NASA TECHNICAL NOTE

NASA TN D-4617

NASA TN D-4617

COMPUTER PROGRAM FOR SYMBOLIC REDUCTION OF BLOCK DIAGRAMS USING FORMAC

by Carl F. Lorenzo and Paul Swigert

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1968

COMPUTER PROGRAM FOR SYMBOLIC REDUCTION OF BLOCK DIAGRAMS USING FORMAC

By Carl F. Lorenzo and Paul Swigert

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

ABSTRACT

Techniques have been established and a computer program has been written (in the experimental language, FORMAC) to symbolically reduce arbitrary block diagrams for desired transfer functions. Symbolic solutions are determined in several forms including an expanded form in terms of the driving frequency and system constants. Programs are written to numerically evaluate the symbolic solutions for real and imaginary parts and magnitude ratio and phase angle. The programs have been applied to several research problems which include both lumped and distributed parameter systems. The latter forms are built into the program and are handled automatically.

STAR Category 08

CONTENTS

Page

SUMMARY 1

INTRODUCTION 1

GENERAL SCHEME 2

Algebriac Manipulations 6

Numerical Manipulations 9

Problem Types 10

IMPLEMENTATION 12

FORMAC Program (REDUCE) 12

FORMAC Input/Output 15

FORTRAN Program (EVAL) 15

FORTRAN Input/Output 15

APPLICATIONS 16

Application 1: Interacting-Control System 16

Application 2: Mass-Spring-Damper Dynamics 18

Application 3: Hydraulic-Line-Dynamics Study 22

Other Applications 26

CONCLUDING REMARKS 26

APPENDIXES

A - SYMBOL LIST 28

B - USER'S MANUAL 29

C - DESCRIPTIONS, LISTINGS, AND FLOW CHARTS FOR COMPUTER PROGRAMS 41

D - METHOD OF DETERMINING ORDER 72

REFERENCES 74

COMPUTER PROGRAM FOR SYMBOLIC REDUCTION OF BLOCK DIAGRAMS USING FORMAC

by Carl F. Lorenzo and Paul Swigert

Lewis Research Center

SUMMARY

Techniques have been established and a computer program has been written (in the experimental language, FORMAC) to symbolically reduce arbitrary block diagrams for desired transfer functions. Symbolic solutions are determined in several forms including an expanded form in terms of the driving frequency and system constants.

Programs are written to numerically evaluate the symbolic solutions for real and imaginary parts and magnitude ratio and phase angle.

The programs have been applied to several research problems which include both lumped and distributed parameter systems. The distributed-parameter forms are built into the program and are handled automatically.

INTRODUCTION

The use of block diagrams has become widespread for the analysis of dynamic problems. The classes of dynamic problems handled with this technique include controls and servomechanisms, various physical dynamic problems from acoustics to structural dynamics, and such engineering tasks as transient heat-transfer and circuit analyses.

In cases of block diagrams with few blocks or paths', symbolic solution is rather routine. However, as the path and block complexity increases, manual symbolic solution becomes more difficult and, indeed, prohibitive. The manual reduction of a block diagram is an algebraic task which is at best routine and can often be difficult and/or tedious. Also, the process is susceptible to human error and is time consuming.

In the past, these more complex problems have been attacked by use of computer programs to allow at least numerical solutions. These programs have used matrix methods and require a complete matrix manipulation for each frequency (refs. 1 to 3).

The purpose of this study is to evolve a digital computer program that can reduce any block diagram symbolically to obtain a transfer function of interest, that is, to generate expressions for the transfer function either in terms of the G's (the transfer functions for the blocks), or S (the Laplace operator), or ω (the driving frequency).

A further goal is to supply programs by which the symbolic solutions can be efficiently evaluated numerically for magnitude ratio, phase angle, and real and imaginary parts as functions of frequency.

Such programs would allow the engineer to focus his attention on detailed evaluation of the model (contents of each block) and interpretation of the end results of the analysis.

The value of the symbolic expression is that (1) it may be used in research applications where such expressions are useful for some further analytical purpose, (2) the mathematical expression is a most compact manner for communicating the desired information (transfer function) for reports, etc., and (3) the analytical expression allows efficient numerical evaluation for cases of particular interest.

The present work differs from the previous efforts (ref. 1) in that symbolic solutions will be generated (in minimum form) using FORMAC. These symbolic solutions will be evaluated directly for numerical results, therein bypassing the matrix manipulation.

This report is organized in the following manner. The first section will be general background information with some theoretical preliminary material to show how block diagrams are reduced. The next two sections will give detailed information as to how the general ideas are implemented into digital programs and how those programs function. An applications section demonstrates program results for both lumped parameter and distributed parameter systems. A users manual is presented which gives a step-by-step use of the program with an example.

GENERAL SCHEME

The basic elements composing block diagrams are

(1) Summers - add or subtract n-signals

(2) Blocks - accept <u>one</u> signal and modify it by an S or frequency-sensitive operator

(3) Nodes - split a signal into two or more parts

Any linear block diagram can be composed of these elements in various combinations.

The information contained in the block diagram when the blocks have some generalized form (G-form) is basically topological information (ref. 4). When the expressions for the G's are put into the diagram, the information becomes system information.

Consider the problem of finding an arbitrary transfer function X_a/X_b for some given block diagram. (Symbols are defined in appendix A.) For a sufficiently general block diagram, there will be several solutions: (1) the forward transfer function, (2) the

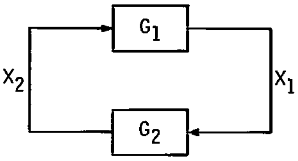


Figure 1. - Two-element block diagram.

external-loop transfer function, and (3) the transfer function with respect to a given input. The forward and external-loop transfer functions are illustrated in the simple block diagram of figure 1. It is characterized by the equations:

$$X_1 = G_1X_2$$

and

$$X_2 = G_2X_1$$

To form the transfer function X_1/X_2, clearly, two results are possible:

$$\frac{X_1}{X_2} = G_1$$

and

$$\frac{X_1}{X_2} = \frac{1}{G_2}$$

using the first and second equations, respectively. This is also true of larger, more complex diagrams. Generally, it is the forward transfer function ($X_1/X_2 = G_1$) that is of interest in the solution of controls problems. When a block diagram is reduced manually, the problem does not usually occur because the engineer is discriminatory in his selection of equations (blocks or summers) used to reduce the system; that is, he takes into account the direction of signal flow. It is undesirable to account for signal flow direction in the computer program. Therefore, it is best to avoid this problem by forming only transfer functions involving an input, as opposed to those containing two internal signals in the block diagram.

The effect of an input on the solution is related to the question of forward and external-loop transfer functions. To illustrate, the block diagram of figure 2 has two

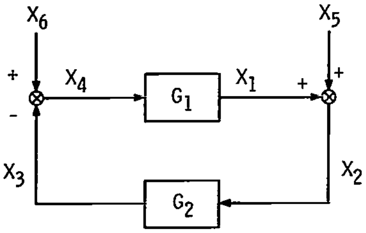


Figure 2. - Two-input block diagram.

inputs. As a result of this, there will be two transfer function relations for X_2/X_4, for example. Consider the system equations:

$$X_1 - G_1X_4 = 0 \tag{1}$$

$$X_3 - G_2X_2 = 0 \tag{2}$$

$$X_6 - X_3 - X_4 = 0 \tag{3}$$

$$X_5 + X_1 - X_2 = 0 \tag{4}$$

It is easily shown that for X_5 as input ($X_6 = 0$)

$$\frac{X_2}{X_5} = \frac{1}{1 + G_1G_2}$$

also,

$$\frac{X_4}{X_5} = \frac{-G_2}{1 + G_1G_2}$$

Hence,

$$\frac{X_2}{X_4} = \frac{\dfrac{1}{1 + G_1G_2}}{\dfrac{-G_2}{1 + G_1G_2}} = -\frac{1}{G_2} \qquad \text{for } X_5 \text{ as an input}$$

It can be further shown that for X_6 as input ($X_5 = 0$)

$$\frac{X_2}{X_6} = \frac{G_1}{1 + G_1G_2}$$

and

$$\frac{X_4}{X_6} = \frac{1}{1 + G_1G_2}$$

Hence,

$$\frac{X_2}{X_4} = \frac{\dfrac{G_1}{1 + G_1G_2}}{\dfrac{1}{1 + G_1G_2}} = G_1 \qquad \text{for } X_6 \text{ as an input}$$

It is concluded, therefore, that the transfer function between two arbitrary points (neither an input) in a block diagram requires the specification of an input as reference; that is, total specification of a transfer function should be X_a/X_b with X_c as an input (which implies that all other inputs are zero). Or how does X_a respond to X_b when X_c (only) is stimulated? It is interesting to note that, when both inputs $X_5 = X_6 = 0$, equations (1) and (4) yield $X_2/X_4 = G_1$ and equations (2) and (3) yield $X_2/X_4 = -1/G_2$. These, by analogy to the previous example, would be considered the forward and external-loop transfer functions. This also indicates that, if all the equations are not required for a reduction, the solution is probably not unique. Were it not for this fact, it would be difficult to determine whether the solutions for large complex diagrams were unique.

The problem of nonunique solutions is easily remedied in the programs which follow by only forming transfer functions with respect to an input. Experience indicates that this results in a unique solution. Ratios of the proper results form the arbitrary transfer functions; that is,

$$\frac{\dfrac{X_a}{X_m}}{\dfrac{X_b}{X_m}} = \frac{X_a}{X_b} \qquad \text{for input } X_m$$

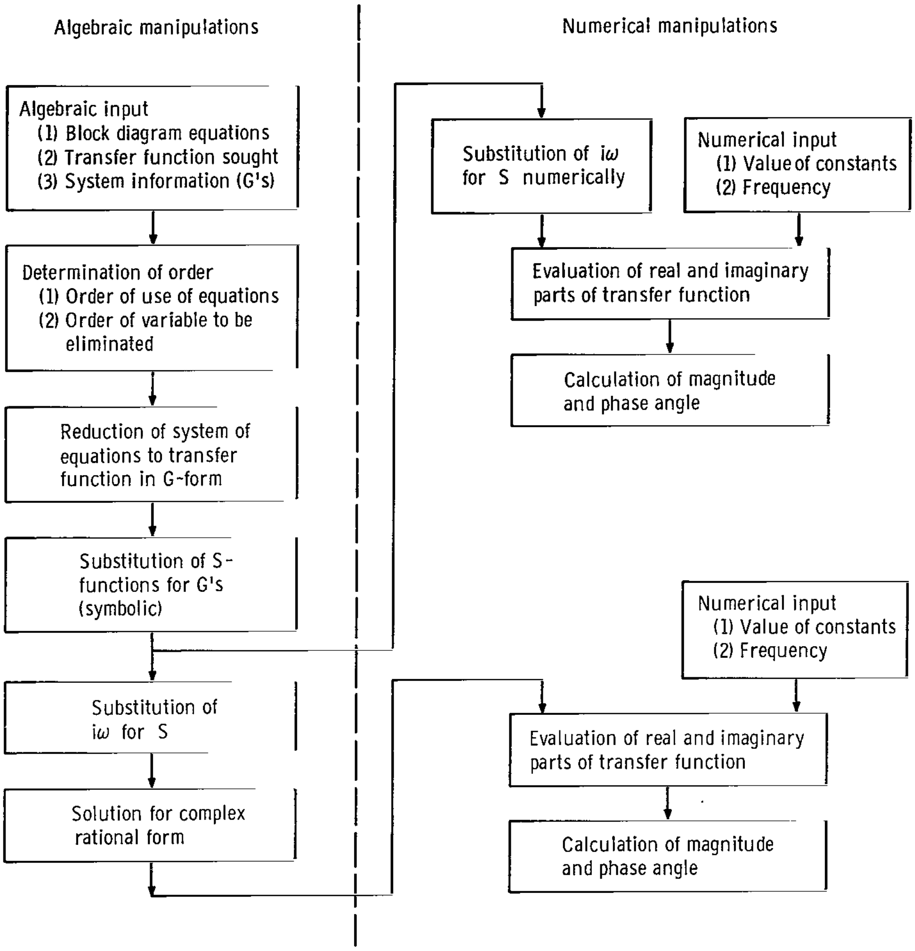

Algebraic manipulations
Numerical manipulations
Algebraic input
(1) Block diagram equations
(2) Transfer function sought
(3) System information (G's)
Determination of order
(1) Order of use of equations
(2) Order of variable to be eliminated
Reduction of system of equations to transfer function in G-form
Substitution of S-functions for G's (symbolic)
Substitution of iω for S
Solution for complex rational form
Substitution of iω for S numerically
Numerical input
(1) Value of constants
(2) Frequency
Evaluation of real and imaginary parts of transfer function
Calculation of magnitude and phase angle
Numerical input
(1) Value of constants
(2) Frequency
Evaluation of real and imaginary parts of transfer function
Calculation of magnitude and phase angle

Figure 3. - Simplified flow chart for block-diagram-reduction technique.

It is further noted that, if minimum form solutions are attained for X_a/X_m and X_b/X_m, the denominators (characteristic parts) will generally be identical. Hence, only the ratio of the numerators are needed for the desired result. This can be validated by applications of Cramer's rule.

A simplified flow chart for the process of reduction of a block diagram (to a desired transfer function) is presented in figure 3. The flow chart is composed of two basic parts: (1) those processes involving algebraic manipulations and (2) those processes involving numerical manipulations.

Algebraic Manipulations

The algebraic manipulations, of course, precede the numerical. The first block (fig. 3) indicates the information required for the algebraic (or symbolic) reduction.

These are (1) the equations corresponding to each element of the block diagram (one equation per element), (2) the transfer function sought from the reduction, and (3) the contents of the blocks of the diagram, that is, the equations expressing the G's in terms of the Laplace variable S and the various time constants and natural frequencies, etc. These data are sufficient to allow reduction to a symbolic form.

Before the actual algebriac reduction can take place, the order in which the variables are to be eliminated and the equations to be used for such eliminations must be determined. For a manual reduction this is usually done by observation or intuition. The technique of ordering will be discussed more fully in appendix D.

Having a desirable order of elimination, the reduction takes place in the following manner. The first variable to be eliminated is solved for in the equation indicated by the order determination. This equation is then substituted into all other equations of the set containing that variable, thereby eliminating it. It is important that the variable be substituted in all equations; otherwise, the possibility of looping occurs. (Looping is repeated substitution of variables without achieving elimination.) This substitution process is continued for all the remaining variables except the two variables involved in the desired transfer function. This leaves a single equation in the two variables which is now manipulated to form a ratio only involving the G's. For example:

$$\frac{X_a}{X_b} = \frac{1}{1 + G_1 G_2} \tag{5}$$

This is the so-called G-form of the transfer functions solution. The G-functions can always be expressed as a ratio, that is, $G_a = N_a/D_a$. This ratio is now substituted into equation (5) which results in

$$\frac{X_a}{X_b} = \frac{1}{1 + \frac{N_1}{D_1}\frac{N_2}{D_2}} \tag{6}$$

Solving for a simple ratio for X_a/X_b gives

$$\frac{X_a}{X_b} = \frac{D_1 D_2}{D_1 D_2 + N_1 N_2} \tag{7}$$

for the example taken. In this form, the system information is easily introduced into the result. Assume that

$$\left.\begin{array}{ll} & G_1 = \dfrac{\tau_1 S + 1}{\tau_2 S + 1} \\ \text{and} & \\ & G_2 = \dfrac{\tau_3 S + 1}{\tau_4 S^2 + 1} \end{array}\right\} \tag{8}$$

The fourth block (flow chart, fig. 3) indicates that equation (8) is to be substituted in equation (7), thus,

$$\frac{X_a}{X_b} = \frac{\left(\tau_2 S + 1\right)\left(\tau_4 S^2 + 1\right)}{\left(\tau_2 S + 1\right)\left(\tau_4 S^2 + 1\right) + \left(\tau_1 S + 1\right)\left(\tau_3 S + 1\right)} \tag{9}$$

This is the S-form solution for the transfer function (unexpanded). From this form, either numerical evaluation is possible or further symbolic reduction can be done. For the symbolic result, S is now replaced by $i\omega$ (a step which is valid for wide set of conditions). (See ref. 5.) Hence,

$$\frac{X_a}{X_b} = \frac{\left(\tau_2 i\omega + 1\right)\left(\tau_4 i^2\omega^2 + 1\right)}{\left(\tau_2 i\omega + 1\right)\left(\tau_4 i^2\omega^2 + 1\right) + \left(\tau_1 i\omega + 1\right)\left(\tau_3 i\omega + 1\right)}$$

Now, $i^2 = -1$, thus

$$\frac{X_a}{X_b} = \frac{\left(\tau_2 i\omega + 1\right)\left(-\tau_4\omega^2 + 1\right)}{\left(\tau_2 i\omega + 1\right)\left(-\tau_4\omega^2 + 1\right) + \left(\tau_1 i\omega + 1\right)\left(\tau_3 i\omega + 1\right)}$$

and

$$\frac{X_a}{X_b} = \frac{\left(1 - \tau_4\omega^2\right) + i\left(-\tau_2\tau_4\omega^3 + \tau_2\omega\right)}{\left[2 - \left(\tau_1\tau_3 + \tau_4\right)\omega^2\right] + i\left[\left(\tau_1 + \tau_2 + \tau_3\right)\omega - \tau_2\tau_4\omega^3\right]}$$

This has been called the complex rational form in the sixth block of the flow chart (fig. 3), that is, the form

$$\frac{X_a}{X_b} = \frac{A + iB}{C + iD} \tag{10}$$

where A, B, C, and D are functions of the excitation frequency and system parameters, τ's, ω_n's, K's, etc. This form is very important in many research studies. From this form, the magnitude ratio and phase angle follow directly because

$$\operatorname{Re}\left(\frac{X_a}{X_b}\right) = \frac{AC + BD}{C^2 + D^2} \tag{11}$$

$$\operatorname{Imag}\left(\frac{X_a}{X_b}\right) = \frac{CB - AD}{C^2 + D^2} \tag{12}$$

therefore,

$$\text{Phase angle} = \tan^{-1}\left(\frac{CB - AD}{AC + BD}\right) \tag{13}$$

also,

$$\text{Magnitude of } \frac{X_a}{X_b} = \left|\frac{X_a}{X_b}\right| = \frac{\sqrt{A^2 + B^2}}{\sqrt{C^2 + D^2}} = \sqrt{\frac{A^2 + B^2}{C^2 + D^2}} \tag{14}$$

Numerical Manipulations

For the numerical manipulations, when the path starting with the S-form is considered, it is merely necessary to substitute the numerical values for the frequency and the system parameters then to use complex arithmetic to obtain the numerical equivalent of equation (10). From this, the quantities of equations (11) to (14) are readily evaluated.

The path from the complex rational form is similar except that the substitutions are made into equation (10) directly.

A further note on the importance of the order used in the reductions: if the proper order is not used in the reduction scheme presented, it is possible to introduce a common factor into both the numerator and denominator (of eq. (9), e.g.). Now, although this result is still technically correct, this could potentially introduce hundreds or thousands of extra terms in the symbolic solution. This is important in the computer program which follows because it seriously affects storage capabilities and, hence, problem size capability.

Problem Types

The type of problems that can be handled using these techniques is limited to linear systems. Hence, the contents of the blocks (the G's) can only be functions of S and constants; that is, time varying constants (system characteristics) are not permissible.

For the so-called lumped parameter systems, the G's are sums and products of KS terms and constant terms.

For distributed parameter systems, namely, systems where the blocks are solutions to the wave, diffusion, or beam equation or where they are dead times, the G's will generally occur in the following forms (one dimensional):

$$\sinh\left(d\sqrt{as^2 + bs + c}\right) \qquad (15)$$

$$\cosh\left(d\sqrt{as^2 + bs + c}\right) \qquad (16)$$

$$\sin\left(d\sqrt{as^2 + bs + c}\right) \qquad (17)$$

$$\cos\left(d\sqrt{as^2 + bs + c}\right) \qquad (18)$$

$$\exp\left(d\sqrt{as^2 + bs + c}\right) \qquad (19)$$

$$\left(\sqrt{as^2 + bs + c}\right) \qquad (20)$$

Here again, this is true provided the partial differential equation involved is not of the time varying coefficient type.

The preceding forms are built into the programs described in the next section. The following identities are useful when these forms are involved in either a manual reduction or in the programs. The program symbolic output is RSINH (real part of sinh), ISINH (imaginary part of sinh), etc., when $z = x + iy$.

$$\sinh z = \sinh x \cos y + i \cosh x \sin y \tag{21}$$

$$\cosh z = \cosh x \cos y + i \sinh x \sin y \tag{22}$$

$$\sin z = \sin x \cosh y + i \cos x \sinh y \tag{23}$$

$$\cos z = \cos x \cosh y - i \sin x \sinh y \tag{24}$$

$$e^{z} = e^{x} \cos y + i\, e^{x} \sin y \tag{25}$$

These equations can be found in reference 6. In the event of a square-root argument, the following equations are helpful:

$$\sqrt{z} = \sqrt{x + iy} = u + iv \tag{26}$$

If $x \geq 0$

$$\left.\begin{aligned} u &= \sqrt{\frac{1}{2}|x| + \frac{1}{2}|x + iy|} \\ v &= \frac{y}{2u} \end{aligned}\right\} \tag{27}$$

If $x \leq 0$

$$\left.\begin{aligned} u &= y/2v \\ v &= (\operatorname{sgn} y)\sqrt{\frac{1}{2}|x| + \frac{1}{2}|x + iy|} \end{aligned}\right\} \tag{28}$$

The forms (21) to (25) result when the various partial differential equations are solved in rectangular coordinate systems. When this is not the case, other forms can be introduced, for example, Bessel functions and Legendre functions, etc. When this is the case, it becomes desirable to have some arbitrary unassigned functions which can represent such a solution. It is only necessary that expressions (similar to (21) to (25)) can be written for the real and imaginary parts. It is interesting to note that the sinh, cosh, sin, and cos will not have square-root arguments unless damped versions of the original partial differential equations (wave or beam) are solved. Whereas, the square-root form is usually required for the diffusion equation solutions, that is, $\sinh\left(X\sqrt{S}\right)$ and $\exp\left(d\sqrt{S}\right)$. (See ref. 7 for typical results.)

IMPLEMENTATION

The complete solution of the block-diagram reduction is obtained through the use of two separate computer programs: the algebriac solution and the numerical solution. The algebriac computer program required the use of the FORmula MAnipulation Compiler (FORMAC) (ref. 8). The FORMAC language is an experimental extension of FORTRAN IV for the IBM 7094 computer. This extension provides the capability for doing algebriac manipulation by using symbol manipulation. The numerical solution of the FORMAC expressions is performed by a program written in FORTRAN IV. Appendix B presents a User's Manual for these programs and appendix C gives listings, flow charts, and descriptions of the programs.

Two separate programs were necessary for two reasons: (1) FORMAC is an interpretive language that requires relatively large computer execution times for repetitive numerical evaluations of algebriac expressions and (2) computer storage must be saved because the FORMAC program requires large amounts of storage to form the algebriac expressions. These two considerations made the separate programs necessary even though the user was given the inconvenience of making two computer runs for the complete solution.

The solution obtained from the FORMAC program is communicated to the FORTRAN program by equations punched on cards. These equations are punched by the FORMAC program in a form that can be compiled by the FORTRAN IV compiler. The cards are simply inserted in the proper place in a FORTRAN subroutine.

FORMAC Program (REDUCE)

Basically, the FORMAC program consists of four major steps: (1) reduction of the system of equations (of the block diagram) to one equation containing the two variables of the desired transfer function, (2) solution of this equation for the transfer function in terms of the G-functions, (3) substitution of the system information for the G-functions, and (4) expansion of this equation and collection of terms of the real and imaginary parts.

The reduction proceeds by choosing a variable to eliminate, solving for that variable in one of the equations, and substituting the solution for that variable in the remaining equations. Prior to the actual reduction, the inputs to the block diagram, supplied by the user, are replaced by zero. If an input occurs in the desired transfer function, the input is not made zero. With this method, the number of equations and variables is reduced by one at each step. The variables that appear in the transfer function are, of course, never eliminated. The final equation contains only the variables of the transfer function and the G-functions.

TABLE I. - STEP USED BY FORMAC PROGRAM TO SOLVE HEAD RESPONSE BLOCK DIAGRAM FOR TRANSFER FUNCTION X_1/X_2

Equation	Step				
	0	1	2	3	4
1	$-X_1 + X_2 - X_6 = 0$				$X_6 = -X_1 + X_2$
2	$G_1X_2 - X_3 = 0$			$G_1G_3X_2 - X_1 + G_2G_3X_6 = 0$	$G_1G_3X_2 - X_1 - G_2G_3X_1 + G_2G_3X_2 = 0$
3	$G_2X_6 - X_4 = 0$	$X_4 = G_2X_6$			
4	$X_3 + X_4 - X_5 = 0$	$X_3 + G_2X_6 - X_5 = 0$	$G_3X_3 + G_2G_3X_6 - X_1 = 0$	$X_3 = \frac{X_3 - G_2G_3X_6}{G_3}$	
5	$G_3X_5 - X_1 = 0$		$X_5 = \frac{X_1}{G_3}$		

This reduction method is depicted, using the example of appendix B, in table I. Here, the transfer function desired is X_1/X_2. The equations to be solved are listed at step 0. As the reduction proceeds through each step, the number of variables and equations is reduced by one until only one equation remains. One can see that equation (2), as it appears in step 4, is indeed the desired solution.

This method will always give a correct result. However, the answer may be of a form that is overly complicated; that is, the numerator and denominator may contain like factors. Because these like factors are difficult to detect in FORMAC, it was necessary to choose the variables to eliminate and the equations used to solve for these variables in some specific order. The order desired is the one that yielded transfer functions which had no common factor in the numerator and denominator. In other words, the minimum form of the solution is desired. The method for determining this order is described in appendix D.

The FORMAC program has certain limitations because of programming considerations and computer memory size. There is an upper limit to the number of equations, the number of variables, and the number of G-functions that it can handle. These limits are 30, 63, and 20, respectively. There are also limitations on the number and type of functions and variables that may be used to describe the system information. All limitations are described in the user's manual (appendix B). The limitations are due partly to the method employed to order the equations for solution and partly to conserve memory locations. It is felt that the limits are large enough so that the user is not unduly restricted in the size of problem that may be solved. However, many of the limits may be increased by changing dimension statements in the program. Care must be taken in changing the dimensions so that too much storage space is not removed from the space needed to store the FORMAC expressions.

Because FORMAC stores expressions in the computer memory locations not used by the programs, it is desirable to keep the unused storage as large as possible. Subroutines referenced by FORMAC programs but not executed in this program are not loaded at execution time. This is accomplished by the use of a dummy subroutine with entry point names the same as those subroutines that would be loaded but not executed.

In order to expand the transfer function into real and imaginary parts, it was necessary to represent the allowed system functions symbolically as complex numbers. This is accomplished in the program by prefixing the function names with R and I, where the names prefixed with R stand for the real part of the complex number and I, the imaginary part. The term S is considered as a pure imaginary number $i\omega$.

The expansion of the transfer function into real and imaginary parts may not always be accomplished by the program. The limiting factor here is memory size. Because of this fact, provision is made in the FORTRAN program to accept transfer functions in either the S form or the complex rational form.

The FORMAC program uses the following function subprograms that are not supplied with either FORMAC or FORTRAN. These subprograms are not given in this report and must be supplied by the user.

Function	Description
ALS(N,X)	Accumulator left shift of X, N binary places
IARS(N,X)	Accumulator right shift of X, N binary places
LGR(N,X)	Logical left shift of X, N binary places
AND(X1,X2)	Logical intersection of X1 and X2
OR(X1,X2)	Logical union of X1 and X2

FORMAC Input/Output

The input to the FORMAC program consists of two parts. The first is the transfer function desired and the system of equations from the block diagram. The second part is the system information expressions. After the equations are read, they are reduced to the G-form and output printed and punched. The G-form is then transformed into the N/D-form and output. A logical variable is read to determine whether the user wants system information substituted for the N/D's. If substitution is desired, this information is read and the substitution and the output of the S-form takes place. Another logical variable is read and tested before expansion to the complex rational form is performed. A more detailed description of the input and output is given in the users manual (appendix B).

For the equations punched by the FORMAC program to be acceptable to the FORTRAN IV compiler, the floating point powers and the trailing $ had to be removed. This is done by the program just prior to the punching of equations.

The execution time for the FORMAC program is difficult to predict. However, execution times cited in the next section are typical. All execution times mentioned in this report are for the IBM 7094II-7044 direct-couple system.

FORTRAN Program (EVAL)

The FORTRAN program can accept punched output from the FORMAC program in one of two forms: either the unexpanded (or S-form), or the expanded (or complex rational form). The FORTRAN program is designed to handle all the different types of system information acceptable to the FORMAC program except the arbitrary function.

To be able to handle the arbitrary function, the user must supply a complex function subprogram F(I) that evaluates the desired function. The subprogram that defines the other functions (appendix C) may be used as an example of how this is to be done.

FORTRAN Input/Output

The equations punched by the FORMAC program are inserted into one of two subroutines, depending on the form of the equation. This subroutine must be compiled and executed with the main program and the function evaluation subprogram. The numerical input is read by the main program, and the subroutine containing the FORMAC expressions is executed. The numerical input is determined by the type of functions used in the system information expressions. A more detailed discussion of this may be found in appendix B.

Output from the FORTRAN program consists of all numerical input and the following data from the transfer function at each frequency specified by the user:

(1) Values of the real and imaginary parts of the numerator and denominator normalized such that either the real or imaginary part of the numerator is one. The normalization is necessary because some terms may involve large values of frequency raised to powers.

(2) The real and imaginary parts

(3) The complex absolute value (magnitude)

(4) The complex argument (phase angle)

The execution time of the FORTRAN program depends on the complexity of the transfer function and is therefore difficult to predict. The execution time, however, is somewhat larger for the S-form than the complex rational form.

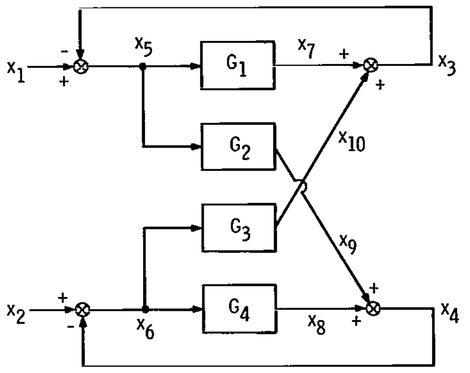
APPLICATIONS

Several applications have been chosen to demonstrate various aspects and potentialities of the block-diagram reducing programs. The first example is that of an interacting-control system. It is a lumped parameter example in which the forms of the solutions are of some interest. A second example, representing a lumped mass, spring-damper system, is presented. Here, the symbolic solution is of particular interest. Finally, a distributed parameter is presented in which the numerical results are of primary interest.

Application 1: Interacting-Control System

A block diagram of an interacting-control system is presented in figure 4(a). Although the values for the G's are not specified here, the block diagram might be representative of the interacting-control problem for jet engines (e.g., attempting to control engine speed and turbine-outlet temperature simultaneously). The algebriac information required, the program, and the solutions desired are indicated in figure 4(b). Because the G's are not specified, the solutions cannot give system information, but it does give the topological information associated with the form of the diagram.

Because there are eight components in the block diagram (summation points and blocks) there are eight equations required to define the system. (Nodes are not considered components because the equation $X_a = X_a$ is not needed by the program.)



(a) Block diagram.

$x_1 - x_3 - x_5 = 0$ $\qquad$ $x_7 - G_1x_5 = 0$

$x_2 - x_4 - x_6 = 0$ $\qquad$ $x_9 - G_2x_5 = 0$

$x_7 + x_{10} - x_3 = 0$ $\qquad$ $x_{10} - G_3x_6 = 0$

$x_8 + x_9 - x_4 = 0$ $\qquad$ $x_8 - G_4x_6 = 0$

Block-diagram inputs: x_1 and x_2

Solutions desired: $\frac{x_3}{x_1}, \frac{x_3}{x_2}, \frac{x_4}{x_1}, \frac{x_4}{x_2}$

(b) Algebraic information required.

Figure 4. - Interacting control system.

The solution for the transfer functions are as follows:

$$\frac{X_4}{X_1} = \frac{G_2}{G_1 + G_1G_4 - G_2G_3 + G_4 + 1}$$

$$\frac{X_3}{X_1} = \frac{G_1 + G_1G_4 - G_2G_3}{G_1 + G_1G_4 - G_2G_3 + G_4 + 1}$$

$$\frac{X_4}{X_2} = \frac{G_1G_4 - G_2G_3 + G_4}{G_1 + G_1G_4 - G_2G_3 + G_4 + 1}$$

$$\frac{X_3}{X_2} = \frac{G_3}{G_1 + G_1G_4 - G_2G_3 + G_4 + 1}$$

These are the results as generated by the computer. Should the user be interested in the transfer function X_4/X_3, it follows that, for X_1 as the reference input,

$$\frac{X_4}{X_3} = \frac{G_2}{G_1 + G_1G_4 - G_2G_3}$$

and, for X_2 as the reference input,

$$\frac{X_4}{X_3} = \frac{G_1G_4 - G_2G_3 + G_4}{G_3}$$

clearly not the same result. For this reason, the user should form the transfer functions as indicated herein so that the reference input will be known.

The total computer execution time required for these four solutions was 0.30 minute. It will be noted also that the solutions are in minimum form. Also, the equations are not put into the program in any particular order, but the order in which they are used is determined in the program.

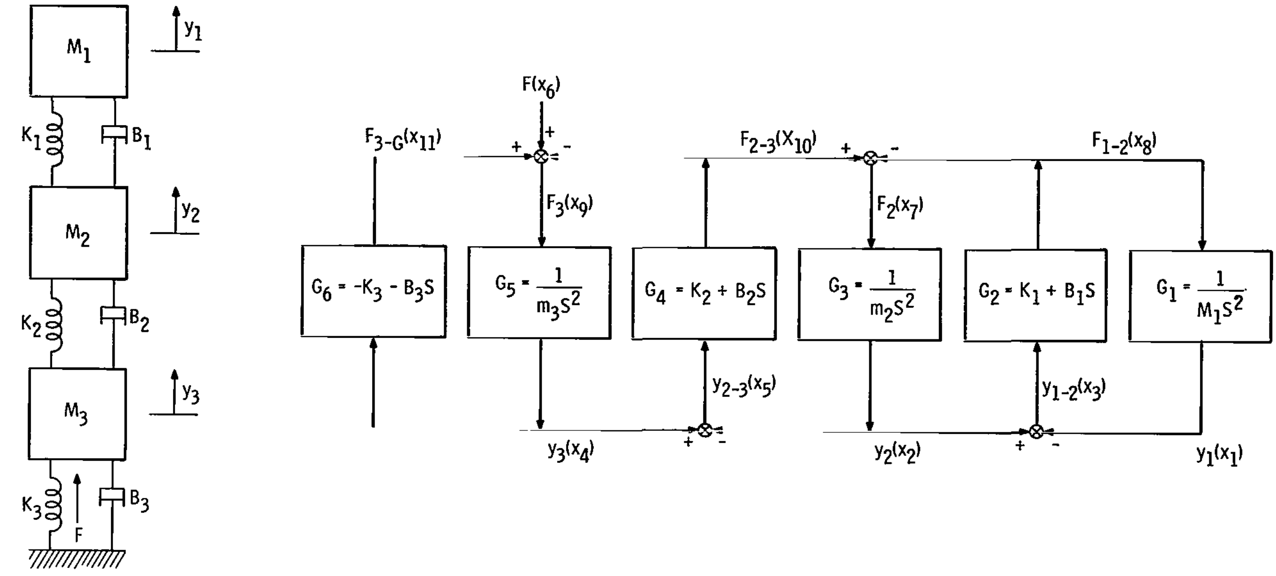
Application 2: Mass-Spring-Damper Dynamics

This particular application was taken from a study in which the symbolic solution was of primary interest for use in further analysis. (See ref. 9.) The object of that research was to relate the individual dampers of n-mass systems (n = 3, fig. 5(a)) to a modal equivalent damper. To accomplish this required the symbolic solution in the complex rational form.

The block diagram for the system is presented in figure 5(b). The values for the G's are given in the blocks and variable names used in the program are parenthetically noted after each variable.

For this problem, the following symbols apply:

B	damper constant, (sec)(lb)/ft; (sec)(N)/m
F	force, lb; N
K	spring constant, lb/ft; N/m
M	mass, $(\sec^2)$(lb)/ft; $(\sec^2)$(N)/m
y	displacement, ft; m



(a) Physical system.

(b) Block diagram.

$G_1x_8 - x_1 = 0$

$G_2x_3 - x_8 = 0$

$x_2 - x_1 - x_3 = 0$

$G_3x_7 - x_2 = 0$

$x_{10} - x_8 - x_7 = 0$

$G_4x_5 - x_{10} = 0$

$x_4 - x_2 - x_5 = 0$

$G_5x_9 - x_4 = 0$

$x_{11} + x_6 - x_{10} - x_9 = 0$

$G_6x_4 - x_{11} = 0$

Block diagram input: x_6

Solution required: $\frac{x_1}{x_6}$

(c) Algebraic information required.

Figure 5. - Three-mass spring-damper.

Specifically required in the study were the coefficients of the various powers of ω in the characteristic equation (denominator of the complex rational form).

The input equations required by the program are shown in figure 5(c). This is a 10-component (10 equations) system and the solution required is for $X_1/F = (X_1)/(X_6)$. The solutions as generated by the program are as follows:

The G-form of the transfer functions.

$$X_1/X_6 = G_1G_2G_3G_4G_5 \Big/ (G_1G_2 + G_1G_2G_3G_4 - G_1G_2G_3G_4G_5G_6 + G_1G_2G_4G_5 - G_1G_2G_5G_6$$

$$+ G_2G_3 + G_2G_3G_4G_5 - G_2G_3G_5G_6 + G_3G_4 - G_3G_4G_5G_6 + G_4G_5 - G_5G_6 + 1.0)$$

After program substitution of the G equations:

$$
\begin{aligned}
X_1/X_6 = (K_1 + SZ_1)(K_2 + SZ_2)\Big/\Big[&-(K_1 + SZ_1)(K_2 + SZ_2)(-K_3 - SZ_3) + (K_1 + SZ_1) \\
&\times (K_2 + SZ_2)S^2T_1 + (K_1 + SZ_1)(K_2 + SZ_2)S^2T_2 + (K_1 + SZ_1)(K_2 + SZ_2)S^2T_3 \\
&- (K_1 + SZ_1)(-K_3 - SZ_3)S^2T_1 - (K_1 + SZ_1)(-K_3 - SZ_3)S^2T_2 + (K_1 + SZ_1)S^4 \\
&\times T_1T_3 + (K_1 + SZ_1)S^4T_2T_3 - (K_2 + SZ_2)(-K_3 - SZ_3)S^2T_1 + (K_2 + SZ_2)S^4 \\
&\times T_1T_2 + (K_2 + SZ_2)S^4T_1T_3 - (-K_3 - SZ_3)S^4T_1T_2 + S^6T_1T_2T_3\Big]
\end{aligned}
$$

After program substitution of S = iω and expansion to the complex rational form:

$$
\begin{aligned}
X_1/X_6 = K_1K_2 - \omega^2Z_1Z_2 + i(K_1Z_2 + K_2Z_1)\omega\Big/\Big\{&K_1K_2K_3 - T_1T_2T_3\omega^6 + (K_1T_1T_3 + K_1T_2T_3 \\
&+ K_2T_1T_2 + K_2T_1T_3 + K_3T_1T_2 + T_1Z_1Z_2 + T_1Z_1Z_3 + T_1Z_2Z_3 + T_2Z_1Z_2 \\
&+ T_2Z_1Z_3 + T_3Z_1Z_2)\omega^4 + (-K_1K_2T_1 - K_1K_2T_2 - K_1K_2T_3 - K_1K_3T_1 - K_1K_3T_2 \\
&- K_1Z_2Z_3 - K_2K_3T_1 - K_2Z_1Z_3 - K_3Z_1Z_2)\omega^2 + i\Big[(T_1T_2Z_2 + T_1T_2Z_3 + T_1T_3Z_1 \\
&+ T_1T_3Z_2 + T_2T_3Z_1)\omega^5 + (-K_1T_1Z_2 - K_1T_1Z_3 - K_1T_2Z_2 - K_1T_2Z_3 - K_1T_3Z_2 \\
&- K_2T_1Z_1 - K_2T_1Z_3 - K_2T_2Z_1 - K_2T_3Z_1 - K_3T_1Z_1 - K_3T_1Z_2 - K_3T_2Z_1 \\
&- Z_1Z_2Z_3)\omega^3 + (K_1K_2Z_3 + K_1K_3Z_2 + K_2K_3Z_1)\omega\Big]\Big\}
\end{aligned}
$$

The actual program output form corresponding to this equation is shown in table II.

The coefficients generated here, with $T_n = M_n$ and $B_n = Z_n$, compare exactly with those generated manually in reference 9. Thus, the tedious hours required to the various cases presented in that study can now be avoided by use of this program.

The symbolic form of the solution is fairly complex; however, parametric studies using the solution are now easily made. In this case, for example, numerical values for the B's and the K's can be substituted in the complex rational form equation. This substitution would leave an expression involving only the Z's (or damping terms). From

TABLE II. - PROGRAM OUTPUT FORM

```
THE NUMERATOR
      (K(1)+S*Z(1))*(K(2)+S*Z(2))

THE DENOMINATOR
      -(K(1)+S*Z(1))*(K(2)+S*Z(2))*(-K(3)-S*Z(3))+(K(1)+S*Z(1))*(K(2)+S*
      Z(2))*S**2  *T(1)+(K(1)+S*Z(1))*(K(2)+S*Z(2))*S**2  *T(2)+(K(1)+S*
      Z(1))*(K(2)+S*Z(2))*S**2  *T(3)-(K(1)+S*Z(1))*(-K(3)-S*Z(3))*S**
      2  *T(1)-(K(1)+S*Z(1))*(-K(3)-S*Z(3))*S**2  *T(2)+(K(1)+S*Z(1))*S
      **4  *T(1)*T(3)+(K(1)+S*Z(1))*S**4  *T(2)*T(3)-(K(2)+S*Z(2))*(-K(3
      )-S*Z(3))*S**2  *T(1)+(K(2)+S*Z(2))*S**4  *T(1)*T(2)+(K(2)+S*Z(2))
      *S**4  *T(1)*T(3)-(-K(3)-S*Z(3))*S**4  *T(1)*T(2)+S**6  *T(1)*T(2'
      *T(3)

THE NUMERATOR (REAL)

      K(1)*K(2)

      -W**2  *Z(1)*Z(2)

THE NUMERATOR (IMAGINARY)

      (K(1)*Z(2)+K(2)*Z(1))*W

THE DENOMINATOR (REAL)

      K(1)*K(2)*K(3)

      -T(1)*T(2)*T(3)*W**6

      (K(1)*T(1)*T(3)+K(1)*T(2)*T(3)+K(2)*T(1)*T(2)+K(2)*T(1)*T(3)+K(3)*
      T(1)*T(2)+T(1)*Z(1)*Z(2)+T(1)*Z(1)*Z(3)+T(1)*Z(2)*Z(3)+T(2)*Z(1)*Z
      (2)+T(2)*Z(1)*Z(3)+T(3)*Z(1)*Z(2))*W**4

      (-K(1)*K(2)*T(1)-K(1)*K(2)*T(2)-K(1)*K(2)*T(3)-K(1)*K(3)*T(1)-K(1)
      *K(3)*T(2)-K(1)*Z(2)*Z(3)-K(2)*K(3)*T(1)-K(2)*Z(1)*Z(3)-K(3)*Z(1)*
      Z(2))*W**2

THE DENOMINATOR (IMAGINARY)

      (T(1)*T(2)*Z(2)+T(1)*T(2)*Z(3)+T(1)*T(3)*Z(1)+T(1)*T(3)*Z(2)+T(2)*
      T(3)*Z(1))*W**5

      (-K(1)*T(1)*Z(2)-K(1)*T(1)*Z(3)-K(1)*T(2)*Z(2)-K(1)*T(2)*Z(3)-K(1)
      *T(3)*Z(2)-K(2)*T(1)*Z(1)-K(2)*T(1)*Z(3)-K(2)*T(2)*Z(1)-K(2)*T(3)*
      Z(1)-K(3)*T(1)*Z(1)-K(3)*T(1)*Z(2)-K(3)*T(2)*Z(1)-Z(1)*Z(2)*Z(3))*
      W**3

      (K(1)*K(2)*Z(3)+K(1)*K(3)*Z(2)+K(2)*K(3)*Z(1))*W
```

this expression, the effect of the individual dampers on the response of interest could be assessed.

The computer time required to execute this study was 0.39 minute, the user preparation (for the program) was minimal because it merely required listing the equations (the G-expressions) and transfer functions sought and placement of appropriate punched cards in the program.

Also, with the numerical evaluation program, frequency response for cases of special interest (M's, K's, and B's) are quickly generated.

Application 3: Hydraulic-Line-Dynamics Study

This study is an example of a distributed parametric study. For this study, only the numerical results were of interest. For the program, of course, it is necessary to generate the symbolic answers at least to the S-form to obtain the numerical answers.

The following symbol list applies to this application:

C_n line capacitance per unit length, in.; m

C_m compliance, in.^{2}; m^2

L_n line inductance per unit length, sec^2/in.^{3}; sec^2/m^3

l line length, in.; m

m index number for termination impedance

n index number for characteristic impedance

P pressure, lb/in.^{2}; N/m^2

R_n line resistance per unit length, sec/in.^{3}; sec/m^3

R_m resistance, sec/in.^{2}; sec/m^2

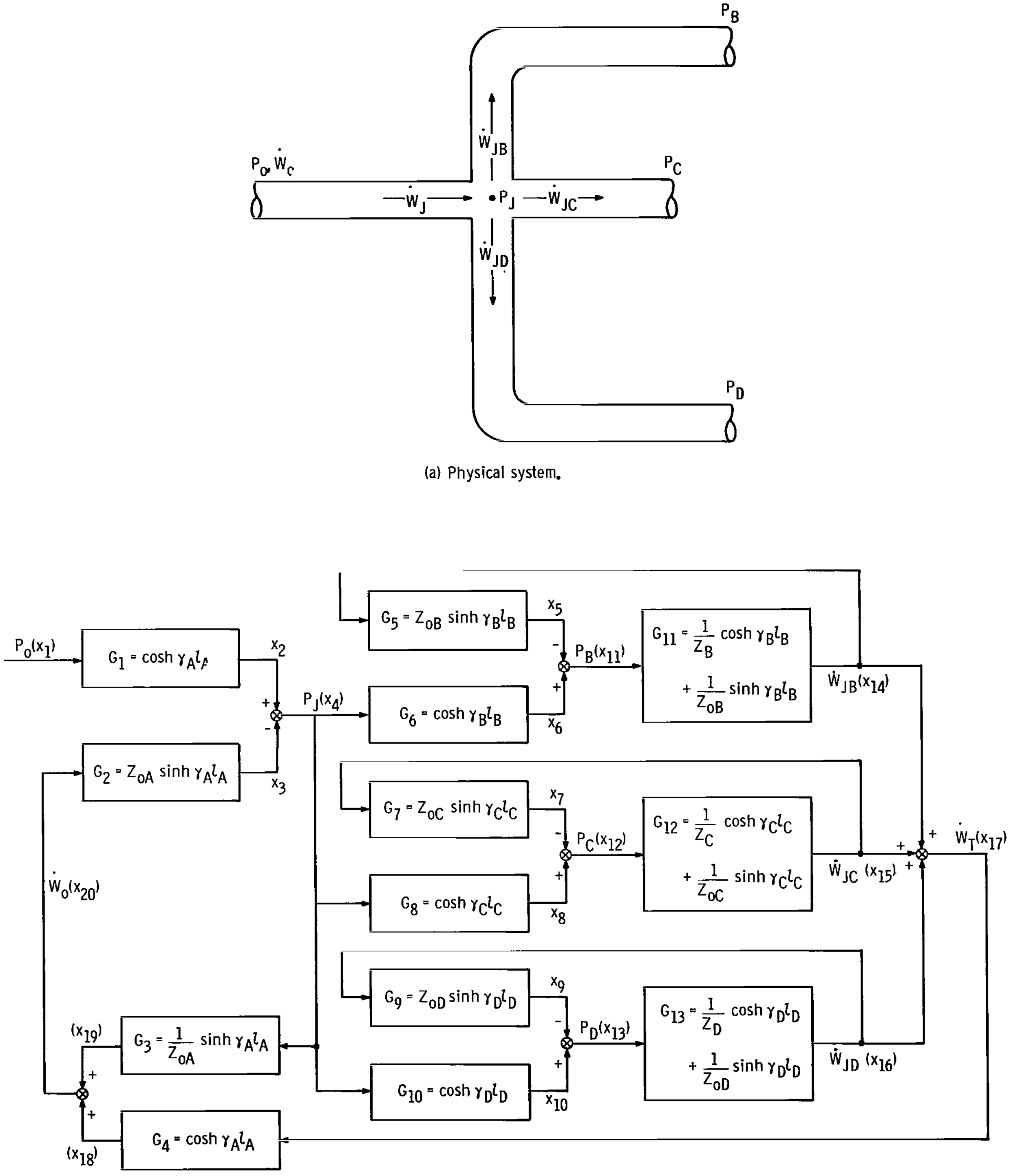
$\dot{W}$ weight flow, lb/sec; N/sec

Z_{on} characteristic impedance equal to $1/S\sqrt{L_n/C_n\left[S^2 + (R_n/C_n)S\right]}$, sec/in.^{2}; sec/m^2

Z_m termination impedance equal to $R_m/(R_m C_m S + 1)$, sec/in.^{2}; sec/m^2

γ_n propagation constant equal to $\sqrt{L_n C_n S^2 + R_n C_n S}$, 1/in.; 1/m

The physical system is shown schematically in figure 6(a). The system is composed of a main feed line OJ feeding three subsidiary ducts, JB, JC, and JD. The dynamic problem is to determine the response of the end pressures in the subsidiary ducts in re-



P_B
$\dot{W}_{JB}$
$P_o, \dot{W}_c$
P_C
$\dot{W}_J$
P_J
$\dot{W}_{JC}$
$\dot{W}_{JD}$
P_D
$P_o(x_1)$
$G_1 = \cosh \gamma_A l_A$
x_2
$G_2 = Z_{0A} \sinh \gamma_A l_A$
x_3
$P_J(x_4)$
$G_5 = Z_{0B} \sinh \gamma_B l_B$
x_5
$P_B(x_{11})$
$G_6 = \cosh \gamma_B l_B$
x_6
$G_{11} = \frac{1}{Z_B} \cosh \gamma_B l_B + \frac{1}{Z_{0B}} \sinh \gamma_B l_B$
$\dot{W}_{JB}(x_{14})$
$G_7 = Z_{0C} \sinh \gamma_C l_C$
x_7
$P_C(x_{12})$
$G_8 = \cosh \gamma_C l_C$
x_8
$G_{12} = \frac{1}{Z_C} \cosh \gamma_C l_C + \frac{1}{Z_{0C}} \sinh \gamma_C l_C$
$\dot{W}_{JC}(x_{15})$
$\dot{W}_T(x_{17})$
$\dot{W}_o(x_{20})$
$G_9 = Z_{0D} \sinh \gamma_D l_D$
x_9
$P_D(x_{13})$
$G_{10} = \cosh \gamma_D l_D$
x_{10}
$G_{13} = \frac{1}{Z_D} \cosh \gamma_D l_D + \frac{1}{Z_{0D}} \sinh \gamma_D l_D$
$\dot{W}_{JD}(x_{16})$
(x_{19})
$G_3 = \frac{1}{Z_{0A}} \sinh \gamma_A l_A$
$G_4 = \cosh \gamma_A l_A$
(x_{18})
+
-

(a) Physical system.

(b) Block diagram.

Figure 6. - Line dynamic response.

$x_2 - G_1x_1 = 0$

$x_3 - G_2x_{20} = 0$

$x_{19} - G_3x_4 = 0$

$x_{18} - G_4x_{17} = 0$

$x_5 - G_5x_{14} = 0$

$x_6 - G_6x_4 = 0$

$x_7 - G_7x_{15} = 0$

$x_8 - G_8x_4 = 0$

$x_9 - G_9x_{16} = 0$

$x_{10} - G_{10}x_4 = 0$

$x_{14} - G_{11}x_{11} = 0$

$x_{15} - G_{12}x_{12} = 0$

$x_{16} - G_{13}x_{13} = 0$

$x_2 - x_3 - x_4 = 0$

$x_{19} + x_{18} - x_{20} = 0$

$x_6 - x_5 - x_{11} = 0$

$x_8 - x_7 - x_{12} = 0$

$x_{10} - x_9 - x_{13} = 0$

$x_{14} + x_{15} + x_{16} - x_{17} = 0$

Block diagram input: x_1

Solution required: $\frac{P_c}{P_o} = \frac{x_{12}}{x_1}$

Auxiliary information:

$$\gamma_n = \sqrt{L_nC_nS^2 + R_nC_nS}$$

$$Z_{on} = \frac{1}{S}\sqrt{\frac{L_n}{C_n}\left(S^2 + \frac{R_n}{L_n}S\right)}$$

$$Z_m = \frac{R_m}{R_mC_mS + 1}$$

(c) Algebraic information required.

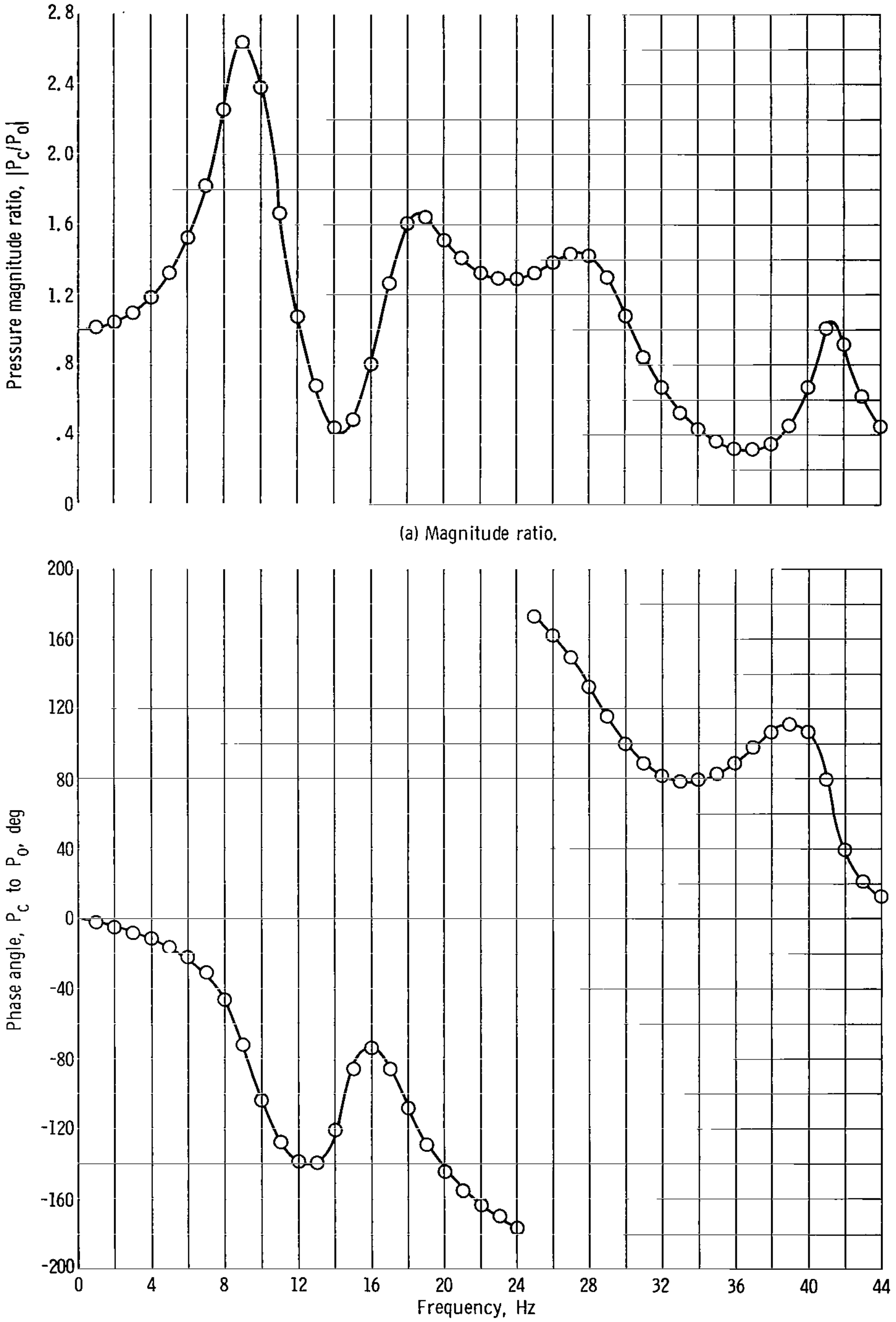
Line	Length		Line inductance per unit length, L_n		Line capacitance per unit length, C_n		Line resistance per unit length, R_n		Resistance, R_m		Resistance times compliance R_mC_m, sec
	in.	cm	sec^2/in.^{3}	sec^2/m^3	in.	m	sec/in.^{3}	sec/m^3	sec/in.^{2}	sec/m^2	
A	197	5.0	2.73x10^{-5}	1.66	12.38x10^{-5}	31.44x10^{-7}	0	0	-----	----------	--------
B	72	1.83	13.14	8.018	.94	2.388	↓	↓	8.48	13x144x10^3	0.0697
C	75	1.90	6.71	4.094	1.78	4.521	↓	↓	2.61	4.045	.0212
D	32	.813	6.71	4.094	1.78	4.521	↓	↓	2.61	4.045	.0212

(d) Numerical information required.

Figure 6. - Concluded.

sponse to pressure disturbance at the far end of the feeder duct, for example, to determine the frequency response of P_c/P_o.

The system block diagram is shown in figure 6(b). The contents of the blocks, because each line is a distributed system, are hyperbolic functions of square-root arguments. The form of these G's represents four-terminal network solutions for the wave equation for each section of line. The hyperbolic functions are multiplied by impedances representing the line terminations. These impedance forms together with the equations for the block diagram are given in figure 6.



(a) Magnitude ratio.

(b) Phase angle (principle valve).

Figure 7. - Numerical solution for line dynamic response.

Using the above information, a computer solution for the S-form transfer function was generated in 1.10 minutes of computing time. (The solution requires two pages of printed output for the symbolic answers which are not presented herein.) A deck of punched cards is also output which contains the symbolic solution.

For the numerical values of the line characteristics presented in figure 6(d), the frequency response was numerically evaluated for a range of frequencies of 0 to 50 hertz in 1-hertz steps. The amplitude ratio and phase-angle results are plotted in figure 7. The execution time required to generate these data was 0.50 minute.

Other Applications

The applications stemming from having a transfer function readily available in symbolic form are many. For example, if one is dealing strictly in lumped parameter systems, a program could be evolved which would evaluate system stability quickly for a large number of numerical cases by application of Routh's criterion to the characteristic equation (which are now available in symbolic form directly from the program).

System stability can also be determined using the slope of the amplitude ratio plot as it goes through the zero dB line, or by applying Nyquist's criterion to the real and imaginary parts.

Nonlinearities can be introduced into the general analysis by use of describing function and small perturbation techniques, both of which give linear approximations for the block transfer functions. (See ref. 10.)

CONCLUDING REMARKS

Techniques were established and a program was written to reduce arbitrary block diagrams to obtain symbolic expressions for any transfer function with respect to an input. Further evaluation programs are presented which allow these symbolic solutions to be evaluated for a desired set of frequencies and numerical values of system constants. These evaluations yield magnitude ratio, phase angle, and real and imaginary parts of the transfer function.

The advantages of the present study over previous techniques are as follows: (1) the results are given in symbolic form (3 forms: G, S, and complex rational) and (2) arrangement of the input data in the form of arrays and the matrix manipulation for each frequency point are not required. (The symbolic solution need be generated only once and then reevaluated for each frequency.)

The most serious limitation of the work is the computer storage required. The FORMAC routine itself requires a large amount of storage, and, as the systems size increases, the transfer function equations grow very large and, hence, the storage capabilities are affected. Specifically, as written, the program can handle a system of 30 equations, containing as many as 63 variables.

The computing time required is modest even for large systems. Also, the user preparation time is minimal.

The program is capable of handling block diagrams for any linear constant coefficient system. The system can be either a lumped parameter or a distributed parameter system so long as an expression for the real and imaginary parts is known. Furthermore, the ability to handle distributed parameter block in rectangular coordinate form is built into the program.

Finally, a criterion is presented which allows ordering of the equations and variables to be eliminated so that a minimum form solution can be achieved. This technique has given minimum form solutions for many trial cases.

The technique involved for the algebraic reduction is simple and handles the problem of looping.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, February 26, 1968,
180-31-01-01-22.

APPENDIX A

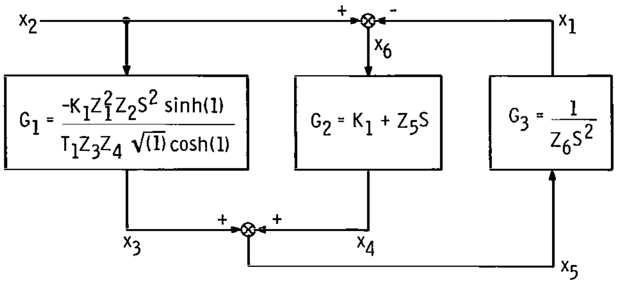
SYMBOL LIST

A, a B, b C, c D, d	constants for complex rational form and for program functional form
D	denominator of G
G	transfer function for a block in a block diagram
i	$\sqrt{-1}$
K	a system constant that can be used in REDUCE
N	numerator of G
S	Laplace variable
T	a system constant that can be used in REDUCE
u	real part of a complex quantity
v	imaginary part of a complex quantity
WN	a system constant that can be used in REDUCE
X	signal or variable in a block diagram
x	real part of a complex quantity
y	imaginary part of a complex quantity
Z	a system constant that can be used in REDUCE
z	a complex quantity
ω	circular frequency, rad/sec
Subscripts:	
m	index number 1, 2, . . .
n	index number 1, 2, . . .

APPENDIX B

USER'S MANUAL

This appendix describes the use of both the FORMAC and FORTRAN programs presented in this report. The description of the FORMAC program use is given first and the FORTRAN program second. The example used in this appendix is taken from the block diagram and system information given in figure 8.


x_2
+
-
x_1
x_6
$G_1 = \frac{-K_1 Z_1^2 Z_2 S^2 \sinh(1)}{T_1 Z_3 Z_4 \sqrt{(1)} \cosh(1)}$
$G_2 = K_1 + Z_5 S$
$G_3 = \frac{1}{Z_6 S^2}$
+
+
x_3
x_4
x_5

(a) Block diagram.

$x_3 - G_1 x_2 = 0$

$x_4 - G_2 x_6 = 0$

$x_1 - G_3 x_5 = 0$

$\sinh(1) \rightarrow \sinh\left(d\sqrt{as^2 + bs}\right)$

$\cosh(1) \rightarrow \cosh\left(d\sqrt{as^2 + bs}\right)$

$\sqrt{(1)} \rightarrow \sqrt{as^2 + bs}$

$x_2 - x_1 - x_6 = 0$

$x_3 + x_4 - x_5 = 0$

Solution required: x_1/x_2

(b) Algebraic information required.

$K_1 = 763\,359.38$

$Z_1 = 0.75$

$Z_2 = 62.4$

$Z_3 = 6.480 \times 10^7$

$Z_4 = 32.2$

$Z_5 = 10.0$

$Z_6 = 1.0$

$T_1 = 0.010416$

$a = 0.4100625 \times 10^{-5}$

$b = 0.0001$

$d = 3.33333$

(c) Numerical data required.

Figure 8. - User's manual sample problem.

INPUT TO REDUCE

Several points must be considered while preparing input to the FORMAC program.

(1) All algebriac expressions must end with a $.

(2) Blanks are ignored in the expressions and therefore may be used freely.

(3) All 80 card columns of a card may be used for an expression.

(4) An expression may take up to and including five cards.

(5) The expressions of the block diagram may only be of the form

$$\pm X(a) \pm X(b) \pm X(c) \ . \ . \ . \ \$$$

or

$$\pm X(a) \pm X(b) * G(c)\$$$

or linear combinations, as

$$\pm X(a) \pm X(b) \pm \ . \ . \ . \ \pm X(m) * G(j) \pm X(n) * G(k) \ \pm \ . \ . \ . \ \$$$

where a, b, c, etc., are integer numbers. The number of terms in the first form must be greater than one. The =0 that makes the preceding expressions equations is implied and is not to be given. The total number of terms in all the block-diagram expressions may not be greater than 100.

(6) The integer subscripts used in the X terms must start at one and increase sequentially and must not be greater than 63.

(7) The integer subscripts used in the G terms must start at one and increase sequentially. Also, the largest subscript must not be greater than 20.

(8) The numerator and denominator of the system information that is to replace the N's and D's are input as separate expressions. These expressions are read sequentially; that is, the numerator for G(1), N(1), is read first, the denominator, D(1), is read second, the numerator for G(2) is read third, and so on.

(9) The expressions of the system information may contain any of the following items:

Item	Description
I	$\sqrt{-1}$, i
W	omega, ω
S	$i\omega$

T(a), Z(a), K(a), WN(a)	variables
EXPF(b)	complex exponential
SINF(b)	complex sine
COSF(b)	complex cosine
SINHF(b)	complex hyperbolic sine
COSHF(b)	complex hyperbolic cosine
SQRTF(b)	complex square root
F(b)	arbitrary function
numbers	any number in FORTRAN notation, such as 1.0, 6.23E+2

The subscripts a and b stand for integer numbers. These subscripts must begin at one and increase sequentially for each function and variable used. The maximum subscript allowed is a = 10 (for variables) and b = 5 (for functions).

Keeping the above points in mind, the input must be in the following order:

(1) The first input card contains three integer numbers, (a) the number of variables (number of x's), (b) the number of equations, and (c) the number of block-diagram inputs. These numbers are punched on the card such that the last digit of the first number appears in card column 5, the second in card column 10, and the last in card column 15.

(2) The second card contains X(a)$,X(b)$,X(c)$, . . ., where a,b,c, . . . are integers that refer to the block-diagram inputs.

(3) The third card contains X(a)$ and X(b)$, where a is the integer referring to the output variable of the transfer function desired and the integer b refers to the input variable of the transfer function. The ratio X(a)/X(b) is formed.

(4) The block diagram expressions are punched on the next cards. One card may not contain more than one expression.

(5) The word TRUE or FALSE is punched on the next card beginning in card column two. The word TRUE indicates that system information expressions are to follow. The word FALSE indicates that no system information follows, and items (6) and (7) are to be omitted from the input.

(6) The system information expressions are punched on the next cards. One card may contain only one expression.

(7) The word TRUE or FALSE is punched on the next card beginning in card column two. The word TRUE indicates that the program is to expand the transfer function into real and imaginary parts. The word FALSE indicates that expansion is not desired.

Consecutive problems may be run by simply repeating the above steps.

The following example of input to the FORMAC program is taken from figure 8(a) and (b):

```
$DATA
    6     5     1
 X(2)$
 X(1)$ X(2)$
 X(2)-X(1)-X(6)$
 X(2)*G(1)-X(3)$
 X(6)*G(2)-X(4)$
 X(3)+X(4)-X(5)$
 X(5)*G(3)-X(1)$
 TRUE
 -K(1)*Z(1)**2*Z(2)*S**2*SINHF(1)$
 T(1)*Z(3)*Z(4)*SQRTF(1)*COSHF(1)$
 K(1)+Z(5)*S$
 1$
 1$
 Z(6)*S**2$
 TRUE
```

REDUCE OUTPUT

Twelve errors are detected by the FORMAC program. Any of these errors will cause an error message to be printed and the termination of execution. Two warning messages may also be printed. The warning does not terminate execution. A list of the messages and their causes follow:

Messages (errors)	Cause
(1) MORE THAN 63 VARIABLES	The number of variables exceeded the maximum allowed.
(2) MORE THAN 30 EQUATIONS	The number of equations exceeded the maximum allowed.
(3) MORE THAN 5 BLOCK DIAGRAM INPUTS	The number of block diagram inputs exceeded the maximum allowed.
(4) MORE THAN 5 CARDS FOR A BLOCK DIAGRAM EXPRESSION	Block diagram expressions may not use more than 5 cards.
(5) MORE THAN 5 CARDS FOR A SYSTEM INFORMATION EXPRESSION	System information expressions may not use more than 5 cards.

Messages (errors)	Cause
(6) THE TOTAL NUMBER OF TERMS EXCEEDS 100	The number of terms in all block diagram expressions exceeded the maximum allowed.
(7) THE OUTPUT VARIABLE COULD NOT BE FOUND	The output variable specified in the transfer function was not in the block diagram expressions. Check input equations.
(8) THE INPUT VARIABLE COULD NOT BE FOUND	The input variable specified in the transfer function was not in the block diagram expressions. Check input equations.
(9) SYSTEM TOO LARGE	The internal array that contains the order in which the reduction is to proceed was filled.
(10) THE NUMBER OF TERMS INVOLVING X(_) IN EQUATION ___ EXCEEDS 63	During ordering the referenced variable appeared in more than 63 reductions in the equation referenced.
(11) THE NUMBER OF TERMS INVOLVING X(_) IN ALL EQUATIONS EXCEEDS 63	During ordering the referenced variable appeared in more than 63 reductions.
(12) SYSTEM CANNOT BE SOLVED	All variables were eliminated from the final equation. Check input equations.

Messages (warnings)	Cause
(1) ONLY ONE TERM IN EQUATION ___	The equation referenced contained only one variable during reduction. This may be a result of an error in the input equations.
(2) SOLUTION BELOW MAY NOT BE UNIQUE	All block diagram equations were not used for the reduction.

The printed output of the FORMAC program lists the following items:
(1) The number of variables
(2) The number of block-diagram equations

(3) The transfer function requested

(4) The block-diagram equations after the variables that were specified as block-diagram inputs have been made zero

(5) The variables that were specified as block-diagram inputs

(6) The solution in the G-form

(7) The solution in the N/D-form

(8) The system information, if any

(9) The solution in the S-form, if requested

(10) The solution in the complex rational form, if requested

Functional forms preceded by R or I indicate the real or imaginary parts associated with that function. Refer to equations (20) to (28). The punched output contains the solution in all the different forms. This output is punched in a form that is acceptable to the FORTRAN IV compiler. The S and complex rational forms are compatible with the FORTRAN programs given in this report.

The examples of printed and punched output which follow were generated by REDUCE using the input listed in the Input to REDUCE section.

Printed Output from FORMAC Program

```
THE NUMBER OF VARIABLES IS    6
THE NUMBER OF EXPRESSIONS IS    5

THE SOLUTION IS FOR X(1)$ TO X(2)$

THE BLOCK DIAGRAM EXPRESSIONS

   1  -X(1)+X(2)-X(6)$
   2  G(1)*X(2)-X(3)$
   3  G(2)*X(6)-X(4)$
   4  X(3)+X(4)-X(5)$
   5  G(3)*X(5)-X(1)$

THE BLOCK DIAGRAM INPUTS

    X(2)$
```

```
THE NUMERATOR
     G(1)*G(3)+G(2)*G(3)

THE DENOMINATOR
     G(2)*G(3)+1.0

THE NUMERATOR
     D(1)*N(2)*N(3)+D(2)*N(1)*N(3)

THE DENOMINATOR
     D(1)*D(2)*D(3)+D(1)*N(2)*N(3)
```

```
N( 1)= -K(1)*S**2.0*SINHF(1)*Z(1)**2.0*Z(2)$

D( 1)= COSHF(1)*SQRTF(1)*T(1)*Z(3)*Z(4)$

N( 2)= K(1)+S*Z(5)$

D( 2)= 1.0$

N( 3)= 1.0$

D( 3)= S**2.0*Z(6)$
```

```
THE NUMERATOR
     COSHF(1)*(K(1)+S*Z(5))*SQRTF(1)*T(1)*Z(3)*Z(4)-K(1)*S**2   *SINHF(1
     )*Z(1)**2   *Z(2)

THE DENOMINATOR
     COSHF(1)*(K(1)+S*Z(5))*SQRTF(1)*T(1)*Z(3)*Z(4)+COSHF(1)*S**2   *
     SQRTF(1)*T(1)*Z(3)*Z(4)*Z(6)

THE NUMERATOR (REAL)

     -ICOSH(1)*ISQRT(1)*K(1)*T(1)*Z(3)*Z(4)+K(1)*RCOSH(1)*RSQRT(1)*T(1)
     *Z(3)*Z(4)

     K(1)*RSINH(1)*W**2   *Z(1)**2   *Z(2)
     (-ICOSH(1)*RSQRT(1)*T(1)*Z(3)*Z(4)*Z(5)-ISQRT(1)*RCOSH(1)*T(1)*Z(3
     )*Z(4)*Z(5))*W

THE NUMERATOR (IMAGINARY)

     ICOSH(1)*K(1)*RSQRT(1)*T(1)*Z(3)*Z(4)+ISQRT(1)*K(1)*RCOSH(1)*T(1)*
     Z(3)*Z(4)

     ISINH(1)*K(1)*W**2   *Z(1)**2   *Z(2)

     (-ICOSH(1)*ISQRT(1)*T(1)*Z(3)*Z(4)*Z(5)+RCOSH(1)*RSQRT(1)*T(1)*Z(3
     )*Z(4)*Z(5))*W

THE DENOMINATOR (REAL)

     -ICOSH(1)*ISQRT(1)*K(1)*T(1)*Z(3)*Z(4)+K(1)*RCOSH(1)*RSQRT(1)*T(1)
     *Z(3)*Z(4)

     (ICOSH(1)*ISQRT(1)*T(1)*Z(3)*Z(4)*Z(6)-RCOSH(1)*RSQRT(1)*T(1)*Z(3)
     *Z(4)*Z(6))*W**2

     (-ICOSH(1)*RSQRT(1)*T(1)*Z(3)*Z(4)*Z(5)-ISQRT(1)*RCOSH(1)*T(1)*Z(3
     )*Z(4)*Z(5))*W

THE DENOMINATOR (IMAGINARY)

     ICOSH(1)*K(1)*RSQRT(1)*T(1)*Z(3)*Z(4)+ISQRT(1)*K(1)*RCOSH(1)*T(1)*
     Z(3)*Z(4)

     (-ICOSH(1)*RSQRT(1)*T(1)*Z(3)*Z(4)*Z(6)-ISQRT(1)*RCOSH(1)*T(1)*Z(3
     )*Z(4)*Z(6))*W**2

     (-ICOSH(1)*ISQRT(1)*T(1)*Z(3)*Z(4)*Z(5)+RCOSH(1)*RSQRT(1)*T(1)*Z(3
     )*Z(4)*Z(5))*W
```

Listing of FORMAC Punched Output

The G-form:

```
C     THE NUMERATOR
      N0=
     *G(1)*G(3)+G(2)*G(3)
C     THE DENOMINATOR
      D0=
     *G(2)*G(3)+1.0
```

The N/D-form:

```
C     THE NUMERATOR
      N1=
     *D(1)*N(2)*N(3)+D(2)*N(1)*N(3)
C     THE DENOMINATOR
      D1=
     *D(1)*D(2)*D(3)+D(1)*N(2)*N(3)
```

The S-form:

```
C     THE NUMERATOR
      N2=
     *COSHF(1)*(K(1)+S*Z(5))*SQRTF(1)*T(1)*Z(3)*Z(4)-K(1)*S**2   *SINHF(1
     *)*Z(1)**2   *Z(2)
C     THE DENOMINATOR
      D2=
     *COSHF(1)*(K(1)+S*Z(5))*SQRTF(1)*T(1)*Z(3)*Z(4)+COSHF(1)*S**2   *
     *SQRTF(1)*T(1)*Z(3)*Z(4)*Z(6)
```

The complex rational form:

```
C     THE NUMERATOR (REAL)
      A( 1)=
     *-ICOSH(1)*ISQRT(1)*K(1)*T(1)*Z(3)*Z(4)+K(1)*RCOSH(1)*RSQRT(1)*T(1)
     **Z(3)*Z(4)
      A( 2)=
     *K(1)*RSINH(1)*W**2   *Z(1)**2   *Z(2)
      A( 3)=
     *(-ICOSH(1)*RSQRT(1)*T(1)*Z(3)*Z(4)*Z(5)-ISQRT(1)*RCOSH(1)*T(1)*Z(3
     *)*Z(4)*Z(5))*W
      NA= 3
```

```
C     THE NUMERATOR (IMAGINARY)
      B( 1)=
     *ICOSH(1)*K(1)*RSQRT(1)*T(1)*Z(3)*Z(4)+ISQRT(1)*K(1)*RCOSH(1)*T(1)*
     *Z(3)*Z(4)
      B( 2)=
     *ISINH(1)*K(1)*W**2  *Z(1)**2  *Z(2)
      B( 3)=
     *(-ICOSH(1)*ISQRT(1)*T(1)*Z(3)*Z(4)*Z(5)+RCOSH(1)*RSQRT(1)*T(1)*Z(3
     *)*Z(4)*Z(5))*W
      NB= 3
C     THE DENOMINATOR (REAL)
      C( 1)=
     *-ICOSH(1)*ISQRT(1)*K(1)*T(1)*Z(3)*Z(4)+K(1)*RCOSH(1)*RSQRT(1)*T(1)
     **Z(3)*Z(4)
      C( 2)=
     *(ICOSH(1)*ISQRT(1)*T(1)*Z(3)*Z(4)*Z(6)-RCOSH(1)*RSQRT(1)*T(1)*Z(3)
     **Z(4)*Z(6))*W**2
      C( 3)=
     *(-ICOSH(1)*RSQRT(1)*T(1)*Z(3)*Z(4)*Z(5)-ISQRT(1)*RCOSH(1)*T(1)*Z(3
     *)*Z(4)*Z(5))*W
      NC= 3
C     THE DENOMINATOR (IMAGINARY)
      D( 1)=
     *ICOSH(1)*K(1)*RSQRT(1)*T(1)*Z(3)*Z(4)+ISQRT(1)*K(1)*RCOSH(1)*T(1)*
     *Z(3)*Z(4)
      D( 2)=
     *(-ICOSH(1)*RSQRT(1)*T(1)*Z(3)*Z(4)*Z(6)-ISQRT(1)*RCOSH(1)*T(1)*Z(3
     *)*Z(4)*Z(6))*W**2
      D( 3)=
     *(-ICOSH(1)*ISQRT(1)*T(1)*Z(3)*Z(4)*Z(5)+RCOSH(1)*RSQRT(1)*T(1)*Z(3
     *)*Z(4)*Z(5))*W
      ND= 3
```

INPUT TO EVAL

The user may evaluate the transfer function by using either the S-form or the complex rational form. If the S-form is to be evaluated, the S-form on punched cards is inserted between the two comment cards that contain asterisks in program COEFFS (see appendix C). The complex rational form requires the insertion of the punched cards between the two comment cards that contain asterisks in program COEFFR. The numerical results from both forms are identical. When the expression punched by EVAL exceeds 19 continuation cards, the user must manually divide the expression into two or more parts. Each of these parts must contain less than 19 continuation cards to comply with the FORTRAN IV compiler.

If the user has made use of the arbitrary function in REDUCE, he must supply a function subprogram for his particular function. The numerical input consists of specifying values for the explicit variables used in the system information and values for the variables implied in the arguments of the allowed functions. A number corresponding to the maximum subscript employed in each variable and function used must be supplied.

The frequency range and increment desired must also be supplied.

The first card of the EVAL input must contain $INPUT beginning in card column 2, and the last card must contain a $ in card column 2. The following input forms may appear in any order but all must begin in card column 2:

(1) NEXP=number, EXPA=list, EXPB=list, EXPC=list, EXPD=list,

(2) NSIN=number, SINA=list, SINB=list, SINC=list, SIND=list,

(3) NCOS=number, COSA=list, COSB=list, COSC=list, COSD=list,

(4) NSINH=number, SINHA=list, SINHB=list, SINHC=list, SINHD=list,

(5) NCOSH=number, COSHA=list, COSHB=list, COSHC=list, COSHD=list,

(6) NSQRT=number, SQRTA=list, SQRTB=list, SQRTC=list,

(7) NK=number, K=list,

(8) NT=number, T=list,

(9) NZ=number, Z=list,

(10) NWN=number, WN=list,

(11) FSTART=initial frequency, FEND=final frequency, DELTAF=increment,

Any of these forms may use more than one card as long as each card ends with a comma. The suffixes A, B, C, and D in the function names represent the variables implied in the function argument. (Note: no D in SQRT.) For example, EXPA, EXPB, EXPC, and EXPD represent the variables a, b, c, and d in the function

$$\exp\left(d\sqrt{as^2 + bs + c}\right)$$

(See eqs. (15) to (20) for additional forms.) The term "number," as used in the preceding forms, means an integer number which has the value of the maximum subscript used for that function or variable. For the functions, this number must not be greater than 5; for variables, not greater than 10. The term "list" means a list of numbers that the variables are to take on. The elements of this list must be separated by commas and must be in the proper order. For example, NK=2, K=1, 6.25E-2 would set K(1)=1.0 and K(2)=0.0625. For consecutive runs (runs where the transfer function does not change) only the changed numbers need to be given on the input. For example,

```
$INPUT
K(2)=1.25E-2,
$
```

In the following example of input to EVAL, it is understood that either the S-form of the complex rational form of the transfer function (given in the FORMAC Sample Output) from REDUCE is to be inserted in the proper FORTRAN subroutine. These numerical data are taken from figure 8(c).

```
$DATA
 $INPUT
 NSINH=1,SINHA=4.100625E-6,SINHB=1.0E-4,SINHC=0,SINHD=3.33333,
 NCOSH=1,COSHA=4.100625E-6,COSHB=1.0E-4,COSHC=0,COSHD=3.33333,
 NSQRT=1,SQRTA=4.100625E-6,SQRTB=1.0E-4,SQRTC=0,
 NK=1,K=.7633593RE6,
 NT=1,T=1.0416E-2,
 NZ=6,Z=0.750,62.4,6.48E7,32.2,10,1,
 FSTART=1,FEND=50,DELTAF=1,
 $
```

EVAL OUTPUT

The output printed by the FORTRAN program lists the following items:

(1) All numerical input

(2) For each frequency requested

(a) The frequency

(b) The real and imaginary parts of the numerator and denominator of the transfer function normalized so that either the real or imaginary part of the numerator is one

(c) The real and imaginary parts of the transfer function

(d) The complex absolute value of the transfer function

(e) The complex amplitude (phase angle) of the transfer function

An example of the EVAL output follows. This output was obtained by using the input given in the previous section.

```
FORTRAN EVALLATION OF FORMAC OUTPUT

    I      SINHA          SINHB          SINHC             SINHD
    1    0.41006E-05    0.10000E-03    0                 3.33333

    I      COSHA          COSHB          COSHC             COSHD
    1    0.41006E-05    0.10000E-03    0                 3.33333

    I      SQRTA          SQRTB          SQRTC
    1    0.41006E-05    0.10000E-03    0

    I      K
    1    0.76336E 06

    I      T
    1    0.10416E-01

    I      Z
    1    0.75000
    2    62.4000
    3    0.64800E 08
    4    32.2000
    5    10.0000
    6    1.00000



FSTART=     1.00000          FEND= .   50.0000          DELTAF=     1.00000

             F              A            B              C              D             REAL           IMAG          ABS            ANGLE
    1    1.00000        1.00000      1.29988        0.99974        1.29954       1.00026       -0.52105E-06   1.00026       -0.29846E-04
    2    2.00000        1.00000      1.66699        0.99893        1.66523       1.00106       -0.41571E-05   1.00106       -0.23793E-03
    3    3.00000        1.00000      2.09351        0.99759        2.08853       1.00239       -0.14119E-04   1.00239       -0.80704E-03
    4    4.00000        1.00000      2.57234        0.99567        2.56145       1.00426       -0.33830E-04   1.00426       -0.19301E-02
    5    5.00000        1.00000      3.10066        0.99314        3.08010       1.00669       -0.66941E-04   1.00669       -0.38099E-02
    6    6.00000        1.00000      3.68043        0.98997        3.64520       1.00970       -0.11756E-03   1.00970       -0.66712E-02
    7    7.00000        1.00000      4.31822        0.98608        4.26174       1.01329       -0.19034E-03   1.01329       -0.10763E-01
    8    8.00000        1.00000      5.02519        0.98138        4.93900       1.01751       -0.29062E-03   1.01751       -0.16365E-01
    9    9.00000        1.00000      5.81798        0.97575        5.69107       1.02237       -0.42466E-03   1.02237       -0.23799E-01
   10    10.0000        1.00000      6.72039        0.96902        6.53839       1.02792       -0.59990E-03   1.02792       -0.33438E-01
   11    11.0000        1.00000      7.76657        0.96093        7.51047       1.03420       -0.82526E-03   1.03420       -0.45720E-01
   12    12.0000        1.00000      9.00666        0.95114        8.65074       1.04127       -0.11116E-02   1.04127       -0.61164E-01
   13    13.0000        1.00000      10.5169        0.93907        10.0253       1.04917       -0.14721E-02   1.04917       -0.80391E-01
   14    14.0000        1.00000      12.4188        0.92384        11.7398       1.05799       -0.19232E-02   1.05800       -0.10415
   15    15.0000        1.00000      14.9189        0.90396        13.9734       1.06782       -0.24854E-02   1.06782       -0.13336
   16    16.0000        1.00000      18.3976        0.87664        17.0570       1.07876       -0.31847E-02   1.07876       -0.16915
   17    17.0000        1.00000      23.6435        0.83610        21.6759       1.09093       -0.40539E-02   1.09094       -0.21291
   18    18.0000        1.00000      32.5991        0.76814        29.5185       1.10449       -0.51356E-02   1.10450       -0.26641
   19    19.0000        1.00000      51.6918        0.62572        46.1724       1.11963       -0.64851E-02   1.11964       -0.33186
   20    20.0000        1.00000      122.467        0.10475        107.753       1.13656       -0.81755E-02   1.13659       -0.41214
   21    21.0000        1.00000     -333.826        3.44122       -288.852       1.15558       -0.10305E-01   1.15562       -0.51092
   22    22.0000        1.00000     -70.0887        1.50742       -59.5307       1.17702       -0.13006E-01   1.17710       -0.63310
   23    23.0000        1.00000     -38.7821        1.27455       -32.2645       1.20136       -0.16464E-01   1.20147       -0.78514
   24    24.0000        1.00000     -26.5304        1.18090       -21.5644       1.22914       -0.20937E-01   1.22932       -0.97585
   25    25.0000        1.00000     -19.9393        1.12841       -15.7868       1.26112       -0.26799E-01   1.26140       -1.21735
   26    26.0000        1.00000     -15.7847        1.09352       -12.1290       1.29828       -0.34603E-01   1.29875       -1.52672
   27    27.0000        1.00000     -12.9000        1.06766       -9.57693       1.34195       -0.45187E-01   1.34271       -1.92856
   28    28.0000        1.00000     -10.7606        1.04701       -7.67463       1.39394       -0.59868E-01   1.39522       -2.45929
   29    29.0000        1.00000     -9.09548        1.02959       -6.18662       1.45674       -0.80793E-01   1.45898       -3.17447
   30    30.0000        1.00000     -7.75048        1.01426       -4.97902       1.53289       -0.11162       1.53794       -4.16203
   31    31.0000        1.00000     -6.63146        1.00034       -3.97007       1.63032       -0.15891       1.63805       -5.56699
   32    32.0000        1.00000     -5.67758        0.98738       -3.10700       1.75264       -0.23512       1.76834       -7.64072
   33    33.0000        1.00000     -4.84777        0.97509       -2.35419       1.90785       -0.36544       1.94253       -10.8434
   34    34.0000        1.00000     -4.11324        0.96325       -1.68675       2.09418       -0.60306       2.17928       -16.0647
   35    35.0000        1.00000     -3.45320        0.95171       -1.08672       2.25445       -1.05416       2.48873       -25.0604
   36    36.0000        1.00000     -2.85222        0.94036       -0.54081       2.10994       -1.81967       2.78623       -40.7754
   37    37.0000        1.00000     -2.29859        0.92911       -0.38975E-01   1.17801       -2.42455       2.69558       -64.0865
   38    38.0000        1.00000     -1.78321        0.91790        0.42655       0.15352       -2.01405       2.01989       -85.6412
   39    39.0000        1.00000     -1.29891        0.90667        0.86185      -0.13599       -1.30335       1.31043       -95.9567
   40    40.0000        1.00000     -0.83991        0.89538        1.27181      -0.71441E-01   -0.83658       0.83962       -94.8811
   41    41.0000        1.00000     -0.40149        0.88399        1.66036       0.61437E-01   -0.56957       0.57287       -83.8435
   42    42.0000        1.00000      0.20303E-01    0.87248        2.03075       0.18704       -0.41207       0.45253       -65.5869
   43    43.0000        1.00000      0.42878        0.86081        2.38568       0.29285       -0.31350       0.42900       -46.9507
   44    44.0000        1.00000      0.82680        0.84897        2.72744       0.38041       -0.24823       0.45424       -33.1263
   45    45.0000        1.00000      1.21687        0.83692        3.05799       0.45347       -0.20291       0.49679       -24.1065
   46    46.0000        1.00000      1.60122        0.82466        3.37901       0.51540       -0.17016       0.54276       -18.2707
   47    47.0000        1.00000      1.98184        0.81216        3.69202       0.56885       -0.14572       0.58721       -14.3684
   48    48.0000        1.00000      2.36061        0.79941        3.99836       0.61578       -0.12699       0.62874       -11.6521
   49    49.0000        1.00000      2.73926        0.78639        4.29923       0.65769       -0.11230       0.66721       -9.68961
   50    50.0000        1.00000      3.11947        0.77307        4.59578       0.69569       -0.10057       0.70292       -8.22560

                                                                                    REC=  00000 FIL=
```

APPENDIX C

DESCRIPTIONS, LISTINGS, AND FLOW CHARTS FOR COMPUTER PROGRAMS

FORMAC Program

REDUCE (main program). - This FORMAC program is the body of the FORMAC programs. It reads the input, reduces the equations, performs the substitutions, expands and separates the real and imaginary parts, and calls other subroutines for initialization and output.

Subroutine SOUT. - This FORMAC subroutine writes and punches the symbolic transfer function. To do this, SOUT calls the subroutine that removes the $ and floating point powers from the symbolic expressions.

Subroutine INIT, subroutine SORT. - The purpose of subroutine INIT is to determine the order in which the block diagram equations are to be reduced. For a detailed description of the method used, see appendix D.

Subroutine SORT sorts an array of integer numbers into an array of numbers that increase in magnitude. Subroutine SORT is used by subroutine INIT.

Subroutine COS, SIN, THNH, ATAN, ALOG, DUMP, PDUMP, FMCDIF, EXDDMP, FMCDMP. - This is a dummy subroutine that is not executed. Its purpose is to provide entry point names that are the same as the subprogram's that are referenced by FORMAC, but not executed by the programs in this report. This causes the programs that are not needed not to be loaded.

Subroutine REMOVE, subroutine DELETE, function NBR, function FC. - This set of subprograms removes the $ and floating point powers from the BCD representation of FORMAC expressions. They are called just prior to the punching of any expression.

FORTRAN Program

EVAL (main program). - Program EVAL reads and writes all the numerical input and output. It also calls the subprogram that evaluates the symbolic transfer function.

Function EXPFX, SINFX, COSFX, SINHFX, COSHFX, SQRTFX. - This function subprogram evaluates the allowed system functions. Different entry point names are used for each function.

Subroutine COEFF. - Two subroutines with the name COEFF are supplied. The deck names are used to differentiate between them. The subroutine with the deck name COEFFS is used for transfer functions in the S-form. Deck COEFFR is used for the complex rational form of the transfer function. In both cases, the symbolic transfer function is to be inserted between the two comment cards that contain asterisks.

The rest of this appendix gives listings and flow charts (figs. 9 to 14) for the programs. Flow charts are only given for the programs in which the flow warrants a chart.

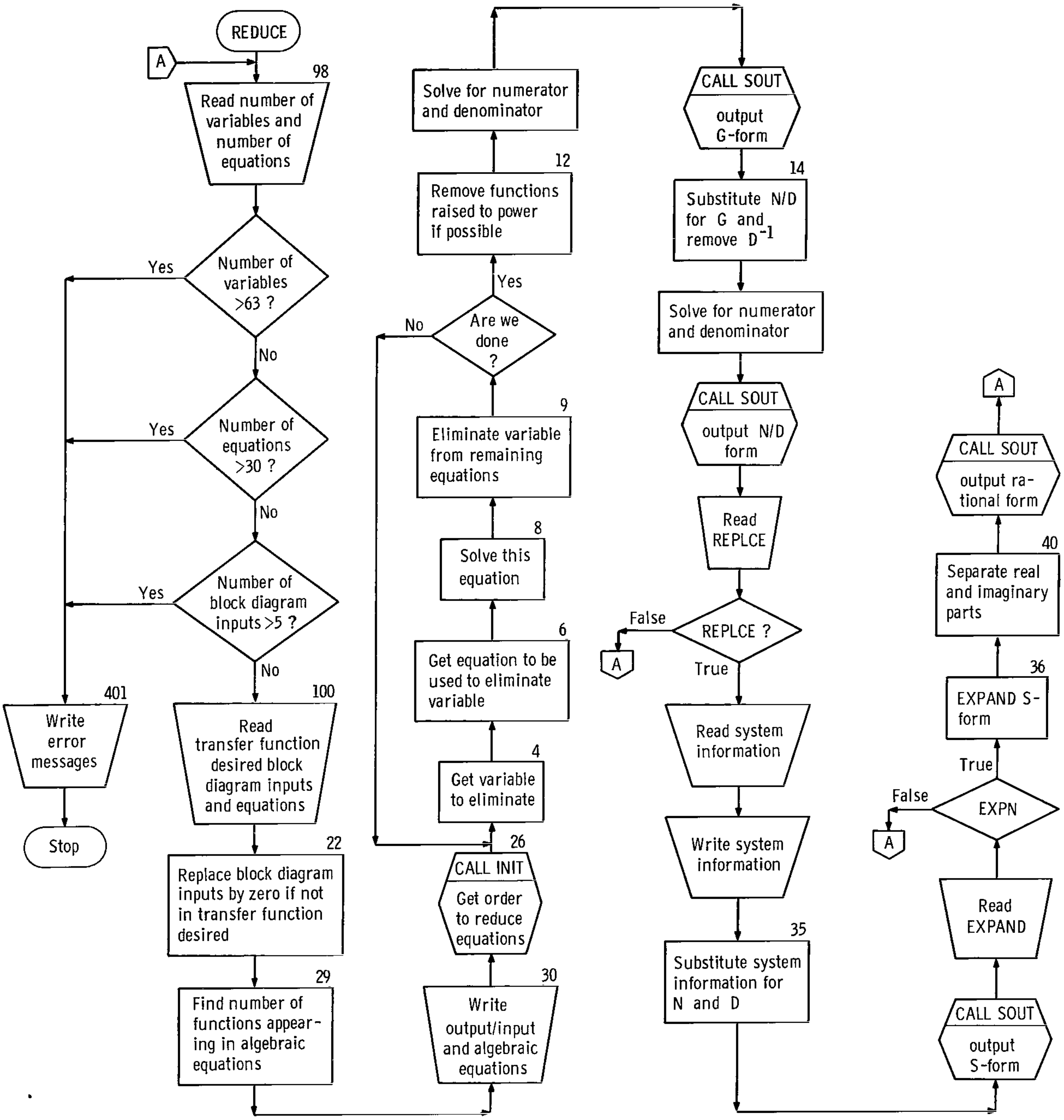

REDUCE
A
98
Read number of variables and number of equations
Number of variables >63 ?
Yes
No
Number of equations >30 ?
Yes
No
Number of block diagram inputs >5 ?
Yes
No
401
Write error messages
Stop
100
Read transfer function desired block diagram inputs and equations
22
Replace block diagram inputs by zero if not in transfer function desired
29
Find number of functions appearing in algebraic equations
30
Write output/input and algebraic equations
26
CALL INIT
Get order to reduce equations
4
Get variable to eliminate
6
Get equation to be used to eliminate variable
8
Solve this equation
9
Eliminate variable from remaining equations
Are we done ?
No
Yes
12
Remove functions raised to power if possible
Solve for numerator and denominator
CALL SOUT
output G-form
14
Substitute N/D for G and remove D^{-1}
Solve for numerator and denominator
CALL SOUT
output N/D form
Read REPLCE
REPLCE ?
False
A
True
Read system information
Write system information
35
Substitute system information for N and D
CALL SOUT
output S-form
Read EXPAND
EXPN
False
A
True
36
EXPAND S-form
40
Separate real and imaginary parts
CALL SOUT
output rational form
A

Figure 9. - Flow chart of REDUCE.

Computer Listings

REDUC

```
$IBFMC REDUCE
      SYMARG
      ATOMIC X(63),G(20),N(20),D(20),I,T(10),Z(10),K(10),WN(10),W,S,
     *EXPF(5),REXP(5),IEXP(5),SINF(5),RSIN(5),ISIN(5),COSF(5),RCOS(5),
     *ICOS(5),SINHF(5),RSINH(5),ISINH(5),COSHF(5),RCOSH(5),ICOSH(5),
     *SQRTF(5),RSQRT(5),ISQRT(5),F(5),RF(5),IF(5)
      DIMENSION IN(60),OUT(17),NEQ(150),EQ(30),INPUT(5),SN(20),SD(20)
      EQUIVALENCE (IN(1),OUT(1),NEQ(1))
      LOGICAL LOCATE,TEST,REPLCE,EXPND
      REAL I,K,N,IS,IT,IEXP,ISIN,ICOS,ISINH,ICOSH,NUM,INPUT,ISQRT,IF
 CMPX PARAM (I**L,I),(I**(L+1),-1),(I**(L+2),-I),(I**(L+3),1)
 FNCT PARAM (EXPF(J),REXP(J)+I*IEXP(J)),
     *            (SINF(J),RSIN(J)+I*ISIN(J)),
     *            (COSF(J),RCOS(J)+I*ICOS(J)),
     *            (SINHF(J),RSINH(J)+I*ISINH(J)),
     *            (COSHF(J),RCOSH(J)+I*ICOSH(J)),
     *            (SQRTF(J),RSQRT(J)+I*ISQRT(J)),
     *            (F(J),RF(J)+I*IF(J))
      DATA DR/6H$00000/
C***********************************************************************
C     READ ALGEBRAIC EQUATIONS
C***********************************************************************
   98 READ (5,101) M,M1,IK
      IF(M.GT.63)  GO TO 401
      IF(M1.GT.30)  GO TO 402
      IF(IK.GT.5)  GO TO 403
      JJ = 0
      READ (5,102) (IN(J),J=1,12)
      DO 96 J=1,IK
      LET INPUT(J) = ALGCON IN(1),JJ
   96 CONTINUE
      JJ = 0
      READ (5,102) (IN(J),J=1,12)
      LET NUM = ALGCON IN(1),JJ
      LET DEN = ALGCON IN(1),JJ
      DO 100 L=1,M1
      JJ = 0
      DO 99 J=1,5
      K1 = 12*(J-1)+1
      K2 = K1+12-1
      READ (5,102) (IN(KK),KK=K1,K2)
      DO 99 KK=1,72
   99 IF(FC(IN(K1),KK).EQ.DR)  GO TO 97
      WRITE (6,230)
      STOP
   97 LET EQ(L) = ALGCON IN(1),JJ
      LET EQ(L) = EXPAND EQ(L)
  100 CONTINUE
C***********************************************************************
C     INITIALIZE AND SET UP EQUATIONS FOR SOLUTION
C***********************************************************************
C
C     FIND THE NUMBER OF FUNCTIONS USED
C
      DO 29 L=1,20
      DO 27 J=1,M1
      LET LOCATE = FIND EQ(J),APP,ALL,(G(L))
   27 IF(LOCATE)  GO TO 28
      GO TO 30
   28 MG = L
   29 CONTINUE
C
C     REPLACE THE INPUTS BY ZERO IF NOT DEN OR NUM
C
   30 DO 22 L=1,M1
      DO 21 J=1,IK
      LET TEST = MATCH ID,NUM,INPUT(J)
      IF(TEST)  GO TO 21
      LET TEST = MATCH ID,DEN,INPUT(J)
      IF(TEST)  GO TO 21
      LET EQ(L) = SUBST EQ(L),(INPUT(J),0)
   21 CONTINUE
   22 CONTINUE
C***********************************************************************
C     WRITE ALGEBRAIC EQUATIONS
C***********************************************************************
      WRITE (6,201) M,M1
      Q = 0.0
      LET Q = BCDCON NUM,OUT,17
      WRITE (6,202) OUT(2)
      Q = 0.0
      LET Q = BCDCON DEN,OUT,17
      WRITE (6,203) OUT(2)
      WRITE (6,204)
      DO 301 L=1,M1
      Q = 0.0
   51 LET Q = BCDCON EQ(L),OUT,10
      WRITE (6,208) L,(OUT(J),J=2,10)
      IF(Q.NE.0.0)  GO TO 51
  301 CONTINUE
   24 WRITE (6,213)
      DO 25 L=1,IK
      Q = 0.0
      LET Q = BCDCON INPUT(L),OUT,17
   25 WRITE (6,206) OUT(2)
C***********************************************************************
C     REDUCE THE EQUATIONS
C***********************************************************************
   26 CALL INIT(NEQ,X,M,EQ,M1,NUM,DEN)
C
C     GET A VARIABLE TO ELIMINATE
C
      II = 0
    4 II = II+1
      J = NEQ(II)
      IF(J.EQ.0)  GO TO 12
C
C     GET AN EQUATION CONTAINING THE VARIABLE
C
    6 II = II+1
      L = NEQ(II)
C
C     SOLVE THE EQUATION
C
    8 LET RR = EQ(L)
      ERASE EQ(L)
      LET IT = COEFF RR,X(J)
      LET RR = EXPAND RR-IT*X(J)
C
C     ELIMINATE THE VARIABLE IN THE REMAINING EQUATIONS
```

```
C
    9 II = II+1
      L = NEQ(II)
      IF(L.EQ.0)  GO TO 10
      LET IS = COEFF EQ(L),X(J)
      LET EQ(L) = EXPAND EQ(L)-IS*X(J)
      LET EQ(L) = EXPAND IS*RR-IT*EQ(L)
      GO TO 9
   10 ERASE IS,IT
      GO TO 4
C
C     REMOVE FUNCTIONS RAISED TO A POWER IF POSSIBLE
C
   12 II = II+1
      L = NEQ(II)
      LET R = EQ(L)
      ERASE EQ(L)
      DO 13 L=1,MG
      J = 0
      LET IS = COEFF R,G(L)**J,V1,V2
      LET TEST = MATCH ID,IS,0.0
      ERASE IS
      IF(.NOT.TEST)  GO TO 13
      LET R = EXPAND R*G(L)**(-V1)
   13 CONTINUE
C***********************************************************************
C     SOLVE FOR NUMERATOR AND DENOMINATOR
C***********************************************************************
      LET RD = COEFF R,NUM
      LET RN = COEFF R,DEN
      LET RD = EXPAND -1*RD
C***********************************************************************
C     WRITE NUMERATOR AND DENOMINATOR
C***********************************************************************
      WRITE (6,205)
      PUNCH 221
      CALL SOUT(RN,W,2,2HNO)
      WRITE (6,207)
      PUNCH 222
      CALL SOUT(RD,W,2,2HDO)
      ERASE RN,RD
C***********************************************************************
C     SUBSTITUTE 'N(L)/D(L)' FOR 'G(L)' AND REMOVE D(L)**(-1)
C***********************************************************************
      DO 14 L=1,MG
      LET LOCATE = FIND R,APP,ALL,(G(L))
      IF(.NOT.LOCATE)  GO TO 14
      J = 0
      LET IS = COEFF R,G(L)**J,V1,V2
      ERASE IS
      LET R = SUBST R,(G(L),N(L)/D(L))
      LET R = EXPAND R*D(L)**V2
   14 CONTINUE
C***********************************************************************
C     SOLVE FOR NUMERATOR AND DENOMINATOR
C***********************************************************************
      LET RD = COEFF R,NUM
      LET RN = COEFF R,DEN
      LET RD = EXPAND -1*RD
      ERASE R
```

```
C***********************************************************************
C     WRITE NUMERATOR AND DENOMINATOR
C***********************************************************************
      WRITE (6,205)
      PUNCH 221
      CALL SOUT(RN,W,2,2HN1)
      WRITE (6,207)
      PUNCH 222
      CALL SOUT(RD,W,2,2HD1)
C***********************************************************************
C     READ AND WRITE THE SYSTEM INFORMATION
C***********************************************************************
      READ (5,103) REPLCE
      IF(.NOT.REPLCE)  GO TO 98
      I1 = 0
      MG2 = 2*MG
      DO 61 L=1,MG2
      LL = MOD(L,2)+1
      JJ = 0
      DO 58 J=1,5
      K1 = 12*(J-1)+1
      K2 = K1+12-1
      READ (5,102) (IN(KK),KK=K1,K2)
      DO 58 KK=1,72
   58 IF(FC(IN(K1),KK).EQ.DR)  GO TO (60,59),LL
      WRITE (6,231)
      STOP
   59 I1 = I1+1
      LET SN(I1) = ALGCON IN(1),JJ
      GO TO 61
   60 LET SD(I1) = ALGCON IN(1),JJ
   61 CONTINUE
      Q = 0.0
      DO 34 L=1,MG
      WRITE (6,214) L
   32 LET Q = BCDCON SN(L),OUT,9
      WRITE (6,215) (OUT(J),J=2,9)
      IF(Q.NE.0.0)  GO TO 32
      WRITE (6,216) L
   33 LET Q = BCDCON SD(L),OUT,9
      WRITE (6,215) (OUT(J),J=2,9)
      IF(Q.NE.0.0)  GO TO 33
   34 CONTINUE
C***********************************************************************
C     SUBSTITUTE THE SYSTEM INFORMATION FOR 'N(L)' AND 'D(L)'
C***********************************************************************
      DO 35 L=1,MG
      LET RN = SUBST RN,(N(L),SN(L)),(D(L),SD(L))
      LET RD = SUBST RD,(N(L),SN(L)),(D(L),SD(L))
      ERASE SN(L),SD(L)
   35 CONTINUE
C***********************************************************************
C     WRITE NUMERATOR AND DENOMINATOR
C***********************************************************************
      WRITE (6,205)
      PUNCH 221
      CALL SOUT(RN,W,2,2HN2)
      WRITE (6,207)
      PUNCH 222
      CALL SOUT(RD,W,2,2HD2)
```

```
C***********************************************************************
C     READ EXPND
C***********************************************************************
      READ (5,103) EXPND
      IF(.NOT.EXPND)  GO TO 98
C***********************************************************************
C     SUBSTITUTE REAL FORMS FOR COMPLEX FORMS AND EXPAND
C***********************************************************************
      DO 38 J=1,5
      LET RN = SUBST RN,FNCT
      LET RD = SUBST RD,FNCT
   38 CONTINUE
      LET RN = SUBST RN,(S,W*I)
      LET RD = SUBST RD,(S,W*I)
      LET RN = EXPAND RN
      J = 0
      LET IS = COEFF RN,I**J,V1,V2
      ERASE IS
      K1 = V2
      IF(K1.LT.2)  GO TO 40
      DO 36 L=1,K1,4
      LET RN = SUBST RN,CMPX
   36 CONTINUE
C***********************************************************************
C     SEPERATE INTO REAL AND IMAGINARY PARTS AND WRITE
C***********************************************************************
   40 LET B1 = COEFF RN,I
      LET A1 = EXPAND RN-I*B1
      ERASE RN
      WRITE (6,217)
      PUNCH 223
      CALL SOUT(A1,W,1,1HA)
      ERASE A1
      WRITE (6,218)
      PUNCH 224
      CALL SOUT(B1,W,1,1HB)
      ERASE B1
      LET RD = EXPAND RD
      J = 0
      LET IS = COEFF RD,I**J,V1,V2
      ERASE IS
      K1 = V2
      IF(K1.LT.2)  GO TO 42
      DO 41 L=1,K1,4
      LET RD = SUBST RD,CMPX
   41 CONTINUE
   42 LET D1 = COEFF RD,I
      LET C1 = EXPAND RD-I*D1
      ERASE RD
      WRITE (6,219)
      PUNCH 225
      CALL SOUT(C1,W,1,1HC)
      ERASE C1
      WRITE (6,220)
      PUNCH 226
      CALL SOUT(D1,W,1,1HD)
      ERASE D1
      GO TO 98
C***********************************************************************
  401 WRITE (6,227)
```

```
      GO TO 500
  402 WRITE (6,228)
      GO TO 500
  403 WRITE (6,229)
  500 STOP
  101 FORMAT (3I5)
  102 FORMAT (13A6,A2)
  103 FORMAT (L5)
  201 FORMAT (1H1,//,27H THE NUMBER OF VARIABLES IS,I5,/,29H THE NUMBER
     *OF EXPRESSIONS IS,I5)
  202 FORMAT (21HKTHE SOLUTION IS FOR ,A6,3HTO )
  203 FORMAT (1H+,29X,A6)
  204 FORMAT (30HKTHE BLOCK DIAGRAM EXPRESSIONS,//)
  205 FORMAT (1HK,55X,13HTHE NUMERATOR)
  206 FORMAT (5X,9A6)
  207 FORMAT (1HK,55X,15HTHE DENOMINATOR)
  208 FORMAT (3X,I2,2X,9A6)
  213 FORMAT (25HKTHE BLOCK DIAGRAM INPUTS,//)
  214 FORMAT (3H N(,I2,2H)=)
  215 FORMAT (1H+,7X,8A6,//)
  216 FORMAT (3H D(,I2,2H)=)
  217 FORMAT (1HK,55X,20HTHE NUMERATOR (REAL))
  218 FORMAT (1HK,55X,25HTHE NUMERATOR (IMAGINARY))
  219 FORMAT (1HK,55X,22HTHE DENOMINATOR (REAL))
  220 FORMAT (1HK,55X,27HTHE DENOMINATOR (IMAGINARY))
  221 FORMAT (1HC,5X,13HTHE NUMERATOR)
  222 FORMAT (1HC,5X,15HTHE DENOMINATOR)
  223 FORMAT (1HC,5X,20HTHE NUMERATOR (REAL))
  224 FORMAT (1HC,5X,25HTHE NUMERATOR (IMAGINARY))
  225 FORMAT (1HC,5X,22HTHE DENOMINATOR (REAL))
  226 FORMAT (1HC,5X,27HTHE DENOMINATOR (IMAGINARY))
  227 FORMAT (33H **ERROR** MORE THAN 63 VARIABLES)
  228 FORMAT (33H **ERROR** MORE THAN 30 EQUATIONS)
  229 FORMAT (43H **ERROR** MORE THAN 5 BLOCK DIAGRAM INPUTS)
  230 FORMAT (59H **ERROR** MORE THAN 5 CARDS FOR A BLOCK DIAGRAM EXPRES
     *SION)
  231 FORMAT (64H **ERROR** MORE THAN 5 CARDS FOR A SYSTEM INFORMATION E
     *XPRESSION)
      END
```

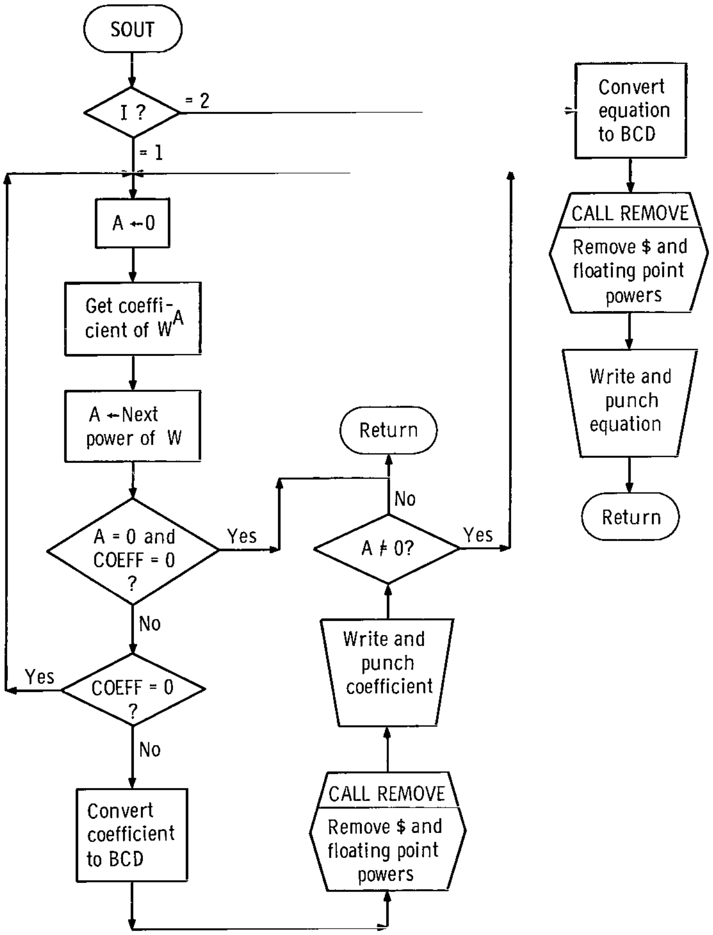


Figure 10. - Flowchart for subroutine SOUT.

SOUT

```
$IBFMC SPLIT
      SUBROUTINE SOUT(X,W,I,NAME)
      DIMENSION OUT(12)
      EQUIVALENCE (OUT(1),MOUT)
      LOGICAL TEST
      SYMARG X,W
      ATOMIC W
      GO TO (4,5),I
    4 A = 0
      J = 0
    1 LET Y = COEFF X,W**A,V1,V2
      LET Y = Y*W**A
      A = V2
      LET TEST = MATCH ID,Y,0.0
      IF(TEST.AND.A.EQ.0.0)  GO TO 7
      IF(TEST)  GO TO 1
      WRITE (6,203)
      J = J+1
      PUNCH 204,NAME,J
    2 LET Q = BCDCON Y,OUT,12
      CALL REMOVE(OUT)
    3 WRITE (6,201) (OUT(L),L=2,12)
      PUNCH 202,(OUT(L),L=2,12)
      IF(Q.NE.0.0)  GO TO 2
      ERASE Y
      IF(A.NE.0.0)  GO TO 1
    7 PUNCH 205,NAME,J
      RETURN
    5 PUNCH 206,NAME
    6 LET Q = BCDCON X,OUT,12
      CALL REMOVE(OUT)
      WRITE (6,201) (OUT(L),L=2,12)
      PUNCH 202,(OUT(L),L=2,12)
      IF(Q.NE.0.0)  GO TO 6
      RETURN
  201 FORMAT (61X,11A6)
  202 FORMAT (5X,1H*,11A6)
  203 FORMAT (1HK)
  204 FORMAT (6X,A1,1H(,I2,2H)=)
  205 FORMAT (6X,1HN,A1,1H=,I2)
  206 FORMAT (6X,A2,1H=)
      END
```

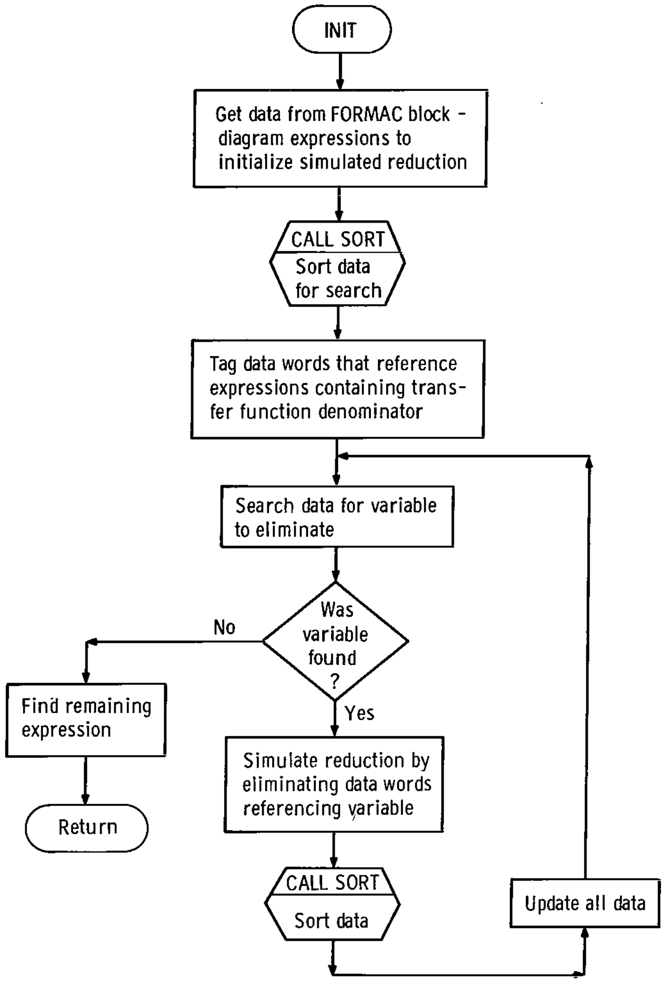

INIT
Get data from FORMAC block - diagram expressions to initialize simulated reduction
CALL SORT
Sort data for search
Tag data words that reference expressions containing transfer function denominator
Search data for variable to eliminate
Was variable found ?
No
Yes
Find remaining expression
Return
Simulate reduction by eliminating data words referencing variable
CALL SORT
Sort data
Update all data

Figure 11. - Flow chart for subroutine INIT.

INIT

```
$IBFMC INIT.
      SUBROUTINE INIT(NEQ,X,M,EQ,M1,NUM,DEN)
      DIMENSION NEQ(150),EQ(30),NQ(100)
      LOGICAL LOCATE,OUT1,OUT2
      REAL NUM
      INTEGER DN,S,SS,SSS,S2,S3,S4,S5,S6,EX4,EX5,EX6,EX
      DATA S2,S3,S4,S5,S6,SSS/0000000000100,0000000010000,
     *0000001000000,0000100000000,0010000000000,0377777777777/
      DATA MASK1,MASK2/0000077777777,0000000000077/
      IFIXO(IY) = IY
      MASK(IX) = IFIXO(AND(MASK1,IX))
      EX(IX,N) = IFIXO(AND(MASK2,IARS((N-1)*6,IABS(IX))))
      SYMARG X,EQ,NUM,DEN
      ATOMIC X(63)
C     INITIALIZE NM,DN, AND THE NQ ARRAY
      NM = 0
      DN = 0
      K = 0
      OUT1 = .FALSE.
      OUT2 = .FALSE.
      DO 5 I=1,M
      L = 0
      DO 1 J=1,M1
      LET LOCATE = FIND EQ(J),APP,ALL,(X(I))
      IF(.NOT.LOCATE)  GO TO 1
      K = K+1
      IF(K.GT.100)  GO TO 301
      L = L+1
      NQ(K) = S4+S5*I+S6*J
    1 CONTINUE
      S = L+S2*L
      N1 = K-L+1
      N2 = K
      DO 2 J=N1,N2
    2 NQ(J) = NQ(J)+S
      IF(NM.NE.0)  GO TO 4
      LET LOCATE = FIND NUM,APP,ALL,(X(I))
      IF(.NOT.LOCATE)  GO TO 4
      NM = I
    4 IF(DN.NE.0)  GO TO 5
      LET LOCATE = FIND DEN,APP,ALL,(X(I))
      IF(.NOT.LOCATE)  GO TO 5
      DN = I
    5 CONTINUE
      IF(NM.EQ.0)  GO TO 302
      IF(DN.EQ.0)  GO TO 303
      MNQ = K
      CALL SORT(NQ,MNQ)
      NQ(MNQ+1) = SSS
      K = 1
      DO 9 I=1,M1
      L = 1
      LOCATE = .FALSE.
    6 IF(EX(NQ(K),6).GT.I)  GO TO 7
      IF(EX(NQ(K),5).EQ.DN)  LOCATE = .TRUE.
      K = K+1
      L = L+1
      GO TO 6
    7 N1 = K-L+1
      N2 = K-1
      S = S3*(L-1)
      DO 8 J=N1,N2
      NQ(J) = NQ(J)+S
    8 IF(LOCATE)  NQ(J) = -NQ(J)
    9 CONTINUE
      MNEQ = 0
      MNQ1 = 0
C     SEARCH FOR A VARIABLE TO ELIMINATE
  100 MIN = SSS
      IF(MNEQ.GT.146)  GO TO 304
      DO 10 I=1,MNQ
      S = EX(NQ(I),5)
      IF(S.EQ.NM.OR.S.EQ.DN.OR.S.EQ.63)  GO TO 10
      MSK = MASK(NQ(I))
      IF(NQ(I).LT.0)  MSK = MSK+S5
      IF(MIN.LE.MSK)  GO TO 10
      IMIN = I
      MIN = MSK
   10 CONTINUE
C     ARE WE DONE
      IF(MIN.EQ.SSS)  GO TO 25
C     ELIMINATE VARIABLE FROM EQUATIONS
      NO = EX(NQ(IMIN),3)
      EX6 = EX(NQ(IMIN),6)
      IF(NO.LT.2)  WRITE (6,405) EX6
      EX5 = EX(NQ(IMIN),5)
      MNEQ = MNEQ+1
      NEQ(MNEQ) = EX5
      MNEQ = MNEQ+1
      NEQ(MNEQ) = EX6
      MNQ2 = MNQ
      DO 11 I=1,MNQ
   11 IF(EX(NQ(I),6).EQ.EX6)  GO TO 12
   12 N1 = I-1
      NQ(IMIN) = SSS
      SS = EX6*S6
      DO 14 I=1,MNQ
      IF(EX(NQ(I),5).NE.EX5)  GO TO 14
      IEQ = EX(NQ(I),6)
      MNEQ = MNEQ+1
      NEQ(MNEQ) = IEQ
      LOCATE = NQ(I).LT.0
      NQ(I) = SSS
      MNQ1 = MNQ1+1
      S = S6*IEQ-SS
      DO 13 J=1,NO
      L = N1+J
      IF(NQ(L).GE.SSS)  GO TO 13
      MNQ2 = MNQ2+1
      NQ(MNQ2) = IABS(NQ(L))+S
      IF(LOCATE)  NQ(MNQ2) = -NQ(MNQ2)
   13 CONTINUE
   14 CONTINUE
      MNEQ = MNEQ+1
      NEQ(MNEQ) = 0
      DO 15 J=1,NO
      L = N1+J
      MNQ1 = MNQ1+1
   15 NQ(L) = SSS
      CALL SORT(NQ,MNQ2)
      MNQ = MNQ2-MNQ1
      MNQ1 = 0
      K = 1
C     UPDATE S3,S4 AND COMBINE LIKE VARIABLES
      DO 21 I=1,M1
      L = 1
   16 IF(EX(NQ(K),6).GT.I)  GO TO 17
      K = K+1
      L = L+1
      GO TO 16
   17 IF(L.EQ.1)  GO TO 21
      N1 = K-L+1
      N2 = K-1
      N = 0
      DO 19 J=N1,N2
      IF(NQ(J).GE.SSS)  GO TO 19
      EX5 = EX(NQ(J),5)
      EX4 = EX(NQ(J),4)
      DO 18 L=N1,N2
      IF(L.EQ.J)  GO TO 18
      IF(EX(NQ(L),5).NE.EX5)  GO TO 18
      S = EX(NQ(L),4)
      IF(S+EX4.LE.63)  GO TO 171
      S = 63-EX4
      IF(.NOT.OUT1)  WRITE (6,406) EX5,I
      OUT1 = .TRUE.
  171 S = S*S4
      IF(NQ(J).LT.0)  S = -S
      NQ(J) = NQ(J)+S
      NQ(L) = SSS
      N = N+1
      MNQ1 = MNQ1+1
   18 CONTINUE
   19 CONTINUE
      S = (N2-N1+1-N)*S3
      DO 20 J=N1,N2
      SS = EX(NQ(J),3)
      IF(SS.EQ.63)  GO TO 20
      SS = S-SS*S3
      IF(NQ(J).LT.0)  SS = -SS
      NQ(J) = NQ(J)+SS
   20 CONTINUE
   21 CONTINUE
C     UPDATE S1 AND S2
      DO 24 I=1,M
      J = 0
      K = 0
      DO 22 L=1,MNQ
      IF(EX(NQ(L),5).NE.I)  GO TO 22
      J = J+1
      K = K+EX(NQ(L),4)
   22 CONTINUE
      IF(J.EQ.0)  GO TO 24
      IF(K.LE.63)  GO TO 221
      K = 63
      IF(.NOT.OUT2)  WRITE (6,407) I
      OUT2 = .TRUE.
  221 S = J*S2+K
      DO 23 L=1,MNQ
      IF(EX(NQ(L),5).NE.I)  GO TO 23
      SS = S-S2*EX(NQ(L),2)-EX(NQ(L),1)
      IF(NQ(L).LT.0)  SS = -SS
      NQ(L) = NQ(L)+SS
   23 CONTINUE
   24 CONTINUE
      CALL SORT(NQ,MNQ)
      MNQ = MNQ-MNQ1
      MNQ1 = 0
      GO TO 100
C     RETURN
   25 MNEQ = MNEQ+1
      NEQ(MNEQ) = 0
      DO 26 I=1,MNQ
   26 IF(NQ(I).LT.SSS)  GO TO 27
      GO TO 308
   27 EX6 = EX(NQ(I),6)
      MNEQ = MNEQ+1
      NEQ(MNEQ) = EX6
      DO 28 I=1,MNQ
      IF(NQ(I).GE.SSS)  GO TO 28
      IF(EX(NQ(I),6).NE.EX6)  GO TO 29
   28 CONTINUE
      RETURN
   29 WRITE (6,409)
      RETURN
C     ERROR
  301 WRITE (6,401)
      GO TO 500
  302 WRITE (6,402)
      GO TO 500
  303 WRITE (6,403)
      GO TO 500
  304 WRITE (6,404)
      GO TO 500
  308 WRITE (6,408)
  500 STOP
  401 FORMAT (48H **ERROR** THE TOTAL NUMBER OF TERMS EXCEEDS 100)
  402 FORMAT (49H **ERROR** THE OUTPUT VARIABLE COULD NOT BE FOUND)
  403 FORMAT (48H **ERROR** THE INPUT VARIABLE COULD NOT BE FOUND)
  404 FORMAT (27H **ERROR** SYSTEM TOO LARGE)
  405 FORMAT (39H **WARNING** ONLY ONE TERM IN EQUATION ,I2)
  406 FORMAT (45H **WARNING** THE NUMBER OF TERMS INVOLVING X(,I2,14H) I
     *N EQUATION ,I2,12H EXCEEDED 63,/,12X,36H SOLUTION MAY NOT BE IN MI
     *NIMUM FORM)
  407 FORMAT (45H **WARNING** THE NUMBER OF TERMS INVOLVING X(,I2,30H) I
     *N ALL EQUATIONS EXCEEDED 63,/,12X,36H SOLUTION MAY NOT BE IN MINIM
     *UM FORM)
  408 FORMAT (34H **ERROR** SYSTEM CANNOT BE SOLVED)
  409 FORMAT (45H **WARNING** SOLUTION BELOW MAY NOT BE UNIQUE)
      END
```

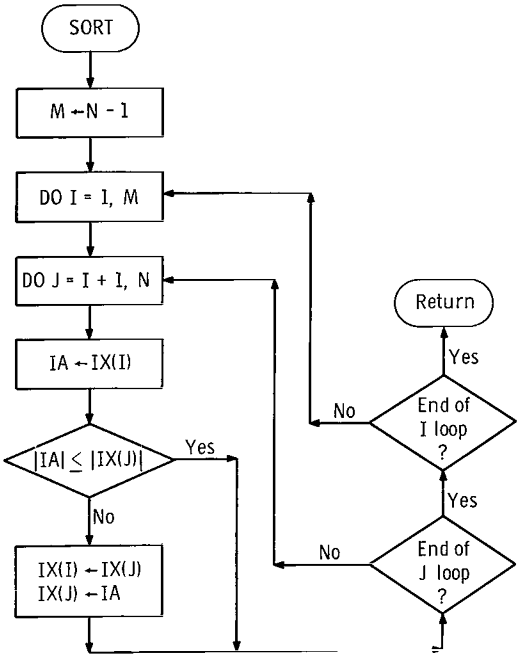


Figure 12. - Flow chart for subroutine SORT.

SORT

```
$IBFTC SORT.
      SUBROUTINE SORT(IX,N)
      DIMENSION IX(1)
      M = N-1
      DO 1 I=1,M
      K = I+1
      DO 1 J=K,N
      IA = IX(I)
      IF(IABS(IA).LE.IABS(IX(J)))  GO TO 1
      IX(I) = IX(J)
      IX(J) = IA
    1 CONTINUE
      RETURN
      END
```

COS

```
$IBFTC TAKOUT
      SUBROUTINE COS(X)
      ENTRY SIN(X)
      ENTRY TANH(X)
      ENTRY ATAN(X)
      ENTRY ALOG(X)
      ENTRY DUMP(X)
      ENTRY PDUMP(X)
      ENTRY FMCDIF(X)
      ENTRY EXPDMP(X)
      ENTRY FMCDMP(X)
      STOP
      END
```

NBR

```
$IBFTC AID1
      LOGICAL FUNCTION NBR(X,L)
      REAL      X(1)
      N=LGR(30,FC(X,L))
      NBR=.FALSE.
      IF((N.GE.0).AND.(N.LE.9)) NBR=.TRUE.
      RETURN
      END
```

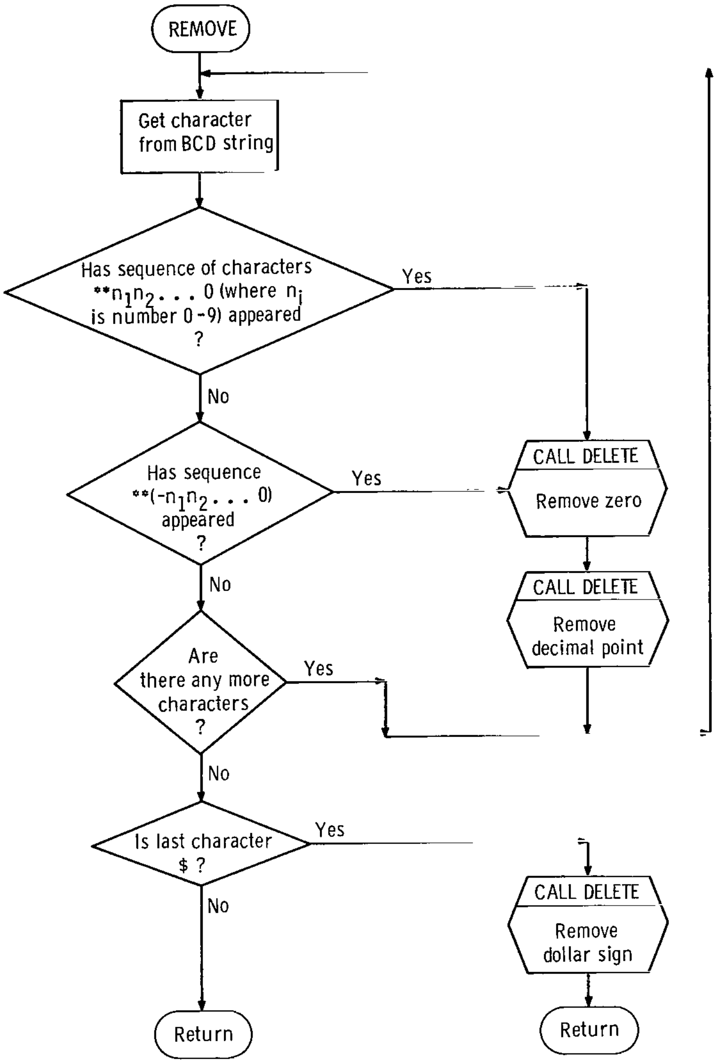

REMOVE
Get character from BCD string
Has sequence of characters $**n_1n_2...0$ (where n_i is number 0-9) appeared ?
Yes
No
Has sequence $**(-n_1n_2...0)$ appeared ?
Yes
No
CALL DELETE
Remove zero
CALL DELETE
Remove decimal point
Are there any more characters ?
Yes
No
Is last character $?
Yes
No
CALL DELETE
Remove dollar sign
Return
Return

Figure 13. - Flow chart for subroutine REMOVE.

REMOVE

```
$IBFTC AID2
      SUBROUTINE REMOVE(X)
      INTEGER X(1)
      LOGICAL NBR
      DATA AAST,AMINUS,AE,ALP,ARP,ADP,AZERO/6H*00000,6H-00000,
     1 6HE00000,6H(00000,6H)00000,6H.00000,6H000000/,MSW/1/
      DATA ADLR,K/6H$00000,1/
      LMAX = X(1)+6
      L = 7
  109 GO TO (110,111,112,113),K
  110 IF(FC(X,L).NE.AAST)  GO TO 201
      GO TO 202
  111 IF(FC(X,L).NE.AAST)  GO TO 201
      GO TO 203
  112 IF(FC(X,L).EQ.ALP)  GO TO 204
  120 IF(NBR(X,L))GO TO 140
      GO TO 201
  113 IF(FC(X,L).NE.AMINUS)  GO TO 201
      L=L+1
      MSW=2
      GO TO 120
  140 L=L+1
      IF(NBR(X,L))GO TO 140
  114 IF(FC(X,L).NE.ADP)  GO TO 201
      L=L+1
  115 IF(FC(X,L).NE.AZERO)  GO TO 201
      L=L+1
  116 GO TO (150,160J,MSW
  150 IF(L.EQ.LMAX+1)  GO TO 170
      IF(NBR(X,L).OR.FC(X,L).EQ.AE)  GO TO 201
      GO TO 170
  160 MSW=1
      IF(L.EQ.LMAX+1)  GO TO 170
      IF(FC(X,L).NE.ARP)  GO TO 201
  170 CALL DELETE(X,L-1)
      CALL DELETE(X,L-2)
  201 K = 1
      GO TO 209
  202 K = 2
      GO TO 209
  203 K = 3
      GO TO 209
  204 K = 4
  209 L = L+1
      IF(L.LE.LMAX)  GO TO 109
  210 IF(FC(X,LMAX).NE.ADLR)  RETURN
      K = 1
      CALL DELETE(X,LMAX)
      RETURN
      END
```

Function FC

```
$IBFTC AID3
      FUNCTION FC(X,L)
      DIMENSION X(1)
      DATA MASK/0770000000000/
      LM1=L-1
      IW=LM1/6+1
      FC=AND(ALS(6*MOD(LM1,6),X(IW)),MASK)
      RETURN
      END
```

DELETE

```
$IBFTC AID4
      SUBROUTINE DELETE(X,L)
      DIMENSION X(L),ZERO(6),BLANK(6)
      DATA ZERO /0007777777777,0770077777777,0777700777777,
     1           0777777007777,0777777770077,0777777777700/,
     2    BLANK /0600000000000,0006000000000,0000060000000,
     3           0000000600000,0000000006000,0000000000060/
      LM1=L-1
      IW=LM1/6+1
      IP=MOD(LM1,6)+1
      X(IW)=OR(AND(X(IW),ZERO(IP)),BLANK(IP))
      RETURN
      END
```

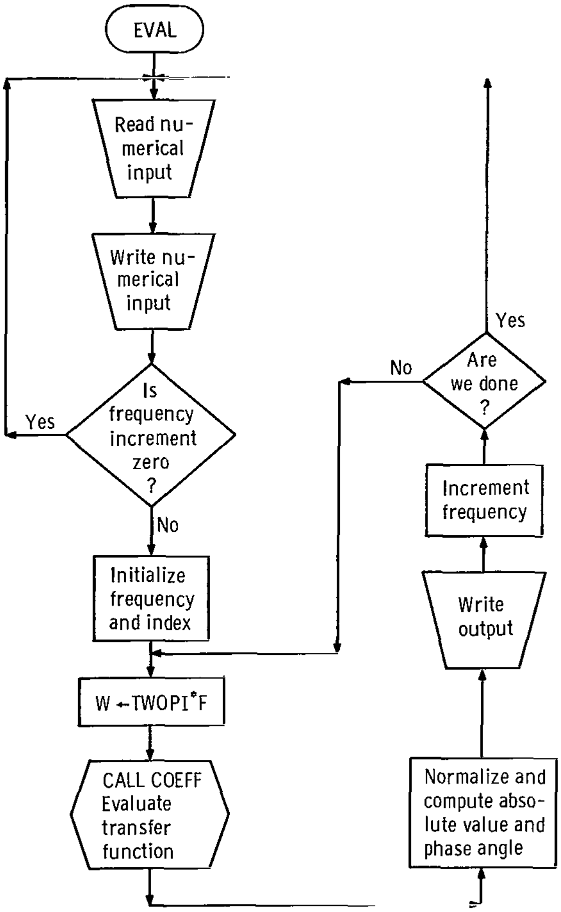

EVAL
Read numerical input
Write numerical input
Is frequency increment zero ?
Yes
No
Initialize frequency and index
W ←TWOPI*F
CALL COEFF Evaluate transfer function
Normalize and compute absolute value and phase angle
Write output
Increment frequency
Are we done ?
No
Yes

Figure 14. - Flow chart of EVAL (main program).

EVAL

```
$IBFTC EVAL
      REAL ITMP,K
      DATA TWOPI/6.2831853/
      DATA NEXP,NSIN,NCOS,NSINH,NCOSH,NSQRT,NK,NT,NZ,NWN/10*0/
      COMMON /CEVAL/T(10),Z(10),WN(10),K(10),NEXP,NSIN,NCOS,NSINH,NCOSH,
     *              NSQRT
      COMMON /OMEGA/W,A,B,C,D
      COMMON /NONLIN/ EXPA(5),SINA(5),COSA(5),
     *                EXPB(5),SINB(5),COSB(5),
     *                EXPC(5),SINC(5),COSC(5),
     *                EXPD(5),SIND(5),COSD(5),
     *                SINHA(5),COSHA(5),SQRTA(5),
     *                SINHB(5),COSHB(5),SQRTB(5),
     *                SINHC(5),COSHC(5),SQRTC(5),
     *                SINHD(5),COSHD(5)
      NAMELIST /INPUT/ NEXP,NSIN,NCOS,
     *                 EXPA,SINA,COSA,
     *                 EXPB,SINB,COSB,
     *                 EXPC,SINC,COSC,
     *                 EXPD,SIND,COSD,
     *                 NSINH,NCOSH,NSQRT,
     *                 SINHA,COSHA,SQRTA,
     *                 SINHB,COSHB,SQRTB,
     *                 SINHC,COSHC,SQRTC,
     *                 SINHD,COSHD,
     *                 NK,K,NT,T,NZ,Z,NWN,WN,FSTART,FEND,DELTAF
    1 READ (5,INPUT)
      WRITE (6,201)
      IF(NEXP.NE.0)  WRITE (6,202) (I,EXPA(I),EXPB(I),EXPC(I),EXPD(I),
     *I=1,NEXP)
      IF(NSIN.NE.0)  WRITE (6,203) (I,SINA(I),SINB(I),SINC(I),SIND(I),
     *I=1,NSIN)
      IF(NCOS.NE.0)  WRITE (6,204) (I,COSA(I),COSB(I),COSC(I),COSD(I),
     *I=1,NCOS)
      IF(NSINH.NE.0)  WRITE (6,205) (I,SINHA(I),SINHB(I),SINHC(I),
     *SINHD(I),I=1,NSINH)
      IF(NCOSH.NE.0)  WRITE (6,206) (I,COSHA(I),COSHB(I),COSHC(I),
     *COSHD(I),I=1,NCOSH)
      IF(NSQRT.NE.0)  WRITE (6,207) (I,SQRTA(I),SQRTB(I),SQRTC(I),
     *I=1,NSQRT)
      IF(NK.NE.0)  WRITE (6,208) (I,K(I),I=1,NK)
      IF(NT.NE.0)  WRITE (6,209) (I,T(I),I=1,NT)
      IF(NZ.NE.0)  WRITE (6,210) (I,Z(I),I=1,NZ)
      IF(NWN.NE.0)  WRITE (6,211) (I,WN(I),I=1,NWN)
      WRITE (6,212) FSTART,FEND,DELTAF
      IF(DELTAF.EQ.0.0)  GO TO 1
      F = FSTART
      I = 0
    2 W = TWOPI*F
      CALL COEFF
      DIV = A
      IF(DIV.EQ.0.0)  DIV = B
      IF(DIV.EQ.0.0)  DIV = 1.0
      A = A/DIV
      B = B/DIV
      C = C/DIV
      D = D/DIV
      TEMP = C**2+D**2
      RTMP = (A*C+B*D)/TEMP
      ITMP = (B*C-A*D)/TEMP
      ATMP = SQRT((A**2+B**2)/TEMP)
      PTMP = ATAN2(ITMP,RTMP)*57.2957795
      I = I+1
      WRITE (6,213) I,F,A,B,C,D,RTMP,ITMP,ATMP,PTMP
      F = FSTART+FLOAT(I)*DELTAF
      IF(F.GT.FEND)  GO TO 1
      GO TO 2
  201 FORMAT (1H1,//,36H FORTRAN EVALUATION OF FORMAC OUTPUT)
  202 FORMAT (1H0,3X,1HI,5X,5HEXPA ,10X,5HEXPB ,10X,5HEXPC ,10X,
     *5HEXPD ,/,(I5,4G15.5))
  203 FORMAT (1H0,3X,1HI,5X,5HSINA ,10X,5HSINB ,10X,5HSINC ,10X,
     *5HSIND ,/,(I5,4G15.5))
  204 FORMAT (1H0,3X,1HI,5X,5HCOSA ,10X,5HCOSB ,10X,5HCOSC ,10X,
     *5HCOSD ,/,(I5,4G15.5))
  205 FORMAT (1H0,3X,1HI,5X,5HSINHA,10X,5HSINHB,10X,5HSINHC,10X,
     *5HSINHD,/,(I5,4G15.5))
  206 FORMAT (1H0,3X,1HI,5X,5HCOSHA,10X,5HCOSHB,10X,5HCOSHC,10X,
     *5HCOSHD,/,(I5,4G15.5))
  207 FORMAT (1H0,3X,1HI,5X,5HSQRTA,10X,5HSQRTB,10X,5HSQRTC,/,(I5,3G15.5
     *))
  208 FORMAT (1H0,3X,1HI,5X,1HK,/,(I5,G15.5))
  209 FORMAT (1H0,3X,1HI,5X,1HT,/,(I5,G15.5))
  210 FORMAT (1H0,3X,1HI,5X,1HZ,/,(I5,G15.5))
  211 FORMAT (1H0,3X,1HI,5X,2HWN,/,(I5,G15.5))
  212 FORMAT (1H1,//,8H FSTART=,G15.5,5X,5HFEND=,G15.5,5X,7HDELTAF=,
     *G15.5,//,13X,1HF,13X,1HA,13X,1HB,13X,1HC,13X,1HD,13X,4HREAL,10X,
     *4HIMAG,10X,3HABS,11X,5HANGLE,/)
  213 FORMAT (I5,9G14.5)
      END
```

EXPF

```
$IBFTC FCTNS
      COMPLEX FUNCTION EXPFX(I)
      COMPLEX X,Y
      COMMON /OMEGA/W,A,B,C,D
      COMMON /NONLIN/ EXPA(5),SINA(5),COSA(5),
     *                EXPB(5),SINB(5),COSB(5),
     *                EXPC(5),SINC(5),COSC(5),
     *                EXPD(5),SIND(5),COSD(5),
     *                SINHA(5),COSHA(5),SQRTA(5),
     *                SINHB(5),COSHB(5),SQRTB(5),
     *                SINHC(5),COSHC(5),SQRTC(5),
     *                SINHD(5),COSHD(5)
      X = CMPLX(EXPC(I)-EXPA(I)*W**2,EXPB(I)*W)
      Y = EXPD(I)*CSQRT(X)
      EXPFX = CEXP(Y)
      GO TO 1
      ENTRY SINFX(I)
      X = CMPLX(SINC(I)-SINA(I)*W**2-SINB(I)*W)
      Y = SIND(I)*CSQRT(X)
      EXPFX = CSIN(Y)
      GO TO 1
      ENTRY COSFX(I)
      X = CMPLX(COSC(I)-COSA(I)*W**2,COSB(I)*W)
      Y = COSD(I)*CSQRT(X)
      EXPFX = CCOS(Y)
      GO TO 1
      ENTRY SINHFX(I)
      X = CMPLX(SINHC(I)-SINHA(I)*W**2,SINHB(I)*W)
      Y = SINHD(I)*CSQRT(X)
      X1 = REAL(Y)
      Y1 = AIMAG(Y)
      EXPFX = CMPLX(SINH(X1)*COS(Y1),COSH(X1)*SIN(Y1))
      GO TO 1
      ENTRY COSHFX(I)
      X = CMPLX(COSHC(I)-COSHA(I)*W**2,COSHB(I)*W)
      Y = COSHD(I)*CSQRT(X)
      X1 = REAL(Y)
      Y1 = AIMAG(Y)
      EXPFX = CMPLX(COSH(X1)*COS(Y1),SINH(X1)*SIN(Y1))
      GO TO 1
      ENTRY SQRTFX(I)
      X = CMPLX(SQRTC(I)-SQRTA(I)*W**2,SORTB(I)*W)
      EXPFX = CSQRT(X)
    1 RETURN
      END
```

COEFF

```
$IBFTC COEFFR
      SUBROUTINE COEFF
C     THIS SUBROUTINE ACCEPTS THE FORMAC OUTPUT IN THE COMPLEX RATIONAL
C     FORM
      REAL IEXP,ISIN,ICOS,ISINH,ICOSH,ISQRT,K
      COMPLEX X,EXPFX,SINFX,COSFX,SINHFX,COSHFX,SQRTFX
      DIMENSION REXP(5),IEXP(5),RSQRT(5),ISQRT(5),
     *          RSIN(5),ISIN(5),RCOS(5),ICOS(5),
     *          RSINH(5),ISINH(5),RCOSH(5),ICOSH(5),
     *          A(20),B(20),C(20),D(20)
      COMMON /CEVAL/T(10),Z(10),WN(10),K(10),NEXP,NSIN,NCOS,NSINH,NCOSH,
     *              NSQRT
      COMMON /OMEGA/W,AA,BB,CC,DD
      IF(NEXP.EQ.0)  GO TO 2
      DO 1 I=1,NEXP
      X = EXPFX(I)
      REXP(I) = REAL(X)
    1 IEXP(I) = AIMAG(X)
    2 IF(NSIN.EQ.0)  GO TO 4
      DO 3 I=1,NSIN
      X = SINFX(I)
      RSIN(I) = REAL(X)
    3 ISIN(I) = AIMAG(X)
    4 IF(NCOS.EQ.0)  GO TO 6
      DO 5 I=1,NCOS
      X = COSFX(I)
      RCOS(I) = REAL(X)
    5 ICOS(I) = AIMAG(X)
    6 IF(NSINH.EQ.0)  GO TO 8
      DO 7 I=1,NSINH
      X = SINHFX(I)
      RSINH(I) = REAL(X)
    7 ISINH(I) = AIMAG(X)
    8 IF(NCOSH.EQ.0)  GO TO 10
      DO 9 I=1,NCOSH
      X = COSHFX(I)
      RCOSH(I) = REAL(X)
    9 ICOSH(I) = AIMAG(X)
   10 IF(NSQRT.EQ.0)  GO TO 12
      DO 11 I=1,NSQRT
      X = SQRTFX(I)
      RSQRT(I) = REAL(X)
   11 ISQRT(I) = AIMAG(X)
   12 CONTINUE
C***********************************************************************
C     THE COMPLEX RATIONAL FORM OF THE FORMAC OUTPUT IS TO BE INSERTED
C     AT THIS POINT
C***********************************************************************
      AA = 0.0
      DO 13 J=1,NA
   13 AA = AA+A(J)
      BB = 0.0
      DO 14 J=1,NB
   14 BB = BB+B(J)
      CC = 0.0
      DO 15 J=1,NC
   15 CC = CC+C(J)
      DD = 0.0
      DO 16 J=1,ND
   16 DD = DD+D(J)
      RETURN
      END
```

COEFF

```
$IBFTC COEFFS
      SUBROUTINE COEFF
C     THIS SUBROUTINE ACCEPTS THE FORMAC OUTPUT IN THE S FORM
      REAL K
      COMPLEX N2,D2,EXPFX,SINFX,COSFX,SINHFX,COSHFX,SQRTFX,S
      COMPLEX EXPF(5),SINF(5),COSF(5),SINHF(5),COSHF(5),SQRTF(5)
      COMMON /CEVAL/T(10),Z(10),WN(10),K(10),NEXP,NSIN,NCOS,NSINH,NCOSH,
     *              NSQRT
      COMMON /OMEGA/W,A,B,C,D
      S = CMPLX(0.0,W)
      IF(NEXP.EQ.0)  GO TO 2
      DO 1 I=1,NEXP
    1 EXPF(I) = EXPFX(I)
    2 IF(NSIN.EQ.0)  GO TO 4
      DO 3 I=1,NSIN
    3 SINF(I) = SINFX(I)
    4 IF(NCOS.EQ.0)  GO TO 6
      DO 5 I=1,NCOS
    5 COSF(I) = COSFX(I)
    6 IF(NSINH.EQ.0)  GO TO 8
      DO 7 I=1,NSINH
    7 SINHF(I) = SINHFX(I)
    8 IF(NCOSH.EQ.0)  GO TO 10
      DO 9 I=1,NCOSH
    9 COSHF(I) = COSHFX(I)
   10 IF(NSQRT.EQ.0)  GO TO 12
      DO 11 I=1,NSQRT
   11 SQRTF(I) = SQRTFX(I)
   12 CONTINUE
C************************************************************************
C     THE S FORM OF THE FORMAC OUTPUT IS TO BE INSERTED AT THIS POINT
C************************************************************************
      A = REAL(N2)
      B = AIMAG(N2)
      C = REAL(D2)
      D = AIMAG(D2)
      RETURN
      END
```

APPENDIX D

METHOD OF DETERMINING ORDER

The method of determining the order of equation reduction simulates the FORMAC reduction. Initially, certain data are obtained for each equation and variable. These data are updated at each step of the simulated reduction and are used to determine the variable to be eliminated. This method was chosen because the FORMAC expressions need be examined only once to initialize the simulated reduction. Also, the data retained at each step may be easily updated for the next step.

Four pieces of information are retained at each step of the simulated reduction for each variable in each equation. Listed in order of importance, they are

(A) The number of times the variable has been combined in the equation

(B) The number of variables in the equation

(C) The number of equations in which the variable appears

(D) The number of times the variable has been combined in all equations

These data are initially extracted from the FORMAC block-diagram expressions. The data are updated through each step of the simulated reduction by eliminating variables and equations and by combining like variables in equations.

The variable to eliminate and the equation used to eliminate the variable is chosen by finding the smallest from the list of numbers made by forming the number $(8^3A + 8^2B + 8C + D)$ (where A, B, C, and D are octal numbers). This, in effect, finds the set of smallest A, then reduces this set by finding the intersection between this set and the set of smallest B, and so on for C and D. The variable and equation number associated with the ABCD found are then the ones desired. The variables of the transfer function are, of course, never eliminated. Equations that contain input variables are only used to eliminate a variable if there is no other choice.

A list is formed consisting of the variable eliminated, the equation used to eliminate the variable, and all other equations that contain the variable. This occurs at each step of the simulated reduction. The list is then used for the actual FORMAC reduction.

The data, A, B, C, and D are packed into one computer word along with the equation and variable number. This word is tagged with a negative sign if the equation associated with it contains an input variable. This packing saved considerable storage space; however, limitations to the magnitude of A, B, C, and D, the number of equations, and the number of variables were introduced. These magnitudes, 1 to 30 for the number of equations and 1 to 63 for the other data, were considered large enough for most problems.

To illustrate this method, the set of block-diagram equations given in table I is carried through the simulated reduction in table III.

TABLE III. - STEPS USED IN SIMULATED REDUCTION TO OBTAIN ORDER OF REDUCTION

[Solution is for transfer function X_1/X_2]

Step 0						Step 1						Step 2						Step 3						Step 4					
Equation	Variable	A	B	C	D	Equation	Variable	A	B	C	D	Equation	Variable	A	B	C	D	Equation	Variable	A	B	C	D	Equation	Variable	A	B	C	D
1	1	1	3	2	2	1	1	1	3	2	2	1	1	1	3	2	2	1	1	1	3	2	2	2	1	2	2	1	2
1	2	1	3	2	2	1	2	1	3	2	2	1	2	1	3	2	2	1	2	1	3	2	2	2	2	2	2	1	2
1	6	1	3	2	2	1	6	1	3	2	2	1	6	1	3	2	2	1	6	1	3	2	2						
2	2	1	2	2	2	2	2	1	2	2	2	2	2	1	2	2	2	2	1	1	3	2	2						
2	3	1	2	2	2	2	3	1	2	2	2	2	3	1	2	2	2	2	2	1	3	2	2						
3	4	1	2	2	2	4	3	1	3	2	2	4	1	1	3	2	2	2	6	1	3	2	2						
3	6	1	2	2	2	4	5	1	3	2	2	4	3	1	3	2	2												
4	3	1	3	2	2	4	6	1	3	2	2	4	6	1	3	2	2												
4	4	1	3	2	2	5	1	1	2	2	2																		
4	5	1	3	2	2	5	5	1	2	2	2																		
5	1	1	2	2	2																								
5	5	1	2	2	2																								

	Step 0	Step 1	Step 2	Step 3	Step 4
Variable to eliminate	4	5	3	6	
Equation to eliminate variable	3	5	4	1	
Other equations containing variable	4	4	2	2	

REFERENCES

1. Kvick, P. G.: CMPXMAT, A Transfer Function Evaluation Program. Pratt & Whitney Aircraft (AEC Rep. No. TIM-940) July 23, 1965.

2. Pavkovich, John M.: The Solution of Large Systems of Algebriac Equations. Tech. Rep. No. 33, Stanford Univ., Dec. 6, 1963. (Available from DDC as AD-427753.)

3. Robichaud, Louis P. A.; Boisvert, Maurice, and Robert, Jean: Signal Flow Graphs and Applications. Prentice-Hall, Inc., 1962.

4. Berge, Claude (Alison Doig, trans.): The Theory of Graphs and its Applications. John Wiley & Sons, Inc., 1962.

5. Bower, John L.; and Schultheiss, Peter M.: Introduction to the Design of Servomechanisms. John Wiley & Sons, Inc., 1958.

6. Abramowitz, Milton; and Stegun, Irene A., eds.: Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Appl. Math. Series No. 55, National Bur. Standards, June 1964.

7. Chruchill, Ruel V.: Operational Mathematics. Second ed., McGraw-Hill Book Co., Inc., 1958.

8. Bond, E. R.: User's Preliminary Reference Manual for FORMAC. Boston Advanced Programming Dept., Int. Business Machines, Feb. 1964.

9. Koenig, Harry J.; and Drain, Daniel I.: Method of Relating Modal Damping to Local Dampers in Lumped-Parameter Systems. NASA TN D-3637, 1966.

10. Graham, Dunstan; and McRuer, Duane: Analysis of Nonlinear Control Systems. John Wiley & Sons, Inc., 1961.

NASA-Langley, 1968 — 8 E-4305

> *"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*
>
> —NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546