NASA Technical Memorandum 85690

IMAGES - A Digital Computer Program for Interactive Modal Analysis and Gain Estimation for Eigensystem Synthesis

Ruben L. Jones

FEBRUARY 1984

NASA Technical Memorandum 85690

IMAGES - A Digital Computer Program for Interactive Modal Analysis and Gain Estimation for Eigensystem Synthesis

Ruben L. Jones
Langley Research Center
Hampton, Virginia

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1984

SUMMARY

An interactive digital computer program for modal analysis and gain estimation for eigensystem synthesis has been written. Both the mathematical and the operational considerations are described; however, the mathematical presentation is limited to those concepts essential to the operational capability of the program. The program is capable of both modal and spectral synthesis of multi-input control systems. It is also user friendly with scratchpad capability and dynamic memory and can be used to design either state or output feedback systems.

INTRODUCTION

Srinathkumar (refs. 1, 2, and 3) developed several algorithms for the synthesis of control systems by using the concept of eigensystem (or pole) placement. He later programmed one of the algorithms (modal synthesis) for use on the Control Data Corporation (CDC) CYBER computer system and solved several example problems. But that program was written as a subroutine and was intended to operate in a "batch" mode. Moreover, the program was not formally documented for public use and did not include the algorithm for spectral synthesis.

This document describes a digital computer program which is based on the work of Srinathkumar. This program takes advantage of the interactive capability of an on-line terminal within the modern computer complex. It also includes those algorithms for both modal and spectral synthesis as well as numerous other options. The program is called IMAGES, which is an acronym for Interactive Modal Analysis and Gain Estimation System.

This paper describes the interactive program and the mathematical basis for the major algorithms. The reader is referred to references 1 and 2 for those details omitted here. The present paper is intended to introduce the interactive program IMAGES to the scientific community and to serve as a user's guide. Hence, program structure, interactive discourse, and other user-related information are emphasized. An example problem is included to clarify operational procedures and to demonstrate ease of application. It is also intended to clarify certain points not clearly evident within references 1 and 2.

SYMBOLS

A	system matrix (numerical subscripts indicate particular partitions of this matrix)
B	input matrix (numerical subscripts indicate particular partitions of this matrix)
C	matrix defined in equations (23)
C_i,C_k	matrices defined by equations (10b) and (15b), respectively

D	matrix defined in equations (23)
e_r	vector with the rth entry equal to unity and the remaining elements equal to zero
F	matrix defined in equations (9)
$f_r^{(k-1)}$	vector defined in equation (21)
G	matrix defined in equations (9)
$g_r^{(k)}$	vector defined in equation (21)
$h_r^{(k-1)}$	vector defined in equation (21)
I	order identity matrix
i,k	indices
$K,\bar{K},\tilde{K}$	feedback matrices
$M^{(k)}$	matrix defined in equation (16)
m	number of inputs
$m_a^{(k)},m_b^{(k)}$	vectors defined in equation (16)
N	matrix of $(k - 1)$ eigenvectors
n	number of states
P	matrix defined below equations (25)
$Q^{(k-1)}$	matrix defined by equation (18)
$R^{(k-1)}$	matrix defined by equation (14)
r	number of outputs in output feedback vector
$r \in \Delta^{(k)}$	implies r is an element of the set $\Delta^{(k)}$
S	matrix defined in equations (9)
t	time
u	input vector defined by equation (2)
$u \in R^m$	u is an $m \times 1$ real vector
u_p	external reference input
V	modal matrix (matrix of eigenvectors)
v_i,v_k	eigenvectors

w_i, w_k	vectors representing partitions of v_i and v_k, respectively
x	state vector defined by equation (1)
$x \in R^n$	x is an $n \times 1$ real vector
y	output vector defined by equations (23)
z_i, z_k	vectors representing partitions of v_i and v_k, respectively
$z_i \in R^m$	z is an $m \times 1$ real vector
$\Delta^{(k)}$	set of indices defined in equation (17)
δz_k	perturbed quantities
Λ	eigenvalue matrix
λ_i, λ_k	eigenvalues
σ, σ_k	scalar quantity indicating limiting value

Superscripts:

(1),(2),...,(k)	indicates an iterative step
T	matrix transpose
-1	matrix inverse

Subscripts:

o	initial value
ref	reference value

Uppercase letters denote matrices; matrix subscripts indicate partitioned quantities. A dot over a symbol denotes a derivative with respect to time; det is an abbreviation for determinant. A tilde ($\tilde{\ }$), caret ($\hat{\ }$), or bar ($\bar{\ }$) over a variable or matrix denotes a transformed quantity.

GENERAL DISCUSSION

Modal Synthesis

Srinathkumar (refs. 1, 2, and 3) both developed and programmed an algorithm for the synthesis of control systems by using the concept of eigensystem (eigenvalue/eigenvector) or pole placement. He derived constraining relations for calculating those feedback gains which will provide the desired modal characteristics of a linear system with multi-inputs. Although this analysis is described in detail in reference 1 and to a lesser degree in reference 2, multivariable synthesis by eigensystem assignment is briefly discussed herein.

The analysis begins with the controllable system

$$\dot{x} = Ax + Bu \tag{1}$$

where $x \in R^n$ and $u \in R^m$. Then, if B is assumed to be full rank, the state variable feedback law can take the form

$$u = Kx + u_p \tag{2}$$

where K is the feedback matrix, and u_p is an external reference input. Hence, the problem is to select a K so that the closed-loop system matrix A + BK satisfies

$$[A + BK]v_i = \lambda_i v_i \qquad (i = 1, 2, \ldots, n) \tag{3}$$

where λ_i is the ith eigenvalue, and v_i is the corresponding eigenvector.

Now, since m represents the number of inputs, equation (3) can be partitioned as

$$\left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} [K_1 \mid K_2] \right) \begin{Bmatrix} z_i \\ w_i \end{Bmatrix} = \lambda_i \begin{Bmatrix} z_i \\ w_i \end{Bmatrix} \qquad (i = 1, 2, \ldots, n) \tag{4}$$

where A_{11}, B_1, and K_1 are m × m matrices, and the remaining matrices are compatibly dimensioned. Moreover, $z_i \in R^m$, and

$$v_i^T = \left[z_i^T \mid w_i^T \right]$$

where z_i is the design mode shape (user-specified part of mode shape). Then, completing the multiplication of these partitioned quantities yields

$$[A_{11} + B_1K_1]z_i + [A_{12} + B_1K_2]w_i = \lambda_i z_i \tag{5}$$

$$[A_{21} + B_2K_1]z_i + [A_{22} + B_2K_2]w_i = \lambda_i w_i \tag{6}$$

Hence, it follows from equation (5) that

$$K_1z_i + K_2w_i = B_1^{-1}[\lambda_i z_i - A_{11}z_i - A_{12}w_i] \tag{7a}$$

$$K_1z_i + K_2w_i = B_2^{-1}[\lambda_iw_i - A_{22}w_i - A_{21}z_i] \tag{7b}$$

provided B_1 is either nonsingular or can be made nonsingular by reordering the state variables. That is, on combining equations (7a) and (7b) to eliminate K_1 and K_2,

$$[\lambda_iI_{n-m} - F]w_i = [G + \lambda_iS]z_i \qquad (i = 1, 2, \ldots, n) \tag{8}$$

where I_{n-m} is an (n - m)th-order identity matrix, and

$$\left.\begin{aligned} S &= B_2B_1^{-1} \\ G &= A_{21} - SA_{11} \\ F &= A_{22} - SA_{12} \end{aligned}\right\} \tag{9}$$

Equation (8) is the fundamental eigenvector-constraining relationship used by Srinathkumar in his algorithm for the synthesis of multi-input control systems. It constitutes a set of $n - m$ linear equations in n unknowns, each of which represents an element of an eigenvector. Then, if λ_i is not an eigenvalue of F, the $n - m$ elements of w_i are uniquely related to the remaining m eigenvector elements, each of which may be assigned arbitrarily. These assigned elements correspond to the z_i vector, while the $n - m$ elements which correspond to the w_i vector must be computed. That is,

$$w_i = C_iz_i \qquad (i = 1, 2, \ldots, n) \tag{10a}$$

where

$$C_i = [\lambda_iI_{n-m} - F]^{-1}[G + \lambda_iS] \tag{10b}$$

is defined as that modal coupling matrix corresponding to λ_i. Hence, after all n eigenvalues are assigned, up to nm eigenvector elements can also be arbitrarily assigned through the use of state variable feedback, provided the resulting modal matrix

$$V = \left[v_1 \mid v_2 \mid \cdots \mid v_n\right]$$

is nonsingular. The reader, however, must be aware that those choices of λ_i and z_i depend on the desired characteristics of the control system.

Equation (8) can be applied in a variety of ways. Aside from that option of synthesizing those eigenvectors achievable by state feedback, the analysis can be extended to the problem of output feedback. Moreover, equation (8) can be used to evaluate the achievability of certain preconceived objectives. For example, there are occasions where those eigenvalues and eigenvectors chosen to achieve certain design objectives will not satisfy those constraints imposed by equation (8) on the choices of λ_i and z_i. In these instances, it will be necessary to perturb slightly either λ_i or z_i to achieve these design objectives. The option which provides this capability is the spectral synthesis option. Equation (8) can also be used to establish the parametric equivalence between the nonunique feedback matrix K and the arbitrary modal entries. Each of these options is included within IMAGES and is described briefly in this paper. But, the reader is, once again, referred to references 1 and 2 for additional information.

The feedback matrix K is easily evaluated by returning to equation (3) or (4). If a nonsingular modal matrix V and an eigenvalue matrix Λ should be chosen to satisfy equation (8), then λ_i will not be an eigenvalue of F, and the closed-loop system matrix $\hat{A} = A + BK$ is uniquely determined by

$$\hat{A} = \left[\begin{array}{c|c} \hat{A}_{11} & \hat{A}_{12} \\ \hline \hat{A}_{21} & \hat{A}_{22} \end{array}\right] = V\Lambda V^{-1} \tag{11}$$

Moreover, that feedback matrix K required to yield this closed-loop matrix $\hat{A}$ is computed using the relations

$$\left.\begin{aligned} K_1 &= B_1^{-1}\left[\hat{A}_{11} - A_{11}\right] \\ K_2 &= B_1^{-1}\left[\hat{A}_{12} - A_{12}\right] \end{aligned}\right\} \tag{12}$$

where

$$K = \left[\begin{array}{c|c} K_1 & K_2 \end{array}\right]$$

Hence, given a modal matrix V and an eigenvalue matrix Λ, it is possible to evaluate the feedback matrix K, provided the matrices A and B are known.

Spectral Synthesis

The spectral synthesis option is more complicated. This option also assumes that one may be given those matrices A and B along with the desired eigenvalues λ_i and the eigenvector elements z_i. In this instance, however, it is required to synthesize a nonsingular modal matrix of eigenvectors, each of which not only satisfies equation (8) but also is mutually independent of the others. The latter condition is satisfied provided the kth eigenvector v_k is chosen so that

$$R^{(k-1)} v_k \neq v_k \tag{13}$$

where $R^{(k-1)}$ is the projector of that subspace spanning the previous (k - 1) eigenvectors. That is, if

$$N = \left[v_1 \;\vdots\; v_2 \;\vdots\; \cdots \;\vdots\; v_{k-1} \right]$$

represents the (k - 1) eigenvectors, then

$$R^{(k-1)} = N[N^T N]^{-1} N^T \tag{14}$$

represents the projector of that subspace spanning these eigenvectors, and equation (13) ensures a nonsingular modal matrix.

It is also essential that each eigenvalue λ_k not be an eigenvalue of F if equation (8) is to be satisfied. If the desired eigenvalue should not combine with F so that

$$\det[\lambda_k I_{n-m} - F] \neq 0 \tag{15a}$$

then

$$C_k = [\lambda_k I_{n-m} - F]^{-1} [G + \lambda_k S] \tag{15b}$$

is not defined. Spectral synthesis is an iterative procedure which avoids this difficulty by use of equation (8) to guarantee a solution for w_k which will combine with z_k so that v_k satisfies equation (13) and λ_k satisfies equation (15a). It requires that each eigenvector be evaluated sequentially and allows for perturbations within both λ_k and z_k.

The algorithm depends on constructing an n × n matrix $M^{(k)}$ such that

$$M^{(k)} = \left[\begin{array}{c|c|c} I_{n-r} & m_a^{(k)} & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & m_b^{(k)} & I_{r-1} \end{array} \right] \tag{16}$$

In equation (16), both $m_a^{(k)}$ and $m_b^{(k)}$ are vectors which must be evaluated, and the superscript (k) refers to the kth index chosen from a subset $\Delta^{(k)}$ of indices 1, 2, ..., n. These indices represent the corresponding values of λ_i and z_i, and the superscript represents the kth permutation of these indices. Hence, a particular index will no longer be resident within the subset $\Delta^{(k)}$ once it is used in the construction of $M^{(1)}$, $M^{(2)}$, ..., $M^{(k)}$.

Now, $M^{(k)}$ is defined so that equation (13) may be satisfied without evaluating $R^{(k-1)}$ explicitly. The construction is such that $v_k^{(k)}$ is transformed into canonical form by

$$M^{(k)} v_k^{(k)} = \sigma_k e_r \qquad (r \in \Delta^{(k)}) \tag{17}$$

where $\sigma_k \neq 0$, and e_r is an n vector with the rth entry equal to unity and all other entries equal to zero. Then, both the vectors $m_a^{(k)}$ and $m_b^{(k)}$ must be constructed so that

$$\sigma_k = v_{rk}^{(k)}$$

Hence, r is both the rth element of $v_k^{(k)}$ and the rth row of $M^{(k)}$. With these definitions, an n × m Q matrix is defined as

$$Q^{(k-1)} = M^{(k-1)} M^{(k-2)} \ldots M^{(1)} \qquad (k > 1) \tag{18}$$

where

$$Q^{(0)} = I_n \tag{19}$$

Hence,

$$v_k^{(k)} = Q^{(k-1)} v_k \tag{20}$$

The construction of $M^{(k)}$ and hence $Q^{(k)}$ is such that a nonsingular modal matrix is assured. The definition of $Q^{(k-1)}$ also permits considerable freedom in the choice of z_k and λ_k from the list of desired values denoted by z_i and λ_i. These values may be permuted at will to find the best combination of independent eigenvectors to achieve the desired design objectives. The procedure also permits perturbations within both z_k and λ_k to ensure that both equation (13) and equation (15a) are satisfied. These and other points should become clearer as the remaining steps are outlined.

The v_kth eigenvector is synthesized by using the rth row of equation (18). It is first required to compute

$$[g_r^{(k)}]^T = f_r^{(k-1)} + h_r^{(k-1)} C_k \tag{21}$$

where

$$\left[f_r^{(k-1)} \;\middle|\; h_r^{(k-1)} \right]$$

is the rth row of the transformation $Q^{(k-1)}$, and $h_r^{(k-1)}$ is an (n - m) row vector, while C_k is evaluated using equation (15b). Then,

$$\sigma_k = [g_r^{(k)}]^T z_k \tag{22}$$

which must not be equal to zero. (See eq. (17).) In the event $\sigma_k = 0$, the user must either select another r, perturb z_k to $z_k + \delta z_k$, or return to perturb λ_k once again. (See eqs. (15).) Normally, the order of application is as listed (i.e., select another r, etc.), with all remaining values of r being tried before perturbing either z_k or λ_k. Hence, on finding those values of z_k and λ_k which will satisfy those conditions specified, w_k can be evaluated by using equation (8). The results will be a v_k which is independent of those eigenvectors computed earlier. Moreover, subsequent eigenvectors computed in the same manner will also be independent. But, $M^{(k)}$ must be constructed from equation (17) before these steps are repeated for subsequent eigenvectors and eigenvalues.

Output Feedback Control System

The synthesis can be extended to an output feedback system. For example, consider systems of the form

$$\left.\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}\right\} \tag{23}$$

where y is an r × 1 output vector with $r \leqslant n$. Then, an approximation to the state feedback law (eq. (2)) can be found using the control law

$$u = \tilde{K}y + u_p \tag{24}$$

That is, on combining equations (23) and (24), the closed-loop system takes the form

$$\left.\begin{aligned} \dot{x} &= [A + B\bar{K}]x + \hat{B}u_p \\ y &= [C + D\bar{K}]x + \hat{D}u_p \end{aligned}\right\} \tag{25}$$

where $\bar{K} = P\tilde{K}C$, $\hat{B} = BP$, $\hat{D} = DP$, and $P = [I_m - \tilde{K}D]^{-1}$. Hence, P is the equivalent feedforward matrix, and K is the equivalent state feedback matrix. Then, for every state feedback matrix K derived by using the control law $u = Kx + u_p$, there exists an output feedback approximation

$$\tilde{K} = \bar{K}[C + D\bar{K}]^{\dagger} \tag{26}$$

where the superscript dagger (†) indicates the generalized inverse.

These algorithms are more general than is apparent from the preceding descriptions. Throughout the preceding discussion, the emphasis has been on real variables. It is, however, possible to extend the theory to complex variables. In fact, Srinathkumar has shown in references 1 and 2 that complex pairs can be treated using the same equations with very little modification. Those modifications have been included within the program and will not be discussed herein. Both real and complex eigenvalues and eigenvectors may be entered into the program without regard for order. The user, however, must be aware that the expected number of eigenvalues is diminished by one for each complex number entered. This information will be especially useful when running IMAGES.

PROGRAM DESCRIPTION

The digital computer program IMAGES takes advantage of the interactive capability of the modern computer complex in the modal analysis and gain estimation for control system synthesis. The emphasis is on simplicity with a minimum of demand on the user. "Control" variables and other decision-making logic have been reduced to a series of numbered lists from which the user must select one. Moreover, dynamic memory is utilized to eliminate any need to redimension any DIMENSION statement. These and other ideas have been combined with those algorithms of reference 1 as described herein to develop a truly interactive program with scratchpad capability.

The program was developed using that MODAL synthesis package written by Srinathkumar. That program, however, suffered greatly in two important points. First, it was based only on equations (1) through (12) and equations (23) through (25). This limitation eliminated spectral synthesis as a viable option within the program. As an interactive tool, this option may not only be useful but also essential. Second, MODAL was a card-oriented program (subroutine) possessing numerous options which required the user to resubmit the job for processing with each change in either options or model. The interactive program eliminates these time-consuming requirements.

Srinathkumar's program (or, more accurately, subroutine) was extremely flexible and well written. It was capable of estimating either eigenvectors, eigenvalues, or gains for either state feedback or output feedback control systems given those matrices indicated by either control system. (See eqs. (1) and (2) and eqs. (23) and (24).) Although the modal matrix would normally be synthesized by using equation (10), another option computes this matrix, along with its associated matrix of eigenvalues Λ, for the given system matrix A. The gain matrix K was estimated by using equation (11). It was also possible to compute K on entering both the modal matrix and its associated Λ matrix as input to the program; the gains would be estimated using both equations (10) and (11). Other options were also available to the user. But, the semimodular nature of the batch-oriented subroutine MODAL imposed a requirement by the control logic for several different codes, each of which could assume several different values.

Those code words used in MODAL along with their possible values are given in table I. As can be seen, one can be easily confused by the many possible combinations required to solve a particular problem. These codes, however, were necessary if the program was to be both efficient and useful in a batch computational mode. They are also essential to a lesser degree in the interactive program. They activate that control logic essential to provide the flexibility needed to accomplish many different objectives by using the same source code.

IMAGES uses a modified version of Srinathkumar's program. The interactive program does not require the user to provide that FORTRAN logic needed for a calling program. Neither is it required to maintain a close watch over returning output as well as an ever-changing data deck. Punched card input has been replaced by either keyboard or disk file entries. Control words and other variables associated with control logic are automatically assigned within IMAGES and are no longer the responsibility of the user. These objectives are accomplished through a series of question and answer diagnostics, which are described in greater detail later. These questions and answers determine which codes will be set along with the appropriate entry point within MODAL. This has been achieved by restructuring the existing subroutine by using entry points. Hence, the original subroutine MODAL has essentially been decomposed into four modules: (1) an input module, (2) a synthesis module, (3) a gain module, and (4) a response module. These modules are described graphically by the flow charts shown in figures 1 through 4. They represent the major entry points within the present version of MODAL.

These charts are somewhat self-explanatory. The various entries and operations have been grouped into those modules which best describe their functions. Addresses of selected READ's and other transfer points have also been included to aid the user who may have a listing of the program. (Both the program and the listing are available through COSMIC - Computer Software Management and Information Center, Suite 112, Barrow Hall, University of Georgia, Athens, GA 30602.) For example, the numbers 1030, 1040, 1037, and 1045 in the gain module (see fig. 3) indicate addresses of transfer points within MODAL. Numbers less than 10 on the figures are associated with those codes selected automatically by the various interactive responses. These responses are shown in figure 5 and are discussed later. Hence, that path taken within each module depends on the particular option chosen by the user.

One should not be confused by the subroutine names but should be continuously aware of the differences between modal and spectral synthesis. The spectral synthesis subroutine SPECTRL - which is called from the synthesis module within MODAL - was not included within the original version of the pole placement package. It is based on that discussion centered around equations (13) through (22) and depends on subroutine SYNTHS. The original subroutine was named to reflect its primary function of modal synthesis. But, with this addition of the spectral synthesis algorithm, both the spectral synthesis and the modal synthesis are options within the synthesis module. Hence, MODAL is now an executive subroutine, and SYNTHS is that subroutine on which both algorithms depend. Both algorithms require the user to provide m elements of each eigenvector along with its corresponding eigenvalue. Modal synthesis will not adjust these design values. But, spectral synthesis permits the user to perturb these values in such a way as to satisfy within a certain tolerance (denoted by σ) the condition of mutual independence.

The remaining subroutines will neither be described graphically nor discussed herein, since the program listing is not included as a part of this document. These subroutines are, however, listed in table II along with a brief description of their functions. Hence, the remaining discussion is restricted to operational considerations.

IMAGES is a user-oriented, interactive, digital computer program. Many different questions, messages, and other diagnostics are included within this program. The various matrices and other critical variables are printed on the terminal screen as they are generated, with sufficient pauses to permit the user to examine his or her results. In many instances, the user is permitted to modify these values. This is particularly true with those matrices and vectors serving as input. The user may also permute the A, B, C, and D matrices. These, as well as the several other options, provide considerable flexibility in the design and analysis of control systems.

The interactive discourse is given in the flow chart shown in figure 5. Most of these diagnostics are included within the executive program IMAGES and the executive subroutine MODAL. Other diagnostics are also present, but these are not considered to be essential to the discussion. That path taken by the program in response to a given command is indicated by its number. Clearly, the user has many different choices.

The program begins with the "master menu." It is indicated by the bold border in figure 5. All other blocks, with only one exception, are subordinate to this block and must ultimately return to it. Hence, each of those options within the master menu is a major program branch. Moreover, options within the submenus on one or more of those program modules are described elsewhere herein.

The user is essentially free to choose among the several options as needed. Each is, hopefully, self-explanatory, and the order of selection is usually unimportant. But, the user should be aware that the printed order of options is suggestive of that sequence of decisions a researcher would ordinarily make during the synthesis of an eigensystem for gain estimation. This is particularly true for new problems. The user must always "begin problem," "enter input," "enter computational mode," and "stop," or terminate, the program. The remaining options are utilities which enhance the overall flexibility and operational capability of the program. The "save problem" option, in particular, allows the user to automatically save all those program variables needed to restart the program in the solution for a particular problem at some future date without penalty. This is provided as a safety feature as well as for computational flexibility. The save option automatically files all the current program variables and other data onto STRTUP - a random-access mass-storage disk file - under a name chosen by the user for future reference. It is also provided within each of the two remaining branches: (1) enter input and (2) enter computational mode.

On beginning a problem, the user must first specify whether the problem is an old or a new problem and supply a file name, which can be as many as seven characters. (Note: A first-time user must first establish STRTUP as a viable storage file within the computer complex.) The program will then determine if the name has been used previously. If so, the user must decide if the name is to be retained. In the event his problem is to be a new example, the user may either purge the existing file or rename the new file. If the problem is an old one, the program will automatically find and load all those values associated with all steps completed up to the most recent save command using that file name.

System matrices can be reviewed, modified, or both reviewed and modified on entering the enter input module. The user may also choose to go directly to the computational mode. If the problem is a new one, however, the user must enter those matrices required for his problem. Diagnostics specify the expected format for these new data. Provision is also made for correcting the data. All these diagnostics are self-explanatory and follow in a natural sequence to minimize the demand on the user.

The user need not be concerned with array dimension statements. The program uses dynamic memory. Program dimensions and memory requirements are automatically allocated during execution. Moreover, they are automatically adjusted during a session to accommodate any other problem the user may choose to analyze. This capability is possible through the use of the Control Data Corporation (CDC) Common Memory Manager, which is a part of their operating system.

The Common Memory Manager is a set of several FORTRAN-callable subroutines, each of which begins with the prefix CMM. These subroutines return to the calling program the first word address (FWA) of any array name. Another returns the offset needed to reposition this array at the end of the program, while a third routine sets aside that memory needed to execute the program. A fourth routine releases this space, and a fifth routine extends it. These routines are extremely useful within a variable dimension program like IMAGES and are used extensively therein.

All array variables are listed in table III along with their memory assignments. This table provides the user with a quick index to each program variable and its position within the memory table. The variable name A is used extensively within both IMAGES and its executive subroutine MODAL. Other variables are listed in table III along with their algebraic equivalents. These variables can be correlated with those algebraic expressions discussed earlier.

The "computational mode" is much more complex than either of the other modes. It is the backbone of IMAGES. All computations are initiated from this module. Some options require the user to choose between other suboptions, as can be seen in figure 5. In some instances, the user is required to provide additional input. For example, the synthesis option will usually require the user to provide both the modal and the Λ matrices. In fact, the spectral synthesis returns to these inputs repeatedly until the specified tolerance σ is satisfied. The "response" option is another example where additional input is required. In this instance, the user must specify either a step input or initial condition input as well as other parameters such as time increments. Other options will simply provide the specified computational result. The user should be cautioned, however, to save needed results in all instances before entering the response option. In fact, the user should save results after each operation if it is desired to retain the data from completed steps for future use.

The response option is provided as an analysis aid only. This option will provide the user with a graphical display of system response to various types of input. These results may not be saved by the user. They are intended only as design aids. Hence, a particular design can be modified repeatedly until the desired response is obtained.

The program is designed to provide the user with considerable numerical results. Hence, automatic scrolling could present difficulties for some terminals. Line limitations vary considerably from terminal to terminal. This is controlled internally, to some extent, within the program by using a PAUSE command. This command stops execution until the user enters a carriage return. The user can employ this command to control the rate of execution. It also allows the user to examine the returning output. If the numerical results are not progressing as expected, the user can return to an earlier step to modify his input.

An example session is given in the appendix. It is not an example of a complete design problem. This example represents the end result of several sessions where the desired eigenvalues and eigenvectors have been refined to achieve the desired results. The user is encouraged to try this example. The program is extremely fast and requires less than 70K octal memory words to run. Response time is essentially instantaneous for most options. The various diagnostics will quickly become second nature with only a few practice sessions such as this. Moreover, it will provide a valuable check on program compatibility. The reader is referred to the references for that logic used in selecting desired eigenvalues and eigenvectors in a practical design problem.

CONCLUDING REMARKS

An interactive digital computer program for modal analysis and gain estimation through eigensystem synthesis has been described. The program is modular in design and is extremely flexible. It is capable of both modal and spectral synthesis of multi-input control systems. The numerous options along with the incorporation of both memory management and random access have been combined to produce a truly user friendly, scratchpad program for the design of multi-input control systems.

The program is very fast and requires less than 70K octal memory words to run. Some facilities impose limitations on programs to maximize the number of jobs which can be processed. Hence, these characteristics can be an added advantage to potential users.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
November 30, 1983

APPENDIX

EXAMPLE OF AN INTERACTIVE SESSION

```
 13.43.21.RETURN(IMAGES)
 13.43.22.ATTACH(FTNMLIB/UN=LIBRARY)
 13.43.26.REWIND(LMAP)
 13.43.27.LDSET(LIB=FTNMLIB,PRESETA=INDEF,MAP=SBEX/LMAP)


  MASTER MENU
   CHOOSE OPTION:
    1--BEGIN PROBLEM
    2--ENTER INPUT
    3--ENTER COMPUTATIONAL MODE
    4--SAVE PROBLEM
    5--RENAME FILE
    6--DUPLICATE FILE
    0--STOP

? 1
 THERE ARE   2 PROBLEMS ON FILE. THESE ARE AS FOLLOWS:

  SAMPLE              TEST


  IS THIS A NEW EXAMPLE?
  CHOOSE ONE:
    1--NEW
    2--OLD
    3--NEITHER
? 1


  ENTER SEVEN (7) OR LESS CHARACTER FILE NAME:
? sample


  PROBLEM NAME IS ON FILE!!
  DO YOU WISH TO PURGE?
    CHOOSE ONE:?
      1--YES
      2--NO

? 2


  IS THIS A NEW EXAMPLE?
  CHOOSE ONE:
    1--NEW
    2--OLD
    3--NEITHER
? 1
```

APPENDIX

```
  ENTER SEVEN (7) OR LESS CHARACTER FILE NAME:
? example



  MASTER MENU
   CHOOSE OPTION:
    1--BEGIN PROBLEM
    2--ENTER INPUT
    3--ENTER COMPUTATIONAL MODE
    4--SAVE PROBLEM
    5--RENAME FILE
    6--DUPLICATE FILE
    0--STOP

? 2


  PROBLEM IS A NEW ONE. ARRAYS MUST BE DIMENSIONED!
  DO YOU WANT A GRAPHIC DISPLAY OF MATRICES & THEIR DIMENSIONS
    CHOOSE ONE:
      1--YES
      2--NO

? 1
   ***
  *****
   ***


*       *               *     *       *           *****     *     *
 *     *                       *     *            *    *    *     *
  *   *                * *      *   *       *     *    *    *     *
   * *      ******               * *        *     *   *     *     *
    *                 *   *       *       *****   ****      *     *
   * *      ******    *****      * *        *     *   *     *     *
  *   *              *     *    *   *       *     *    *    *     *
 *     *                       *     *            *    *    **   **
*       *           *       * *       *           *****      *****



  N X 1              N X N      N X 1              N X M     M X 1
                     SYSTEM     STATE              INPUT     INPUT
 VECTOR              MATRIX    VECTOR              MATRIX    VECTOR


  PLEASE ENTER THE NUMBER OF STATE ELEMENTS, N.
? 4
```

APPENDIX

THE CONTROL LAW MAY BE EITHER AN INPUT OR AN OUTPUT CONTROL:

```
*     *             *     *   *       *           *     *
*     *             *    *     *     *            *     *
*     *             *   *       *   *       *     *     *
*     *     ******  *  *         * *        *     *     *
*     *             * *           *       *****   *     *
*     *     ******  ** *         * *        *     *     *
*     *             *   *       *   *       *     *     *   ****
**   **             *    *     *     *            **   **   *   *
 *****              *     *   *       *            *****    *   *
                                                            ****
                                                            *
                                                            *
  M X 1                M X N     N X 1               M X 1   *
  INPUT              FEEDBACK    STATE               INPUT
  VECTOR              MATRIX    VECTOR               VECTOR
```

OR

```
*     *             *     *   *       *           *     *
*     *             *    *     *     *            *     *
*     *             *   *       *   *       *     *     *
*     *     ******  *  *         * *        *     *     *
*     *             * *           *       *****   *     *
*     *     ******  ** *          *         *     *     *
*     *             *   *         *         *     *     *   ****
**   **             *    *        *               **   **   *   *
 *****              *     *       *                *****    *   *
                                                            ****
                                                            *
                                                            *
  M X 1             M X (R =N)(R =N) X 1            M X 1   *
  INPUT              FEEDBACK   OUTPUT              INPUT
  VECTOR              MATRIX    VECTOR              VECTOR
```

BUT, THE NUMBER OF INPUTS WILL REMAIN UNCHANGED.

APPENDIX

```
  PLEASE ENTER THE NUMBER OF INPUTS, M.


? 2



  PLEASE ENTER THE NUMBER OF OUTPUTS, R.


? 4


  DESCRIBE YOUR INPUT (CHOOSE ONE)
    1--THE SYSTEM MATRICES A & B
    2--OUTPUT MATRICES (I.E., Y=CX+DU)
    3--BOTH "1" &"2"
    4--EIGENVALUE/EIGENVECTOR MATRICES
    5--MODAL COUPLING MATRIX
    6--ENTER COMPUTATIONAL MODE
    7--SAVE INPUT
    8--PRINT A & B MATRICES
    9--PRINT C & D MATRICES
    0--EXIT

? 1

                         ...EIGENVALUE/EIGENVECTOR ASSIGNMENT ALGORITHM     ...


           ...SYSTEM PARAMETERS...

                NO OF STATES =   4

                NO OF INPUTS =   2


  READ SYSTEM MATRICES A AND B
  ENTER A MATRIX
  ENTER BY ROWS:


? -.56758,.859871,-2.192935,0.0
? -.006922,-.147531,.230069,0.0
? .195034,-.970592,-.079899,.123380
? 1.0,.140541,0.0,0.0
  ENTER B MATRIX
  ENTER BY ROWS:
```

```
? .829520,.210565
? -.013460,-.115903
? -.006502,.0110501
? 0.0,0.0

     : A : MATRIX



            -.567580          .859871      -2.192935        0.000000

            -.006922         -.147531        .230069        0.000000

             .195034         -.970592       -.079899         .123380

            1.000000          .140541       0.000000        0.000000


PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?

     : B : MATRIX



             .829520          .210565

            -.013460         -.115903

            -.006502          .011050

            0.000000         0.000000


PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?
```

APPENDIX

```
  DO YOU LIKE A & B?
   CHOOSE ONE:
     1--YES
     2--NO

 ? 1
   DO YOU WISH TO PERMUT A & B MATRICES?
     CHOOSE ONE:
       0--NO
       1--YES

 ? 0


   DESCRIBE YOUR INPUT (CHOOSE ONE)
     1--THE SYSTEM MATRICES A & B
     2--OUTPUT MATRICES (I.E., Y=CX+DU)
     3--BOTH "1" &"2"
     4--EIGENVALUE/EIGENVECTOR MATRICES
     5--MODAL COUPLING MATRIX
     6--ENTER COMPUTATIONAL MODE
     7--SAVE INPUT
     8--PRINT A & B MATRICES
     9--PRINT C & D MATRICES
     0--EXIT

 ? 7


   DESCRIBE YOUR INPUT (CHOOSE ONE)
     1--THE SYSTEM MATRICES A & B
     2--OUTPUT MATRICES (I.E., Y=CX+DU)
     3--BOTH "1" &"2"
     4--EIGENVALUE/EIGENVECTOR MATRICES
     5--MODAL COUPLING MATRIX
     6--ENTER COMPUTATIONAL MODE
     7--SAVE INPUT
     8--PRINT A & B MATRICES
     9--PRINT C & D MATRICES
     0--EXIT

 ? 0


   MASTER MENU
    CHOOSE OPTION:
     1--BEGIN PROBLEM
     2--ENTER INPUT
     3--ENTER COMPUTATIONAL MODE
     4--SAVE PROBLEM
     5--RENAME FILE
     6--DUPLICATE FILE
     0--STOP
```

APPENDIX

```
  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 1
  CURRENT EPS IS  0.              WHAT IS ITS NEW VALUE?
? 0.000001


     : B1 INVERSE :



            1.242133       2.256626


            -.144251      -8.889970



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?


     :DETERMINANT OF B1 =     -.93310E-01



     : S : MATRIX



            -.009670       -.112908


            0.000000       0.000000



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?
```

```
     : F : MATRIX



            .075129       -.123380


           0.000000        0.000000



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?


     : G : MATRIX



            .188764       -.978934


           1.000000         .140541



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?


  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 5
```

APPENDIX

```
  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 2


  CHOOSE ONE:
    1--SPECTRAL SYNTHESIS
    2--MODAL SYNTHESIS
    3--EXIT

? 2
    CHOOSE ONE:
      0--MODAL MATRIX
      1--MODAL COUPLING MATRIX

? 0
SYSTEM MODES (AND THE OPTIONAL MODAL MATRIX) ARE REQUIRED.
                  ENTER DESIRED MODES



   A MAXIMUM OF   4 REAL MODES ARE EXPECTED
COMPLEX MODES WILL AUTOMATICALLY DIMINISH THIS NUMBER


ENTER EACH IN TABLE FORM WITH COMMA (,) SEPARATORS:


          TYPE     ,     REAL PART     ,     IMAG PART

SPACING IS NOT IMPORTANT AND TYPE CODES ARE AS FOLLOWS:
0--IMPLIES REAL MODE;  1--IMPLIES COMPLEX MODE;  2--ENDS INPUT


? 0,-.72,0.0
? 1,-.15,.4
? 0,-.031,0.0
```

```
                 DESIRED MODES

          TYPE          REAL PART          IMAG PART


           0           -.7200E+00         0.

           1           -.1500E+00          .4000E+00

           0           -.3100E-01         0.

   CHOOSE ONE:
     1--CORRECT MODES
     2--CONTINUE

 ? 2


  ENTER THE M X N DESIGN MATRIX " Z " BY ROWS


   DO YOU WANT A GRAPHIC DISPLAY OF MATRICES & THEIR DIMENSIONS
     CHOOSE ONE:
       1--YES
       2--NO

 ? 1
                                                                       *****
                                                                         *
                                                                         *
                    *****                                    *****       *
                    *                                            *       *
*       *           *         *********            *        *    *       *
                    *                *      **                   *
 *     *            *               *       **                   *
            ******  *              *                *      *     *
  *   *             *             *                     *        *
            ******  *            *                               *
   * *              *           *           **       * * * *     *
                    *          *            **                   *
    *               *         *********               *   *      *
                    *                                            *
                    *****                                    *****

   N X 1                          M X 1                  (N-M) X 1
  EIGEN-                        SPECIFIED                ASSIGNED
  VECTOR                        ELEMENTS                 ELEMENTS
  ENTER ROW   1
? 1.0,0.0,0.0,-.045841
  ENTER ROW  2
? 0.0,.5,.5,.10927
```

```
          DESIGN PARAMETER MATRIX " Z "



             1.000000        0.000000        0.000000        -.045841


             0.000000         .500000         .500000         .109270



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?
  CHOOSE ONE:
    1--CORRECT  "Z"
    2--CONTINUE
? 2





  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 5


  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 3
```

APPENDIX

```
  FEEDBACK GAINS
    CHOOSE ONE:
      1--ENTER STATE FEED BACK GAINS
      2--CALCULATE FEEDBACK GAINS
      3--EXIT

? 2


  CHOOSE SYSTEM:
    1--STATE FEEDBACK
    2--OUTPUT FEEDBACK

? 1

     MATRIX OF EIGENVALUES



             -.720000       0.000000       0.000000       0.000000

             0.000000       -.150000        .400000       0.000000

             0.000000       -.400000       -.150000       0.000000

             0.000000       0.000000       0.000000       -.031000


PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?

     : V : MATRIX (MODAL MATRIX)



              .584162       0.000000       0.000000       -.045841

             0.000000        .269093        .269093        .109271

             -.022072       -.579930        .709355        .137647

             -.811337        .051806       -.113974        .983368
```

APPENDIX

```
PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?


DETERMINANT OF : V : =        .19181E+00

     : V-INVERSE :



           1.827418        .007014        .010662        .082917

           -.203078       2.034071       -.791655       -.124680

           -.394926       1.645819        .736485       -.304382

           1.472654        .089380        .135863       1.056615

PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?

                              NUMERICAL CONDITION OF SOLUTION
                              (NORM OF RECIPROCAL VECTORS)

                        MODE                     CONDITION NUMBER

            -.72000E+00(+/-) J    0.               .18293E+01
            -.15000E+00(+/-) J     .40000E+00      .28846E+01
            -.31000E-01(+/-) J    0.               .18198E+01

      : A + BK : MATRIX



             -.766513       -.002823       -.004291       -.033373

             -.001501       -.190628        .166251       -.005603

              .196346       -.957383       -.093858        .124335

             1.000000        .140541        .000000        .000000
```

```
PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?



      : K : MATRIX



              -.234868      -1.168835       2.574575       -.054098

              -.019499        .507576        .251622        .054626



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?

      : ERROR MATRIX :



              -.000000        .000000        .000000       -.000000

              -.000000       0.000000        .000000        .000000



PROGRAM EXECUTION IS TEMPORARILY INTERRUPTED.PRESS  'ENTER'  TO CONTINUE EXECUTION.
   PAUSE
?

          ...MESSAGE ...

              IF ELEMENTS OF ERROR MATRIX HAVE NOT APPROACHED ZERO
              SIGNIFICANT ROUND OFF ERRORS HAVE OCCURRED IN COMPUTATION.
              PROBABLE CAUSES ...1) MATRIX B1 IS ILL CONDITIONED
                                   TRY PERMUTING SYSTEM MATRICES TO GENERATE BETTER B1 MATRIX

                                  2) MODAL  MATRIX IS ILL CONDITIONED
                                   CHANGE DESIGN VECTORS
```

APPENDIX

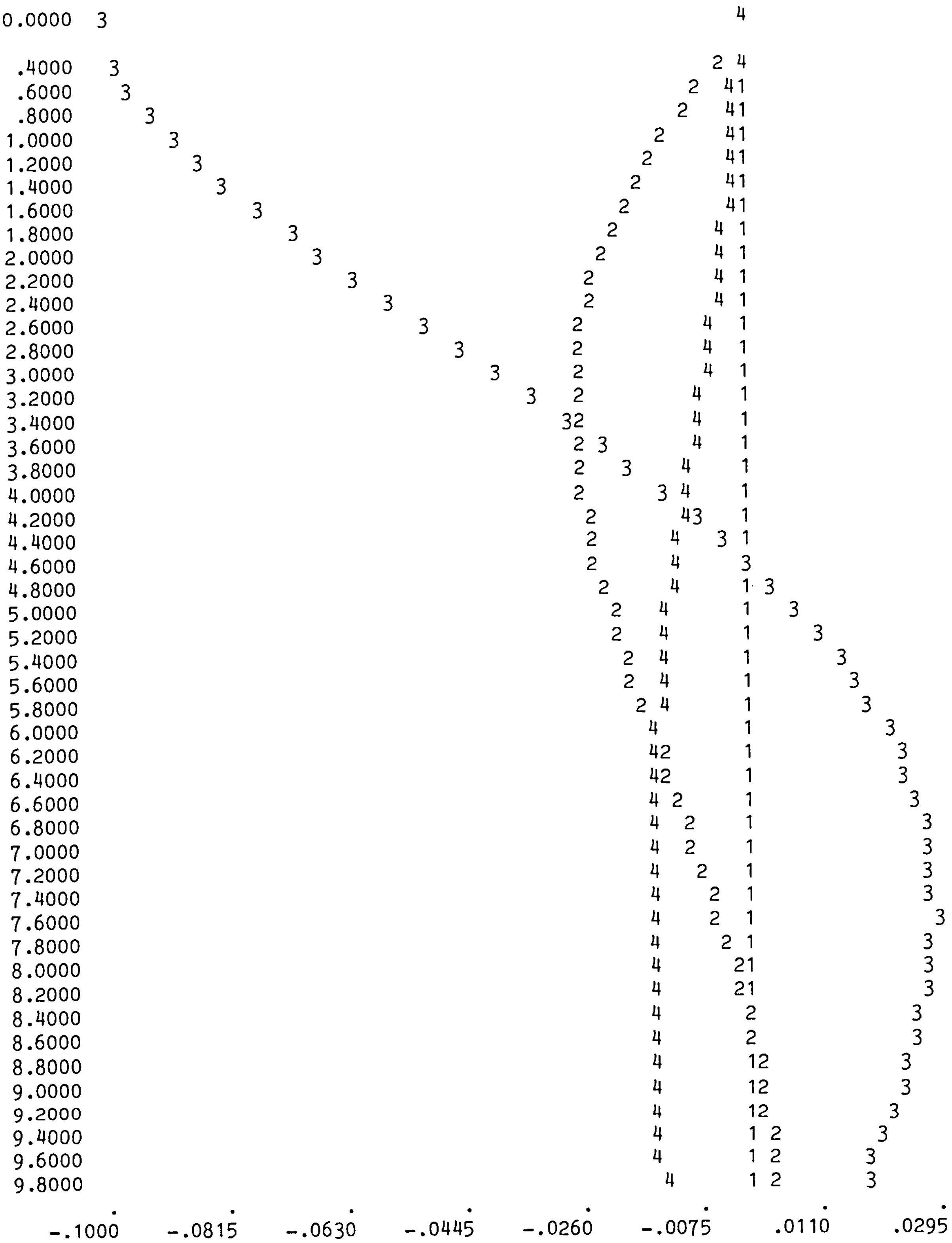
```
  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 5


  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 4


  CHOOSE OPTION:
    0--EXIT
    1--STATE RESPONSE
    2--STATE+CONTROL RESPONSE
    3--STATE+OUTPUT RESPONSE
    4--STATE+OUTPUT+CONTROL RESPONSE

? 1


  ENTER THE NUMBER OF TIME STEPS:
? 50
  ENTER NUMBER OF OUTPUT RESPONSES, R
? 4
WHAT RUN NUMBER IS THIS?
? 1
ENTER THE TIME INCREMENT
? .2
CHOOSE THE DESIRED TYPE OF RESPONSE:
  0--INITIAL CONDITION RESPONSE
  1--STEP RESPONSE
? 0
ENTER THE "  4" X "  1" INITIAL STATE VECTOR
? 0.0,0.0,-.1,0.0
```

APPENDIX

```
1                                                        STATE VARIABLE RESPONSE

         TIME RESPONSE PARAMETERS


         NO OF STEPS =  50
         TYPE OF RESPONSE =   0
         TIME INCREMENT =  .200     SECONDS


                                                       ...INITIAL CONDITION RESPONSE ...


              ... I.C. VECTOR ...


                   0.000        0.000         -.100        0.000
```

APPENDIX

CHART 1



```
  CHOOSE OPTION:
    0--EXIT
    1--STATE RESPONSE
    2--STATE+CONTROL RESPONSE
    3--STATE+OUTPUT RESPONSE
    4--STATE+OUTPUT+CONTROL RESPONSE

? 0


  COMPUTATIONAL MODE:
    CHOOSE ONE--
      1--CONSTRAINING RELATION
      2--SYNTHESIS
      3--GAINS
      4--TIME RESPONSE
      5--SAVE
      6--EXIT

? 6


  MASTER MENU
   CHOOSE OPTION:
    1--BEGIN PROBLEM
    2--ENTER INPUT
    3--ENTER COMPUTATIONAL MODE
    4--SAVE PROBLEM
    5--RENAME FILE
    6--DUPLICATE FILE
    0--STOP

? 0
 HAS THE FILE BEEN SAVED?
 (YES/NO)
? yes
 14.16.55.REWIND(LGO)
 14.16.56.RETURN(FTNMLIB)
 14.16.57.REPLACE,STRTUP.
```

REFERENCES

1. Srinathkumar, S.: Spectral Characterization of Multi-Input Dynamic Systems. Ph. D. Diss., Oklahoma State Univ., 1976.

2. Srinathkumar, S.: Modal Control Theory and Application to Aircraft Lateral Handling Qualities Design. NASA TP-1234, 1978.

3. Srinathkumar, S.: Eigenvalue/Eigenvector Assignment Using Output Feedback. NASA TP-1118, 1978.

TABLE I.- CONTROL WORDS AND POSSIBLE VALUES

Name	Function or description	Possible values
EPS	Tolerance parameter	If $\lvert \det(A) \rvert \leq$ EPS, the matrix A is assumed to be singular
IENTRY	Control parameter	-2 - Computes time response for a specified V and Λ, -1 - Computes eigenvalues/eigenvectors of matrix A, numerical condition (sensitivity) of modal matrix, and time response 0 - Normal setting for both modal and spectral synthesis 1 - Normal setting for gain estimation 2 - Setting for feedback synthesis
INPUT	Controls type of response	0 - Initial condition response 1 - Step response
IPRMUT	Reorders state variables to improve numerical condition of B_1 matrix	0 - No 1 - Yes
IPUNCH	Controls print	0 - No 1 - Yes
ITSOLN	Controls time response	0 - No time response 1 - State variable response 2 - State + control response 3 - State + output response 4 - State + control + output response
MODE(I)	Type of mode (complex modes occur in pairs)	0 - Real mode 1 - Complex mode
NPHASE	Control parameter	-2 - Permutes A & B and C & D -1 - Computes B_1^{-1}, F, S, and G matrices 0 - Selects modal synthesis 1 - Computes modal coupling matrices C_i for each specified mode λ_i 2 - Selects spectral synthesis
NRUN	Identification number	

TABLE II.- DESCRIPTION OF SUBROUTINES

Name	Function or description
CONDN	Computes numerical ill-conditioning (sensitivity) of modal matrix V
DIAG	System routine for generating the quasi-diagonal equivalent of the eigenvalue and eigenvector matrices
EIG	System routine for finding eigenvalues and eigenvectors
ENT	Reads a matrix
LUDECP & SUBST	Solves a system of linear equations
MATEQN	Inverts a square matrix
MINV	System routine for inverting matrices
MULT	Multiplies two matrices
NORM	Normalizes a vector
OGAIN	Computes the output feedback gains approximation $(\bar{K})$ to a state feedback design (K)
PERMUT	Permutes rows and columns of a matrix
PLOT	System routine for generating printed plots
PRT	Prints a matrix
SGAIN	Computes the equivalent state feedback gains (K) for an output feedback design
SPECTRL	Interactive subroutine for eigensystem placement
SYNTHS	Computes modal coupling matrices and eigenvectors for specified λ_i and z_i $(i = 1, 2, \ldots, n)$
TFUNC	Evaluates $e^{\Lambda t}$
TRANSF	Computes the closed-loop system after applying output feedback
USOLN	Computes actuator response $u = K_x + u_{ref}$
XSOLN	Computes state variable response time histories for $x(t) = Ve^{\Lambda t}V^{-1}x(0)$
YSOLN	Computes designated output response $y_o = C_o x + D_o u$

TABLE III.- ARRAY NAMES, ALGEBRAIC EQUIVALENTS, AND MEMORY ASSIGNMENTS

Variable	Algebraic equivalent	Memory assignment		
		First word address	Length	Upper limit
A	A	IOFF	N × N	I1
AE		I13	N × N	I14
AHAT	A	I12	N × N	I13
B	B	I1	N × M	I2
B1	B_1	I2	M × M	I3
B2	B_2	I3	L × M	I4
BINV	B_1^{-1}	I4	M × M	I5
BTEMP		I5	N × M	I6
C	C	I17	N × N	I18
CCT	$\mathrm{Im}\{C_i\}$	I47	LL × MM	I48
CH		I57	K × N	I58
CI		I59	K × K	I60
CM	$\mathrm{Im}\{\lambda_i\}$	I36	NM	I37
CMK	$\mathrm{Im}\{M^{(k)}\}$	I78	KK × KK	I79
CO		I50	K × N	I51
CP		I42	LL × LL	I43
CQ		I43	LL × MM	I44
CT		I58	K × K	I59
CW		I62	N × K	I63
CY		I69	NO × N	I70
D	D	I9	N × N	I10
DG		I55	K × N	I56
DO		I51	K × M	I52
DT		I56	K × M	I57
E		I14	L × N	I15
EI		I75	N	I76
ER		I74	N	I75
F		I15	N × N	I16
FC		I85	2 × M	I86
FR	$f_r^{(k-1)}$	I92	N	I93
FRC		I88	2 × NM	I89
G	K	I18	M × N	I19
G1	K_1	I19	M × M	I20
G2	K_2	I20	M × L	I21
GH		I52	M × M	I53
GO		I53	M × K	I54
GOT		I54	M × K	I55
GRT	$[g_r^{(k)}]^T$	I81	L	I84
GRTC		I87	2 × KK	I88
GT		I21	M × N	I22

TABLE III. - Continued

Variable	Algebraic equivalent	Memory assignment		
		First word address	Length	Upper limit
H		I16	N × N	I17
HC		I93	2 × NM	I94
HR	$h_r^{(k-1)}$	I83	NM	I84
HRC		I89	2 × NM	I90
IK		I60	K	I61
IL		I41	L	I42
ILL		I48	LL	I49
IM		I24	M	I25
IN		I22	N	I23
INO		I71	NO	I72
IRCCK		I84	2L	I85
IRCK		I82	L	I83
JK		I61	K	I62
JLL		I49	LL	I50
JM		I25	N	I26
JN		I23	N	I24
MODE		I34	NM	I35
Q		I79	L × L	I80
QC		I80	KK × KK	I81
QR	$Q^{(k-1)}$	I91	N × N	I92
R	F	I6	L × L	I7
RE	$\mathrm{Re}\{\lambda_i\}$	I35	NM	I36
RM		I90	N × N	I91
RMK	$\mathrm{Re}\{M^{(k)}\}$	I77	L × L	I78
RRT	$\mathrm{Re}\{Q_2\}$	I40	L × M	I41
S	S	I7	L × M	I8
SA	A_{11}	I30	M × M	I31
SB	A_{21}	I31	M × L	I32
SC		I32	L × M	I33
SD		I33	L × L	I34
T	G	I8	L × M	I9
U		I10	N × N	I11
UC		I65	M	I66
UO		I64	M	I65
UPLT		I67	LU	I68
UREF		I72	M	I73
V		I11	N × N	I12
VC		I86	NN	I87
WC	$\mathrm{Im}\{w_i\}$	I46	MM	I47
WK		I76	NN	I77
WR	$\mathrm{Re}\{w_i\}$	I38	L	I39

TABLE III.- Concluded

Variable	Algebraic equivalent	Memory assignment		
		First word address	Length	Upper limit
X		I26	N	I27
XI		I29	N	I30
XO		I27	N	I28
XPLT		I66	LX	I67
XR		I28	N	I29
YC		I44	LL	I45
YO		I73	NO	I74
YPLT		I68	LY	I69
YR		I37	L	I38
YU		I70	NO	I71
Z		I63	N	I64
ZC	$\mathrm{Im}\{z_i\}$	I46	MM	I47
ZR	$\mathrm{Re}\{z_i\}$	I39	M	I40

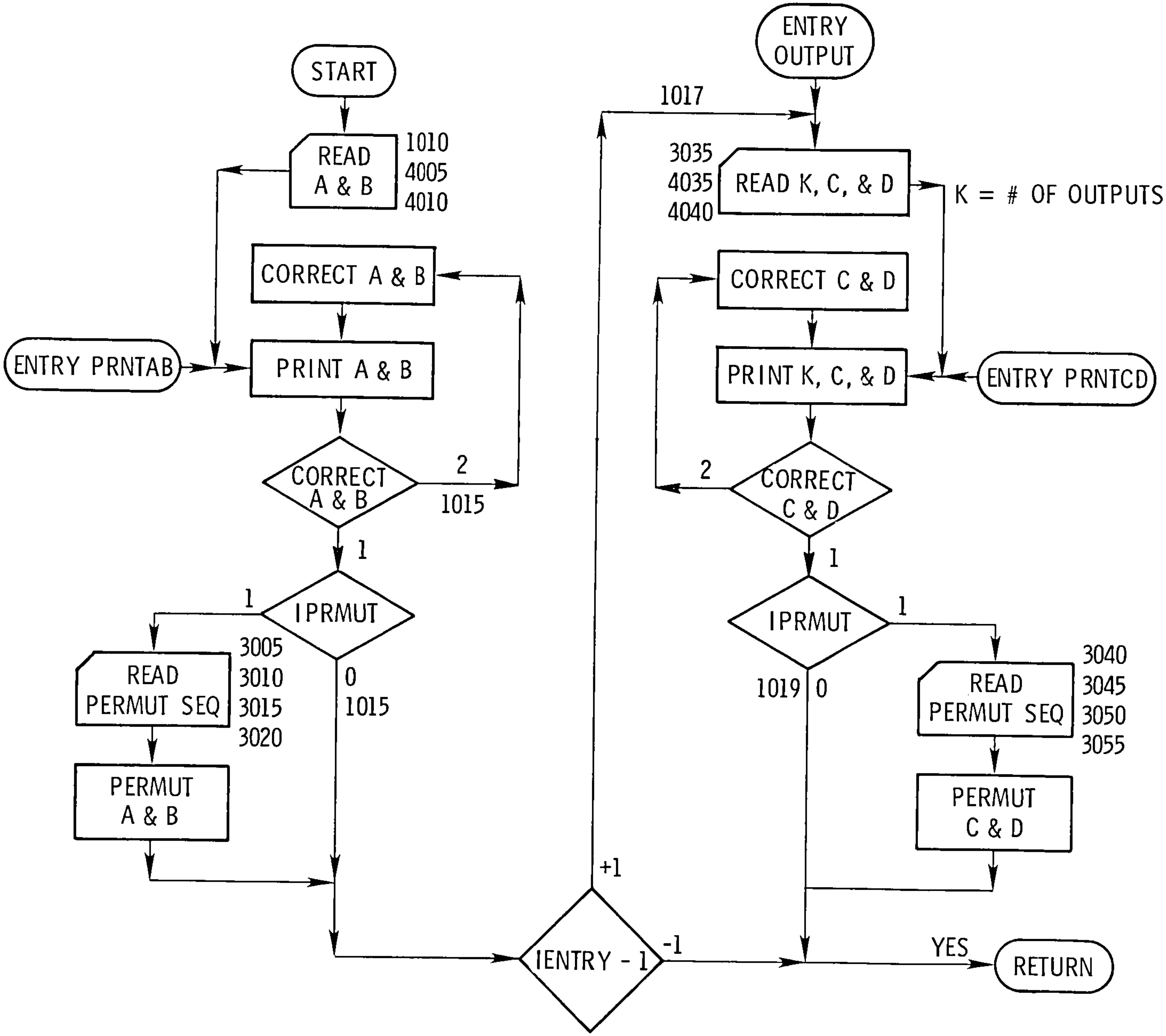


Figure 1.- Flow chart for input module.

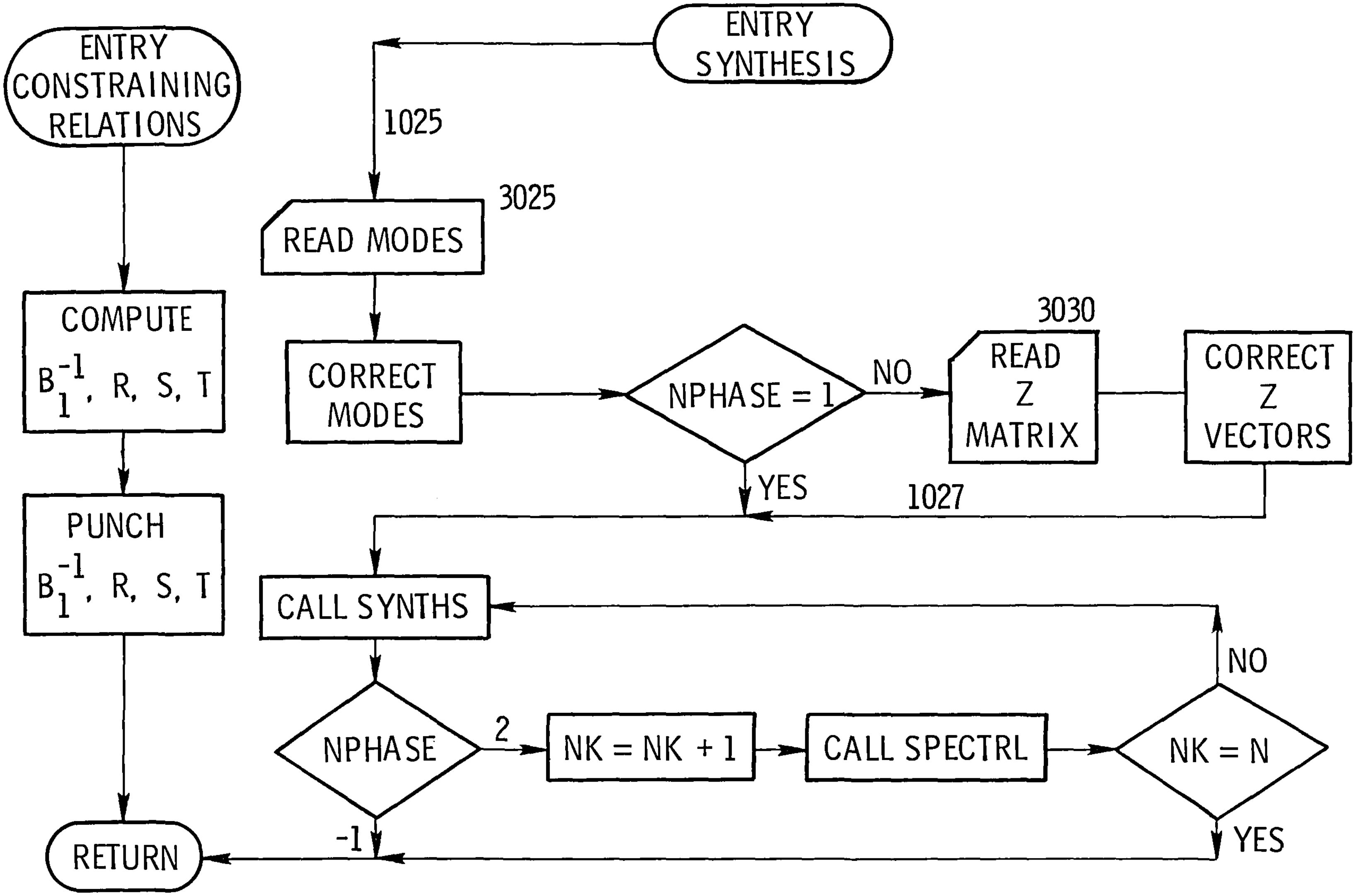


Figure 2.- Computational flow for eigensystem synthesis.

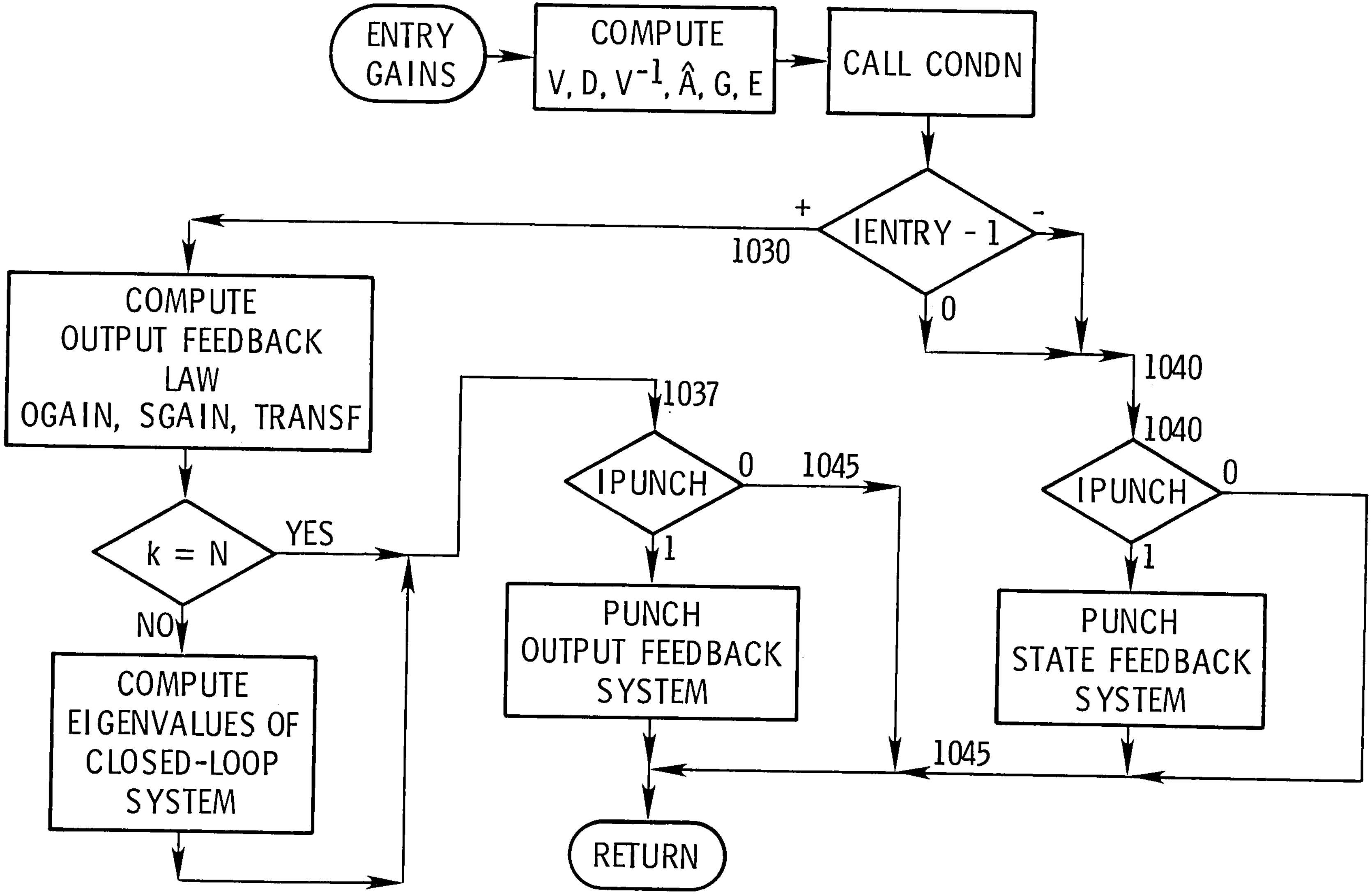

ENTRY GAINS
COMPUTE V, D, V^{-1}, $\hat{A}$, G, E
CALL CONDN
IENTRY - 1
+
-
0
1030
1040
COMPUTE OUTPUT FEEDBACK LAW OGAIN, SGAIN, TRANSF
k = N
YES
NO
COMPUTE EIGENVALUES OF CLOSED-LOOP SYSTEM
1037
IPUNCH
0
1045
1
PUNCH OUTPUT FEEDBACK SYSTEM
1040
IPUNCH
0
1
PUNCH STATE FEEDBACK SYSTEM
1045
RETURN

Figure 3.- Computational flow for gain estimation.

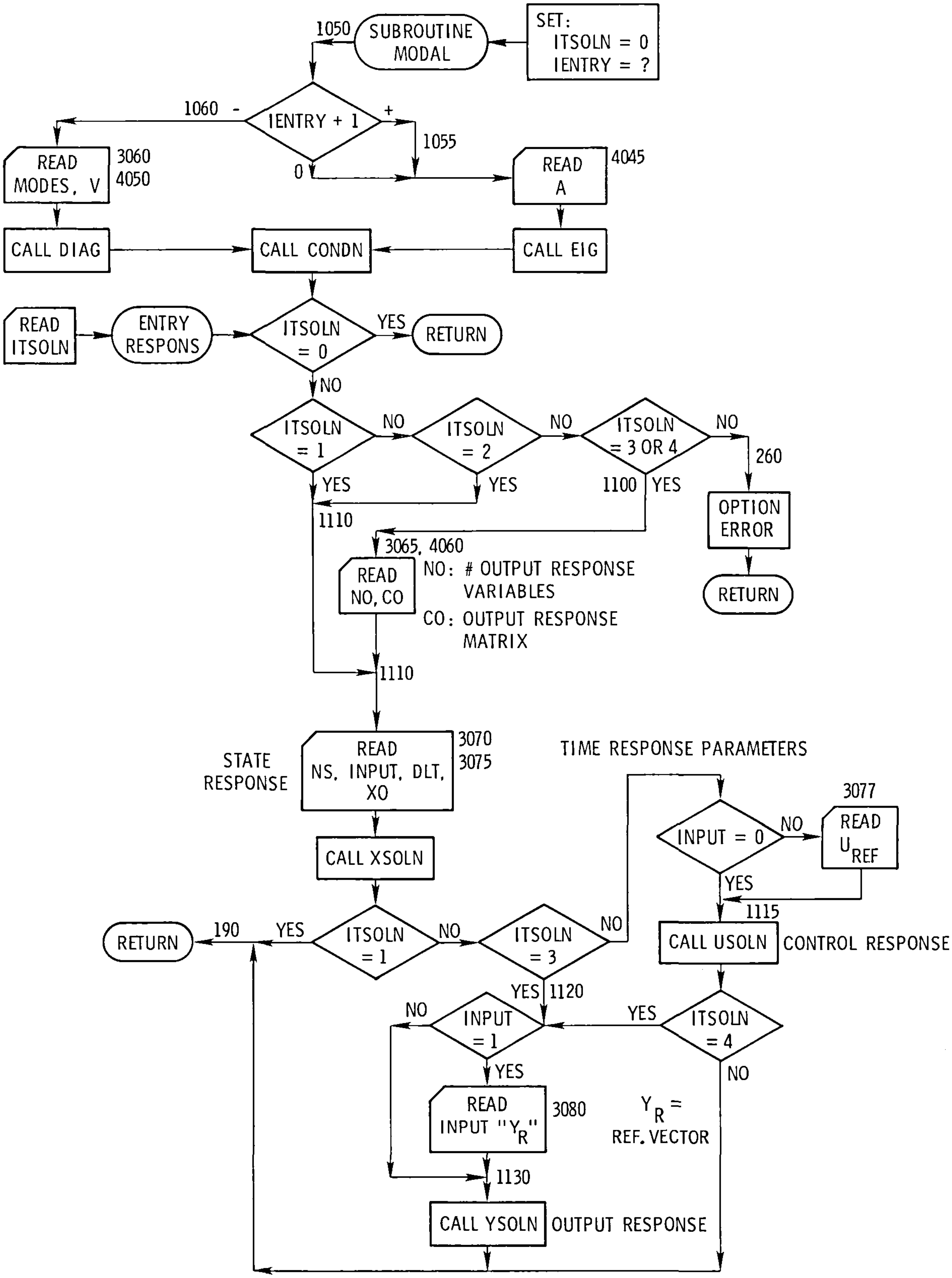


Figure 4.- Flow of response computations.

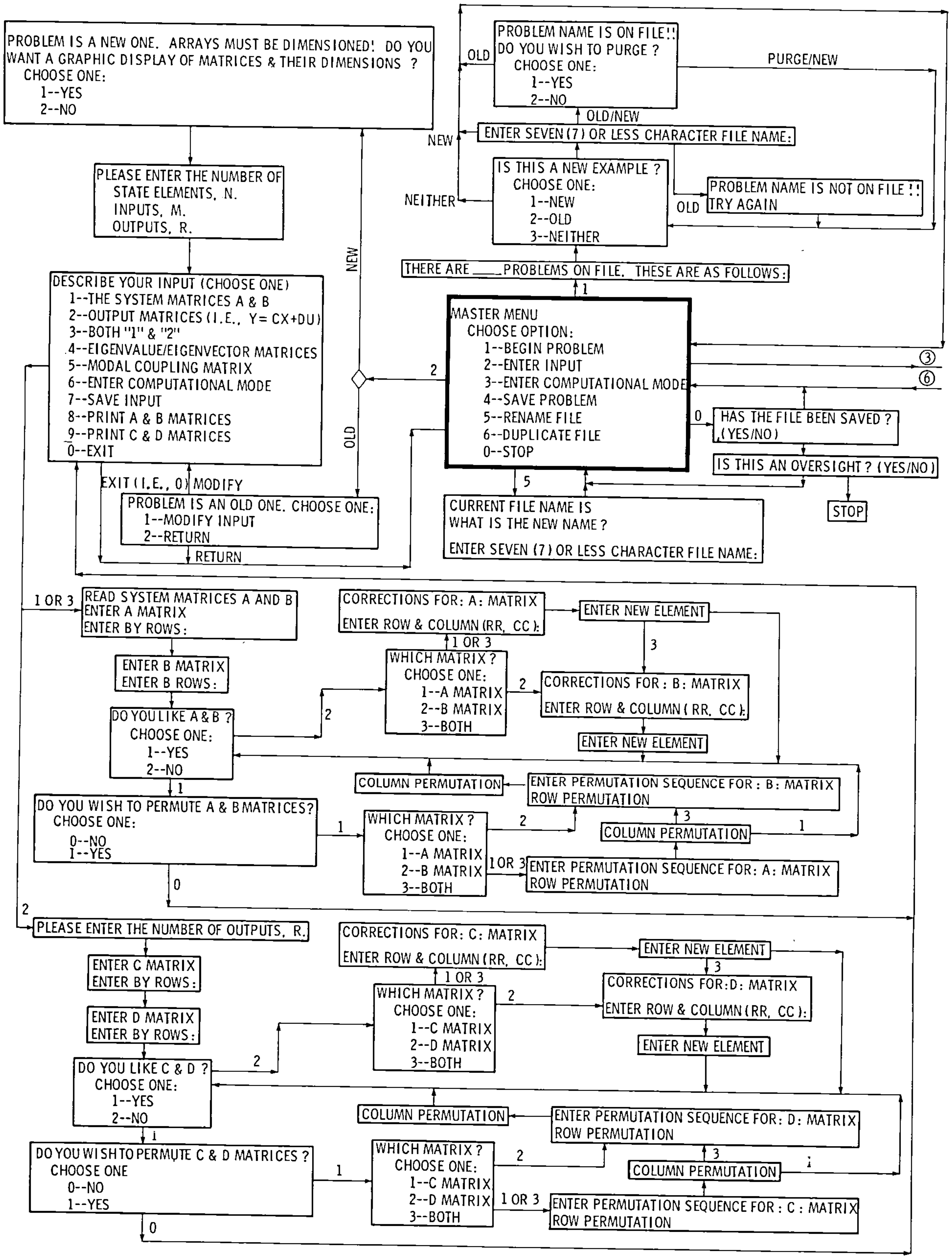

PROBLEM IS A NEW ONE. ARRAYS MUST BE DIMENSIONED! DO YOU WANT A GRAPHIC DISPLAY OF MATRICES & THEIR DIMENSIONS ? CHOOSE ONE: 1--YES 2--NO
PLEASE ENTER THE NUMBER OF STATE ELEMENTS, N. INPUTS, M. OUTPUTS, R.
DESCRIBE YOUR INPUT (CHOOSE ONE) 1--THE SYSTEM MATRICES A & B 2--OUTPUT MATRICES (I.E., Y= CX+DU) 3--BOTH "1" & "2" 4--EIGENVALUE/EIGENVECTOR MATRICES 5--MODAL COUPLING MATRIX 6--ENTER COMPUTATIONAL MODE 7--SAVE INPUT 8--PRINT A & B MATRICES 9--PRINT C & D MATRICES 0--EXIT
NEW
OLD
EXIT (I.E., 0)
MODIFY
PROBLEM IS AN OLD ONE. CHOOSE ONE: 1--MODIFY INPUT 2--RETURN
RETURN
PROBLEM NAME IS ON FILE!! DO YOU WISH TO PURGE ? CHOOSE ONE: 1--YES 2--NO
OLD
PURGE/NEW
OLD/NEW
NEW
ENTER SEVEN (7) OR LESS CHARACTER FILE NAME:
IS THIS A NEW EXAMPLE ? CHOOSE ONE: 1--NEW 2--OLD 3--NEITHER
NEITHER
OLD
PROBLEM NAME IS NOT ON FILE !! TRY AGAIN
THERE ARE ____PROBLEMS ON FILE. THESE ARE AS FOLLOWS:
1
MASTER MENU CHOOSE OPTION: 1--BEGIN PROBLEM 2--ENTER INPUT 3--ENTER COMPUTATIONAL MODE 4--SAVE PROBLEM 5--RENAME FILE 6--DUPLICATE FILE 0--STOP
2
③
⑥
0
HAS THE FILE BEEN SAVED ? (YES/NO)
IS THIS AN OVERSIGHT ? (YES/NO)
STOP
5
CURRENT FILE NAME IS WHAT IS THE NEW NAME ? ENTER SEVEN (7) OR LESS CHARACTER FILE NAME:
1 OR 3
READ SYSTEM MATRICES A AND B ENTER A MATRIX ENTER BY ROWS:
ENTER B MATRIX ENTER B ROWS:
DO YOU LIKE A & B ? CHOOSE ONE: 1--YES 2--NO
2
1
CORRECTIONS FOR: A: MATRIX ENTER ROW & COLUMN (RR, CC):
ENTER NEW ELEMENT
1 OR 3
3
WHICH MATRIX ? CHOOSE ONE: 1--A MATRIX 2--B MATRIX 3--BOTH
2
CORRECTIONS FOR: B: MATRIX ENTER ROW & COLUMN (RR, CC):
ENTER NEW ELEMENT
DO YOU WISH TO PERMUTE A & B MATRICES? CHOOSE ONE: 0--NO 1--YES
1
0
COLUMN PERMUTATION
ENTER PERMUTATION SEQUENCE FOR : B: MATRIX ROW PERMUTATION
WHICH MATRIX ? CHOOSE ONE: 1--A MATRIX 2--B MATRIX 3--BOTH
2
3
COLUMN PERMUTATION
1
1 OR 3
ENTER PERMUTATION SEQUENCE FOR: A: MATRIX ROW PERMUTATION
2
PLEASE ENTER THE NUMBER OF OUTPUTS, R.
ENTER C MATRIX ENTER BY ROWS:
ENTER D MATRIX ENTER BY ROWS:
DO YOU LIKE C & D ? CHOOSE ONE: 1--YES 2--NO
2
1
CORRECTIONS FOR: C: MATRIX ENTER ROW & COLUMN (RR, CC):
ENTER NEW ELEMENT
1 OR 3
3
WHICH MATRIX ? CHOOSE ONE: 1--C MATRIX 2--D MATRIX 3--BOTH
2
CORRECTIONS FOR:D: MATRIX ENTER ROW & COLUMN (RR, CC):
ENTER NEW ELEMENT
COLUMN PERMUTATION
ENTER PERMUTATION SEQUENCE FOR: D: MATRIX ROW PERMUTATION
DO YOU WISH TO PERMUTE C & D MATRICES ? CHOOSE ONE 0--NO 1--YES
1
0
WHICH MATRIX ? CHOOSE ONE: 1--C MATRIX 2--D MATRIX 3--BOTH
2
3
COLUMN PERMUTATION
1
1 OR 3
ENTER PERMUTATION SEQUENCE FOR : C : MATRIX ROW PERMUTATION

Figure 5.- Flow chart of iterative discourse.

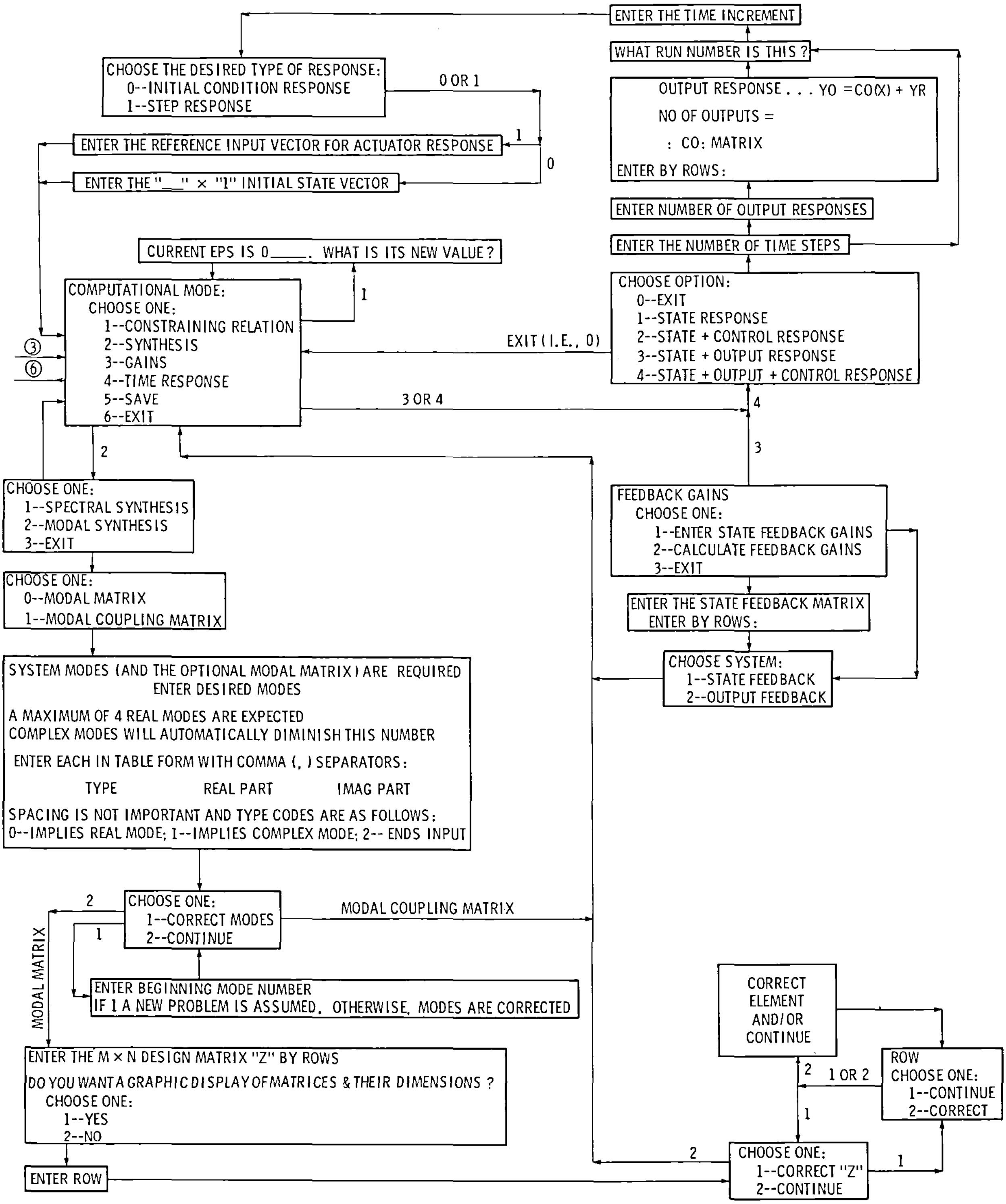

ENTER THE TIME INCREMENT
WHAT RUN NUMBER IS THIS ?
OUTPUT RESPONSE . . . YO =CO(X) + YR
NO OF OUTPUTS =
: CO: MATRIX
ENTER BY ROWS:
ENTER NUMBER OF OUTPUT RESPONSES
ENTER THE NUMBER OF TIME STEPS
CHOOSE THE DESIRED TYPE OF RESPONSE:
0--INITIAL CONDITION RESPONSE
1--STEP RESPONSE
0 OR 1
ENTER THE REFERENCE INPUT VECTOR FOR ACTUATOR RESPONSE
1
0
ENTER THE "__" × "1" INITIAL STATE VECTOR
CURRENT EPS IS 0____. WHAT IS ITS NEW VALUE?
COMPUTATIONAL MODE:
CHOOSE ONE:
1--CONSTRAINING RELATION
2--SYNTHESIS
3--GAINS
4--TIME RESPONSE
5--SAVE
6--EXIT
1
③
⑥
EXIT (I.E., 0)
CHOOSE OPTION:
0--EXIT
1--STATE RESPONSE
2--STATE + CONTROL RESPONSE
3--STATE + OUTPUT RESPONSE
4--STATE + OUTPUT + CONTROL RESPONSE
3 OR 4
4
3
2
CHOOSE ONE:
1--SPECTRAL SYNTHESIS
2--MODAL SYNTHESIS
3--EXIT
FEEDBACK GAINS
CHOOSE ONE:
1--ENTER STATE FEEDBACK GAINS
2--CALCULATE FEEDBACK GAINS
3--EXIT
CHOOSE ONE:
0--MODAL MATRIX
1--MODAL COUPLING MATRIX
ENTER THE STATE FEEDBACK MATRIX
ENTER BY ROWS:
CHOOSE SYSTEM:
1--STATE FEEDBACK
2--OUTPUT FEEDBACK
SYSTEM MODES (AND THE OPTIONAL MODAL MATRIX) ARE REQUIRED
ENTER DESIRED MODES
A MAXIMUM OF 4 REAL MODES ARE EXPECTED
COMPLEX MODES WILL AUTOMATICALLY DIMINISH THIS NUMBER
ENTER EACH IN TABLE FORM WITH COMMA (,) SEPARATORS:
TYPE REAL PART IMAG PART
SPACING IS NOT IMPORTANT AND TYPE CODES ARE AS FOLLOWS:
0--IMPLIES REAL MODE; 1--IMPLIES COMPLEX MODE; 2-- ENDS INPUT
2
CHOOSE ONE:
1--CORRECT MODES
2--CONTINUE
1
MODAL COUPLING MATRIX
MODAL MATRIX
ENTER BEGINNING MODE NUMBER
IF 1 A NEW PROBLEM IS ASSUMED. OTHERWISE, MODES ARE CORRECTED
CORRECT ELEMENT AND/OR CONTINUE
ENTER THE M × N DESIGN MATRIX "Z" BY ROWS
DO YOU WANT A GRAPHIC DISPLAY OF MATRICES & THEIR DIMENSIONS ?
CHOOSE ONE:
1--YES
2--NO
2
1 OR 2
ROW
CHOOSE ONE:
1--CONTINUE
2--CORRECT
1
2
CHOOSE ONE:
1--CORRECT "Z"
2--CONTINUE
1
ENTER ROW

Figure 5.- Concluded.

1. Report No.
NASA TM-85690

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
IMAGES - A DIGITAL COMPUTER PROGRAM FOR INTERACTIVE MODAL ANALYSIS AND GAIN ESTIMATION FOR EIGENSYSTEM SYNTHESIS

5. Report Date
February 1984

6. Performing Organization Code
505-34-03-02

7. Author(s)
Ruben L. Jones

8. Performing Organization Report No.
L-15655

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An interactive digital computer program for modal analysis and gain estimation for eigensystem synthesis has been written. Both mathematical and operational considerations are described; however, the mathematical presentation is limited to those concepts essential to the operational capability of the program. The program is capable of both modal and spectral synthesis of multi-input control systems. It is user friendly, has scratchpad capability and dynamic memory, and can be used to design either state or output feedback systems.

17. Key Words (Suggested by Author(s))
Pole placement
Modal synthesis
Spectral synthesis
IMAGES
Interactive control design

18. Distribution Statement
Unclassified - Unlimited

Subject Category 08

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
46

22. Price
A03

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1984