising blinders to his diagnostic acumen - this in contrast to blind spots

in his diagnostic acumen that may always have been blank because of rarity of

disease or absence of experience.

In choosing a group of diseases to test the proposition of the use of simple statisti-

cal procedures in making diagnoses, it was felt that a group of diseases diagnosable

with certainty by some objective method was the first requirement. The second

requirement for the test group of diseases was that they have enough overlapping

symptomatology that diagnosis by history and physical examination presented some difficulties. The next problem was that of converting the subjective and intuitive information given by the patient to the physician to black, white, and reasonable shades of gray. Finally, and most difficult, is the construction of the statistical pattern of the disease to serve as a template with which to compare the individual patient's data. This particular difficulty serves to focus attention upon the need for the collection and collation of dependable statistics about the occurrence of major and minor manifestations of a given disease.

The diseases chosen are those involving the upper gastrointestinal tract and its appendages and all causing intermittent upper abdominal pain. The validity of the proposal was then ready for examination in the trial here reported. The construction of the statistical templates and the collection of the data from the individual patients for the trial pointed up additional problems such as the relative importance of individual symptoms and combinations of symptoms which bore greater weight in the consideration of one diagnosis versus others in the group. In this respect the attempts to objectify and "to mechanize" the subjective process will serve to further objectify diagnostic methods and procedures.

IV. STEPS IN DEVELOPMENT OF THE TECHNIQUE

The following is considered to be a logical sequence of steps toward evolving a technique and testing the idea.

1. Test Area Selection

   This is selection of a group of diseases or disorders which are
   symptomatically similar and difficult to diagnose; and for which
   a sizable body of data in the form of actual case histories exists.

2. Test Data Compilation

   A statistical number of cases from the literature on the selected
   category must be reviewed. The symptom data must be extracted
   in sufficiently uniform format to permit derivation of correlation
   constants.

3. Derivation of Constants

   This is the "dog work" of actually counting up the incidence of each
   manifestation as associated with each disorder within the selected
   category.

4. Test Computations and Evaluation

   First trials of the method should be made using the cases upon which
   the table was based. This enables development of the technique, as
   described in the following steps, without introduction of new variables.

5. Development of the Correlation Technique

   Here the so called nonlinearities or irregularities in the preparation
   and use of the correlation constant table will be tried and introduced
   as they are found to be effective in improving the resolution of the
   technique.

6. <u>Establishment of Data Format and Computer Programs</u>

   This is the adaptation of the process to high-speed machine use. The two major tasks are selection of an appropriate form for input information, and programming.

7. <u>Parallel Use With Conventional Diagnosis in New Cases</u>

   With no change in technique, doctors will make diagnoses as they are now being routinely made, while in parallel with that the computed diagnosis technique would be utilized and the effectiveness compared.

V. PROGRESS TO DATE

For the first trial of this idea, the authors selected a set of five abdominal disorders, -- cholelithiasis (gall stones), cancer of the stomach, peptic ulcer, hiatus hernia, and pylorospasm. In the hope of making an inexpensive first trial, we elected to derive the initial set of correlation constants by asking several specialists in internal medicine to report their estimates of the various constants. Each was given a blank table with the five disorders, each heading a column; and 48 rows for the 48 symptoms of significance in this set of disorders. The correlation constants which they reported were averaged to obtain a trial set of constants. These were then tested with a few case histories. The results were correct; but the difference between the correct and the incorrect probability index numbers was very little. Furthermore, in some of the manifestations, the doctors reported rather markedly different estimates of the appropriate correlation; and this cast doubt on the validity of the particular trial. It was evident that the rather laborious formulation of a true statistical basis would have to be performed.

This has been done, utilizing a statistical number of case histories for each of the five disorders. With the resulting table of constants, Table 3, the technique is now being tested on the same case histories. The first results are shown in Figures 1 and 2.

Figure 1 shows the result for 23 cases of hiatus hernia and 22 of peptic ulcer, in all of which the actual disorder was determined by X-ray. Each case is represented by a vertical column of the five different symbols. The vertical position of a symbol indicates N, the relative probability index number. The horizontal position for each column has no significance except that within each group the cases were plotted arbitrarily in descending order of highest N for the case. This places at the left the cases in which many symptoms were noted, proceeding to those with fewer symptoms at the right.

If the method were perfectly effective the flagged symbols would all be at the top of the mass in both sets. This is fairly closely approached in the peptic ulcer set; but with hiatus hernia the results are less encouraging. It is interesting to note that in about one-sixth of all the cases, the total for hiatus hernia was closely approximated by that of cholelithiasis. In all of these there was a history of gall bladder surgery or an established diagnosis of gall stones in addition to the hiatus hernia.

Figure 2 is an effort to summarize the most significant data of Figure 1. For each case the ratio of N for the correct diagnosis to the highest of the other four N values was computed; and the distribution of these ratio values is represented by Figure 2. Where this ratio $N_C : N_{H4}$ is greater than one,

the correct diagnosis has been defined by the technique. The cross-hatched bars represent groups of cases where the ratio value is less than one and the technique has failed.

These figures, based on very few cases, still make it pretty clear that refinements of the method, suggested in Section II, will indeed be essential.

VI. COMPUTER CONSIDERATIONS

The block diagram, Figure 3 shows the organization of a digital calculator considered suitable to this problem. Assuming for the moment that the machine would be used for the analysis of individual cases, one at a time, then the determination of the diagnosis probability index numbers would involve the following steps:

1. Place in the working storage of the computer the input information which consists of the disorder category, selected by the physician, and the manifestations noted and reported by him.

2. According to the disorder category, search the bulk storage for the appropriate table of correlation constants and transfer it to the working storage.

3. Modify any of the constants in the table according to any of the various nonlinear effects which were noted in Section II.

4. Extract the appropriate correlation constants under the heading of each of the disorders in the category, and accumulate the sum for each of the disorders.

5. Report the appropriate sum for each disorder.

The bulk storage requirement shown in the diagram might be satisfied by a magnetic tape unit, a large capacity magnetic drum, or a large capacity random access memory unit such as the I. B. M. Ramac. A typical table of correlation constants is expected to require perhaps one thousand decimal digits. Thus, a moderate capacity magnetic drum as working storage could easily accept one such table of correlation constants and a fairly detailed program.

If we look at the diagnosis operation as a whole, it becomes evident that the time requirement for manual preparation of the input information for the individual diagnosis is going to be of the order of several minutes. Thus, it would seem that allowance of two minutes for the total computer operation would not be excessive. On that scale, the actual arithmetic operations time becomes practically insignificant; and the most of the time can be allotted to the extraction of the appropriate correlation constants table and placing it in the working storage. If something like one minute and a half is allotted to this, then a magnetic tape servo should be sufficiently fast.

This line of thought leads to the conclusion that an inexpensive medium size general purpose machine with magnetic tape facility, such as the Bendix G-15, would be entirely appropriate.

It may turn out to be desirable to treat diagnosis problems in groups, by accumulation of a number of cases, prepared for machine input, and all to be handled in a continuous machine run. In this method, the machine time requirement per case may become much more important, and faster bulk storage, working storage, and arithmetic unit may be called for.

With such a small machine time requirement per case, the use of a single machine serving many hospitals, clinics, and independent doctors is a leading possibility. This scheme substitutes a communications requirement for the local machine requirement. By intelligent selection of coding and format, the information content of these messages will be kept low enough such that existing communications media such as teletype will be adequate. Thus it appears almost certain that the central computer service will be the most practical way of applying the technique.

You may note that the real problem in this project is not the computer, nor the program, but rather the development of statistical correlation procedures. That is true; and we admit that we haven't yet made use of an electronic calculator in the experimental work. From these facts one might conclude that high speed machine use is only incidental to the idea. Of course, that isn't so.

The statistical procedure proposed cannot be applied practically unless it can be performed rapidly. Thus in a sense we have a classic computer application, one in which the machine is not just a convenience nor a luxury, but rather an essential requirement. The job simply can't be done without it.

VII. CONCLUSIONS

No doubt it is quite evident from the limited progress which we reported in Section 5, that we are hardly in a position to state many conclusions on the effectiveness of this scheme. There are two statements which we can make in which we feel very confident. One is that, qualitatively at least, the scheme is definitely effective although some doubt still remains about the capability of the idea, as it is now visualized, to make a sharp and unmistakable differentiation of the correct from the incorrect diagnosis.

Our second conclusion is in a sense a by-product one arising from the observations during the first preparation of correlation constants. This is that the opportunity for improving the doctor's basis for diagnosis is greater than we expected. Present day diagnosis ideally is based upon all the information to which the diagnostician has been exposed, -- his total textbook and classroom content plus all the data from his daily practice. This ideal is not realized because of the limitations of the human memory, and the differences in constants reported by several internists suggested that the natural distillation of memory yields quite variable results in various men.

On the other hand, the digital computing machine can remember and utilize all the data it is given; and that fact is the foundation for our strong confidence in the need for and effectiveness of the proposed technique.

This is the real essence of our notion that here is another way that automatic computers can contribute to the good of mankind.

BIBLIOGRAPHY

Dr. K. Brodman, Dr. A. J. van Woerkom, Dr. A. J. Erdmann, Jr., and Dr. L. S. Goldstein, "Interpretation of Symptoms with a Data-Processing Machine," Reprinted from the A. M. A. Archives of Internal Medicine, May 1959, Vol. 103, pp. 776-782.

R. S. Ledley and L. B. Lusted, "Reasoning Foundations of Medical Diagnosis," Science, Vol. 130, No. 3366, pp. 9-21, July 3, 1959.

D. M. Cleary, "The Role of Automation in Diagnostics," Current Medical Digest, April 1959, pp. 57-63.

"Dr. Automation," Time, Vol. 74, No. 6, p. 48, August 10, 1959. On work at System Development Corporation under C. J. Roach.

M. Lipkin and J. D. Hardy, "Mechanical Correlation of Data in Differential Diagnosis of Hematological Diseases," Journal of American Medical Association, Vol. 166, No. 2, pp. 113-125, Jan. 11, 1958.

C. L. Chiang, J. L. Hodges, Jr., and J. Yerushalny, "Statistical Problems in Medical Diagnosis," Third Berkeley Symposium on Mathematical Statistics and Probability, 1954-1955, Vol. 4, pp. 121-123, University of California Press, Berkeley, 1956.

"Dr. Robot," Time, Vol. 68, p. 44, August 20, 1956. Punched card machine diagnosis of diseases of cornea, by Dr. Francois Paycha.

Charbonnier, Cyffers, Schwartz, and Vessereau, "Discrimination entre ictères médicaux et chirurgicaux a partir des résultats de l'analyse électrophorétique des protéines du sérum," Revue Internationale d'Hepatologie, Vol. 5, No. 7, 1955.

C. F. Paycha, "Memoire diagnostique," Montpellier Med., Vol. 47, p. 588, 1955.

FIGURE LIST

TABLE I

FORM OF THE TABLE OF CORRELATION CONSTANTS

DISORDERS →	D_A	D_B	D_C	D_D
SYMPTOMS ↓				
S_1	C_{1A}	C_{1B}	C_{1C}	C_{1D}
S_2	C_{2A}	C_{2B}	C_{2C}	C_{2D}
S_3	C_{3A}	C_{3B}	C_{3C}	C_{3D}
S_4	C_{4A}	C_{4B}	C_{4C}	C_{4D}
S_5	C_{5A}	C_{5B}	C_{5C}	C_{5D}
S_6	C_{6A}	C_{6B}	C_{6C}	C_{6D}
S_7	C_{7A}	C_{7B}	C_{7C}	C_{7D}
S_8	C_{8A}	C_{8B}	C_{8C}	C_{8D}

TABLE 2

PROCESS OF OBTAINING RELATIVE PROBABILITY INDEX NUMBERS

DISORDERS →	D_A	D_B	D_C	D_D
SYMPTOMS ↓				
S_2	C_{2A}	C_{2B}	C_{2C}	C_{2D}
S_3	C_{3A}	C_{3B}	C_{3C}	C_{3D}
S_5	C_{5A}	C_{5B}	C_{5C}	C_{5D}
S_8	C_{8A}	C_{8B}	C_{8C}	C_{8D}
	N_A	N_B	N_C	N_D

TABLE 3

COMPUTED DIAGNOSIS TEST _____ Case Number _____

True Diagnosis _____

	Symptoms Reported	Stomach Cancer	Choli- thiasis	Gas- tritis	Hiatus Hernia	Peptic Ulcer	Pyloro- spasm
PAIN LOCATION:							
Epigastrium	1	55	45		60	94	85
Right upper quadrant	2	15	76		52	40	10
Radiation to scapula	3	5	43		0	28	0
Radiation to back	4	15	28		7	28	5
PAIN RELIEF BY FOOD	5	0	0		20	75	12
PAIN AGGRAVATED BY FOOD	6	57	42		50	10	43
PAIN AGGRAVATED BY LYING DOWN	7	0	0		78		3
PAIN TIMING:							
Worse in a.m.	8	10	3		70	0	18
Odd infrequent interval attacks	9	10	62		20	0	86
Under 1 hour post prandial	10	65	22		80	5	68
Over 1 hour post prandial	11	47	83		20	95	10
FOOD INTOLERANCE:							
Spices	12	20	7		0	15	33
Cabbage family	13	10	28		0	5	37
Gravy	14	25	57		0	10	28
Fried Foods	15	40	60		0	10	30
Belching	16	14	25		50	65	50
Flatus	17	17	8		70	50	23
Jaundice	18	3	27		0	0	0
Fever	19	5	37		0	0	0
Constipation	20	28	10		0	50	32
Diarrhea	21	4	20		0	2	35
Pellet stools	22	0	10		0	20	10
Nausea	23	65	75		60	35	70
Vomiting	24	53	67		57	68	30
Anorexia	25	22	20		70	50	78
Multiple children	26	0	37			0	0
Obesity	27	0	35		39	5	13
Sex: Male	28	78	40		40	75	47
Female	29	22	68		60	25	62
Alcoholism-Under 4 oz. per day	30	0	62		0	0	62
AGE GROUP: Under 20	31	2	7		1	15	13
20 - 40	32	10	21		7	62	62
40 - 65	33	80	56		84	20	25
Over 65	34	8	13		8	3	10
Emotional stress related to Sx	35	0	0		0	30	80
PHYSICAL EXAM:							
Tender epigastrium	36	55	32		15	75	40
Hyperesthesia epigastrium	37	0	20		10	35	10
Tender right upper quadrant	38	0	70		5	40	15
Abdominal mass	39	28	18		0	0	3
Laboratory: Anemia	40	30	7		10	0	7
Leucocytosis	41	5	55		0	0	0
B.D.	42	0	62		0	0	0
Hyperacidity	43	0	10		10	90	18
Hypoacidity	44	84	8		0	0	8
X-RAY: Chest	45	0	0		15	0	0
Gallbladder	46	0	94		25	0	10
Stomach	47	91	15		81	95	50
COEFFICIENT TOTAL							

FIGURE I
PROBABILITY INDEX NUMBERS FOR
45 CASES

FLAGGED SYMBOL INDICATES CORRECT DIAGNOSIS

- □ STOMACH CANCER
- ○ CHOLILITHIASIS
- ● HIATUS HERNIA
- ▲ PEPTIC ULCER
- △ PYLOROSPASM

23 CASES OF
HIATUS HERNIA

22 CASES OF
PEPTIC ULCER

— 700 —
— 600 —
— 500 —
— 400 —
— 300 —
— 200 —

INDEX
NUMBER

FIGURE 2

DISTRIBUTION OF $\frac{N_c}{N_{H4}}$

23 CASES OF HIATUS HERNIA

22 CASES OF PEPTIC ULCER

FIGURE 3

MACHINE ORGANIZATION

$$\begin{pmatrix} \text{TABLES OF CONSTANTS} \\ \text{FOR ALL CATEGORIES} \end{pmatrix}$$

```
                    ┌──────────────┐
                    │    BULK      │
                    │   STORAGE    │
                    └──────────────┘
                          ↑  ↓
┌──────────────┐    ┌──────────────┐    ┌──────────────┐
│   WORKING    │ →  │   CONTROL    │ →  │  ARITHMETIC  │
│   STORAGE    │ ←  │    UNIT      │ ←  │    UNIT      │
└──────────────┘    └──────────────┘    └──────────────┘
                          ↑  ↓
```

$$\begin{pmatrix} \text{SYMPTOM} \\ \text{LIST AND} \\ \text{SELECTED} \\ \text{CATEGORY} \end{pmatrix} \qquad \begin{pmatrix} \text{RELATIVE} \\ \text{PROBABILITY} \\ \text{NUMBERS} \end{pmatrix}$$

```
                    ┌──────────────┐
                    │    INPUT-     │
                    │   OUTPUT      │
                    └──────────────┘
```

November 9, 1959

Mr. Harlan E. Anderson
E.J.C.C. Publication Committee
Digital Equipment Corporation
Maynard, Massachusetts

Dear Mr. Anderson:

In accordance with your request, I enclose herewith four copies of my paper titled "The Automatic Transcription of Machine Shorthand" which will be presented at the Eastern Joint Computer Conference next month. I also enclose a 100 word biography.

I have enclosed only one copy of each figure. All figures are glossy prints with the exception of figures 3 and 4 which have been typed. If you require additional prints of the figures I would appreciate your letting me know. I can be reached at UN.8-7600, extension 570.

I shall be using standard 3-1/4" by 4-1/4" slides.

Thank you for your attention.

Sincerely yours,

Gerard Salton
Instructor in
Applied Mathematics

GS:lm

THE AUTOMATIC TRANSCRIPTION OF MACHINE SHORTHAND

Gerard Salton

Computation Laboratory, Harvard University

ABSTRACT

The transformation of speech input into written output is an important part of many data-processing problems. Since digitalized speech input cannot be readily recognized, it is suggested that machine shorthand notes might be usable instead as input to a computer process for the automatic production of written output.

The basic theory of machine shorthand is briefly described. Thereafter, the principal steps needed for the transcription of machine shorthand are discussed. Many problems arising in the transcription process are found to be similar to problems of interest in the automatic translation of languages and in the analysis of speech. In particular, the problems posed by multiple correspondences and the resulting requirement for syntactic and semantic analysis of the translated texts also arise in automatic translation; on the other hand, the correct determination of word boundaries which is not normally a problem in language translation is a necessary element of the transcription of machine shorthand as well as of speech analysis. Computer routines for the production of comprehensible, although not syntactically or semantically unambiguous, English are described and computer output is demonstrated. Suggestions are made for improving the quality of the English output.

THE AUTOMATIC TRANSCRIPTION OF MACHINE SHORTHAND*

Gerard Salton†

Introduction

The recognition of digitalized speech signals is an important data processing problem, still completely unsolved. In fact, speech is the basic input to many data processing problems, and before any processing can take place, the speech signals must first be transformed into digital form, recognized, and identified. Typical applications include: the processing of court testimony, the recording of data in medical practice, the handling of business correspondence, and the monitoring of radio broadcasts.

In spite of the fact that considerable research has been conducted within the past few years,[1,2,3,4,5,6] the problems which arise in the recognition of speech are not, as yet, close to solution. For this reason, it must be expected that speech cannot in the near future be made available automatically as an input for data processing purposes.

Since stenographers are trained to record spoken information at high speed, it may be possible to solve a substantial part of

the problem of transforming spoken information into written form by using a stenographic transcript as machine input. Indeed, stenographic notes are written in standard form and rules exist for transforming the phonemic stenograms into the standard written language. Machine shorthand is the most useful form of input in this connection, since the problem of recognizing handwritten characters is eliminated. Moreover, in machine shorthand, the notes are furnished on a paper tape which may serve as a direct input to a computer handling the translation process. A study of methods for transcribing machine shorthand is thus of interest from a practical point of view to speed up the production of written output, and also because it provides opportunities for the analysis of both the written and the spoken language.

The Stenotype Machine

Most machine shorthand methods presently in use are based on a machine developed by a shorthand reporter, W. S. Ireland,[7] in the beginning of this century. The machines resemble miniature typewriters and are marketed under such names as "Stenotype" or "Stenograph."

The keyboard developed by Ireland comprises 22 keys and a numeral bar. As each key is depressed, a character corresponding to that key is printed on a paper tape and a clutch driven roller

advances the tape. The various keys can be depressed either singly
or together; if depressed together, the corresponding characters
all print on the same line on the tape. During actual operation
an average of five or six characters are thus written across the
width of the tape in a single stroke. Unlike an ordinary typewriter,
the stenotype machine does not have a movable carriage; therefore,
the characters always print in the same relative position on the
tape when the corresponding keys are depressed on the keyboard.
The paper tape folds into a pull-out tray at the rear of the device.
The keys are unmarked and writing is done entirely by touch as in
ordinary typewriting. A sample stenotype tape is shown in Fig. 1.

The standard stenotype keyboard is shown in Fig. 2(a) and
the fixed printing order is shown in Fig. 2(b). If the numeral bar
is not depressed, the lower case character shown on each key is
written; if the numeral bar is depressed, a decimal digit will print
in some instances while the lower case character is printed unchanged
in others. Thus for 12 of the 22 character positions across the
width of the tape, the character appearing on the tape is unique,
while for the remaining ten positions two possible characters can
be printed in each position.

The 16 different letters of the alphabet which can be
printed by means of the 22 keys are arranged in three groups: a
set of seven consonants on the left-hand side of the keyboard, a
set of four vowels and an asterisk in the middle, and a set of ten

consonants on the right-hand side of the keyboard. In order to keep the keyboard small and the fingering simple, ten letters of the alphabet are not provided at all and must therefore be represented by other letters or combinations of letters.

Basic Rules of Stenotypy

Before describing the problems which arise in the transcription of stenotypy, the basic rules of machine shorthand will be briefly reviewed.[8,9] While there exist many variations of the basic theory,[10,11] resulting partly from the preferences of individual reporters and stenographers, and partly from the desire to cope with specialized subject matter and to increase reporting speed, all such variations are based on a common set of rules. It is this standard theory which is described here.

A. Consonant and Vowel System

In machine shorthand, consonants are generally written according to sound. Of the 24 consonant phonemes which are recognized in most English dialects,[12] 22 can occur initially in English words and 21 can occur finally. Direct representations are assigned on the stenotype keyboard for seven initial and for nine final consonant phonemes. Special letter combinations are used to represent those phonemes which are not available directly on the keyboard. The character combinations actually used are assigned for ease of

fingering, rather than for economy in the number of keys which must be depressed.

The vowel system is more complicated in that vowels are written partly according to spelling and partly according to sound. A single vowel occurring in a word is generally written in accordance with the English spelling. Thus the phoneme /əh/ is reproduced as the letter E in HER and as the letter U in BURN. A vowel cluster is generally transcribed as a single vowel if only one vowel in the cluster is prominently sounded. Thus one writes BOT for "bought" or "boat," and BET for "beat" and "beet." Some vowel clusters do not follow the general rule: for example, the cluster "oo" as in "book" is transcribed AO, and the phoneme /ɔj/ as in "noise" is transcribed OEU.

Words are formed in stenotyping by merely juxtaposing the transcriptions of the individual phonemes. Thus,

"one" is written WOPB (WON),

"queer" is written KWER,

and

"tax" is written TABGS (TAKS).

B. The Elimination of Strokes

Since only one vowel cluster can be printed on any given line, it is in general not possible to write more than one syllable per stroke. Thus, the word "interviewed" is written

EUPB = TER = SRUD (IN = TER = VUD)

in three strokes, the equality sign denoting the separation of
strokes.

Furthermore, because of the fixed order of the letters on
the keyboard, some words require more than one stroke even though
only one vowel cluster is present. Thus it is necessary to use
two strokes for words such as

TH = WART   (TH = WART),

PWRAPB = FP   (BRAN = CH),

and

TKPWOL = F   (GOL = F).

Theoretically, at least two strokes should thus be required on the
average to render a given English word on a stenotype tape. In
practice, however, the number of stenotype strokes is only slightly
larger on the average than the number of English words.

The reduction from the expected number of strokes to the
actual number of strokes is achieved by a set of rules designed to
save strokes. Seven principal rules are used to eliminate strokes:

(1) Special keyboard letter combinations are assigned to
some phoneme clusters such as /ŋk/ and /nʒ/ which
would normally require more than one stroke.

(2) Certain high frequency consonant clusters ending in
"t" cannot be written in one stroke because of the
arrangement of the letters on the keyboard or because
the required fingering would be too uncomfortable.
For this reason final T is omitted in words ending in
/st/, /kst/, and /kt/.

(3) Completely silent letters are treated as redundant and are therefore not written.

(4) An "unimportant" or "unaccented" vowel occurring in the middle of a word may be omitted if a stroke is thereby saved. Vowels which are not essential to the understanding of a word are therefore generally not written. Thus,

"easily" is written ES = HREU (ES = LI),

"believe" is written PWHREF (BLEV),

and

"commit" is written KPHEUT (KMIT).

(5) Special letter combinations are assigned to some high frequency affixes. In particular, special combinations are assigned to the prefix "ex" and to the suffixes "ing," "ity," "ment," "sion" and "tion," and "ction."

(6) Special abbreviations are used to represent about 300 high frequency words. These abbreviated forms can be used separately or as part of longer derived words. For example:

TA stands for "take,"

but

PHEUS = TA (MIS = TA) stands for "mistake."

Similarly,

TKEU (DI) stands for "difficulty,"

but

TKEUS (DIS) stands for "difficulties."

(7) Certain phrases consisting of more than one high-frequency word may be written in one stroke by juxtaposing the corresponding abbreviations. Thus in a single stroke one writes phrases such as

THAUFB --> that you have been,
(THA   U   F   B)

SWELS --> as well as,
(S WEL S)

and

KWRUR --> why you are
(KWR U   R) .

The Problems of Stenotype Translation

A number of problems must be considered if a stenotype input text is to produce usable English output. First of all, it is necessary to find procedures for mechanizing the input of stenotype texts. Second, methods must be sought for dealing with the large number of multiple English correspondents which are generated in the translation process. Finally, it is desirable to restitute correct English word forms for the mutilated forms which are derived. These problems will be dealt with in order.

A. The Input Problem

The paper tape prepared by the stenographer should clearly be used as a direct input medium during an automatic transcription process. Two main procedures are available for this purpose. The first one consists in modifying the stenotype machine in such a way that holes are punched into the tape between adjacent lines of

printed information. The tape can then be treated in the same manner as ordinary punched paper tape; at the same time, the printed information can still be read as before. The second possible procedure consists in using the characters printed on the tape directly, by means of some character recognition process. This second method requires no equipment modifications in the recording device itself.

For the twelve character positions which are associated with a unique character it is sufficient to detect a hole immediately above the given character position on the tape. The presence of the hole will indicate the presence of the given character on the line printed immediately below. Alternatively, in the character recognition procedure it is sufficient to discriminate between the presence and absence of a character in the given character position. If a character is present, it will then necessarily be that character which is uniquely assigned to the given position on tape.

Each of the remaining ten character positions may contain either of two different characters. The character recognition process must therefore recognize two possible characters in each of these tape positions. This is a relatively simple system in comparison with many other recognition systems now operating, and no insurmountable technological difficulties should arise.

If a punched paper tape system is used, it is sufficient to detect the presence of decimal digits across the tape by using

one extra hole position. If the extra position is punched, the
characters on the line immediately below will be recognized as
decimal digits in all positions where a possible conflict exists.
If the extra hole position is left unpunched, only alphabetics will
be recognized. The required decoding equipment is again extremely
simple.

B.   The Decoding of Abbreviations, Phrases, and Affixes

The number of abbreviated forms in general use is rela-
tively small, of the order of several hundred. It is therefore
possible to construct a table or dictionary of these forms and to
obtain the English correspondents by a table look-up operation.
In principle, the table look-up operation presents no difficulties.
In practice, however, certain ambiguities may have to be resolved.
Indeed, many abbreviations have a number of different correspondents
in English and it is not always easy to discriminate between them.
For example, TPOR is an abbreviation for both "inform" and "infor-
mation"; similarly, W stands for both "were" and "with."

The difficulty is often compounded by the fact that many
clusters may be abbreviated forms in some contexts, but not in
others. Thus, the previously cited TPOR is transliterated "FOR"
and may, therefore, correspond also to "four" and "for." Similarly,
UL is the abbreviated form for "you will" since U stands for "you"
and L stands for "will"; however, in some contexts it may be trans-
lated as merely "ul" as, for example, in TPEBG = UL = EU (effectually).

Phrases are strings of abbreviated forms which may be written in one stroke. The dictionary of abbreviations can be used for the decoding of phrases, provided each phrase is properly partitioned before the table look-up operation. Consider as an example the phrase URBGT. If it is partitioned U/RB/G/T the table look-up operation will produce the correspondents

$$U \longrightarrow you$$
$$RB \longrightarrow shall$$
$$G \longrightarrow go, gone$$
$$T \longrightarrow it, the$$

and the most reasonable translation would be "you shall go the"; if, on the other hand, it is partitioned UR/BGT, the look-up operations produce

$$UR \longrightarrow you are, your$$
$$BGT \longrightarrow account$$

and the correct translation "your account" is then obtained.

While each phrase can generally be partitioned in many different ways, a preferred partitioning method can normally be found by storing certain phrases or parts of phrases in the dictionary, and by using the stored information during the translation process.

The number of affixes in general use is small. However, the decoding of affixes can present considerable problems. In particular, a letter cluster denoting an affix may generally also

stand for a normal word ending, or it may represent an abbreviated form. Consider, for example, the letter G, denoting the suffix "ing." It may be variously translated as "g," "ing," "go," or "gone," or it may be a part of a longer letter cluster requiring still another translation.

Furthermore, a certain amount of conflict is inherent in the assignment of letter clusters to the various suffixes. Thus, the suffix "tion" is represented by "GS"; since G also stands for the suffix "ing," the ending GS can also be decoded as "ings."

In general, it is possible to distinguish three types of ambiguity resulting from the decoding process. First, the generation of improper English word forms as, for example, the translation of OFRGS by "offertion" instead of "offerings." This problem will be considered in the next section. The second is the generation of phrases which are made up of morphologically correct word forms, but are nevertheless syntactically improper. Such syntactically incorrect phrases would result, for example, from the replacement of "issue" by "I shall you," or of "a chief" by "achieve." The last is the generation of semantic ambiguities which cannot be resolved merely by syntactic analyses. Specifically, it is not possible to use English syntax to distinguish "sport" from "export," or "annex" from "annection."

Syntactic difficulties are generally easier to resolve than semantic problems. The former can to some extent be handled by

building a dictionary or table including grammatical designations.
Some sequences of grammatical forms will then be recognized as
legitimate in English, while others will not. Semantic ambiguities,
on the other hand, must be resolved by considering the subject
matter being analyzed. This, in turn, might be attempted by gener-
ating and later identifying certain topic phrases or topic sentences
by means of frequency analyses of the stenotype texts.[13,14] While
frequency studies are helpful in analyzing semantic content, it is
doubtful whether all semantic problems can be easily eliminated.

C. The Transliteration of Stenotype Clusters and the
   Determination of Word Boundaries

One of the required steps in the automatic transcription
of stenotype texts is the substitution of English letters for the
stenotype clusters. For most clusters, this step is mechanical,
since the clusters correspond in general to a unique character in
English. For example, PWEPF will be uniquely transformed into
"BECH" since the transformations

$$PW \rightarrow B,$$

$$E \rightarrow E,$$

and

$$FP \rightarrow CH$$

are uniquely determined.

A few clusters, however, cannot be identified uniquely
without use of the context. Thus SHR corresponds either to "sl"

or to "shr" and both translations are admissible in some cases. For example, SHRED might correspond to "shred" or "sled." Ambiguous clusters of this type are fortunately rather rare.

The transliteration procedure does not restitute missing letters nor restore the correct English spelling. A first approach to this problem might be provided by the use of frequency tables for both stenotype and English digrams and higher order strings. These tables could be used to translate a given stenotype letter cluster into the corresponding English letter cluster with the highest frequency count. In most cases, the correct English correspondent would then be furnished.

In some cases, of course, the frequency tables would not avail, since the most probable correspondent is not always the correct correspondent. This is the case particularly for those words which require more than one stroke to be rendered on the stenotype tape. For example, words like

PA = SHEPBS (PA = SHENS)   ⟶ patience

or

EUPB = TKUS = TREUL (IN = DUS = TRIL) ⟶ industrial

cannot properly be generated only by a frequency analysis of the individual strokes. Moreover, there are many stenotype strokes which may legitimately have more than one correspondent in English and in general, frequency tables cannot be used to make a choice among these correspondents. Consider, for example,

HEUR standing for { hire
                    higher,

A = CUT standing for { a cut
                       acute,

and

PWAR standing for { bar
                     bear
                     bare.

Only the last two words are legitimate homonyms in English. The
other pairs are artificial homonyms which are created in stenotyping
because the language is less redundant than English.

Thus, while digram and trigram frequency tables would prove
helpful in the restoration of English word forms, accurate tables
for longer strings of characters would almost certainly be required
to solve the problem for a large number of strokes. A dictionary
of high-frequency multistroke words could also be used to good
advantage in the recognition of word boundaries. However, since
it is often possible to split English words into stenotype strokes
in several different ways, all possible forms of these multistroke
words would have to be listed in the dictionary.

To summarize, the problems raised by the transcription of
machine shorthand are identical to problems encountered in the
automatic translation of languages with two important exceptions.
In machine shorthand, it is not necessary to change the word order
of the input text, or to alter a sentence by insertion or deletion
of words; idiomatic expressions do not therefore cause any diffi-
culties. On the other hand, in language translation it is not

necessary to generate correct word forms by altering the spelling of words, or to recognize word boundaries. A comparison of the various translation problems is shown in Fig. 3.

The Production of Pseudo-English

As a first step in the analysis of machine shorthand, computer routines were prepared for the production of pseudo-English, that is, comprehensible, although syntactically and semantically ambiguous English. In many applications, such as the monitoring of radio broadcasts or the use of machine shorthand in medical practice, pseudo-English might be an acceptable form of final output; in other applications, imperfect English would be undesirable. However, even then, pseudo-English is a useful intermediate output since it can be used to update the dictionaries, to refine the partitioning methods, and to derive rules for syntactic analyses. Since English is inherently redundant, understandable output sentences can be produced with comparatively little difficulty.

A variety of methods are used to produce pseudo-English. The stenotype strokes are first partitioned into "constituent" pieces using only the structure of each stroke as the criterion. Two short dictionaries are used to look up the correspondents of these constituents and to handle the transliteration process. The word pieces are later reassembled into complete English words or phrases.

No attempt is made to reduce ambiguity by making a choice among many possible English correspondents. Thus, if a given stenotype stroke can reasonably correspond to many different English letter clusters or words, all correspondents will be listed, and the reader is required to choose the most reasonable correspondent using the given context as a guide.

Similarly, the correct English spelling of words not found in the dictionaries is not always restored. Instead, the abbreviated word form, excluding silent letters and unaccented vowels, is used in the transcribed text. This reduces the aesthetic quality of the output although comprehension is not generally impaired.

A.   Analysis of Stenotype Stroke Forms

Depending on the classification of a given stroke, one of five basic translation methods must be used to transform it into the English equivalent.

(1)   To decode abbreviations the complete stenotype stroke must be looked-up in a dictionary of abbreviations; one or more correct English word forms will then be determined for each abbreviation. For example,

T $\longrightarrow$ the, it.

(2)   Phrases must first be partitioned into two or more components, and each component must then be looked-up separately in a dictionary. The juxtaposed English correspondents will then represent the English phrase. For example,

TPHAB $\longrightarrow$ it may be,

since

$$T \longrightarrow it$$

$$PHA \longrightarrow may$$

$$B \longrightarrow be.$$

An analysis of stenotype phrases indicates that many phrases can be translated correctly if partitioned between the middle vowel string and the final consonant string. The above example would then be partitioned into the components TPHA and B, and the English correspondents "it may" and "be" would be found in the dictionary of abbreviations and phrases.

(3) <u>Derivatives of abbreviations</u> are first partitioned to isolate the affix. The correspondent of the complete stroke excluding the affix is then determined by dictionary look-up and the English correspondent of the affix is added. For example,

$$WREUG \longrightarrow write\text{-}ing,$$

since

$$WREU \longrightarrow write$$
$$G \longrightarrow ing.$$

Here WREU is the abbreviation for "write" and G is the suffix.

(4) Stenotype strokes which are not abbreviations or phrases are rendered in English by transforming the stenotype letter clusters into corresponding English letters. The transliteration procedure uses a dictionary of stenotype letter clusters including many of the consonantal and vowel patterns used in English. In order to use the dictionary of letter clusters, it is necessary to partition each stenotype stroke into

initial consonant string, middle vowel string, and final consonant string. For example,

$$KHRERBG \rightarrow cl\text{-}e\text{-}rk,$$

since

$$KHR \rightarrow cl$$

$$E \rightarrow e$$

$$RBG \rightarrow rk.$$

It should be noted that many stenotype clusters appear both in the dictionary of letter clusters and in the dictionary of abbreviations. However, the English correspondents are then necessarily quite different. As an example, it was seen that T is an abbreviation for "it" and "the"; in the dictionary of letter clusters, the correspondents are "t" and "th."

(5) Ordinary stenotype strokes which include a suffix but are not stenotype abbreviations are treated like the strokes of category (4), except that they are partitioned into four parts before look-up in the dictionary of letter clusters: initial consonant string, middle vowel string, final consonant string excepting the suffix, and suffix. As an example,

$$TKPWEUFG \rightarrow giving,$$

since

$$TKPW \rightarrow g$$

$$EU \rightarrow i$$

$$F \rightarrow v$$

$$G \rightarrow ing,$$

G being again the suffix corresponding to "ing."

Since it is not clear _a priori_ how a given stroke is to be classified, every stroke is submitted to the procedures required for all five categories; a correct translation will then necessarily be determined for each stroke. The strokes are first classified as belonging to one of ten _types_. The stroke type depends only on the combination of letter clusters which form the stroke. Specifically, the stroke type indicates whether the corresponding stroke terminates in one of the recognized suffixes and shows which combination of initial consonant, middle vowel, and final consonant strings constitutes the stroke. For example, stroke type 0 represents strokes made up of a vowel string only, while stroke type 2 represents strokes made up of an initial consonant string followed by a vowel string. Depending on the type, each stroke is then partitioned into constituent _parts_. Eight stroke parts including the complete stroke are recognized as shown in Fig. 4. The English correspondents of the various parts are then looked-up in the dictionary of abbreviations and phrases, and in the dictionary of letter clusters. A third dictionary or table is required to handle suffixes.

After the table look-up operations, the English correspondents of the stroke parts must be reassembled to form complete words or phrases. For example, to form complete phrases (category (2) above) stroke parts C and F_1 must be reassembled; similarly, for ordinary transliterated words (category (4)) parts I, V, and F_2

are reassembled. This will be explained further in the description of the machine program.

B. The Machine Program

The Univac I computer at the Harvard Computation Laboratory was used for the production of the pseudo-English output. In order to reduce the programming load, it was desirable to make use of existing machine programs. In particular, it was found that many programs originally written to handle various phases of Russian to English automatic translation could be used for the transcription of machine shorthand.* Some of these programs required small

---

* Several programs written originally by members of the Harvard Automatic Translation Project under sponsorship of the National Science Foundation were used; the writer is indebted to Prof. A. G. Oettinger and to the other project members for having made these programs available.

---

alterations; however, compared with the work required to write new programs, the effort expended in making the required changes was negligible.

In order to construct the two dictionaries, use was made of existing stenotype manuals. The present dictionary of abbreviations includes about 700 stenotype strokes arranged in alphabetical order, and approximately 2000 English correspondents. Punctuation marks are identified by the letter P following the stenotype

cluster, and English correspondents by the digits 1, 2, 3, and so on; a percent (%) sign is used for the last correspondent. An excerpt from the dictionary of abbreviations is shown in Fig. 5.

The dictionary of letter clusters includes only clusters whose English correspondents are different from the stenotype original. To construct this dictionary, lists of consonantal and vowel patterns used in English were consulted;[15] about 100 letter clusters arranged in alphabetical order are included. An excerpt from the dictionary of letter clusters is shown in Fig. 6.

The various steps used to produce pseudo-English are shown schematically in Fig. 7. Each box corresponds to one machine run. Only the partition program and the assembly program shown by double lines were newly programmed.*

---

* The partition and assembly programs were coded by Mr. Rodney W. Thorpe, whose valuable assistance is gratefully acknowledged.

The stenotype text to be translated is first transferred onto magnetic tape. Since equipment for automatic input does not exist at present, the operation is manual. The text is typed like ordinary linear text with spaces between stenotype strokes. English comments are placed between dollar ($) signs to conform with the requirements of the existing programs.[16] No special punctuation marks are used in stenotyping except for a double asterisk to indicate a new paragraph. Instead, punctuation marks are denoted

by special letter clusters which are included in the dictionary of abbreviations. An excerpt of a linear input text is reproduced in Fig. 8.

The itemize program[16] is used first to assign serial numbers to the individual text words. Each text word is transformed into a separate 5-word item. A part of a linear input text is shown in itemized form in Fig. 9. The four low-order digit positions of each serial number, filled with zeros, are later modified during the partitioning process.

The partitioning program uses the itemized text as input and generates a variable number of 5-word items for each input item. Each input stroke is first partitioned into initial consonant, middle vowel and final consonant strings, and a code representing the stroke type is assigned to the complete stroke. A test is also made for the presence of one of the ten recognized suffixes. If a suffix is found the proper suffix number is assigned. The various stroke parts which constitute a complete stroke are then recorded on one of two output tapes (the final consonant string F is actually recorded on both tapes). The dictionary of abbreviations and phrases will be searched during the dictionary look-up routine for the English correspondents of the stroke parts of tape 1, while the dictionary of letter clusters is used for the stroke parts of tape 2. For complete strokes (initial consonant — middle vowel — final consonant clusters) with affix, four stroke parts (W, G, F, S)

are recorded on tape 1 and four stroke parts (I, V, F, E) are recorded on tape 2. If the stroke is truncated (e.g., if the initial or the final consonant string is missing) or if no affix is present, a smaller number of stroke parts will be recorded out. All stroke parts are recorded with a modified serial number to show the stroke type and the stroke part represented by each individual item.

The two tapes are now processed separately through the same sequence of programs. The items are first prepared for the dictionary look-up by an alphabetic sort. The dictionary look-up routine will then expand each 5-word item into a 30-word item and add to each stenotype input stroke the English correspondents from the dictionary. The dictionary of abbreviations is used for the items of tape 1, while the dictionary of letter clusters is used for tape 2. For strokes not found in the dictionary, a dummy 30-word item is generated. The 30-word items are finally sorted back into text order, using the new serial number as a sorting key. The alphabetic sort and the dictionary look-up programs were taken over unchanged from the "Continuous Dictionary Run" used for Russian-English translation.[16]

The next program is a two-way merge program using the serial number as a key and tapes 1 and 2 as inputs. One merging pass is sufficient to transfer all 30-word items onto a single output tape. The merge program used is one of the existing Univac system routines.[17]

The assembly program is now used to assemble those 30-word items which do not correspond to a complete text word. Both the stroke type and the dictionary information obtained during dictionary look-ups are used during the assembly phase to generate complete English correspondents. The stroke type indicates whether it is necessary to add a suffix to the English correspondents found in the dictionary and shows how many items are to be assembled to form complete English correspondents. For example, if the original stroke consisted of an initial consonant and a middle vowel string, the English correspondents of two items must generally be assembled; if a final consonant string was also present, three items are normally assembled.

Up to five different types of assemblies are made. For abbreviations, the English correspondents of the W item are taken intact; for phrases, the correspondents of the C and F_1 items are assembled; for derivatives of abbreviations, the affix A is attached to the correspondents of the S item; for ordinary strokes, the correspondents of the I, V, and F_2 items are attached; finally, for ordinary strokes which include a suffix, an affix A is attached to the correspondents of the assembled I, V, E items. These five types of assemblies are, however, not made for all strokes; the dictionary information obtained during the look-up operation is used to decide whether a given item is to be assembled or not.

Specifically, if the W item is contained in the dictionary, then the correspondents which would normally be generated by assembling

the C-F, S-A, and I-V-E-A items are already contained in the W item, so that no need arises to assemble these other items. Similarly, if C and F_1 are not both in the dictionary, the C-F_1 item is not assembled since this item could not then be an English phrase. Finally, if S-A is to represent an abbreviation followed by a suffix, the S item must have been found in the dictionary; the S-A item is then assembled only under these conditions. The I-V-F_2 item (and I-V-E-A item if different from I-V-F_2) is always assembled, even if one or more of I, V, and F are not found in the dictionary. In that case the English letter cluster is assumed to be identical with the stenotype letter cluster which is therefore assembled as part of the English correspondent. Each assembled stenotype stroke is recorded out on tape as a separate 30-word item.

Many stenotype dictionary entries have more than one English correspondent. In order to avoid too much fragmentation, special provisions are made during the assembly to write such multiple correspondents in parentheses. The reader must then choose one correspondent from each set of correspondents shown in parentheses. Consider, for example, the following sets of English correspondents:

I item: TH,

V item: O,

F_2 item: S, Z, ST.

The English correspondent of the assembled I-V-F_2 item would be

printed out

THO (S, 2, ST),

and the reader would normally choose the first correspondent between the parentheses to form THOS (those). A second output tape to be used for research purposes actually contains all correspondents "multiplied out."

At the end of the assembly each text word is represented by up to four 30-word items each including the stenotype original, the serial number assigned by the itemize routine, and the English correspondents. An output sample from the assembly routine is shown in Fig. 10.

It is now necessary to bring together all English correspondents belonging to a given stenotype stroke. This is done by reducing the assembled 30-word items corresponding to a given stroke to a single 30-word item including all distinct English correspondents. The "homograph package" which was originally written to reduce all duplicate stem entries in the Russian dictionary to a single entry was adapted for this purpose.[13] As part of the "homograph" elimination, duplicate English correspondents which may have been generated during the assembly are condensed so that only distinct correspondents appear on the output tape.

The output edit program of the Continuous Dictionary Run is now used to edit the condensed output of the previous program. Each English correspondent is assigned a 24 character position (two Univac words), and up to five English correspondents of a given

stenotype word are printed one underneath the other, thus making it easy to choose one correspondent for each stenotype word. An edited output text is read from left to right like ordinary English text. Multiple correspondents appearing one underneath the other are scanned, and the best correspondent is chosen.

The output edit program produces two outputs: the "inter-linear translation" which includes the stenotype original with each set of English correspondents, and the "word-by-word translation" which omits the stenotype original. Excerpts of these two types of output are shown in Figs. 11 and 12, respectively.

For research purposes another edited output is prepared. This output is processed in such a way that a separate item is created for each English correspondent. A program originally pre-pared to aid in syntactic analysis and to correct the Harvard Automatic Dictionary Russian file is used.[19] An excerpt showing the listing of all English correspondents is shown in Fig. 13.

The several programs described above can be transformed into a single program by automatically reading into memory at the end of each step the program for the next step and by insuring that servo assignments are not conflicting. A flowchart for such a con-tinuous run is shown in Fig. 14. Each box refers to one of the programs previously described. The programs marked CDR are part of the Continuous Dictionary Run, while the programs marked CST are on a Community Sequence Tape used for linguistic analysis.

The broken line between the inverse sort and the alphabetic sort denotes that this set of programs is used twice with different inputs and different dictionary tapes. Similarly, the broken line between the assembly program and the last box on the chart indicates the path for the second output of the assembly program.

Future Research

A number of methods for the improvement of the English output have already been proposed. These include, in particular, the use of digram and trigram frequency counts of both stenotyping and English to correct the spelling of the output, the inclusion of grammatical information in the dictionaries to remove syntactic ambiguities, and a frequency analysis of stenotype texts to be used in analyzing semantic content.

As a first step, however, the outputs produced by the present program can be used directly to update the dictionaries by reducing the number of possible English correspondents for each stenotype word. A large amount of the ambiguity now existing because of the multiplicity of correspondents might thus be eliminated relatively easily. The more difficult problems of determining word boundaries, and analyzing syntactic and semantic content of the context of a given word can then be tackled at a later time.

A human post-editor first takes the English translation as given on one of the two prints furnished by the output edit

routine.[20,21] The post-edited output can then be used for two
purposes: to determine which of the several English correspondents
listed in the dictionary for each stenotype word were actually used
in obtaining the correct translation, and to determine which of the
assembled items were chosen as correct correspondents.

The first type of information can be used to weed out from
the dictionaries (especially the dictionary of letter clusters)
those correspondents which are not believed to contribute to a
correct English correspondent. Many stenotype strokes will be found
to have invariably the same English correspondent, and no purpose
is served by burdening the dictionaries with unnecessary infor-
mation.

The second type of information can be used to refine the
partitioning and assembly rules. It is believed, for example, that
the I-V-E-A item, when produced, might replace the I-V-F item. At
the present time both items are obtained. If this should be con-
firmed by the post-editing operation a large number of correspondents
would be eliminated.

An automatic system for feeding back information obtained
from the post-editor might be used to improve the translated output
and also to correct the dictionary by deleting or adding infor-
mation.[22]

REFERENCES

1. Davis, K. H., Biddulph, R., and Balashek, S., "Automatic Recognition of Spoken Digits," *Journal of the Acoustical Society of America*, Vol. 24 (1952), pp. 637-42.

2. Forgie, J. W. and Hughes, G. W., "A Real-Time Speech Input System for a Digital Computer," *Journal of the Acoustical Society of America*, Vol. 30 (1958), p. 668.

3. Fry, D. B. and Denes, P., "On Presenting the Output of a Mechanical Speech Recognizer," *Journal of the Acoustical Society of America*, Vol. 29 (1957), pp. 364-67.

4. Hughes, G. W. and Halle, M., "On the Recognition of Speech by Machine," *Proceedings of the International Conference on Information Processing*, Paris (June 1959), to be published.

5. Olson, H. F. and Belar, H., "A Phonetic Typewriter," *Journal of the Acoustical Society of America*, Vol. 28 (1956), pp. 1072-81.

6. Wiren, J. and Stubbs, H. L., "Electronic Binary Selection System for Phoneme Classification," *Journal of the Acoustical Society of America*, Vol. 28 (1956), pp. 1082-91.

7. Ireland, W. S., "Stenotype Reporter," 3d ed., The Stenotype Company, Indianapolis (1914).

8. La Salle Extension University, "The Theory of Stenotypy,"
    Chicago (1949).

9. Stenographic Machines Inc., "Keyboard and Theory for Machine
    Shorthand," Chicago (1957).

10. Horne, B. H., "Stenotype-Stenograph Reporting — A Complete
    Course," rev. ed., New York (1952).

11. Standard Theory Committee, "Recommended Standard Practices,
    Abbreviations and Phrases in Machine Shorthand," Associated
    Stenotypists of America, 3d ed., (1957).

12. Gleason, H. A., An Introduction to Descriptive Linguistics,
    Henry Holt and Co. (1955).

13. Baxendale, P. B., "Machine-made Index for Technical Literature —
    An Experiment," IBM Journal of Research and Development, Vol. 2,
    No. 4 (October 1958).

14. Luhn, H. P., "The Automatic Creation of Literature Abstracts,"
    IBM Journal of Research and Development, Vol. 2, No 2 (April
    1958).

15. Plath, W. J., "The Relative Frequency of English Consonantal
    Patterns," A. B. Honors Thesis, Harvard University, (March 1957).

16. Jones, P. E., Jr., "The Continuous Dictionary Run," Mathematical
    Linguistics and Automatic Translation, Report NSF-2, Harvard
    Computation Laboratory (March 1959).

17. Petonke, Richard, "2- 3- 4-Way General Merge," Univac System Routines, Vol. 4, Remington Rand Univac.

18. Bossert, William, "The Problem of Homographs in the Automatic Dictionary," Papers presented at the Seminar in Mathematical Linguistics, Vol 4, Harvard University (Spring 1958).

19. Frink, Orrin, "Programs for Correcting the Harvard Automatic Dictionary and for Syntactic Study, (Conhadic, Checkhadic, Texthadic, and Freqhadic)," Mathematical Linguistics and Automatic Translation, Report NSF-2, Harvard Computation Laboratory (March 1959).

20. Giuliano, V. E., "A Formula Finder for the Automatic Synthesis of Translation Algorithms," Mathematical Linguistics and Automatic Translation, Report No. NSF-2, Harvard Computation Laboratory (March 1959).

21. Mattingly, I. G., "Logical Post-editing," Mathematical Linguistics and Automatic Translation, Report No. NSF-3, Harvard Computation Laboratory (August 1959).

22. Jones, R. E., Jr., "A Feedback System for the Harvard Automatic Dictionary," Mathematical Linguistics and Automatic Translation, Report No. NSF-3, Harvard Computation Laboratory (August 1959).

THE CROSSED FILM CRYOTRON AND ITS APPLICATION

TO DIGITAL COMPUTER CIRCUITS

V. L. Newhouse
J. W. Bremer
H. H. Edwards

General Electric Company
Schenectady, New York

Abstract

A crossed film cryotron deposited on an insulated super-
conductor is described. It has a time constant of less than one
microsecond and is approximately one hundred times faster than
the original vacuum deposited cryotron.[1] The d-c dissipation is
less than 5 microwatts and the active area of each element is
approximately 5×10^{-4} square centimeters. These cryotrons and
all their interconnecting circuitry can be vacuum deposited at
one and the same time in a few simple steps.

The cryotrons can be applied to both switching and
storage. Some experimental storage and shift register circuits
are described which demonstrate a circuit property unique to
superconductors. A shift register circuit is shown which is
deposited in an area corresponding to 20,000 active elements per
square foot.

Calculations are presented which show that with this
component density a computer or memory containing more than
one million elements can be accommodated in a one-cubic-foot
liquid helium container using presently available refrigeration
methods.

THE CROSSED FILM CRYOTRON AND ITS APPLICATION

TO DIGITAL COMPUTER CIRCUITS

V. L. Newhouse
J. W. Bremer
H. H. Edwards

General Electric Company
Schenectady, New York

Historical Introduction

The name cryotron was applied by the late D. A. Buck to the well known superconductive relay which he invented.[1] Buck's cryotron consisted of a wire of tantalum surrounded by a coil of niobium and was operated at the boiling point of liquid helium at atmospheric pressure. At this temperature the tantalum "gate" wire was only just superconducting. By passing a sufficient current through the niobium control coil, a magnetic field was created which was sufficient to transform the tantalum gate to its resistive state.

It was found by two of the authors that it is possible to produce the cryotron in a geometry suitable for deposition on a flat surface.[2] This will be referred to as the Crossed Film Cryotron (CFC). The CFC was first presented in public at the Electron Device Research Conference at Ithaca in June 1959. Similar devices were described at the same meeting by M. L. Cohen, J. L. Miles, A. E. Slade

and C. R. Smallman of the A. D. Little Company, and by
A. E. Brenneman and R. de Lano of the IBM Research Center.

Summary

Following a brief review of the relevant properties of
superconducting films, the properties of the unshielded cryotron,
which has already been published, will be compared with the
characteristics of the shielded cryotron which is an order of
magnitude faster. The operation of a very simple storage circuit
will be analyzed. This is applied to a shift register. In con-
clusion, an assessment of the role of film cryotrons in computers
will be made.

Superconductive Films

The devices to be described are made up of tin, lead and
insulator films only. Of these, only the tin films change their
state during operation. We can, therefore, confine our attention
mainly to tin films. At temperatures below the so-called critical
temperature T_c, tin and lead become superconducting. For lead
$T_c = 7.2^\circ K$. For the tin films used, $T_c \approx 3.75^\circ K$. The devices
described are operated at approximately $3.6^\circ K$. At this temperature
the tin films can readily be switched from the superconducting to
the normal (resistive) state, but the lead films remain super-
conducting throughout.

Just as in the case of bulk materials, it is possible to
restore a superconductive film to the normal state by the application

of a magnetic field greater than the so-called critical field H_c. The variation of H_c for bulk tin is shown in the insert of Fig. 1.

The main portion of Fig. 1 compares H_c of bulk tin and H_c of a 0.3 micron thick tin film. The film curve was determined experimentally with a uniform magnetic field applied parallel to the film surface. The data points shown in the figure are values of H_c calculated from the electrical characteristics of Crossed Film Cryotrons. These are discussed in connection with Eq. 4 below.

Fig. 1 shows that H_c of the film is higher than for the bulk material. It can be established on the basis of thermodynamics[3] that if the film thickness is of the same order of magnitude or less than the penetration depth, H_c varies inversely with film thickness. The penetration depth is roughly equal to the thickness of the surface layer in which the current flows in a bulk super-conductor. For tin, the penetration depth at absolute zero is approximately 5×10^{-6} cm, but at temperatures close to T_c, it is larger. At 3.6^oK, a typical operating temperature for a Crossed Film Cryotron, the penetration depth is about 0.1 micron.

The variation of resistance with current for a 0.3 micron tin film is shown in Fig. 2. By applying the current in short pulses it is possible to obtain the so-called isothermal transition shown in the broken line. This curve is connected with the actual superconducting behavior of the film and is reasonably independent of other film characteristics such as resistivity and of the substrate properties. If a _slowly_ rising current is passed through

a film, Joule heating causes thermal "propagation" of resistive areas in the film.[4] This behavior is shown in the solid curve of Fig. 2 and is strongly dependent on substrate thermal conductivity and on film resistivity.

The current at which resistance first appears is known as the critical current I_c. For thin films $I_c = i_c W(T_c - T)$ where W is the film width and T the bath temperature, provided that $T_c - T \ll T_c$.[5] i_c increases as film thickness increases and appears to depend somewhat on heat treatment and film substrate. It has been found that i_c is more than doubled if the film in question is deposited on top of an insulated lead "shield" plane.

The explanation of why a tin film which lies adjacent to a lead shield plane has a higher critical current than a similar tin film deposited on glass is believed to be as follows. It can be shown[6] that a film in the shape of a cylinder will carry twice as much current as the same film unwrapped into a flat plate. When current passes through a tin film adjacent to a superconducting shield, surface currents are induced in the shield to prevent flux penetrating into it. It can be shown that these surface currents double the field between the film and the shield and produce an approximately zero field on the opposite side of the film. This field configuration is the same as would occur if the tin film were in the shape of a cylinder. It is to be expected, therefore, that the critical current for a shielded flat film is increased from the value for the unshielded flat film to that for the cylinder.

The mathematical problems of calculating the surface currents induced in a superconducting surface due to the presence of an external current carrying conductor are similar to the problems of calculating the surface charge produced in a perfect conductor due to an external charge. It is found that some of the results of the "method of images" of electrostatics can be carried over to superconductors if an electronic dipole is replaced by a magnetic dipole, and a line of charge by a line of current. For a current carrying wire above a superconducting surface for instance it can be shown that the net field outside the surface is equal to the field of the original current plus that of an equal and opposite shielding current which is the same distance behind the super-conducting surface as the real current is in front. This effect increases the field between the current and the surface but reduces it everywhere else. (It is assumed that the maximum net field is less than the critical field of the superconducting shield plane.) It can be seen therefore that if it is desired to reduce the effective inductance of a wire or length of film, it is simply necessary to place a superconducting plane with a high critical field in close proximity.

The Crossed Film Cryotron

The basic structure of a crossed film cryotron (CFC) is shown in Fig. 3. If a sufficiently large current is passed through the "grid" film, the resulting magnetic field produces a resistive channel across the much wider tin "gate" film. The grid remains

superconducting at all times. The characteristics of the
unshielded cryotron are shown in the broken curves of Fig. 4.
The curve intersecting the left-hand ordinate at 91 ma shows values
of gate and grid current at which resistance just begins to appear.
Its intersection with the ordinate defines I_c for the gate film
used. It can be seen that the presence of gate current "helps"
the grid current to make the gate resistive.

The gain of the CFC can be defined as the ratio of the maximum
current I_c which the gate can carry and remain superconductive, to
the minimum grid current I_G required to make the gate resistive at
low gate currents, i.e.,

$$g = I_c/I_G \tag{1}$$

For the unshielded cryotron I_c = 91 ma and I_G = 31 ma at the
temperature shown. The other broken line curves shown in Fig. 4
refer to the right-hand ordinate and represent the variation of
resistance with grid current at constant gate current. The slope
of these curves can be accounted for to within a factor of 2 in
terms of the known critical field of the film and the film
resistivity.

If the CFC shown in Fig. 3 is covered with an insulating
layer followed by a film of lead, the inductances of the gate and
grid will be reduced as explained above, and the d-c characteristics
will be somewhat changed as shown in the solid lines of Fig. 4.
It can be seen that I_c for the shielded tin film is more than three
times that for the unshielded film. I_G is also somewhat larger.

However, the gain of the shielded cryotron is larger than that of the unshielded one.

It is noteworthy that the curves of resistance as a function of grid current for the shielded CFC approach a saturation value of resistance. This is because the field, due to a grid above a shield plane, falls off very rapidly beyond the edge of the grid because in this region the fields, due to the grid current and the shield current, tend to cancel one another. It is to be expected, therefore, that the portion of the gate film which can be made resistive by grid current action is that portion lying under the grid. The maximum calculated resistance of the shielded CFC is shown dotted and is seen to be in fair agreement with experiment.

We will now show that the gain of the CFC is proportional to the ratio of the gate to the grid widths. As mentioned above, for thin tin films, I_c is proportional to the film width W, i.e.,

$$I_c = i_c W. \tag{2}$$

Here i_c is a constant of the material roughly proportional to $T_c - T$ as long as this is small.

In a current-carrying superconducting film (at least when the film thickness is large compared to the penetration depth of the current), the current density is not uniform over the film surface but is more concentrated near the edges. For a current-carrying film close to a superconducting shield plane, however, the current is distributed more uniformly.

The field H between a film of width w carrying a current I which is assumed distributed uniformly,

and a shield plane is

$$H \approx 0.4\pi I/w. \tag{3}$$

This formula will apply to the field in the space between a grid and its shield plane, which contains the gate film. If the gate film has a critical field H_c at which it becomes resistive, then the grid current I_G at which the gate just becomes resistive will, using Eq. 3, be

$$I_G \approx H_c w/0.4\pi. \tag{4}$$

The points shown in Fig. 1 are values of H_c calculated from experimental values of I_G for representative cryotrons of different grid widths. It is clear that Eq. 4 is at least approximately correct. From 1, 2 and 4 we finally obtain the gain as

$$g = 0.4\pi \frac{i_c}{H_c} \frac{W}{w}. \tag{5}$$

It is seen that the gain is proportional to W/w. $\frac{i_c}{H_c}$ is a property of the material and decreases with gate film thickness. i_c rises linearly with $\Delta T = T_c - T$ close to the critical temperature. However, as Fig. 1 shows, the curve of H_c vs. ΔT has a knee at $\Delta T \approx 0.08^o K$. It is to be expected, therefore, that g should rise strongly as the temperature is decreased below T_c until T goes below the knee of the H_c vs. ΔT curve. This appears to be the case. The cryotrons described here are operated at $\Delta T \approx 0.07^o K$, i.e., just above the "knee" and are not, therefore, operating at their maximum gain.

The speed of a cryotron is, of course, dependent on the mode of operation. In the circuits described below, the cryotron gate is in parallel with the load, which consists of the grid crossing a similar cryotron. The time constant τ at which current will be diverted from a cryotron gate to the grid of the load cryotron is L/R where R is the resistance of the driving cryotron. L is the sum of the grid inductance of the driven cryotron, the gate inductance of the driving cryotron, and the inductance of the connecting circuits.
Out of all these terms, only the grid inductance is important.

The inductance of a grid of width w spaced d cm from the superconducting shield plane can be shown to be $4\pi \times 10^{-9} \frac{d}{w}$ henries/cm. The driven cryotron has a width W, hence its grid has a length W and

$$L = 4\pi \frac{Wd}{w} \times 10^{-9} \text{ henries.} \tag{6}$$

As discussed in connection with Fig. 4 the maximum portion of a shielded cryotron which becomes resistive is that part of the gate film covered by the grid. Hence, the resistance R of the driving cryotron of width W, energized by a grid of width w, is

$$R = \rho\frac{w}{Wt} \text{ ohms} \tag{7}$$

where t is the gate film thickness and ρ the effective bulk resistivity. (For very pure films, ρ is itself a function of t, but for the relatively impure films used here, this dependence can be neglected.)

From 6 and 7 we find that the effective time constant τ of

one cryotron driving another is

$$\tau = L/R$$

$$= 4\pi \frac{td}{\rho}\left(\frac{W}{w}\right)^2 \times 10^{-9} \text{ sec.} \tag{8}$$

Substituting from 5 for W/w in 8 we find the time constant in terms of the gain

$$\tau = 4\pi \frac{td}{\rho}\left(\frac{H_c}{0.4\pi i_c}\right)^2 g^2 \times 10^{-9} \text{ sec.} \tag{9}$$

There are two points of interest in 9. First $\tau \propto t\left(\frac{H_c}{0.4\pi i_c}\right)^2$. This shows that there is an optimum value of the gate thickness t because as we attempt to reduce τ by reducing t, $\frac{H_c}{0.4\pi i_c}$ increases. For solid wires $H_c/0.4\pi i_c \longrightarrow 1$, but for the 0.3 micron tin films presently used $H_c/0.4\pi i_c$ is between 20 and 50. The second point of interest is that τ is not a function of the grid and gate widths. Hence, a reduction in cryotron area will not increase speed.

Present values for the material constants in 9 are as follows: $t \approx 0.3$ microns, $d \approx 1.0$ micron, $\rho \approx 6\text{-}12 \times 10^{-7}$ ohm-cm, $\frac{H_c}{0.4\pi i_c} = 20\text{-}50$. A practical value for the gain is 2. Substituting these values into 8 we obtain a theoretical range of $\tau = 5\text{-}65 \times 10^{-8}$ sec. A typical cryotron, described below, has an experimental time constant of 38×10^{-8} sec at the temperature of operation.

A Simple Storage Circuit

We will now describe a simple storage circuit which makes use of a principle unique to superconducting networks. The principle will be illustrated with the circuit shown in Fig. 5-A.

In one mode of operation of this circuit a current is applied between X and Z. Most of this flows through the path XZ rather than XYZ because the former has much lower inductance. The equivalent circuit is shown in Fig. 5-B.

If current is now passed through the input grid, XZ becomes resistive and I is diverted through the path XYZ. It is now possible to switch off the current through the input grid so that XZ becomes superconducting again. Since L_2 is still superconducting, it is to be expected that I will remain diverted through L_2, even though L_1 has become superconducting again. This does, in fact, happen experimentally. The current in XYZ can conveniently be determined with a d-c measurement by measuring the resistance of MN.

If, after I has been diverted to L_2 and after L_1 has become superconducting again, I is switched off, a circulating current will remain in loop XYZ. Its magnitude can be calculated as follows.

Assume that a current +I has been injected into node X and completely diverted to L_2, through L_1 having been temporarily been made resistive. If now current -I is injected into node X, it will divide itself between XZ and XYZ in the inverse ratio of their inductances. The current along XZ will be $I_{XZ} = -I \dfrac{L_2}{L_1 + L_2}$ and the net current along XYZ will be

$$I_{XYZ} = I - I \frac{L_1}{L_1 + L_2}$$

$$= I \frac{L_2}{L_1 + L_2} .$$

Therefore, $-I_{XZ} = I_{XYZ}$. Hence, the circulating current

$$I_{circ} = \frac{L_2}{L_1 + L_2} I. \tag{10}$$

Eq. 10 has been confirmed experimentally and currents have been stored for several hours.

The results which have been described in connection with the above storage circuit can be generalized. If a current is injected into a network of superconductive elements, it will distribute itself among them in inverse ratio to their **inductances.** By making one or more of these inductors resistive for various lengths of time, current will be diverted from the resistive elements to the superconducting ones. As soon as the resistive elements are made superconducting again, however, the current distribution stops changing, as long as the external injected current remains constant.

It is clear that in a cryotron computer, analog and digital storage will be simpler than in a transistor computer, where positive feedback circuits, or magnetic cores which cannot conveniently supply a continuous output, are required.

The circuit of Fig. 5 provides a convenient way of measuring the effective speed of the cryotrons used in it. The deflection of current from leg XZ to leg XYZ takes place with a time constant

$$\tau = (L_1 + L_2)/R \tag{11}$$

(R is the resistance produced in XZ by the input grid current). By applying individual pulses of known length to the input grid we can measure the current changes in the output grid and obtain an experimental value for τ.

The results of applying 0.05 microsecond current pulses to the input grid of one of the storage loops shown in Fig. 7 is shown in Fig. 6. The experimental curves have a time constant of approximately 0.38 microseconds. The value given by Eq. 11, using calculated values for $L_1 + L_2$ and experimental values for R, is 0.33 microseconds. The curve of rising current is obtained when 150 ma is injected into the cryotron loop and gradually diverted to the output leg. The curve of decreasing current corresponds to a stored current (with the external current switched off) being destroyed by pulses on the input grid.

A Shift Register

The memory circuit described has been applied to a shift register. A short portion of such a register is shown in Fig. 7. The register is of the three stages per bit type and requires two advance and two reset windings. A diagram and a set of calculated waveforms are shown in Fig. 8.

Information travels from left to right. To inject a "1" into the register, the input winding is pulsed while advance current I_1 is on. This diverts I_1 from the first cryotron to its output grid and when I_1 goes off, a circulating current C_1 remains in the first storage cell. I_2 is now injected into the second storage cell. Due to the existence of C_1, I_2 will be diverted to the output grid of the second storage cell. It is necessary at this time to destroy C_1. This is done by passing current through the reset winding R_2. C_1 has to be destroyed before I_2 is switched off since, otherwise, C_2, the circulating current in the second storage cell caused by the effect of C_1 on I_2, would be destroyed. After C_1 has been destroyed and I_2 switched off, the injected "1" is represented by the circulating current C_2 in cell 2. In similar fashion, C_3 is created and C_2 is destroyed. Only now can a new "1" be injected into the first storage cell. The grid of the last storage cell crosses an output cryotron whose resistance is an indication of the presence or absence of the circulating current C_4. As described earlier, the experimental and calculated time constant of the cryotron in this circuit module is 0.38 microseconds. This time constant could probably be reduced to approximately 0.1 microseconds by working at a lower temperature and changing the cryotron dimensions to lower the gain. The register has been used to transfer information. However, it has not yet been operated at high repetition rates

Discussion

A summary of cryotron characteristics is shown in the table.

Time constant	0.4 μsec
Size	6 mm^2
Dissipation (d-c)	$\approx$5 x 10^{-6} watts

The lower limit in size is presently set by difficulties in getting good contact between the CFC gate and its connecting circuitry.

Using our present elements and putting one per square cm^2 (this would leave plenty of area for interconnections) it should be possible to accommodate 1000 circuits on a 1 ft^2 plate. Stacking two of these plates per cm would give a 50,000 element computer in 1 cu ft. Assuming each element to be on for 10 per cent of the time gives the total dissipation as 5 x 10^4 x 5 x 10^{-7} = 0.025 w. Approximately 0.25 w heat enters the liquid helium due to radiation and conduction through the container. It has been calculated that 100 input and output leads 10 of which carry 200 ma dc would contribute less than 0.2 w in conduction and Joule heating. (It was assumed that the upper ends of these leads would be in thermal contact with a liquid nitrogen chamber.) The total heat inflow is therefore just under 0.5 w for the system under discussion.

Helium refrigerators with 1 w capacity for a 1 cu ft volume are presently being designed.

What role will cryotrons play in future computers? They are, of course, small. They appear to constitute the cheapest method of assembling a large number of circuits in a small place, and they are the only active circuit component which can be deposited in a few steps at the same time as the interconnections.

Because CFC's are, in principle, as cheap as magnetic cores, they make possible radically improved memory structures where each storage element can have logic associated with it. Memories, in fact, appear as an attractive first application because their structure is repetitive and because they have many less logic levels than even the simplest computer.

Computers are presently built from plug-in circuit packages where each plug-in unit represents a few logical elements. CFC's make it possible to deposit the equivalent of a present-day rack on a small plate in a few hours. The problems of testing and fault correction in such a complex multi-level logic module are real, but they are not greater for cryotrons than for any other component which would allow similar packing densities. If these problems can be solved economically, and we feel sure that they can be, the next ten years may see the development of cryotron ground-based computers, as well as air-borne computers and memory systems.

References

1. D. A. Buck, "The Cryotron - A Superconductive Computer Component," Proc. I.R.E., Vol. 44, pp. 482-493, (1956)

2. V. L. Newhouse and J. W. Bremer, "High-Speed Superconductive Switching Element Suitable for Two-Dimensional Fabrication," J.A.P., Vol. 30, p. 1458, (September, 1959).

3. D. Shoenberg, "Superconductivity," pp. 171-174, 2nd edition, Cambridge University Press, 1952.

4. J. W. Bremer and V. L. Newhouse, "Thermal Propagation Effect in Thin Superconducting Films," Phys. Rev. Letters, Vol. 1, p. 282, (1958).

5. J. W. Bremer and V. L. Newhouse, "On Current Transitions in Superconductive Tin Films," Phys. Rev., to be published.

6. V. L. Ginzburg, "Critical Currents in Superconducting Films," Soviet Physics "Doklady," Vol. 3, p. 102, (1959).

Captions

Fig. 1 - Critical field of gate film as a function of temperature.
Solid line: Measured directly. Points: Calculated from
cryotron characteristics and grid widths, using Eq. 4.
Grid widths: ▲ 16 microns (unshielded
              □ 65 microns (shielded)
              o 40 microns (shielded)

Fig. 2 - D-c and pulse current-induced transitions for 0.3 micron
thick, 4.05 mm wide tin film on sapphire substrate.
$\Delta T = 0.08^\circ K$.

Fig. 3 - Structure of crossed film cryotron.
Typical dimensions: Gate film - 0.3 microns x 2 mm
                     Insulator - 0.4 microns
                     Grid film - 1 micron x 25 microns.

Fig. 4 - Comparison of electrical characteristics of unshielded
and shielded CFC. Shield insulation thickness ≈ 4 microns.
Grid width - 30 microns
Gate width - 2 mm
$T_c - T \approx 0.07^\circ K$

Fig. 5 - A. Cryotron storage cell
         B. Equivalent circuit

Fig. 6 - Change of stored current due to 0.05 μsec pulses on input grid of storage cell of Fig. 7. Curves are fitted to the data points corresponding to a time constant of 0.38 μsec.

Fig. 7 - Portion of experimental shift register. Note the insulator film separating the grids from the underlying gate films.

Fig. 8 - Calculated waveforms for shift register.

Fig. 1

$\frac{R}{R_{4.2}}$

FILM CURRENT-ma

4 μ SEC. PULSES

Fig 2

TIN GATE

LEAD CONNECTORS

GLASS SUBSTRATE

LEAD GRID

SiO INSULATOR

Fig 3

GATE CURRENT
60 ma

GATE CURRENT
0.91 ma

CALC. R MAX.

GATE CURRENT
120 ma

GATE CURRENT
3.4 ma

SHIELDED

UNSHIELDED

GATE CURRENT—ma

RESISTANCE—MICROHMS

GRID CURRENT—ma

Fig 4

TIN FILM

LEAD FILM

INPUT GATE

X

OUTPUT GATE

M

INPUT GRID

Z

Y

OUTPUT GRID

N

A

I

X

L₁

Y

L₂

R

Z

B

96 EXP. [-t/0.38]

CURRENT SWITCHED
FROM GATE TO GRID
(I_0 = 150 ma)

DESTRUCTION OF
STORED CURRENT
(I_0 = 0)

NO. OF 0.05 μ SEC. 150 ma PULSES OF I_G

I_s
ma

t – TIME μ SEC.

Pb-Sn CONTACT AREA

SUBSTRATE

I MICRON Pb

0.4 MICRON SiO

0.3 MICRON Sn

0   5   10
                      mm

Fig 8

November 12, 1959

Harlan E. Anderson
EJCC Publication Committee
Digital Equipment Corporation
Maynard, Massachusetts

Enclosed find four copies of our paper "Unusual Techniques
Employed in Transient Heat Transfer Programs". Also,
four copies of biographies of ourselves, the abstracts, and
four sets of illustrations with captions are included.

If we have failed to include anything, please let me know,
and I will try to provide it.

It is our plan to use 3 1/4 x 4 colored lantern slides during
the presentation.

I wish to remind you again that both Mrs. Vollenweider and
I are authors, and that both of our names should be included.

D. J. Campbell, Manager
Aerodynamics, Diffusion
and Heat Transfer
Evendale Computations
Building 305
Evendale, Ohio

DJC/ck

Enclosures

Donald J. Campbell is with the Flight Propulsion Laboratory Department of the General Electric Company. He received his Bachelors and Masters degrees from the University of Michigan in 1955 and 1956. After receiving the later degree, he was employed by the Ramo-Wooldridge Corporation. In 1957 he joined General Electric, at Evendale, where he now is manager of Aerodynamics, Diffusion and Heat Transfer Computations.

Deborah B. Vollenweider was graduated from Wellesley College in 1955 and since that date has been employed by the General Electric Company, Flight Propulsion Laboratory Department. Mrs. Vollenweider is a condidate for the degree of master of arts in mathematics at the University of Cincinnati. She is Project Specialist - Computations for Evendale Computations Operation, Cincinnati, Ohio, and has worked on the development of projects in aerodynamics, diffusion, and heat transfer analysis.

UNUSUAL TECHNIQUES EMPLOYED

IN TRANSIENT HEAT TRANSFER PROGRAMS

D. J. Campbell
D. B. Vollenweider

Flight Propulsion Laboratory Department
General Electric Company
Cincinnati, Ohio

ABSTRACT

A method is presented for solving transient and steady state heat transfer
problems. The digital computer program permits the analysis of three-
dimensional problems with arbitrary geometry and various combined modes
of heat transfer. An unusual manner of describing geometric configurations
and of presenting input data in symbolic form is discussed. Aspects of
input processing, testing, and organization are described. Further extensions
of these methods and computational techniques are proposed.

UNUSUAL TECHNIQUES EMPLOYED IN TRANSIENT HEAT TRANSFER PROGRAMS

A program for the IBM 704 to solve general transient and steady state heat transfer problems is described. The computer program was developed by Evendale Computations Operation in co-operation with the Jet Engine Department, General Electric Company, Flight Propulsion Division. Jet Engine Department personnel who were instrumental in the formulation of the problem are Mr. William K. Koffel and Dr. J. M. Botje. Dr. James T. Anderson, of Michigan State University, acted as a consultant and played a central role in the development of the program. The method of solution permits the analysis of problems with arbitrary geometry and several combined modes of heat transfer. To facilitate the description and solution of extensive and complicated problems, many logical and computational techniques not usually applied to engineering calculations are incorporated in the program.

The program yields the temperature distribution at a maximum of 200 points in three-dimensions for homogeneous or composite bodies. Heat transfer by conduction, convection, surface flux, thermal radiation, both solid and gaseous, and internal heat generation is treated. One dimensional fluid flow in several channels can be incorporated. The thermal properties of the system may vary with temperature. Boundary conditions and heat transfer rates for convective exchanges as well as surface flux rates and mass flow rates may vary with time. Internal heat generation is given either as a function of time or of temperature.

The method of solution for the temperature distribution is to divide the geometry into cells enclosing nodes and to write heat balance equations (in finite difference form) for each node. The general heat balance equation that is written for each node is given in figure one. Note that T' indicates temperatures at the beginning of a time interval. The radiation terms are linearized by writing them as a radiation coefficient times a first power temperature difference. The radiation coefficient is computed using the temperatures at the beginning of the time interval, while the linear temperature difference uses a given radiation source temperature. The implicit form of heat balance equations is used in preference to the explicit form, because the stability requirement for the explicit form places a definite limit on the size of time increment which is restrictive for large systems. The implicit form permits more freedom in the choice of time step. The accelerated Gauss-Seidel method is used to solve the system of heat balance equations for the n unknown temperatures. Gauss-Seidel is used in preference to other methods because the convergence criterion for the method is readily met for most applications and because there are a large number of zero elements in the coefficient matrix.

One of the most interesting features of the program is the method used for describing the geometry. Rectangular coordinate systems are not adequate to fully describe the variety and intricacy of shapes commonly encountered in applications of the program.

Instead, the geometry is divided into cells of any shape or size, and the cells are given numbers to identify them. Figure two shows how the plate of a steam iron might be divided into cells. At most six faces are defined for each cell, and the faces are labeled by numbers from one to six. Figure three illustrates the numbering of faces of a cell. In order to write heat balance equations, it is necessary at each cell to give the adjacent cell, if any, for every face through which there is conduction. In the program, this is done by giving the number of the cell adjacent to each face. A convention with respect to adjoining cells makes it unnecessary to give the number of the face of the adjacent cell. The face common to two adjacent cells is, of course, assigned a number in each of the cells. As can be seen from figure four, if a face is labeled "1" for one of the cells, then it must be labeled "3" for the other cell. Similarly, if the face is labeled "2" in one cell, it must be labeled "4" in the other, and if it is labeled face "5" in one cell, it must be labeled face "6" of the other.

Although this method of describing geometry may seem awkward at first glance, it conveniently accommodates unusual patterns of cells and irregular surfaces. Figure five shows some unusual shapes that may be accommodated with this method of describing geometry. It can be seen that cells need not have six faces and can be highly irregular in shape. Any face not needed for conduction exchanges may be used to accommodate special effects. For example, in figure six, the two narrow cells exploded out of position are used to increase the number of faces for the cell that occupies the position in the center of the drawing. Since the cells are assumed to have zero volume and zero thickness, there are no thermal storage terms and no thermal gradients across them. Therefore, they will have no effect on the heat balance except to balance the cell in the center against all of the surrounding cells. These nodes increase the effective number of faces of the central node from six to eleven.

A symbol defining a table of thermal properties is given for each cell in the system. These properties are tabulated as functions of temperature. Given the geometry description and the material symbol just mentioned, sufficient information is available to assemble the conductive heat transfer equations. In describing the dimensions of a cell, a distinction is made between "regular" and "irregular" cells. Regular cells are rectangular parallelepipeds, or linear approximations thereof, and only height, width, and depth of each cell are required as input. For irregular cells, distances from each face to the node point, face areas, and the cell volume must be specified. Physical parameters and boundary conditions are given as functions of time or of temperature. For example, tables of adiabatic wall temperature and fluid film coefficient are given, and the collection of tables is assigned a symbol. The symbol defining the applicable tables of functions is entered for the proper face of the cell which has a convective heat exchange. An analogous method is used to indicate other modes of heat transfer. That is, the functions are defined and identified by a symbol, and the symbols are given at each appropriate cell. Consequently, there are two basic types of information for a cell. One gives the dimensions of the cell and other geometric constants such as configuration factors for thermal radiation. Secondly, symbols are given to define the surrounding geometry, to indicate the modes of heat transfer, and to define the tables of values of the various parameters.

From the symbols describing the cell configuration and corresponding properties, the heat balance equations may be assembled for each time step. However, an assembly directly from symbols would require a large number of sorting procedures on the basis of node numbers and table symbols. To avoid searching for information designated by symbols more than once, a simple device is utilized in the program. Each symbol is replaced by the machine address of the initial memory location of the information defined by the symbol at the beginning of the program, before any temperature calculations are begun. The searches that are undertaken automatically check for completeness. If a table or node has not been given, the search for the location of that table must fail. When such a failure occurs, a comment is printed giving the symbol of the missing table or node.

Further examination of the form of the input indicates that testing the cell connection information is highly desirable. The method of describing geometry requires that connection information for adjoining cells must be given for the designated face of each cell. This may be seen by examining figure seven, a facsimilie of the cell information input form. The entries necessary to define a connection between cells 9 and 13 are indicated. If the connection is not indicated for both cells, erroneous heat balance equations will be generated, and the equations may yield results that appear correct but are in error in the vicinity of the inconsistency. Moreover, the volume of data required for large problems encourages error in input. Consequently, the program checks the consistency of cell connections and makes many other tests for completeness of input data to recognize and identify errors before the equations are assembled. The machine time required for these completeness and consistency tests is negligible when compared with overall computing time.

It can be seen that the method for describing arbitrary cell configurations does necessitate considerable pre-processing, searching, and testing of input data. If the geometry were described by the usual mesh or lattice point scheme, input for the network could be organized by three-dimensional arrays and could be assigned to ordered blocks of machine storage. There would be a simple arithmetic relationship between nodes and the corresponding memory location of information pertaining to nodes. However, if it is necessary to describe and analyze arbitrary patterns of cells enclosing node points, a simple arithmetic formula for computing the address of the machine location of information for another node from the address of a given node cannot in general be established.

Having presented some of the interesting aspects of the program, we would like to describe possible extensions of the techniques and methods employed. In the present program, input data is stored in fixed sections of memory. Consequently, there are specified limits for the maximum number of entries in a table, the number of tables of a given variety, and the maximum number of cells in the division of the problem. These restrictions do not allow the available memory space to be used efficiently for many problems. In a subsequent program, similar input is stored in consecutive memory locations which are computed as the data are read into memory. Since sections of data are not assigned

fixed storage locations, an economical means for finding specified quantities is necessary to accomplish the processing and testing of input data. Figure eight illustrates this memory packing system. Each table begins with two symbols that are shown shaded in the figure. The first symbol identifies the table and defines its type. The second symbol indicates the length of the table. The next table begins in the first unused memory location after the preceding table. Thus, for each table there is information which can be used to locate the next table. By allocating storage as the input data are read into the machine, considerable flexibility in the type of problem acceptable to the program is achieved. There are no pre-assigned limits on the maximum number of such quantities as nodes or physical parameters. Only the total amount of storage available for data limits the size and type of problem that can be handled by the program. An additional advantage to flexible input storage assignment is that the program can be run on a 704 with any memory capacity. In one program this idea was extended further by allowing tables of a particular type to have one of several different formats of varying length. In this case, the next table could begin at any one of several alternate locations. To determine the location of the next table, the symbols are defined in such a manner that the binary representation of these symbols are distinct from ordinary floating-point numbers. Thus, the table following a given table is found by searching each of several alternate locations for a word not in normal floating-point format, that is, for the symbol defining the new table.

Another extension of the transient heat transfer program is to assign an order to the operation on cells. At present, the heat balance equations are assembled and solved in an arbitrary order. Certain applications of the program to large classes of less general problems suggest that methods be incorporated for computing quantities required as input to the present program. A body with two horizontal fluid passages is illustrated in figure nine. In some cases it is desirable to make calculations in the order indicated by the arrows, regardless of the order of the data within the computer. For example, the computation of weight flow and fluid film coefficients from given inlet and outlet pressure for one-dimensional fluid channels could be programmed. If the computer were programmed to operate on the cells in each channel in the direction of flow, calculation of such quantities as fluid film coefficient could be incorporated easily. A straight-forward method for constructing an order of operation on node information is to recognize the desired order from the symbols describing the problem and to store with information for one cell the address of the location of the next cell to be treated. In certain applications of the program, assigning an order to the operation on cells can facilitate the solution of the system of equations. If an order of rows of the matrix is known to yield faster solution, this order can be used in applying the Gauss-Seidel method. Moreover, ordering program operation on cells provides a simple means for utilizing the symmetry of many elements of the matrix. The conduction-convection terms between two connected cells are symmetrical, and considerable computing time is saved if these terms are computed once for each time step. One means for utilizing the symmetry of coefficients is to order the assembly of elements by increasing or decreasing machine address.

In conclusion, several interesting methods and computational techniques are incorporated in the transient heat transfer program to allow its application to a wide range of problems. The program has been used to analyze large problems incorporating unusual shapes and several combined modes of heat transfer. The machine time and consequent cost required for the solution of most problems is very reasonable. For example, typical problems with 150 nodes and 100 transient time steps are run in about 10 minutes of 704 computer time. It is economical to use the heat transfer program to solve extensive problems and to obtain solutions that are considerably more adequate than those resulting from other methods. Consequently, extensions and further application of the ideas used in the present program are planned for the solution of other engineering problems.

Figure 1:   the general heat balance equation. T_0' indicates the temperature

at the beginning of a time interval. Linearized versions of the

radiation terms are actually programmed.

$$\begin{array}{cccc} \text{THERMAL} & \text{NON-SOURCE, SINK} & \text{CONDUCTION} & \text{INTERNAL HEAT} \\ \text{STORAGE} & \text{FLUID HEAT TRANSFER} & \text{\& CONVECTION} & \text{GENERATION} \end{array}$$

$$\frac{C\rho V}{\Delta\tau}(T_O - T_O^1) = cW(T_f - T_O) + \sum_{n=1}^{6}\left[\frac{A_n(T_n - T_O)}{\dfrac{\Delta X_n}{K} + \dfrac{1}{h_c}}\right] + Q_O V$$

$$\begin{array}{ccc} \text{SURFACE} & \text{SOLID} & \text{GASEOUS} \\ \text{FLUX} & \text{RADIATION} & \text{RADIATION} \end{array}$$

$$+ A_s Q_f + \sum_{i=1}^{3}\sum_{s=1}^{3}\left[\sigma A_i F_i (T_h^4 - T_O^4)\right] + \sum_{j=1}^{3}\left[\sigma A_j G_j (\epsilon_g T_g^4 - \alpha_g T_O^4)\right]$$

Figure 2: As an example of the method of dividing an object into cells, the plate of a steam iron is shown. Note the irregular manner in which the cell division has been made.

Figure 3: illustration of the convention for numbering the faces of a cell.

Figure 4: illustration of the face numbering convention for adjacent cells.

The numbers assigned to a face of adjacent cells must be one

of the pairs (1, 3), (2, 4) or (5, 6).

Figure 5: examples of irregular shapes that have been divided into cells.

Figure 6:   illustration of a method for increasing the effective number of

faces of a cell. By including two "zero volume" cells, the

number of faces of the long, inner-most cell is increased to

eleven.

FACE 4   FACE 1   FACE 6

FACE 3

FACE 1
FACE 3

FACE 1

FACE 2   FACE 3   FACE 4   FACE 5

Figure 8: illustration of memory packing system. The shaded areas indicate

symbols defining the type and length of a table. Each table

begins immediately after the preceding table.

COL.

	1n.c	1b.c	3n.c.	3b.c.	2n.c.	2b.c.	4n.c.	4b.c.	5n.c.	5b.c.	6n.c	6b.c.	MAT.	B.Cd.
	10	13	16	19	22	25	28	31	34	37	40	43	46	49
CELL NO. 9	101		2		13		503			A	202		CM	
	3	BC							16				DM	
				AA					31		66		Q	
CELL NO. 13	4	BD	36		102		9		63		7		DM	R

Figure 7:   an example of data describing the geometry of a problem.   The

connection between cells number 9 and 13 is illustrated to

indicate the need for checking its consistency.

MEMORY PACKING SYSTEM FOR FUNCTION TABLES

BCC	A	3				
BCC	B	4				
BCC	F	6				
MT	7	3				
CH	14	2				

START

END

Figure 9: illustration of assigning an order to program operation on cells
to accommodate specialized calculation or to increase the rate
of convergence.

HUGHES

Systems Development Laboratories

HUGHES AIRCRAFT COMPANY ● CULVER CITY ● CALIFORNIA

Telephones: EXmont 8-2711 ● TExas 0-7111

10 November 1959

Mr. Harlan E. Anderson
EJCC Publication Committee
Digital Equipment Corporation
Maynard, Massachusetts

Dear Mr. Anderson:

Please find enclosed our manuscript and illustrations
for publication in the Proceedings of the Eastern Joint
Computer Conference.

Sincerely,

A. Levine

R. B. McGhee

AL:RBM:bc

Encl. (4)

DETERMINATION OF OPTIMUM PRODUCTION TOLERANCES BY ANALOG SIMULATION

by

R. B. McGhee and A. Levine

ABSTRACT

This paper extends the role of analog simulation in the preliminary design of control systems to include optimization of manufacturing tolerances as well as the more common optimization of parameter values. The proposed simulation technique involves random variation of parameters by noise generators along with automatic computation of a performance index. The simulation results in a performance function in terms of the production tolerances assigned to the system parameters as well as the nominal values selected.

When the performance function is considered together with a cost function also depending on parameter tolerances, an optimum tolerance assignment can be made by minimizing cost for a fixed level of performance or maximizing performance for a fixed cost.

The description of the method is followed by an experimental example which refutes some common opinions and practices relating to production tolerance assignments. It is shown that an a priori estimation of the form of the functional dependence of performance on parameter values can lead to serious engineering errors in the assignment of tolerances.

The design of statistical experiments of this type are discussed in the context of the experimental example. It is pointed out that high speed repetitive analog computation is required in a problem of practical complexity. Special simulation considerations such as automatic data processing, multiplexing of noise generators, etc. are also discussed.

DETERMINATION OF OPTIMUM PRODUCTION TOLERANCES BY ANALOG SIMULATION

R. B. McGhee and A. Levine

I. INTRODUCTION

The electronic analog computer has been widely utilized in recent years to obtain optimum parameter values for various types of control systems.[1,2] The procedure usually followed is to determine a control system configuration by simplified analysis and then to program a computer in a realistic way, including all significant nonlinearities and appropriate random input disturbances. Next, a performance criterion of some sort is selected for the control system under study and system parameters (gains, time constants, etc.) are varied until an optimum system is obtained. At this point, simulation activity usually ceases and is not resumed again unless the actual control system either cannot be constructed to obtain the desired parameter values or else fails to attain the level of performance indicated by the earlier simulation.

While this approach has produced many successful systems, it is the contention of the authors that it is seriously deficient in one respect; the essentially statistical nature of an actual system resulting from a manufacturing process is completely ignored. The representation of a control system as a differential equation with known coefficients is not realistic. An accurate simulation must effectively consider an ensemble of control systems, (i.e., a random differential equation) characterized by parameters with common statistical

1. Clarence L. Johnson, "Analog Computer Techniques," McGraw-Hill Book Company, Inc., New York, pp. 45 to 64, 1956.

2. Floyd E. Nixon, "Principles of Automatic Controls," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 287 to 305, 1953.

properties. These statistical properties ordinarily take the form of production tolerances. Such tolerances must be considered concurrently with the search for optimum parameters if the simulator is to provide a reliable evaluation of the system performance.

The addition of random parameter values to a control system simulation not only allows a more realistic determination of performance, but it also permits a more systematic and organized assignment of production tolerances. Conventionally, while the mean values for parameters are selected primarily on the basis of performance, the tolerances are selected primarily on the basis of cost. These two indices are somewhat at odds. For example, if cost were no object, but the desire to maximize performance dominated, then the ideal choice for each tolerance would be zero. On the other hand, if one desired simply to minimize cost per unit, infinite tolerances (one hundred percent accpetance of production) would be the proper policy. In the practical situation, a compromise between performance and cost must be effected. This compromise is too often accomplished on the sometimes dubious basis of intuition and "engineering judgment". It is unfortunate that such methods are used at a decision level so vital to the ultimate performance and cost of a product; more so, since the trade-off can be made more objective through analog simulation. By using a simulator to obtain performance as a function of the tolerance assignment one can choose tolerances so as to maximize performance for a fixed cost or minimize cost for a fixed performance. In statistical terms, one can use the simulator to obtain optimum values for both the mean (nominal value) and standard deviation (production tolerances) for each important parameter.

A first step in this direction has been taken by R. C. Davis.[3] He postulated a linear dependence of a system performance index on parameter variances and obtained the necessary degradation coefficients by regression analysis of an analog computer experiment in which parameters were varied randomly. Although Davis did not explicitly mention cost, the coefficients he derived were utilized in juggling tolerances in such a way as to ease manufacturing difficulties without degrading performance. In principle, it would seem that these coefficients could be used to derive the original tolerances in an optimum manner. While Davis' method is probably the only sensible one if one is restricted to real time simulation, it will be demonstrated with a simple practical example, that there are systems for which such a linearization can lead to wholly erroneous and misleading conclusions. The following paragraphs describe an analog simulation technique leading to an optimum tolerance assignment which avoids the difficulties associated with an a priori assumption concerning the form of the functional dependence of performance on tolerances.

II. PROPOSED TOLERANCE OPTIMIZATION TECHNIQUE

A. Determination of the Performance Function

In order to arrive at an optimum set of tolerances, it is necessary to first determine the dependence of the control system performance index, $P(\vec{\sigma})$, on the tolerance vector $\vec{\sigma}$. It is proposed that $P(\vec{\sigma})$ be determined by simulation according to the following procedure:

---

3. R. C. Davis, "A Statistical Method for Analyzing the Effects of Missile Guidance System Tolerance on Hit Probability," Convair-Pomona, Proceedings of the Third Exploratory Conference on Missile Model Design for Reliability Prediction, White Sands Missile Range, pp 85 - 96, April 1959.

1. Optimum nominal parameters are determined in the usual way.

2. A tolerance vector, $\vec{\sigma}$, whose components are the standard deviations assigned to each parameter, is selected. This vector, together with the mean values determined in Step 1, is used to adjust filters operating on a stationary noise source so as to obtain voltages having the same mean and variance as the parameters under investigation.

3. These voltages are fed to integrators as initial conditions while the computer is in reset. Each integrator in turn provides the input to a multiplier representing the value of the random coefficient in the simulation. When the computer is placed in operate, the integrators have no input. Thus, the value of the coefficient is fixed during the solution of the differential equation but random when the computer is in reset. This effectively simulates a new production sample each time the computer is placed in operate.

4. A large number of runs is made with a fixed tolerance vector. An average performance index is obtained for the whole set of runs (e.g., probability of hit for a missile system).

5. The components of the tolerance vector are varied in a systematic way with a large number of computer runs being made for each new set of tolerances.

6. The results are plotted with the aid of regression analysis using a digital computer. An arbitrarily high order polynomial fit may be used if sufficient data is taken on the analog computer.

The enumerated steps lead to a representation of a system performance function, $P(\vec{\sigma})$, without an a priori assumption regarding the form of the functional dependence of the system performance on parameter values.

It should be noted that step 5 may easily involve hundreds of thousands runs. For this reason, it appears that the proposed simulation technique is economically feasible only when high speed analog computers such as the GPS (General Purpose Simulator Instruments Company) or Philbrick Repetitive Computer are employed.[4] These computers are capable of providing solutions at rates up to 50 cps and can compute average performance indices simultaneously with solution of the differential equations of the control system.

B. Determination of the Cost Function

Once $P(\vec{\sigma})$ is obtained, it is necessary to determine the dependence of the unit cost of the system, $C(\vec{\sigma})$, on the tolerance assignment. This is a difficult problem which has long concerned manufacturing industries. It does not appear that analog simulation can aid in the determination of $C(\vec{\sigma})$. Nevertheless, $C(\vec{\sigma})$ is required in order to optimize $\vec{\sigma}$ so it is therefore necessary to discuss the estimation of costs sufficiently to indicate the difficulties to be expected and the general nature of the cost function.

Among the factors to be taken into account in the determination of $C(\vec{\sigma})$ are the raw cost per unit, salvage value, maintenance costs resulting from tolerances too wide or too narrow, lifetime, etc. Initial estimation of $C(\vec{\sigma})$ based on such considerations can probably best be accomplished by reference to manufacturing experience with previous products similar to the one under consideration. As production proceeds, the original estimate may be refined on the basis of actual

---

4. Johnson, op. cit., pp. 222 - 232.

costs. It addition to the experience accumulated by various manufacturing organizations, considerable literature exists on the subject of cost estimation.[5,6]

While the general problem is much too extensive to treat here, one can observe from experience that $C(\vec{\sigma})$ is generally hyperbolic in nature. That is, reduction of any element of $\vec{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_n)$ to zero will result in infinite cost while minimum cost is achieved by making all tolerances infinite.

C. Optimization of the Tolerance Vector

After $P(\vec{\sigma})$ and $C(\vec{\sigma})$ have been determined, the optimization of $\vec{\sigma}$ may be considered. The first step required is to select a criterion for optimization. An obvious criterion is to optimize performance for a fixed cost. That is, on a surface of constant cost in the $\vec{\sigma}$ space, one seeks the point of maximum performance. On the other hand, it might be more desirable to minimize cost for a fixed performance. More complicated optimization criteria are possible by specifying the cost one is willing to accept as a function of performance, etc.

As more data becomes available, the estimate of the cost function will change, thereby shifting the optimum. However, the original performance function, $P(\vec{\sigma})$, remains valid so the apportionment of tolerances may be altered to a new optimum without further simulation.

III. EXAMPLE

A. Description of Simulation

In order to establish the feasibility of the proposed technique, a simulation of a hypothetical radar homing missile was undertaken. Besides random

---

5. D. H. Evans, "Optimum Tolerance Assignment to Yield Minimum Manufacturing Cost," Bell System Technical Journal, Vol. 37, pp. 461 - 484, March 1958.

6. E. W. Pike and T. R. Silverberg, "Assigning Tolerances for Maximum Economy," Machine Design, Vol. 25, pg. 139, September 1953.

parameters, this simulation included random radar tracking noise and a random missile heading error at launch. The performance index selected was probability of hit, P_h as measured against an idealized strip target. To limit the amount of data to be taken, only two parameters were varied randomly. These parameters were the navigation gain, K, and the principal missile filtering time constant, τ. The missile simulated derived steering information from the angular rate of the line of sight from missile to target so K is given by

$$K \equiv \frac{\text{steady state missile turning rate}}{\text{line of sight angular rate}} \tag{1}$$

Figure 1 is a functional block diagram of the missile simulated.

To measure P_h with sufficient precision, it was necessary to simulate five hundred attacks for each value of the tolerance vector $\vec{\sigma} = (\sigma_K, \sigma_\tau)$. By considering four values for each tolerance, a total of sixteen cases was generated resulting in a grand total of eight thousand simulated attacks. Since a high speed computer was not available at the time, this simulation was performed on a real time basis using automatic sequencing equipment. The total running time was about one week. Thus, this example shows quite clearly that a problem of realistic complexity will require high speed simulation.

In this simulation, it was assumed that the population statistics for both K and τ were Gaussian with mean values K_o and τ_o and standard deviation σ_K and σ_τ respectively. The sample values were therefore obtained by low pass filtering of noise generators using integrators which did not reset. Figure 2 shows the computer program used to randomly vary the navigation gain, K, from run to run.

Figure 3 is a table of the results obtained in the simulation. This table presents the measured P_h as a function of the percentage standard deviation

assigned to the randomized parameters, K and τ. The raw data of this table was subjected to quadratic regression analysis on σ_K and σ_τ using a digital computer; i.e., the data points were fitted in a least square sense by

$$P_h = P_o + a_1\sigma_K + a_2\sigma_\tau + a_{11}\sigma_K^2 + a_{12}\sigma_K\sigma_\tau + a_{22}\sigma_\tau^2 \tag{2}$$

The results are plotted in Figure 4. Figure 5 is a contour map of Figure 4 with hypothetical cost curves included. Examples of minimum cost and maximum performance optima are indicated.

B. Significance of the Experimental Results

The first and most obvious observation that can be made from Figure 4 is that linearization of the dependence of P_h on σ_K and σ_τ is impossible. Linear dependence would result in equally spaced, parallel, straight lines on Figure 4. Replotting with σ_K^2 as the abscissa would show that linearization of the dependence of P_h on σ_K^2 and σ_τ^2 is also impossible. Further examination of the figure reveals the remarkable fact that the performance of this system can actually be degraded by tightening tolerances. For example, when $\sigma_\tau = 30$ percent, the probability of hit is decreased from .28 to .24 when the tolerance on σ_K is reduced from 30 percent to 0 percent. This is quite at odds with the widespread opinion that when tolerances are relaxed in one part of a system, they must be tightened elsewhere to maintain the system performance. In the present example, this apparently anamolous behaviour results from a cancellation of effects when K and τ both vary widely from their nominal values.

A more general and far reaching conclusion arising from this example is concerned with the form of the mathematical model selected to represent the dependence of the system performance index on parameter _values_. In the more conventional

approach to the study of tolerances, one begins by assuming a specific model for this dependence.[7] For example, on the basis of experience gained in the optimization of nominal values, the system designer might decide that an adequate model for the relationship between P_h and the values of the parameters, K and τ, is given by the quadratic expression:

$$P_h = \gamma_0 + \gamma_1(K - K_0) + \gamma_2(\tau - \tau_0) + \gamma_{11}(K - K_0)^2$$
$$+ \gamma_{12}(\tau - \tau_0)(K - K_0) + \gamma_{22}(\tau - \tau_0)^2$$

(3)

Now since the values of K and τ were independent in this experiment (and very often in practice) the average or expected value of P_h measured over the ensemble of missiles is given by

$$P_{h \; ave} = E(P_h) = \gamma_0 + \gamma_{11}E(K - K_0)^2 + \gamma_{22}E(\tau - \tau_0)^2 \quad (4)$$

where the symbol E denotes the expected value or theoretical average resulting from repeated estimation of P_h. The other terms vanish because they have zero expected value. Now, by definition, the ensemble averages in the above expression are just the variances; i.e.,

$$E(K - K_0)^2 \equiv \sigma_K^2 = \text{variance of K} \quad (5)$$

$$E(\tau - \tau_0)^2 \equiv \sigma_\tau^2 = \text{variance of } \tau \quad (6)$$

so

$$P_{h \; ave} = \gamma_0 + \gamma_{11}\sigma_K^2 + \gamma_{22}\sigma_\tau^2 \quad (7)$$

---

7. Evans, op. cit.

This is the analytic origin of the common assumption that performance depends linearly on the variances. If one accepts this assumption, then all that is required is to perform an experiment, selecting many values of the variables and recording the performance index along with the values of the variables. Regression analysis is then accomplished on equation (3) yielding the desired coefficients, Υ_o, Υ_{11}, and Υ_{22}.

While the above approach will succeed if the assumed model (3) is in fact correct, it has already been shown that the resulting conclusion of linear dependence of performance on parameter variances cannot possibly describe the behaviour of the example at hand. The reason is that the effect of the terms σ_1, σ_2, and $\sigma_1\sigma_2$ are absent from (7) whereas a regression analysis of the experimental results based on (2) shows that $\sigma_1\sigma_2$ is in fact the most significant term. This term represents interaction effects which are of prime importance in this system. The actual financial cost of ignoring this term could be considerable since utilization of (7) would lead to both unnecessarily tight tolerances on K and degraded performance when σ_τ is large.

The process of deriving the effects of tolerances from a model for performance dependence on parameter values can be turned around. That is, if one knows the coefficients in an equation like (2)

$$P_h = P_o + a_1\sigma_K + a_2\sigma_\tau + a_{11}\sigma_K^2 + a_{12}\sigma_K\sigma_\tau + a_{22}\sigma_\tau^2 \tag{2}$$

then the effect of parameter values can be inferred (though not uniquely). For example, if it were true that

$$P_h = P_o + a_1'\left|K - K_o\right| + a_2'\left|\tau - \tau_o\right| + a_{11}(K - K_o)^2$$
$$+ a_{12}'\left|K - K_o\right|\left|\tau - \tau_o\right| + a_{22}(\tau - \tau_o)^2 \tag{8}$$

then (2) would follow and the coefficients in (8) could be obtained from a regression on tolerances such as was performed in the present example.

To summarize, in the light of the preceding discussion, it is maintained that one should in general reverse the usual approach to tolerance analysis by:

1. Performing a simulation with randomized coefficients as described in Section II.

2. Derive the mathematical model for the system from a regression analysis of the dependence of system performance on the tolerance vector.

In this way, new insight can be obtained into the relationship between performance and parameter values and serious errors in tolerance assignment can be avoided.

IV. DESIGN OF EXPERIMENTS

A. The Need for Experimental Design

The experiment just described involved only two parameters. In a more realistic problem, it is likely that a much larger number of system parameters would be considered in the assignment of optimum tolerances. Utilization of the method presented in this paper leads to an alarming increase in the total number of runs as the number of parameters increases unless the experiment is carefully designed. The problem of determining just how to take data so as to obtain the most information with a minimum of effort falls within the realm of statistical design of experiment. This field is so extensive that it is neither possible nor desirable to consider it in detail in this paper. Fortunately, there are several excellent texts on the subject which treat a variety of types of experiment.[8]

---

8. Cochran, W. G. and Cox, G. M., "Experimental Designs," John Wiley and Sons, Inc., New York, 1950.

While a general discussion of design of experiment is not appropriate, there are two aspects of this field which are particularily useful in experiments of the type discussed in this paper. In the first place, it is necessary to be able to estimate the total amount of data required to define $P(\vec{\sigma})$ before beginning the experiment. Secondly, in the event that this number is too large, it is very desirable to have a means for reducing this total by a judicious choice of data points in the tolerance vector space. Both of these problems are considered below in connection with the present experiment.

B. Estimation of the Amount of Data

In the missile simulated, the measured values of the performance index, P_h, possess a binomial distribution because they result from repeated trials of the same experiment. One can therefore estimate the variability of successive measurements of P_h.[9] It turns out that if P_h is the true or theoretical probability of hit and $\hat{P}_h$ is the measured ratio of hits to total firings, then

$$P_h = \hat{P}_h \pm 1/\sqrt{n} \tag{9}$$

n = total number of simulated firings

with probability .95 or better. Consequently, since the data in Figure 3 was obtained by making five hundred simulated firings per data point,

$$P_h = \hat{P}_h \pm 1/\sqrt{500} \cong P_h \pm .045 \tag{10}$$

---

9. Hoel, Paul G., "Introduction to Mathematical Statistics," John Wiley and Sons, New York, Chapter 10, 1947.

That is, we can have at least ninety-five percent confidence that the true P_h lies within $\pm$ .045 of the values listed in Figure 3.[10] Utilization of quadratic regression (2) considerably improves the confidence we may have in the final result of Figure 3.

If one were to decide that a precision of $\pm$ .045 with ninety-five percent confidence was required of every point and furthermore that m values of each of k parameters must be considered, then for this example the total number of runs would be given by

$$N = 500 \, m^k \tag{11}$$

This number may be so large as to make a "brute force" determination of $\vec{P(\sigma)}$ impractical, even with a high speed computer. In such an event, some means such as the experimental design described below must be found for reducing this total.

C. Factorial Design

In the example treated in this paper, the data points were taken at uniformly spaced points in a σ_K, σ_τ space (Figure 3). Fortunately, it is not necessary to take data at all of these points to determine a quadratic relationship between $\vec{P(\sigma)}$ and $\vec{\sigma}$. One type of design of experiment which allows fewer data points to be considered is called "fractional factorial design".[11] This design method involves a sequential determination of the points at which observations should

---

10. Methods also exist for the estimation of confidence intervals in the event that the variables do not possess a binomial distribution. Ibid, Chapter 10.

11. J. S. Hunter, "Designing and Interpreting Tests," Control Engineering, pps. 137 - 141, September 1959.

be made and is particularly well suited to the determination of $P(\vec{\sigma})$. The total

number of computer runs may be drastically reduced, amounting possibly to several

orders of magnitude in the total effort required to define $P(\vec{\sigma})$. While the com-

plete theory is somewhat involved, Figure 3 can be used to illustrate

the basic idea of fractional factorial design.

Referring to the experimental example, if one knew the relative importance

of the terms σ_K, σ_τ, and $\sigma_K\sigma_\tau$ in determining $P(\vec{\sigma})$, then a more intelligent

selection of sampling points could be made. Now an <u>estimate</u> of the relative

effects can be obtained by evaluating $P(\vec{\sigma})$ only at the vertices of the table of

Figure 3. Figure 6 is an abreviated table obtained by using only the four corners

of this table. The relative effect of each of the three terms above is obtained

as follows:

1. Estimated effect of σ_K = average of P_h over column (2) - average P_h

    over column (1) = $\dfrac{.29 + .27}{2} - \dfrac{.34 + .24}{2} = -.01$

2. Estimated effect of σ_τ = average of P_h over row (2) - average P_h over

    row (1) = $\dfrac{.24 + .27}{2} - \dfrac{.34 + .29}{2} = -.06$

3. Estimated effect of $\sigma_K\sigma_\tau$ = average difference of row (2) - average

    difference of row (1) = $\dfrac{.27 - .24}{2} - \dfrac{.29 - .34}{2} = .04$

Thus is can be seen that the "interaction effect" of $\sigma_K\sigma_\tau$ is greater than the

"main effect" due to σ_K and nearly as great as the "main effect" due to σ_τ.

While this result is obtained from only four data points, it is (qualitatively)

quite in accord with the final result obtained by regression analysis (Figure 4).

The significance of this test is simply that the $\sigma_K\sigma_\tau$ term cannot be dropped from

the regression equation (2). There is, however, a possibility of dropping the

linear term σ_K. If this were done, then less data would be required to estimate

the remaining regression coefficients. A more sophisticated approach to fractional factorial design must consider confidence intervals and other effects to determine whether or not a given term can be dropped from (2).

When a large number of parameter tolerances are involved in $\vec{\sigma}$, the dropping of even one term from the regression equation may provide a considerable saving in the simulation effort required to obtain $P(\vec{\sigma})$. Even if no terms can be dropped from the regression equation, uniformly spaced data points do not provide the best grid for measurement of $P(\vec{\sigma})$. Further development of fractional factorial design provides a solution to the problem of determining the optimum location for data points in the parameter tolerance space.

V. SPECIAL SIMULATION CONSIDERATIONS

A. Correlation Between Successive Parameter Values

In order to simplify the analysis of data, it is desirable that successive values of parameters selected from noise generators be uncorrelated. For noise sources shaped by first order filtering, the covariance of output samples separated by time T is given by

$$\rho(T) \ = \ e^{-|T|/\tau_1} \tag{12}$$

where τ_1 is the filter time constant.[12] Consequently, to reduce $\rho(T)$ to a small value, T should be more than about $5\tau_1$.

---

12. James L. Lawson and George E. Uhlenbeck, "Threshold Signals," McGraw-Hill Book Company, New York, pg. 42, 1950.

B. Multiplexing of Noise Generators

In a problem of realistic complexity, a sizeable number of uncorrelated noise voltages are required to provide parameter values. These voltages can be obtained from a single noise generator by frequency multiplexing techniques. Rice has shown[13] that an ensemble of samples drawn from a Gaussian noise generator can be described by a Fourier series with uncorrelated random coefficients. Consequently, if such a signal is passed through a bank of bandpass filters, the correlation between filter outputs can be made as small as desired by reducing the spectral overlap of the filters sufficiently. In this process, the correlation between successive samples from a single filter must also be considered. This factor sets a lower limit on the bandwidth of the individual filters.

C. High Speed Data Processing

When solution rates are as high as the 50 cps capability of repetitive computers, data reduction must be at least partially accomplished by the computer. One such computer (GPS) has a probability distribution analyser which automatically provides probability of hit. This means that there is no need to process individual solutions off the computer. It appears that such an instrument or a related one is essential in simulations where hundreds of thousands of runs are required.

VI. <u>CONCLUSIONS</u>

The role of simulation in the preliminary design of control systems can be expanded to include selection of optimum manufacturing tolerances. If a high

---

13. S. O. Rice, "Mathematical Analysis of Random Noise," published in "Selected Papers on Noise and Stochastic Processes," edited by Nelson Wax, Dover Publications, Inc., New York, pp. 157 to 161, 1954.

speed computer is available, one may abandon linear models for the effect of varying tolerances and thereby obtain a realistic method for performing the crucial trade-off between cost and performance. A lesser effort is apt to lead to false conclusions concerning system performance.

PROFESSIONAL BIOGRAPHY

Arnold Levine was born in Brooklyn, New York, on January 26, 1931. He received his B.S. in Physics from M.I.T. in June 1953.

From 1953 to 1954 he worked on inertial guidance systems at the Instrumentation Laboratory of M.I.T. and attended the M.I.T. graduate school. Presently he is studying for his Ph.D. in statistics at the University of California at Los Angeles.

Between 1954 and 1956 he served with the U.S. Army working on the evaluation of the Missile Master System.

Since 1956 he has been employed by the System Development Laboratories, Hughes Aircraft Company, Culver City, California. From 1958 to 1959 he taught probability and statistics on a part-time basis at the University of Southern California.

He is a member of the mathematics honorary society Pi Mu Epsilon.

PROFESSIONAL BIOGRAPHY

Robert B. McGhee was born on June 6, 1929 in Detroit, Michigan. He received the B.S. Degree in engineering physics from the University of Michigan in 1952 and the M.S. degree in electrical engineering from the University of Southern California in 1957. At the present time he is working toward the Ph.D. degree in electrical engineering at the latter university.

From 1952 to 1955 he served in the Ordnance Corps, U.S. Army, at the Ordnance Guided Missile School, Redstone Arsenal, Huntsville, Alabama. Since 1955, he has been employed by System Development Laboratories, Hughes Aircraft Company, Culver City, California. During this time he has been concerned primarily with the analysis and simulation of air-to-air guided missile systems. Since 1957, he has taught courses in servomechanisms on a part-time basis at the University of Southern California.

Mr. McGhee is a member of Tau Beta Pi.