NASA CR-850

A DIGITAL COMPUTER PROGRAM FOR THE ANALYSIS AND DESIGN OF STATE VARIABLE FEEDBACK SYSTEMS

By James L. Melsa

Prepared under Grant No. NsG-490 by
THE UNIVERSITY OF ARIZONA
Tucson, Ariz.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ABSTRACT

This report discusses the development and use of a digital computer program for the analysis and design of linear state variable feedback systems. Because the program is based on the matrix formulation, it retains the generality and flexibility of this approach. The program has been successfully used on a large number of practical problems.

I INTRODUCTION

Although the computations involved in linear state variable design are algebraic, such calculations can still be laborious if the order of the system is greater than three and/or if the numerical values involved are not of the simple textbook variety. For this reason, it is helpful to have a digital computer program to reduce or eliminate the computational load in analyzing and designing state variable feedback systems. This report presents the development and use of such a program.

Although it is assumed that the reader is familiar with the basic concepts of the state variable feedback methods[1],[2], the basic matrix formulation of the problem is presented below in order to acquaint the reader with the specific formulation which is used here and in the program. This presentation has been divided into two parts: the representation of the open-loop plant and the closed-loop system.

Open-Loop Plant: The nth-order plant is assumed to be represented in matrix notation as

$$\dot{\underset{\sim}{x}}(t) = \underset{\approx}{A}\underset{\sim}{x}(t) + \underset{\sim}{b}u(t) \tag{1}$$

and

$$y(t) = \underset{\sim}{c}^T\underset{\sim}{x}(t) \tag{2}$$

where $\underset{\sim}{x}$ is the state vector, u is the scalar input and y is the scalar output.* (See Fig. 1) In terms of this representation, the open-loop plant transfer function is given by

$$\frac{y(s)}{u(s)} = G(s) = \underset{\sim}{c}^T\underset{\approx}{\Phi}(s)\underset{\sim}{b} \tag{3}$$

*It is assumed that if series compensation[1],[2] is to be used, it has already been incorporated into the plant description.

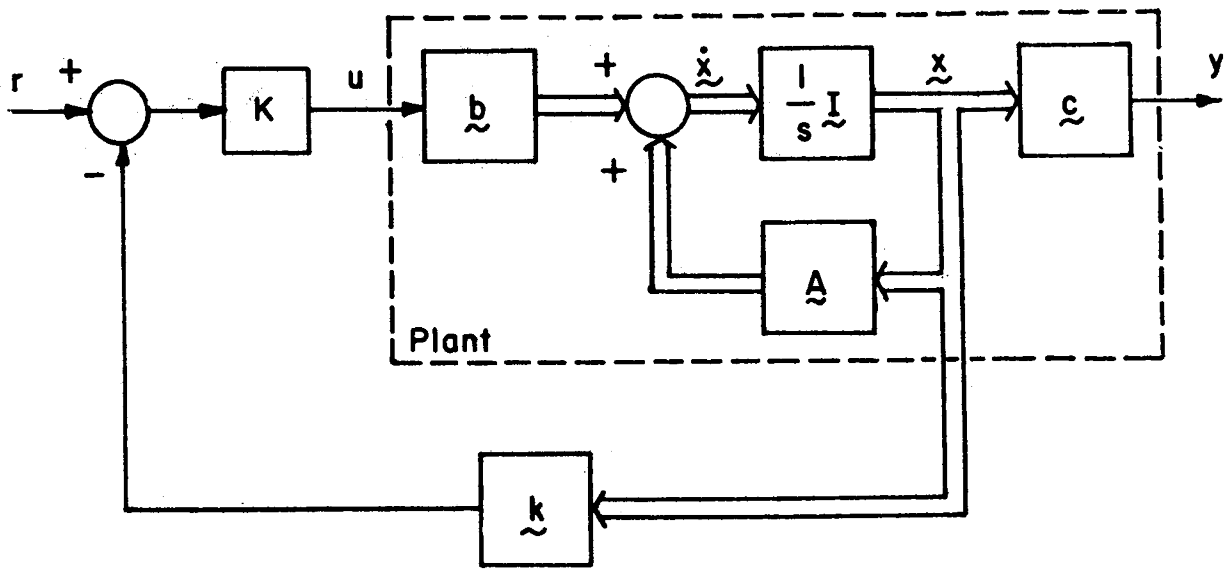

Fig. 1 General System Representation

where $\underset{\sim}{\Phi}(s)$ is the resolvent matrix given by

$$\underset{\sim}{\Phi}(s) = (s\underset{\sim}{I}-\underset{\sim}{A})^{-1} \tag{4}$$

It is also convenient to represent G(s) as a ratio of polynomials in s of the form

$$G(s) = \frac{N(s)}{D(s)} = \frac{n_1 + n_2 s + \cdots + n_m s^{m-1}}{d_1 + d_2 s + d_3 s^2 + \cdots + d_n s^{n-1} + s^n} \tag{5}$$

The basic problem associated with the open-loop plant is the determination of the plant transfer function G(s) from the matrix representation. The use of Eqs. (3) and (4) becomes tedious if the order is greater than two. Although block diagram manipulations may also be used to find G(s), this also becomes laborious if there are many internal feedback or feed-forward loops, as there often are in practical problems. In addition to determining G(s), one may also wish to determine some of the internal transfer functions $x_i(s)/u(s)$.

One might question why G(s) and the internal transfer functions are necessary if one is simply interested in the design of a closed-loop system. It is the author's conviction that one cannot intelligently undertake the design of any closed-loop system without a knowledge of the open-loop transfer function. In addition, the use of internal transfer functions has been shown to be a valuable design aid[3] in the design of closed-loop systems.

Closed-Loop System: The closed-loop form for the control input is given by

$$u(t) = K[r(t) - \underset{\sim}{k}^T \underset{\sim}{x}(t)] \tag{6}$$

where r(t) is the reference input, K is the forward path gain of the controller and $\underset{\sim}{k}$ is the feedback coefficient vector. (See Fig. 1) In terms of this controller structure, the matrix representation of the closed-loop system becomes

$$\dot{\underset{\sim}{x}}(t) = (\underset{\sim}{A}-K\underset{\sim}{b}\underset{\sim}{k}^T)\underset{\sim}{x}(t) + K\underset{\sim}{b}r(t) \tag{7}$$

$$y(t) = \underset{\sim}{c}^T\underset{\sim}{x}(t) \tag{8}$$

and the closed-loop transfer function is given by

$$\frac{y(s)}{r(s)} = K\underset{\sim}{c}^T\underset{\sim}{\Phi}_k(s)\underset{\sim}{b} \tag{9}$$

where

$$\underset{\sim}{\Phi}_k(s) = (s\underset{\sim}{I}-\underset{\sim}{A} + K\underset{\sim}{b}\underset{\sim}{k}^T)^{-1} \tag{10}$$

If we make use of the fictitious $H_{eq}(s)$ representation to find $y(s)/r(s)$ then

$$\frac{y(s)}{r(s)} = \frac{KG(s)}{1 + KG(s)H_{eq}(s)} \tag{11}$$

where

$$H_{eq}(s) = \frac{\underset{\sim}{k}^T\underset{\sim}{\Phi}(s)\underset{\sim}{b}}{\underset{\sim}{c}^T\underset{\sim}{\Phi}(s)\underset{\sim}{b}} \tag{12}$$

As a ratio of polynomials in s, $y(s)/r(s)$ may be written as

$$\frac{y(s)}{r(s)} = \frac{KN(s)}{D_k(s)} = \frac{K(n_1 + n_2s + \cdots + n_ms^{m-1})}{(e_1 + e_2s + \cdots + e_ns^{n-1} + s^n)} \tag{13}$$

Here use has been made of that fact that the zeros of $y(s)/r(s)$ are identical to the zeros of $G(s)$.[1] On the other hand, the poles of $H_{eq}(s)$ are equal to the zeros of $G(s)$ so that $H_{eq}(s)$ becomes

$$H_{eq}(s) = \frac{N_h(s)}{N(s)} = \frac{(h_1 + h_2s + \cdots + h_ns^{n-1})}{(n_1 + n_2s + \cdots + n_ms^{m-1})} \tag{14}$$

In terms of $H_{eq}(s)$, therefore, $y(s)/r(s)$ may be expressed as

$$\frac{y(s)}{r(s)} = \frac{KN(s)/D(s)}{1 + K[N(s)/D(s)][N_h(s)/N(s)]} = \frac{KN(s)}{D(s) + KN_h(s)} \tag{15}$$

so that

$$D_k(s) = D(s) + KN_h(s) = (d_1 + Kh_1) + (d_2 + Kh_2)s + \cdots + (d_n + Kh_n)s^{n-1} + s^n \tag{16}$$

The computational problems associated with the closed-loop system may be divided into two phases: analysis and design. The analysis problem is to determine y(s)/r(s) for a given plant if K and $\underset{\sim}{k}$ are given. This problem is similar to the open-loop problem, and once again, the use of the matrix approach, Eqs. (9) and (10), or the block diagram approach, Eq. (11), can become extremely tedious.

The design problem is associated with the determination of K and $\underset{\sim}{k}$ in order to achieve a desired y(s)/r(s). There are two basic computational problems related to this design task. First, the closed-loop transfer function must be found. In this case the task is even more difficult than in the analysis problem since literal coefficients involving the unknown values of K and $\underset{\sim}{k}$ are involved. Second, once y(s)/r(s) is determined in terms of K and $\underset{\sim}{k}$, the various coefficients must be equated and the resulting equations must be solved. Since these equations are, in general, a set of n simultaneous linear algebra equations, the labor is not trivial.

The purpose of this section has been to introduce the reader to the notation and formulation that is used through this report while at the same time to outline the computational problems associated with the linear state variable feedback method. The program discussed in this report eliminates all of the computational problems discussed above and also provides additional valuable design information. The basic approach of the program is discussed in the next section.

II METHOD OF SOLUTION

Two possible solutions of the computational problems of state variable feedback were discussed briefly in the preceding section: block diagram manipulations[4] and the direct matrix approach. Unfortunately neither of these techniques is well suited to machine computation. The rules of block diagram manipulation, although well suited to hand calculations, are not easy to program. Because of this fact, one must normally force the problem to be formulated in terms of special configurations before results can be obtained. In addition, both the block diagram and matrix methods necessitate the handling of polynomials in s whose coefficients, in the case of closed-loop design problems, may be functions of K and $\underset{\sim}{k}$. Such operations are particularly difficult to treat with a digital computer.

To the credit of the block diagram approach is its close tie with classical engineering procedures and the attendant engineering intuition associated with these procedures. The advantage of the matrix technique, on the other hand, is the generality of its formulation of the problem. An ideal method of solution might, therefore, incorporate the generality of the matrix formulation and the significance of the block diagram method while eliminating the programming problems discussed above.

The method developed in this report, which is referred to as the _indirect matrix_ or _phase variable_ approach, does satisfy these requirements reasonably well. This method is based on the realization that if the plant is represented in phase variables, then both the open- and closed-loop computational problems listed in the previous section may be solved by inspection. In order to illustrate this feature, consider the phase variable representation[1] of the open-loop plant described by the transfer function of Eq. (5).*

$$\dot{\underset{\sim}{x}}^p = \underset{\sim}{A}^p \underset{\sim}{x}^p + \underset{\sim}{b}^p u$$
$$y = (\underset{\sim}{c}^p)^T \underset{\sim}{x}^p \qquad (17)$$

where

$$\underset{\sim}{A}^p = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ -d_1 & -d_2 & -d_3 & \cdots & -d_n \end{bmatrix}, \underset{\sim}{b}^p = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \text{ and } \underset{\sim}{c}^p = \begin{bmatrix} n_1 \\ \vdots \\ n_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (18)$$

A comparison of Eq. (5) and Eqs. (17) and (18) reveals that the plant transfer function may be determined by inspection from the phase variable representation.

In terms of the closed-loop calculations, the task may be similarly accomplished by inspection. If the closed-loop expression for the control is given

$$u(t) = K[r(t) - (\underset{\sim}{k}^p)^T \underset{\sim}{x}(t)]$$

where $\underset{\sim}{k}^p$ is the feedback coefficient vector in terms of phase variables, then the system representation becomes

$$\dot{\underset{\sim}{x}}^p = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ -(d_1+Kk_1^p) & -(d_2+Kk_2^p) & \cdots & -(d_n+Kk_n^p) \end{bmatrix} \underset{\sim}{x}^p + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} r(t) \qquad (19)$$

*The superscript "p" is used throughout to indicate phase variable quantities.

and

$$y(t) = K[n_1\ n_2\ \cdots\ n_m\ 0\ \cdots\ 0]\underset{\sim}{x}^P(t) \tag{20}$$

Therefore the closed-loop transfer function is given by

$$\frac{y(s)}{r(s)} = \frac{K(n_1 + n_2 s + \cdots + n_m s^{m-1})}{(d_1+Kk_1^P) + (d_2+Kk_2^P)s + \cdots + (d_n+Kk_n^P)s^{n-1} + s^n} \tag{21}$$

A simple comparison of Eqs. (13) and (21) reveals that the coefficients of the denominator of y(s)/r(s) may be determined from the simple equation

$$e_i = d_i + Kk_i^P \qquad i = 1,2,\ldots,n \tag{22}$$

Hence, if $\underset{\sim}{k}^P$ and K are known, y(s)/r(s) may be determined directly from Eqs. (21) or (22). Similarly if the d_i's and e_i's are known then K and $\underset{\sim}{k}^P$ may also be obtained directly. In order to make the solution for K and $\underset{\sim}{k}^P$ unique, it is assumed that zero steady-state position error is desired so that K is given by

$$K = \frac{e_1}{n_1} \tag{23}$$

In addition to the above advantages of the phase variable representation, it is easy to show that $H_{eq}(s)$ is also easily related to the phase variable elements by the following equation

$$H_{eq}(s) = \frac{k_1^P + k_2^P s + \cdots + k_n^P s^{n-1}}{n_1 + n_2 s + \cdots + n_m s^{m-1}} \tag{24}$$

The above discussion establishes the fact that the solution of the open- and closed-loop computational problems become trivial if phase variables are used. Since transfer function expressions for G(s) and $H_{eq}(s)$ may also be obtained using the method, the technique retains in part at least the appeal of the block diagram approach.

However, most plants are not naturally or easily described in phase variables. In addition, it is not recommended that an attempt be made to use phase variables to representation the plant since this would destroy the generality of the matrix formulation. Because of these facts, it is obvious that if the phase variable approach is to be used it is necessary to find a technique for transforming the plant to phase variables and then transforming the feedback coefficients back to the original systems of state variables. Fortunately such a procedure exists and is relatively easy to program on a digital computer.

Kalman[5] has shown that it is possible to transform any controllable[1] plant to phase variables by means of a nonsingular linear transformation of variables of the form

$$\underset{\sim}{x} = \underset{\sim}{P}\underset{\sim}{x}^{P} \qquad (25)$$

or since $\underset{\sim}{P}$ is nonsingular

$$\underset{\sim}{x}^{P} = \underset{\sim}{P}^{-1}\underset{\sim}{x} \qquad (26)$$

In terms of the transformation matrix $\underset{\sim}{P}$, the elements of the phase variable representation may be determined from the following equation

$$\underset{\sim}{A}^{P} = \underset{\sim}{P}^{-1}\underset{\sim}{A}\underset{\sim}{P}, \qquad \underset{\sim}{b}^{P} = \underset{\sim}{P}^{-1}\underset{\sim}{b} \qquad \text{and} \qquad \underset{\sim}{c}^{P} = \underset{\sim}{P}^{T}\underset{\sim}{c} \qquad (27)$$

In addition, the feedback coefficients in phase variable and the original variables are related by the following expressions

$$\underset{\sim}{k}^{P} = \underset{\sim}{P}^{T}\underset{\sim}{k} \qquad \text{and} \qquad \underset{\sim}{k} = (\underset{\sim}{P}^{T})^{-1}\underset{\sim}{k}^{P} \qquad (28)$$

Therefore once the matrix $\underset{\sim}{P}$ is known, the complete transformation problem is solved.

A number of recent articles[6]-[10] describe a simple algorithm for determining $\underset{\sim}{P}$ if the coefficients of the characteristic polynomial of $\underset{\sim}{A}$, i.e. the d_i's, are known. This algorithm indicates that if the vectors p^i are defined by the recursion formula,

$$\underset{\sim}{p}^n = \underset{\sim}{b}$$

and

$$\underset{\sim}{p}^{n-i} = \underset{\sim}{A}\underset{\sim}{p}^{n-i+1} + d_{n-i+1}\underset{\sim}{b} \quad i = 1,2,\ldots,n-1 \tag{29}$$

then P is given by

$$\underset{\sim}{P} = [\underset{\sim}{p}^1 \mid \underset{\sim}{p}^2 \mid \cdots \mid \underset{\sim}{p}^n] \tag{30}$$

This procedure is simple and extremely easy to program.

Fortunately, there are a number of simple methods for determining the coefficients of the characteristic polynomial. Two of these methods were investigated: the Leverier algorithm[6] and the principal-minor method[11]. Although the Leverier algorithm is easier to program and uses less computer storage and time than the principal-minor method, unfortunately it does not, in general, yield answers which are as accurate as the principal-minor method. The inaccuracy of the Leverier algorithm is a result of the sequential nature of the algorithm, i.e. each coefficient depends on the previously computed coefficients. Whenever the values of the coefficients vary widely, especially if the later coefficients are much smaller than the earlier ones, round-off error can propogate and cause large errors to arise in the results. Due to this problem with the Leverier algorithm, the principal-minor method was selected for use in the program. The principal-minor method determines the coefficients of the characteristic polynomial of the matrix $\underset{\sim}{A}$ by means of the following expression

$$d_i = (-1)^{n-i} \sum \left[\text{of the } \binom{n}{n-i+1} \begin{array}{l} (n-i+1)\text{-order} \\ \text{principal minors of } \underset{\sim}{A} \end{array} \right] \qquad (31)$$

Once again, this expression is quite easy to program.

After the d_i's have been calculated, $\underset{\sim}{P}$ can be determined from Eqs. (29) and (30) and any of the open- or closed-loop analysis or design problems may be solved by using Eqs. (27) and (28). This procedure has been tested on a wide variety of problems[3],[12] with very successful results.

III CAPABILITIES

In this section, the general capabilities of the computer program are presented and compared to the computational problems posed in Section I. A more detailed discussion of the utilization of the program is presented in the next two sections which are concerned with the preparation of the input data and the interpretation of the output results. In addition, an abbreviated flow chart of the program is contained in Appendix B.

Open-Loop Computations: The basic problem associated with the open-loop plant is the determination of the open-loop transfer function G(s). This problem is automatically solved when the plant is transformed into the phase variable presentation. In addition to determining G(s), the program factors the numerator and denominator polynomials of G(s) so that the locations of the open-loop poles and zeros are known.

The program can determine internal transfer functions of the form $x_i(s)/u(s)$ by simply defining a fictitious $\underset{\sim}{c}$ matrix. If, for example, the transfer function $x_k(s)/u(s)$ is desired then $\underset{\sim}{c}$ is selected with $c_k = 1$ and $c_j = 0$ for $j \neq k$. Any number of these internal transfer functions may be determined at the same time. However, the $\underset{\sim}{c}$ for the actual output y must always be used last so that the forward gain K is selected properly. The denominator polynomial of all the transfer functions is always D(s).

<u>Closed-Loop Computations</u>: The program is capable of performing three types of closed-loop calculations which are indicated by setting the symbol KEY as 1, 2 or 3 respectively. One of these three computations (KEY = 1) is for closed-loop analysis while the other two (KEY = 2 and 3) are for design. Any number of these three types of computations in any combination may be accomplished at one time.

In the analysis mode, KEY = 1, the program must be given K and $\underset{\sim}{k}$ as input. From this input the program determines the coefficients of the closed-loop characteristic polynomial, $D_k(s)$, and the numerator of $H_{eq}(s)$, $N_h(s)$.* In addition, both $D_k(s)$ and $N_h(s)$ are factored and their roots are displayed. This mode of operation is perhaps most useful for sensitivity studies. By varying the values of K and $\underset{\sim}{k}$, it is possible to see the effect of such changes on $y(s)/r(s)$. One may use this procedure for plotting the root locus of $1 + KG(s)H_{eq}(s)$ versus K. In addition, since G(s) and $H_{eq}(s)$ are known in factored form, the root locus may be easily sketched by hand.

The input for the first of the two design modes, KEY = 2, is the desired closed-loop characteristic polynomial, $D_k(s)$. From this information, the program computes K and $\underset{\sim}{k}$ and determines the numerator polynomial of $H_{eq}(s)$. In the second design mode, KEY = 3, the input is

*The reader will recall that the denominator polynomial of $H_{eq}(s)$ is equal to the numerator of G(s), i.e. N(s).

the desired closed-loop pole locations and output is again K, $\underset{\sim}{k}$ and $H_{eq}(s)$. As in the analysis mode, the polynomial $D_k(s)$ and $N_h(s)$ are always given in factored as well as unfactored form.

The reader is reminded that K is always selected so that zero steady-state error for a step input results. If it is desired that some other condition be used to select K, then one may always rescale K and $\underset{\sim}{k}$ to meet such a restriction by hand. Suppose, for example, that it is desired that the d.c. gain be ten rather than one, then one would simply multiply K by ten and divide $\underset{\sim}{k}$ by ten. This procedure leaves the denominator of y(s)/r(s) unchanged but multipltiplies the numerator by ten as desired.

Which of the two design modes is to be used depends on whether the desired y(s)/r(s) is known in factor or unfactored form. It is the author's opinion that the second design mode, i.e. y(s)/r(s) given in factored form, is the more useful. Since the inclusion of the first design mode required only a minor addition to the coding, it is included for added flexibility. A summary of the three modes of closed-loop calculations is given in Table 1.

Special Computations: In addition to the standard open- and closed-loop computations discussed above, the program also makes two special computations to assist the user in evaluating the applicability and accuracy of the program.

Since the program involves transforming the plant to phase variables, it is necessary[5] that the plant be controllable. In addition, the plant must be controllable before any transfer function techniques can be sensibly utilized. Hence it is important to check the controllability of the plant to insure that the results obtained are meaningful. Controllability should be checked even if the physical plant is known to be controllable

TABLE 1

SUMMARY CLOSED-LOOP COMPUTATIONS

KEY	TYPE	Input	Output
1	Analysis	K and $\underset{\sim}{k}$	$N_h(s)$ and $D_k(s)$
2	Design	$D_k(s)$, unfactored	$N_h(s)$, K and $\underset{\sim}{k}$
3	Design	$D_k(s)$, factored	$N_h(s)$, K and $\underset{\sim}{k}$

since the mathematical modeling, especially if linearization has been involved, may have destroyed the property.

The easiest method for determining controllability is the test, originally proposed by Kalman[5], which states that the plant

$$\dot{\underset{\sim}{x}} = \underset{\sim}{A}\underset{\sim}{x} + \underset{\sim}{b}u \tag{1}$$

is controllable if and only if the controllability matrix

$$\underset{\sim}{M}_c = [\underset{\sim}{b} \,\vdots\, \underset{\sim}{A}\underset{\sim}{b} \,\vdots\, \underset{\sim}{A}^2\underset{\sim}{b} \,\vdots\, \cdots \,\vdots\, \underset{\sim}{A}^{n-1}\underset{\sim}{b}] \tag{31}$$

is nonsingular. Although the application of this test is straight-forward, the computational labor involved can, once again, become excessive if the order of the plant is greater than three. On the other hand, it becomes increasingly important to make the controllability test as the order of the plant increases since it becomes more difficult to determine controll-ability by inspection.

The obvious solution to this problem is to include the controllability test as part of the digital program. The additional coding is very minor, and by including the test an integral part of the program, one is assured that controllability is always checked. The controllability test is applied by finding the determinant* of the controllability matrix $\underset{\sim}{M}_c$. If the det($\underset{\sim}{M}_c$) = 0 then plant is indicated as being uncontrollable (See Section V); however all computations, both open- and closed-loop continue to be performed. Extreme caution should be applied in interpreting any results obtained for an uncontrollable plant.

Even when the det($\underset{\sim}{M}_c$) ≠ 0** problems may still arise if the matrix $\underset{\sim}{M}_c$ is ill-conditioned and therefore difficult to invert. In order to check

*Both the determinant and inversion subroutines are already required to determine $\underset{\sim}{P}$ and $\underset{\sim}{P}^{-1}$.

**Note that the exact value of det($\underset{\sim}{M}_c$) is unimportant since it may be set at any value by scaling the original plant equations.

this possibility, the matrix $\underset{\sim}{M}_c$ is inverted, as well as possible, by the numerical inversion subroutine included in the program. If this inverse matrix $\underset{\sim}{M}_c^{-1}$ is multiplied with $\underset{\sim}{M}_c$, the result should be the identity matrix $\underset{\sim}{I}$. The degree to which this product deviates from $\underset{\sim}{I}$ is a measure of the uncontrollability of the plant. If the maximum value of absolute deviation is greater than 10^{-5}[*], the system is identified as being <u>numerically</u> uncontrollable. This terminology was chosen to indicate that while the plant is theoretically controllable, it is uncontrollable in a numerical sense since $\underset{\sim}{M}_c$ cannot be inverted by the program.

Although the numerical controllability of the plant is obviously dependent on the numerical inversion scheme used, a wide variety of programs for inversion have been tried with only minor changes in the results. As in the case of pure uncontrollability, all computations are still completed, and as before, one must be cautious in using these results. In the numerical uncontrollability however, if the maximum deviation is quite small but greater than 10^{-5}, then the results may still be completely satisfactory.

In order to assist in determining the validity of the results in the numerically uncontrollable situation discussed above and to uncover any other numerical problems that might arise, although none have occurred so far in numerous examples of order as high as seventh, a double-checking feature was added to the program. This computational double-check is based on the fact that $D_k(s)$ may be determined by either of the following two expressions.[1]

---

[*]This number was selected somewhat arbitrarily and may be easily altered if desired by the user.

$$D_k(s) = D(s) + KN_h(s) \tag{16}$$

or

$$D_k(s) = \det(s\underset{\sim}{I}-\underset{\sim}{A}+K\underset{\sim}{b}\underset{\sim}{k}^T) \tag{32}$$

The computer program determines $D_k(s)$ by both of these equations, even when KEY = 2 and $D_k(s)$ is given, and compares the results. A normalized error is obtained by dividing the difference between the two values of a coefficient of $D_k(s)$ as determined from Eqs. (16) and (32) by the value of the coefficient. The maximum, over all of the coefficients of $D_k(s)$, of the absolute values of this normalized error is printed at the end of each closed-loop calculation. Because of the dissimilarities of the mathematical techniques involved in Eqs. (16) and (32), this procedure provides a simple and yet meaningful check on the accuracy of the results. The program always lists the coefficients as determined by Eq. (32) since the use of Eq. (16) may be easily accomplished by hand.

IV INPUT FORMAT

The general input and output formats for the program are discussed in this and the following section respectively. While reading these sections, it is suggested that the reader refer often to the specific example problem presented in Appendix A, especially Tables A-1 and A-2.

A primary consideration in the selection of the input format was to make the program as simple as possible to use while still retaining sufficient flexibility. Because of this consideration, the format selected and described in this section contains more cards than necessary to provide the program with input data. On the other hand, the input format is easy to remember since the large majority of the cards have identical forms. In addition, the input deck is closely related to the matrix formulation of the state variable feedback problem so that it is extremely simple to make modifications in the problem if desired.

The program can be used to solve more than one problem in a single run by simply placing the various input decks one after another. In other words, an input deck is prepared for each problem using the format presented in this section and these decks are then added together to form a composite input deck for the program. In addition, it is possible to execute a number of open- and closed-loop calculations as part of one basic problem as long as the plant is not changed.

The operation of the program (See Appendix B) is divided into three phases: 1) basic, 2) open-loop and 3) closed-loop. During the basic phase, the $\underset{\sim}{A}$ and $\underset{\sim}{b}$ matrices are read and controllability is checked. The open-loop phase consists of the open-loop computations discussed in the previous sections and include the determination of D(s), $\underset{\sim}{P}$ and N(s). The closed-loop phase varys depending on which types of closed-loop calculations are desired. In addition, the double check on accuracy is made during the closed-loop phase of operation.

The input and output formats fall into three similar divisions. Because of this fact and in order to simplify the treatment, the discussion of the input and output formats is divided along these same lines.

<u>Basic Input</u>: The input for the basic phase of the programs consists of a set of cards which specify the elements of $\underset{\sim}{A}$ and $\underset{\sim}{b}$. In addition, a card is included which gives problem identification information for easy referencing of the problem and the order of the plant, n. These cards are included for all problems independent of what open- or closed-loop computations are to be made. A detailed description of the preparation of the basic input cards is shown in Table 2.

TABLE 2

CARD DESCRIPTION FOR BASIC INPUT

Card No.	Column Nos.	Description	Format(1) type
1	1-20	Problem identification (any alpha-numeric characters) to be printed out for reference.	4A5,I2
	21-22	n = order of the plant, an interger-right justified in the field.(2)	
2	1-10	a_{11}	8E10.0
	11-20	a_{12}	
	etc	...	
3(3)	1-10	a_{21}	8E10.0
	11-20	a_{22}	
	etc	...	
n+2(3)	1-10	b_1	8E10.0
	11-20	b_2	
	etc	...	

Notes: 1) The format type is listed for those readers familiar with FORTRAN
2) At present, n must be $\leq$ 10, however, this number may be easily increased by altering the dimension statements
3) Cards numbers are correct if n $\leq$ 8, see text

Note that the elements of $\underset{\sim}{A}$ are read one row at a time so that numbers appear much as they do in the original array. The numbers are placed in ten column fields in floating point format. If the order of the plant, n, is greater than eight, the elements continue on the next card. The elements of the next row, however, start on a new card. If $n \leq 8$, there are n + 2 cards in the basic input as shown in Table 2 while if $9 \leq n \leq 16$, there are 2n + 3 cards and so forth. Note that elements of the control vector, $\underset{\sim}{b}$, are read in <u>column</u> form in order to minimize the number of input cards. All vectors, viz. $\underset{\sim}{b}$, $\underset{\sim}{c}$ and $\underset{\sim}{k}$, are read in the same manner.

<u>Open-Loop Input</u>: The input for the open-loop phase consists solely of $\underset{\sim}{c}$ matrices which specify for which transfer functions the numerator polynomials are desired. The reader should recall that fictitious $\underset{\sim}{c}$ matrices may be used to obtain internal transfer functions. The elements of each $\underset{\sim}{c}$ matrix are placed on one or more cards, depending on the value of n, in the same manner as $\underset{\sim}{b}$, i.e. c_1 in column 1-10, c_2 in column 11-20, and so forth. (See Table 3) Any number of $\underset{\sim}{c}$ matrices can be included in any order as long as two simple rules are followed.

First, the last $\underset{\sim}{c}$ matrix must be completely zero, i.e. $c = 0$; this $\underset{\sim}{c}$ matrix of zeros is used only to signal the end of the open-loop calculations and no computations are made with it. Setting $\underset{\sim}{c}$ equal to zero may be easily accomplished by placing one or more blank cards in the deck depending on the value of n.* The second rule is that the $\underset{\sim}{c}$ matrix for the real, physical output must always appear just before the blank $\underset{\sim}{c}$ matrix. This last rule is necessary in order that K be computed properly for zero steady state position error.

*For $n \leq 8$, one card is used; for $9 \leq n \leq 16$, two cards are used and so forth.

TABLE 3

CARD DESCRIPTION FOR THE OPEN-LOOP INPUT
(NOL = number of open-loop calculations desired)

Card No.	Column Nos.	Description	Format type
1	1-10	c_1^1 = first element of the first fictitious $\underset{\sim}{c}$ matrix	8E10.0
	11-20	c_2^1 = second element of the first fictitious $\underset{\sim}{c}$ matrix	
	etc	...	

. . . . Similarly for each remaining fictitious $\underset{\sim}{c}$ matrices

Card No.	Column Nos.	Description	Format type
NOL(1)	1-10	c_1 = first element of the real $\underset{\sim}{c}$(3) matrix	8E10.0
	11-20	c_2	
	etc	...	
NOL+1(1)	1-80	BLANK CARD(S)(2)	

Notes: 1) NOL for $n \leq 8$, 2NOL for $9 \leq n \leq 16$, etc
2) One blank card for $n \leq 8$, two cards for $9 \leq n \leq 16$, etc
3) The real $\underset{\sim}{c}$ matrix always be included just before the blank card(s)

Closed-Loop Input: Each closed-loop calculation is represented by a set of two or more cards whose format depends on which of the three types of closed-loop calculations is desired. (See Tables 4, 5 and 6) These sets, one for each calculation, in any order and of any number, are combined to form the input for the closed-loop phase of the program. For example, if one wished to make three analysis mode (KEY = 1) and two second design mode (KEY = 3) calculations, then he would make up five sets of cards, one for each calculation, using the format appropriate for each calculation. These cards would then be combined by placing the three sets of the analyzing calculations first, the two design sets first, or the analysis and design sets could be intermixed in any order as long as the individual sets remained together as units.

Note that the first card in the format for each type of closed-loop calculation has only one number which is punched in the first column to give the value of KEY. The remaining cards in each set depend on the input that is required for the desired type of calculation.

A single blank card is placed in the input deck after the desired closed-loop calculation sets to signal the end of the closed-loop input. When this blank card is read, the program finds that the value of KEY is zero and returns to the basic input section to find if another problem remains to be solved. The program is stopped by the end-of-file indicator. Even if no closed-loop calculations are desired, this blank card must still be included. In this case, there are two or more blank cards at the end of the input deck since one or more blank cards appear at the end of the open loop input.

In this section, the method of preparing the input deck for state variable feedback program has been discussed in detail. Appendix A contains

TABLE 4

CARD DESCRIPTION FOR THE CLOSED-LOOP INPUT-ANALYSIS MODEL
(KEY = 1)

Card No.	Column Nos.	Description	Format type
1	1	KEY = 1	II
2	1-10	K = Forward path gain	E10.0
3	1-10	k_1	8E10.0
	11-20	k_2	
	etc	...	

TABLE 5

CARD DESCRIPTION FOR THE CLOSED-LOOP INPUT - FIRST DESIGN MODE
(KEY = 2)

Card No.	Column Nos.	Description	Format type
1	1	KEY = 2	II
2	1-10	e_1	8E10.0
	11-20	e_2	
	etc$^{(1)}$	...	

Note: 1) The coefficient, e_{n+1} of s^n is always assumed to be one and is not included on the input cards.

TABLE 6

CARD DESCRIPTION FOR THE CLOSED-LOOP INPUT-SECOND DESIGN MODE
(KEY = 3)

Card No.	Column Nos.	Description	Format type
1	1	KEY = 3	II
2	1-10	real part of the first pole of the desired y(s)/r(s)(1)	2E10.0
	11-20	imaginary part of the first pole of the desired y(s)/r(s)	
3	1-10	real part of the second pole of the desired y(s)/r(s)	2E10.0
	11-20	imaginary part of the second pole of the desired y(s)/r(s)	
etc	...	...	

Note: 1) The pole locations have positive signs if they are in the left half of the s-plane. Only one card is needed for each complex conjugate pair; the program automatically supplies the complex conjugate.

an example illustrating the preparation of the input deck for a specific problem.

V OUTPUT INTERPRETATION

Although the output of the program is almost self-explanatory, a few comments may be desirable to avoid any confusion in interpreting the results. The output format is divided into the same three phases as the input, viz. basic, open-loop and closed-loop. The reader is urged to refer to the specific example discussed in Appendix A especially Table A-2 while reading this section.

Basic Output: The basic output is composed, in the main, of simply a print-out of the information contained in the basic input, i.e. the alphanumeric problem identification, and the $\underset{\sim}{A}$ and $\underset{\sim}{b}$ matrices. The elements of $\underset{\sim}{A}$ are listed one row per line as they would normally appear in writing $\underset{\sim}{A}$. The elements of $\underset{\sim}{b}$ are also printed on one line as they would appear if one were writing $\underset{\sim}{b}^T$. This material is provided solely as reference information for the user since no new results are included. For most problems, these items are the entire basic output. However, for problems which are either uncontrollable or numerically uncontrollable,* an additional item is added to the basic output in order to indicate this complication. The reader should remember that even if the plant is found to be uncontrollable that all open- and closed-loop calculations are still performed. In the case of numerical uncontrollability, the maximum deviation of the matrix $\underset{\sim}{M}_c \underset{\sim}{M}_c^{-1}$ from the identity matrix is also listed for the information of the user.

Open-Loop Output: The open-loop output is composed of the denominator of the plant transfer function, D(s), and the various numerator polynomials

*See Section III for a definition of numerical uncontrollability.

associated with the plant transfer function and any internal transfer functions desired. The coefficients of D(s), as well as those of all other polynomials, are listed with the coefficient of the constant term first. In addition to the coefficients of D(s), the factors of D(s) are also listed.

The various numerator coefficients are identified by printing the $\underset{\sim}{c}$ matrix associated with each. Once again the elements of $\underset{\sim}{c}$ are printed as one would write the elements of $\underset{\sim}{c}^T$. Each numerator polynomial is listed in its factored as well as its unfactored form.

Closed-Loop Output: The output of the closed-loop phase of the program has exactly the same form for each of the three possible types of closed-loop calculations. The first item in the closed-loop output, however, is the value of KEY which identifies the type of computation performed. The next four items in the output are the numerator polynomial of $H_{eq}(s)$, $N_h(s)$, the feedback coefficients, $\underset{\sim}{k}$, the gain K, and the denominator of $y(s)/r(s)$, $D_k(s)$. Both $N_h(s)$ and $D_k(s)$ are given in both factored and unfactored form. Once again the coefficient of s^o is listed first. The vector $\underset{\sim}{k}$ is listed in transposed form with k_1 as the first element.

The last item in the closed-loop output is the value of the maximum normalized error.* The reader will remember that this number is an indication of the numerical accuracy of the program and is obtained by comparing the matrix and transfer function methods for finding $D_k(s)$. If more than one closed-loop computation has been requested, then the output discussed is listed for each computation.

---

*See Section III for a complete discussion.

CONCLUSIONS

This report contains the description of a digital program which performs most of the analysis and design calculations associated with the state variable feedback method. The program, which uses the matrix formulation of the state variable feedback problem, is based on the simplifications achieved by transforming the plant to phase variables. The use of the matrix formulation provides complete flexibility in the type of problems which the program can solve. On the other hand, most of the program results are stated in terms of transfer functions so that the great wealth of classical control theory may also be applied to the problem.

By eliminating the computational labor associated with the state variable feedback method, the program takes the state variable feedback method out of the textbook and makes it available as a powerful tool to the practicing engineer. The program has been employed by the author and numerous other people to solve a wide variety of practical problems. To date, no difficulties have been encountered in the operation of the program.

REFERENCES

1. Schultz, D. G. and J. L. Melsa, State Functions and Linear Control System, McGraw-Hill Book Company, Inc. New York, N. Y. 1967.

2. Schultz, D. G. Control System Design by State Variable Feedback Techniques, NASA Contractor Report, CR-77901, July 1966.

3. Murray, H. S. and J. L. Melsa, State-Variable Feedback Control of Coupled Nuclear Systems, Presented at the ANS National Topical Meeting on Coupled Reactor Kinetics, Texas A & M, January 22-23, 1967.

4. Slivinsky, C. R., L. E. Dellner and D. J. Arpasi, A FORMAC Program to Assist in the State Variable Feedback Design of Control Systems (to be published as a NASA report).

5. Kalman, R. E. Mathematical Description of Linear Dynamical Systems, J.S.I.A.M. Control, Ser. A, Vol. 1, No. 2, pp. 152-192, 1963.

6. Morgan, B. S., Jr. Sensitivity Analysis and Synthesis of Multivariable System, IEEE Trans. of Auto. Control, Vol. AC-11, No. 3, pp. 506-512, July 1966.

7. Tuel, W. G., Jr. On the Transformation to (Phase-Variable) Canonical Form, IEEE Trans. on Auto. Control, Vol. AC-11, No. 3, p. 607, July 1966.

8. Chidambara, M. R. Comment on "On the Transformation to (Phase-Variable) Canonical Form" IEEE Trans. on Auto. Control, Vol. AC-11, No. 3,pp.607-608, July 1966.

9. Rane, D. S. A Simplified Transformation to (Phase-Variable) Canonical Form, IEEE Trans. on Auto. Control, Vol. AC-11, No. 3, p. 608, July 1966.

10. Johnson, C. D. and W. M. Wonham, Another Note on the Transformation to Canonical (Phase-Variable) Form, IEEE Trans. on Auto. Control, Vol. AC-11, No. 3, pp. 609-610, July 1966.

11. Korn, G. A. and T. M. Korn, Mathematical Handbook for Scientists and Engineers, McGraw-Hill Book Company, Inc., 1961, p. 367.

12. Weaver, L. E. and R. E. Vanasse, State Variable Feedback Control of Multiregion Reactors, Nuclear Science and Engineering Publication pending.

APPENDIX A - EXAMPLE PROBLEM

This appendix discusses the application of the state variable feedback program to a simple, but not trivial, third-order example. This example is presented in order to clarify the details of preparing the input deck for and interpreting the print-out of the program. It is suggested that the reader refer often to the main body of the report while reading this appendix in order to achieve maximum understanding.

The state variable representation of the plant for the example is given by

$$\dot{\underset{\sim}{x}} = \begin{bmatrix} -1.0 & 1.0 & 0 \\ 0 & 0 & 1.0 \\ 0 & -3.0 & 0 \end{bmatrix} \underset{\sim}{x} + \begin{bmatrix} 0 \\ 0 \\ 1.0 \end{bmatrix} u \qquad \text{(A-1)}$$

$$y = [1.0 \quad 1.0 \quad 0]\underset{\sim}{x} \qquad \text{(A-2)}$$

The block diagram of the plant is shown in Fig. A-1. From this information it is possible to prepare the basic input portion of the input deck. (See Table A-1) This information must be provided no matter what open- or closed-loop calculations are to be performed.

In terms of open-loop calculations, it is desired to obtain the internal transfer functions $x_1(s)/u(s)$, $x_2(s)/u(s)$ and $x_3(s)/u(s)$ in addition to the plant transfer function $y(s)/u(s)$. Therefore it is necessary to form three fictitious $\underset{\sim}{c}$ matrices which may be used to find these internal transfer functions. These $\underset{\sim}{c}$ matrices are given by

$$\underset{\sim}{c}^1 = \text{col } (1.0,\ 0,\ 0)$$

$$\underset{\sim}{c}^2 = \text{col } (0,\ 1.0,\ 0)$$

$$\underset{\sim}{c}^3 = \text{col } (0,\ 0,\ 1.0)$$

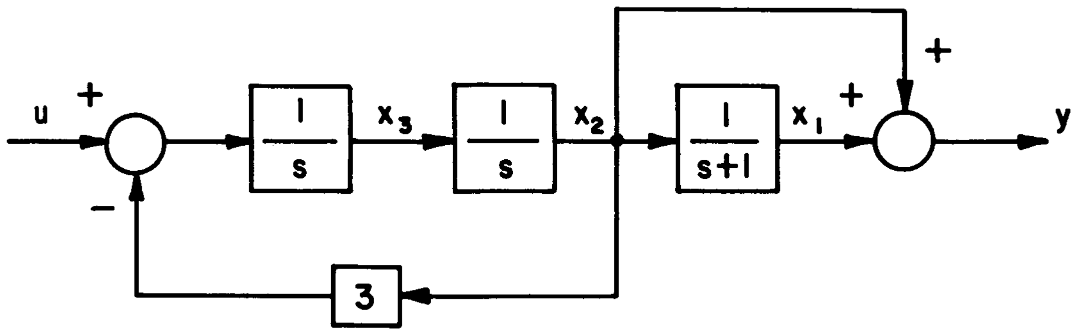

Fig. A−1 Block Diagram of the Plant

TABLE A-1 INPUT DECK

Card No.	Column Nos. 1–10	11–20	21–30	Remarks	
1	E X A M P L E O N	E	3	$\underset{\sim}{A}$	Basic Input
2	- 1 . 0	1 . 0	0 . 0		
3	0 . 0	0 . 0	1 . 0		
4	0 . 0	- 3 . 0	0 . 0		
5	0 . 0	0 . 0	1 . 0	$\underset{\sim}{b}$	
6	1 . 0	0 . 0	0 . 0	$x_1(s)/u(s)$	Open-Loop Input
7	0 . 0	1 . 0	0 . 0	$x_2(s)/u(s)$	
8	0 . 0	0 . 0	1 . 0	$x_3(s)/u(s)$	
9	1 . 0	1 . 0	0 . 0	$y(s)/u(s)$	
10	(B L A N	K C A R D)			
11	1			KEY — 1st Calc.	Closed-Loop Input
12	2 . 0			K	
13	0 . 5	0 . 0	1 . 5	$\underset{\sim}{k}$	
14	2			KEY — 2nd Calc.	
15	4 . 0	6 . 0	4 . 0	$D_k(s)$	
16	3			KEY — 3rd Calc.	
17	1 . 0	1 . 0		$D_k(s)$	
18	2 . 0				
19	(B L A N	K C A R D)			

and the open-loop input takes the form shown in Table A-1. Note that the real $\underset{\sim}{c}$ matrix, $\underset{\sim}{c}$ = col (1.0, 1.0, 0), is placed just before the blank card indicating the end of the open-loop input.

In a practical problem, one would normally run the program one time with no closed-loop calculations, i.e. only a blank card for the closed-loop input, in order to have the open-loop information available for the specification of y(s)/r(s). For the sake of illustration, it is assumed that this open-loop information has been obtained and that the desired y(s)/r(s) has been chosen to be

$$\frac{y(s)}{r(s)} = \frac{2(s+2)}{[(s+1)^2+1^2](s+2)} = \frac{2(s+2)}{(s^3+4s^2+6s+4)}$$

Since $D_k(s)$ is known in both factored and unfactored form, either of the closed-loop design modes could be used to find K and $\underset{\sim}{k}$. In order to illustrate the preparation of the input deck, both of the design modes are used. In addition, the closed-loop analysis mode is utilized by letting

$$K = 2.0 \text{ and } \underset{\sim}{k} = \text{col}(0.5,\ 0,\ 1.5)$$

These values for K and $\underset{\sim}{k}$ are selected, with the advantage of foresight, so that they yield the desired y(s)/r(s). The complete input deck is shown in Table A-1.

This problem is obviously somewhat artificial since both of the design modes are used when only one is necessary and the analysis mode is being run at the same time. The use of all three of the closed-loop modes is strictly for illustrative purposes and should not be construed as typical utilization of the program.

The complete output for this problem is shown in Table A-2. Since there is no indication of uncontrollability in the basic output, it can

TABLE A-2

PROGRAM OUTPUT
(see attached sheets)

```
PROBLEM IDENT.

THE A MATRIX

-1.0000000E  00          1.0000000E  00           0.0000000E  00
 0.0000000E  00          0.0000000E  00           1.0000000E  00
 0.0000000E  00         -3.0000000E  00           0.0000000E  00

THE B MATRIX

 0.0000000E  00          0.0000000E  00           1.0000000E  00

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

OPEN-LOOP CALCULATIONS

DENOMINATOR COEFFICIENTS

 3.0000000E  00      3.0000000E  00       1.0000000E  00       1.0000000E  00

THE ROOTS ARE                 REAL PART                 IMAGINARY PART
                           -0.0000000E  00             -1.7320508E  00
                           -0.0000000E  00              1.7320508E  00
                           -1.0000000E  00              0.0000000E  00


THE C MATRIX   *****

 1.0000000E  00          0.0000000E  00           0.0000000E  00

NUMERATOR COEFFICIENTS

 1.0000000E  00


THE C MATRIX   *****

 0.0000000E  00          1.0000000E  00           0.0000000E  00

NUMERATOR COEFFICIENTS

 1.0000000E  00          1.0000000E  00

THE ROOTS ARE                REAL PART                 IMAGINARY PART
                          -1.0000000E  00             0.0000000E  00


THE C MATRIX   *****

 0.0000000E  00          0.0000000E  00           1.0000000E  00

NUMERATOR COEFFICIENTS
 0.0000000E  00          1.0000000E  00           1.0000000E  00

THE ROOTS ARE                REAL PART                 IMAGINARY PART
                          -1.0000000E  00             0.0000000E  00
                           0.0000000E  00             0.0000000E  00
```

```
THE C MATRIX    *****

 1.0000000E  00          1.0000000E  00           0.0000000E  00

NUMERATOR COEFFICIENTS

 2.0000000E  00          1.0000000E  00

THE ROOTS ARE                 REAL PART              IMAGINARY PART
                           -2.0000000E  00          0.0000000E  00

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

CLOSED-LOOP CALCULATIONS

KEY = 1    *****

THE NUMERATOR OF H-EQUIVALENT

 5.0000000E-01           1.50000000E  00          1.50000000E  00

THE ROOTS ARE                 REAL PART              IMAGINARY PART
                           -5.0000000E-01          -2.8867512E-01
                           -5.0000000E-01           2.8867512E-01

THE FEEDBACK COEFFICIENTS

 5.0000000E-01           0.0000000E  00           1.50000000E  00

THE GAIN = 2.0000000E  00

THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL

 3.9999999E  00       6.0000000E  00       4.0000000E  00       1.0000000E  00

THE ROOTS ARE                 REAL PART              IMAGINARY PART
                           -1.0000000E  00         -1.0000000E  00
                           -1.0000000E  00          1.0000000E  00
                           -1.9999999E  00          0.0000000E  00

MAXIMUM NORMALIZED ERROR = 2.50E-08


KEY = 2    *****

THE NUMERATOR OF H-EQUIVALENT

 5.0000002E-01           1.50000000E  00          1.50000000E  00

THE ROOTS ARE                 REAL PART              IMAGINARY PART
                           -5.0000000E-01          -2.8867514E-01
                           -5.0000000E-01           2.8867514E-01
THE FEEDBACK COEFFICIENTS
 5.0000002E-01           0.0000000E  00           1.50000000E  00

THE GAIN = 1.9999999E  00
THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL

 3.9999999E  00        6.0000000E  00        4.0000000E  00        1.0000000E  00

THE ROOTS ARE                         REAL PART                IMAGINARY PART
                                   -1.0000000E  00            -1.0000000E  00
                                   -1.0000000E  00             1.0000000E  00
                                   -1.9999999E  00             0.0000000E  00

MAXIMUM NORMALIZED ERROR = 5.00E-08


KEY = 3    *****

THE NUMERATOR OF H-EQUIVALENT

 5.0000002E-01           1.5000000E  00            1.5000000E  00

THE ROOTS ARE                         REAL PART                IMAGINARY PART
                                   -5.0000000E-01             -2.8867514E-01
                                   -5.0000000E-01              2.8867514E-01


THE FEEDBACK COEFFICIENTS

 5.0000002E-01           0.0000000E  00            1.5000000E  00

THE GAIN = 1.9999999E  00

THE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL

 3.9999999E  00        6.0000000E  00        4.0000000E  00        1.0000000E  00

THE ROOTS ARE                         REAL PART                IMAGINARY PART
                                   -1.0000000E  00            -1.0000000E  00
                                   -1.0000000E  00             1.0000000E  00
                                   -1.9999999E  00             0.0000000E  00

MAXIMUM NORMALIZED ERROR = 5.00E-08
```

be safely assumed that the plant is controllable and therefore the use of the program is justified. The open-loop portion of the output provides all of the desired open-loop transfer function. The plant transfer is seen to be

$$G(s) = \frac{s+2}{s^3+s^2+3s+3} = \frac{s+2}{[s^2+(\sqrt{3})^2](s+1)}$$

while the internal transfer function, $x_3(s)/u(s)$, for example is

$$\frac{x_3(s)}{u(s)} = \frac{s^2+s}{s^3+s^2+3s+3} = \frac{s}{[s^2+(\sqrt{3})^2]}$$

The closed-loop output of the two design modes (KEY = 2 and 3) indicates that the K and $\underset{\sim}{k}$ necessary to produce the desired y(s)/r(s) are

$$K = 2.0 \text{ and } \underset{\sim}{k} = \text{col}(0.5,\ 0,\ 1.5)$$

The output of the analysis mode, on the other hand, confirms that these values for K and $\underset{\sim}{k}$ do, in fact, provide the desired results. The maximum normalized error for each of the design mode is extremely small indicating that the accuracy of the results are acceptable.

APPENDIX B

FLOW CHARTS OF THE PROGRAM OPERATION

This appendix contains three flow charts which illustrate the operation of the program for each of its three phases: basic, open-loop and closed-loop. No attempt has been made to make these flow charts contain every detail of the program. However, it is hoped that these flow charts with the complete listing given in Appendix C will enable the interested reader to understand the operation of the program.

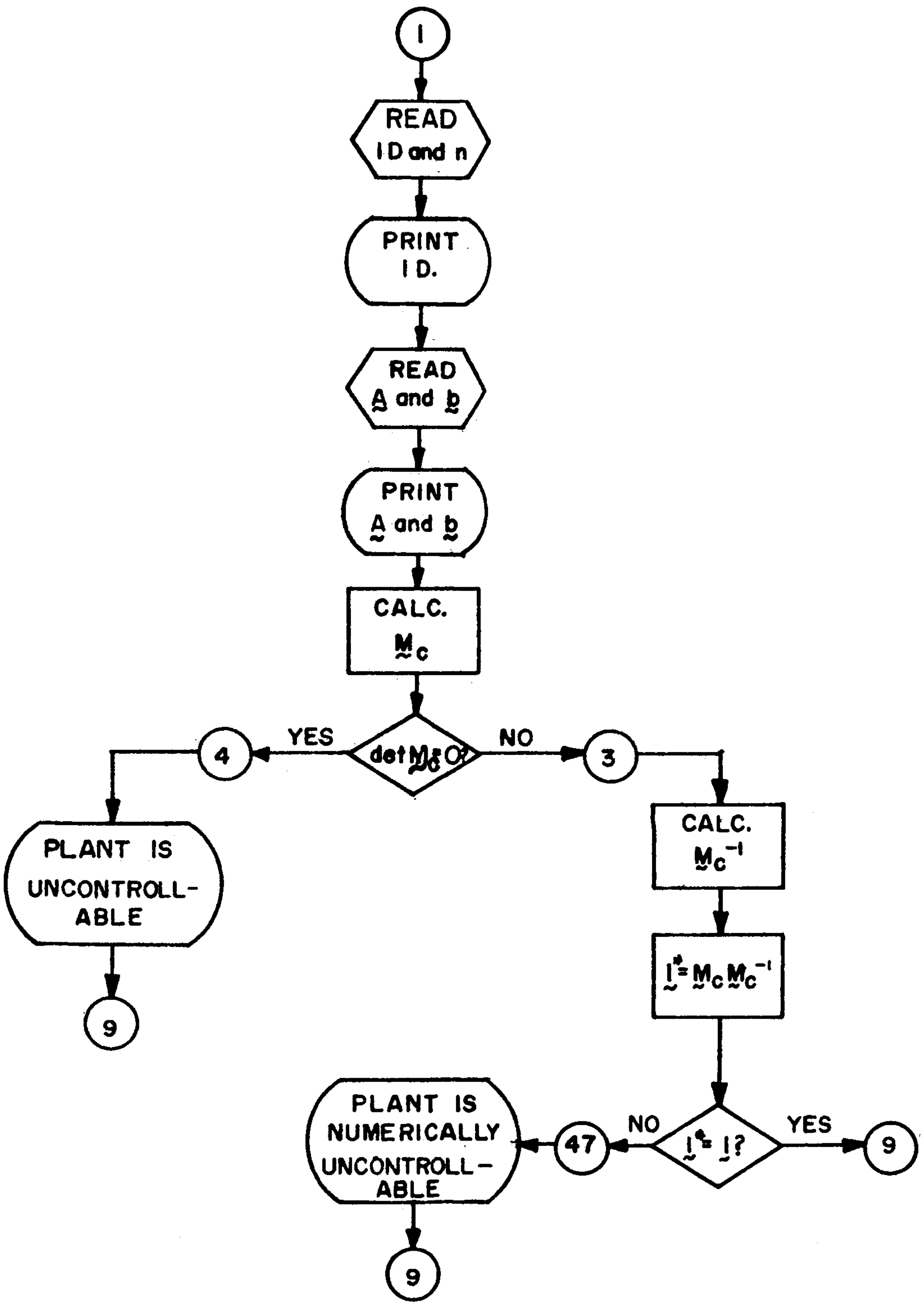
1
READ ID and n
PRINT ID.
READ $\underset{\sim}{A}$ and $\underset{\sim}{b}$
PRINT $\underset{\sim}{A}$ and $\underset{\sim}{b}$
CALC. $\underset{\sim}{M}_c$
det $\underset{\sim}{M}_c = 0$?
YES
NO
4
3
PLANT IS UNCONTROLL-ABLE
9
CALC. $\underset{\sim}{M}_c^{-1}$
$\underset{\sim}{I}^* = \underset{\sim}{M}_c \underset{\sim}{M}_c^{-1}$
$\underset{\sim}{I}^* = \underset{\sim}{I}$?
NO
YES
47
9
PLANT IS NUMERICALLY UNCONTROLL-ABLE
9

Fig. B-1 Flow Chart for the Basic Phase

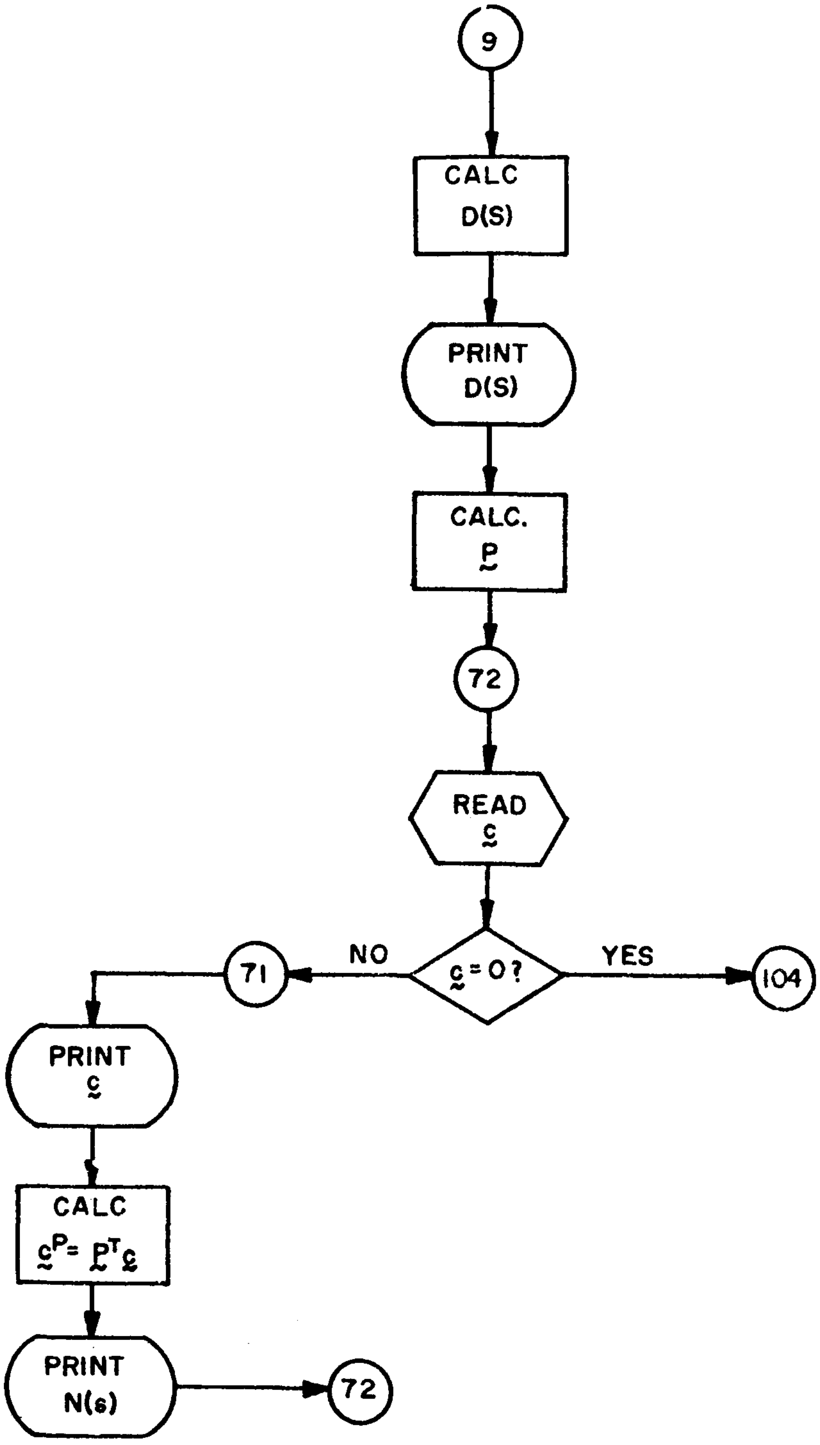
9
CALC D(S)
PRINT D(S)
CALC. $\underset{\sim}{P}$
72
READ $\underset{\sim}{c}$
NO
$\underset{\sim}{c} = 0$?
YES
71
104
PRINT $\underset{\sim}{c}$
CALC $\underset{\sim}{c}^P = \underset{\sim}{P}^T \underset{\sim}{c}$
PRINT N(s)
72

Fig. B-2 Flow Chart for the Open-Loop Phase

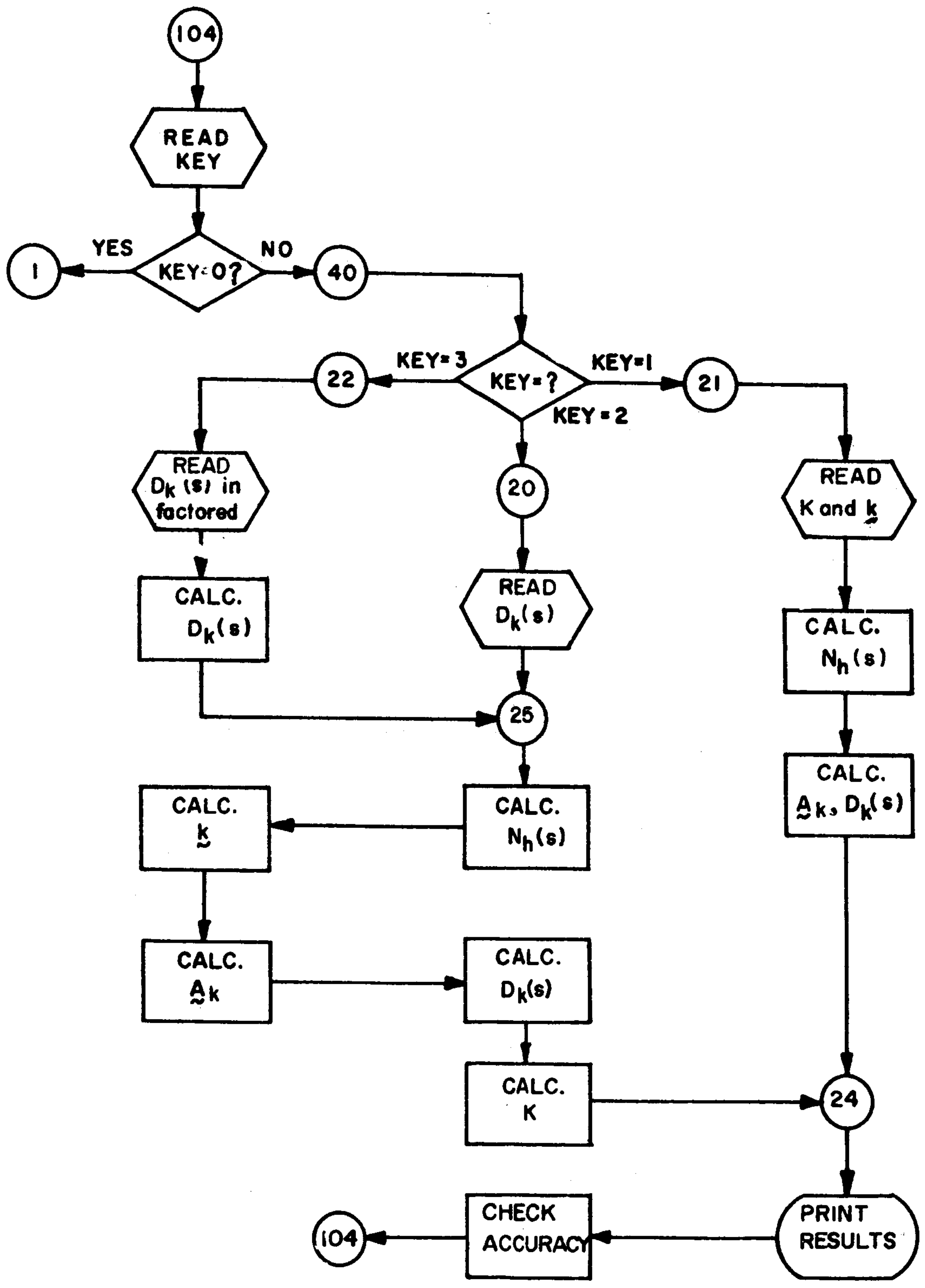

Fig. B-3 Flow Chart for the Closed-Loop Phase

APPENDIX C

PROGRAM LISTING

```
C     STATEVARFDBK
      DIMENSION  A(10,10),B(10),H(10),C(10),NAME(4),AA(10,10),E(10)
      DIMENSION D(10),P(10,10),CC(10),HH(10),PIN(10,10),U(10),V(10)
 2000 FORMAT(45H0    *****************************************)
 2001 FORMAT (4A5,I2)
 2002 FORMAT(8E10.0)
 2003 FORMAT (1H1,5X,14HPROBLEM IDENT.  ,5X,4A5)
 2004 FORMAT(1H0,5X,12HTHE A MATRIX/)
 2005 FORMAT(1P6E20.7)
 2006 FORMAT(1H0,5X,12HTHE B MATRIX/)
 2007 FORMAT(1H0,5X,41HTHE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL/)
 2008 FORMAT(1H0,5X,25HTHE FEEDBACK COEFFICIENTS/)
 2009 FORMAT(1H0,5X,10HTHE GAIN =1PE16.7)
 2010 FORMAT(1H0,5X,12HTHE C MATRIX,5X,5H*****/)
 2011 FORMAT(1H0,5X,24HDEMONINATOR COEFFICIENTS/)
 2012 FORMAT(1H0,5X,22HNUMERATOR COEFFICIENTS/)
 2013 FORMAT(1H0,5X,29HTHE NUMERATOR OF H-EQUIVALENT/)
 2014 FORMAT(1H0,5X,22HOPEN-LOOP CALCULATIONS)
 2015 FORMAT(1H0,5X,26HMAXIMUM NORMALIZED ERROR = 1PE10.2/)
 2016 FORMAT(I1)
 2017 FORMAT(1H0,5X,24HCLOSED-LOOP CALCULATIONS)
 2018 FORMAT(1H0,5X,6HKEY = I1,3X,5H*****)
 2019 FORMAT(1H0,5X, 23HPLANT IS UNCONTROLLABLE5X,10H**********)
 2020 FORMAT(1H0,4X,  35HPLANT IS NUMERICALLY UNCONTROLLABLE10X,
     1 16HMAX. DEVIATION = 1PE10.2,5X,10H**********)
 2021 FORMAT(1H0,5X,14HTHE ROOTS ARE ,13X,9HREAL PART,10X,14HIMAGINARY P
     1ART)
 2022 FORMAT(25X,1P2E20.7)
 2023 FORMAT(1H0)
C     READ INPUT DATA
    1 READ 2001,10,(NAME(I),I=1,4),N
      PRINT 2003,(NAME(I),I=1,4)
      PRINT 2004
      DO 2 I=1,N
      READ 2002,(A(I,J),J=1,N)
    2 PRINT 2005,(A(I,J),J=1,N)
      PRINT 2006
      READ 2002,(B(I),I=1,N)
      PRINT 2005,(B(I),I=1,N)
      NKEY=0
C     CHECK CONTROLLABILITY
      DO 7 I=1,N
    7 AA(I,1)=B(I)
      DO 8 I=2,N
      L=I-1
      DO 8 J=1,N
      AA(J,I)=0.
      DO 8 K=1,N
    8 AA(J,I)=AA(J,I)+A(J,K)*AA(K,L)
      CONTR=DET(AA,N)
      IF(CONTR) 3,4,3
    4 PRINT 2019
```

```
      GO TO 9
    3 CALL SIMEQ(AA,C,N,PIN,C)
      DO 43 I=1,N
      DO 43 J=1,N
      P(I,J)=0.
      DO 43 K=1,N
   43 P(I,J)=P(I,J)+AA(I,K)*PIN(K,J)
      ERROR=0.
      DO 44 I=1,N
      DO 44 J=1,N
      IF(I-J) 45,46,45
   46 ERR = ABSF(P(I,J)-1.0)
      GO TO 44
   45 ERR = ABSF(P(I,J))
   44 ERROR = MAXF(ERR,ERROR)
      IF(ERROR-1.E-5) 9,47,47
   47 PRINT 2020,ERROR
C     OPEN-LOOP CALCULATIONS
    9 PRINT 2000
      PRINT 2014
      NN=N+1
      PRINT 2011
      CALL CHREQ(A,N,D)
      PRINT 2005,(D(I),I=1,NN)
      CALL PROOT(N,D,U,V,+1)
      PRINT 2021
      PRINT 2022,(U(I),V(I),I=1,N)
      DO11 I=1,N
   11 P(I,N)=B(I)
      DO12JJ=2,N
      DO12 I=1,N
      J=N-JJ+1
      K=J+1
      P(I,J)=D(K)*B(I)
      DO12 L=1,N
   12 P(I,J)=P(I,J)+A(I,L)*P(L,K)
   72 READ 2002,(C(I),I=1,N)
      DO 70 I=1,N
      IF(C(I)) 71,70,71
   70 CONTINUE
      GO TO 104
   71 PRINT 2023
      PRINT 2010
      PRINT 2005,(C(I),I=1,N)
   49 DO13 I=1,N
      CC(I)=0.
      DO13 J=1,N
   13 CC(I)=P(J,I)*C(J)+CC(I)
      DO 100 I=1,N
      M=NN-I
      IF(CC(M))101,100,101
  100 CONTINUE
  101 PRINT 2012
      PRINT 2005,(CC(I),I=1,M)
      M=M-1
      IF(M) 105,105,103
  103 CALL PROOT(M,CC,U,V,+1)
      PRINT 2021
      PRINT 2022,(U(I),V(I),I=1,M)
  105 GO TO 72
C     CLOSED-LOOP CALCULATIONS
  104 READ 2016, KEY
      IF (KEY) 40,1,40
   40 IF(NKEY) 120,120,121
  120 NKEY=1
      PRINT 2000
      PRINT 2017
  121 PRINT 2018,KEY
      GO TO (21,20,22),KEY
   21 READ 2002,GAIN
      READ 2002,(HH(I),I=1,N)
      DO 41 I=1,N
      H(I)=0.
      DO 41 J=1,N
   41 H(I)=H(I)+P(J,I)*HH(J)
      DO 42 I=1,N
      DO 42 J=1,N
   42 AA(I,J)=A(I,J)-GAIN*B(I)*HH(J)
      CALL CHREQ(AA,N,E)
      GO TO 24
   22 DO 23 I=1,N
      DO 23 J=1,N
   23 AA(I,J)=0.
      I=0
   34 I=I+1
      READ 2002,RR,RI
      IF(RI) 30,31,30
   31 AA(I,I)=-RR
      GO TO 35
   30 AA(I,I)=-RR
      J=I+1
      AA(I,J)=-RI
      AA(J,I)=RI
      I=J
      AA(I,I)=-RR
   35 IF(I-N) 34,50,50
   50 CALL CHREQ(AA,N,E)
       GO TO 25
   20 READ 2002,(E(I),I=1,N)
   25 DO19 I=1,N
   19 H(I)=E(I)-D(I)
      CALL SIMEQ(P,C,N,PIN,C)
   18 DO16 I=1,N
      HH(I)=0.
      DO16 J=1,N
   16 HH(I)=HH(I)+PIN(J,I)*H(J)
      DO 5 I=1,N
      DO 5 J=1,N
    5 AA(I,J)=A(I,J)-B(I)*HH(J)
      CALL CHREQ(AA,N,E)
      GAIN =E(1)/CC(1)
      DO26 I=1,N
      HH(I)=HH(I)/GAIN
   26 H(I)=H(I)/GAIN
   24 PRINT 2013
      PRINT 2005,(H(I),I=1,N)
      N1=N-1
      CALL PROOT(N1,H,U,V,+1)
      PRINT 2021
      PRINT 2022,(U(I),V(I),I=1,N1)
   17 PRINT 2008
      PRINT 2005,(HH(I),I=1,N)
      PRINT 2009,GAIN
      PRINT 2007
      PRINT 2005,(E(I),I=1,NN)
      CALL PROOT(N,E,U,V,+1)
      PRINT 2021
      PRINT 2022,(U(I),V(I),I=1,N)
C     CHECK ACCURACY
      ERROR =0.
      DO 6 I=1,N
      ERR=(E(I)-GAIN*H(I)-D(I))/E(I)
      ERR= ABSF(ERR)
    6 ERROR=MAXF(ERROR,ERR)
  106 PRINT 2015,ERROR
      GO TO 104
   10 STOP
      END

C     SUBROUTINEPROOT(N,A,U,V,IR)
      SUBROUTINEPROOT(N,A,U,V,IR)
      DIMENSION A(20),U(20),V(20),H(21),B(21),C(21)
      IREV=IR
      NC=N+1
      DO1I=1,NC
    1 H(I)=A(I)
      P=0.
      Q=0.
      R=0.
    3 IF(H(1))4,2,4
    2 NC=NC-1
      V(NC)=0.
      U(NC)=0.
      DO1002I=1,NC
 1002 H(I)=H(I+1)
      GOTO3
  4 IF(NC-1)5,100,5
  5 IF(NC-2)7,6,7
  6 R=-H(1)/H(2)
    GOTO50
  7 IF(NC-3)9,8,9
  8 P=H(2)/H(3)
    Q=H(1)/H(3)
    GOTO70
  9 IF(ABSF(H(NC-1)/H(NC))-ABSF(H(2)/H(1)))10,19,19
 10 IREV=-IREV
    M=NC/2
    DO11I=1,M
    NL=NC+1-I
    F=H(NL)
    H(NL)=H(I)
 11 H(I)=F
    IF(Q)13,12,13
 12 P=0.
    GOTO15
 13 P=P/Q
    Q=1./Q
 15 IF(R)16,19,16
 16 R=1./R
 19 E=5.E-10
    B(NC)=H(NC)
    C(NC)=H(NC)
    B(NC+1)=0.
    C(NC+1)=0.
    NP=NC-1
 20 DO49J=1,1000
    DO21I1=1,NP
    I=NC-I1
    B(I)=H(I)+R*B(I+1)
 21 C(I)=B(I)+R*C(I+1)
    IF(ABSF(B(1)/H(1))-E)50,50,24
 24 IF(C(2))23,22,23
 22 R=R+1.
    GOTO30
 23 R=R-B(1)/C(2)
 30 DO37I1=1,NP
    I=NC-I1
    B(I)=H(I)-P*B(I+1)-Q*B(I+2)
 37 C(I)=B(I)-P*C(I+1)-Q*C(I+2)
    IF(H(2))32,31,32
 31 IF(ABSF(B(2)/H(1))-E)33,33,34
 32 IF(ABSF(B(2)/H(2))-E)33,33,34
 33 IF(ABSF(B(1)/H(1))-E)70,70,34
 34 CBAR=C(2)-B(2)
    D=C(3)**2-CBAR*C(4)
    IF(D)36,35,36
 35 P=P-2.
    Q=Q*(Q+1.)
```

```
      GOTO49
   36 P=P+(B(2)*C(3)-B(1)*C(4))/D
      Q=Q+(-B(2)*CBAR+B(1)*C(3))/D
   49 CONTINUE
      E=E*10.
      GOTO20
   50 NC=NC-1
      V(NC)=0.
      IF(IREV)51,52,52
   51 U(NC)=1./R
      GOTO53
   52 U(NC)=R
   53 DO54I=1,NC
   54 H(I)=B(I+1)
      GOTO4
   70 NC=NC-2
      IF(IREV)71,72,72
   71 QP=1./Q
      PP=P/(Q*2.0)
      GOTO73
   72 QP=Q
      PP=P/2.0
   73 F=(PP)**2-QP
      IF(F)74,75,75
   74 U(NC+1)=-PP
      U(NC)=-PP
      V(NC+1)=SQRTF(-F)
      V(NC)=-V(NC+1)
      GOTO76
   75 U(NC+1)=-(PP/ABSF(PP))*(ABSF(PP)+SQRTF(F))
      V(NC+1)=0.
      U(NC)=QP/U(NC+1)
      V(NC)=0.
   76 DO77I=1,NC
   77 H(I)=B(I+2)
      GOTO4
  100 RETURN
      END

C     SUBROUTINE CHREQ(A,N,C)
      SUBROUTINE CHREQ(A,N,C)
      DIMENSION J(11),C(11),B(10,10),A(10,10),D(300)
      NN=N+1
      DO20 I=1,NN
   20 C(I)=0.
      C(NN) = 1.
      DO14 M=1,N
      K=0
      L=1
      J(1)=1
      GO TO 2
    1 J(L)=J(L)+1
    2 IF(L-M) 3,5,50
    3 MM=M-1
      DO 4 I=L,MM
      II=I+1
    4 J(II)=J(I)+1
    5  CALL FORM(J,M,A,B)
      K=K+1
      D(K)=DET(B,M)
      DO 6 I=1,M
      L=M-I+1
      IF(J(L)-(N-M+L)) 1,6,50
    6 CONTINUE
      M1 = N-M+1
      DO14 I=1,K
   14 C(M1)=C(M1)+D(I)*(-1.)**M
      RETURN
   50 PRINT 2000
 2000 FORMAT (1HO,5X,14HERROR IN CHREQ)
      RETURN
      END

C          FUNCTION DET DETERMINES THE DETERMINATE OF THE  MATRIX 'A'
      FUNCTION DET(A,N)
      DIMENSION A(10,10),B(10,10)
      DO 1 IK=1,N
      DO 1 JK=1,N
    1 B(IK,JK) = A(IK,JK)
      NN = N-1
      D = 1.0
      DO 100 L=1,NN
      LL = L+1
      AMAX = A(L,L)
      IM = L
      JM = L
      DO 15 I=L,N
      DO 15 J=L,N
      IF (AMAX-ABSF(A(I,J))) 10,15,15
   10 IM = I
      JM = J
      AMAX = ABSF(A(I,J))
   15 CONTINUE
      IF(IM-L) 16,20,16
   16 DO 17 J=1,N
      T = A(IM,J)
      A(IM,J) = A(L,J)
   17 A(L,J) = T
      D = -D
   20 IF(JM-L) 21,25,21
   21 DO 22 I=1,N
      T = A(I,JM)
      A(I,JM) = A(I,L)
   22 A(I,L) = T
```

```
      D = -D
   25 DO 30 K1=LL,N
   30 A(L,K1) = A(L,K1)/A(L,L)
      DO 50 J=LL,N
      DO 50 K=LL,N
   50 A(J,K) = A(J,K)-A(J,L)*A(L,K)
  100 CONTINUE
      DO 200 I=1,N
  200 D = D*A(I,I)
      DET = D
      DO 2 IK=1,N
      DO 2 JK=1,N
    2 A(IK,JK) = B(IK,JK)
      RETURN
      END

C     SUBROUTINE FORM(J,M,A,B)
      SUBROUTINE FORM(J,M,A,B)
      DIMENSION A(10,10),B(10,10),J(11)
      DO1 I=1,M
      DO1 K=1,M
      NR=J(I)
      NC=J(K)
    1 B(I,K)=A(NR,NC)
      RETURN
      END

C     SUBROUTINE SIMEQ
      SUBROUTINE SIMEQ (A,XDOT,KC,AINV,X)
      DIMENSION A(10,10),B(10,10),XDOT(10),X(10),AINV(10,10)
      DO1 I=1,KC
      DO1 J=1,KC
      AINV(I,J)=0
    1 B(I,J)=A(I,J)
      DO2 I=1,KC
      AINV(I,I)=1
    2 X(I)=XDOT(I)
      DO3 I=1,KC
      COMP=0
      K=I
    6 IF(ABSF(B(K,I))-ABSF(COMP))5,5,4
    4 COMP=B(K,I)
      N=K
    5 K=K+1
      IF(K-KC)6,6,7
    7 IF(B(N,I))8,51,8
    8 IF(N-I)51,12,9
    9 DO10 M=1,KC
      TEMP=B(I,M)
      B(I,M)=B(N,M)
      B(N,M)=TEMP
      TEMP=AINV(I,M)
      AINV(I,M)=AINV(N,M)
   10 AINV(N,M)=TEMP
      TEMP=X(I)
      X(I)=X(N)
      X(N)=TEMP
   12 X(I)=X(I)/B(I,I)
      TEMP = B(I,I)
      DO13 M=1,KC
      AINV(I,M) = AINV(I,M)/TEMP
   13 B(I,M) = B(I,M)/TEMP
      DO16 J=1,KC
      IF(J-I)14,16,14
   14 IF(B(J,I))15,16,15
   15 X(J)=X(J)-B(J,I)*X(I)
      TEMP=B(J,I)
      DO17 N=1,KC
      AINV(J,N)=AINV(J,N)-TEMP*AINV(I,N)
   17 B(J,N)=B(J,N)-TEMP*B(I,N)
   16 CONTINUE
    3 CONTINUE
      RETURN
   51 PRINT52,I,KC
   52 FORMAT( 18H0 ERROR IN COLUMN I2,2X,9HOF MATRIX ,5X,3HKC=I2//)
      RETURN
      END
```