TIME DOMAIN COMPENSATION

OF LINEAR SYSTEMS

By

JERRY GRAHAM MRAZEK

Bachelor of Science
Oklahoma University
Norman, Oklahoma
1956

Master of Science
Wichita State University
Wichita, Kansas
1964

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1967

TIME DOMAIN COMPENSATION

OF LINEAR SYSTEMS

Thesis Approved:

E. C. Fitch

Thesis Adviser

Paul A. McCollum

K. N. Reid

Charles M. Bacon

D. D. Durham

Dean of the Graduate College

ACKNOWLEDGMENT

I wish to express my sincere appreciation to the people and organizations whose support aided me in the completion of my doctoral program:

A special thanks is extended to Dr. E. C. Fitch, who served as chairman of my graduate committee. His guidance and encouragement have been especially valuable.

My graduate committee, composed of professors Karl N. Reid, Paul A. McCollum, and Charles M. Bacon, whose criticisms and suggestions were of great help.

The Caterpillar Tractor Company, and especially Mr. Frank Winters, for their faith in the judgment of the project personnel and for their financial assistance during this work.

My friends, Dr. James O. Matous, Dr. James E. Bose, and Mr. Stanley Wendt, for their helpful discussions.

Miss Velda Davis and Mr. Allen Ross, for their valuable assistance in the preparation of the manuscript.

Mr. and Mrs. Cecil E. Schultz for their help and understanding throughout this work.

My wife, Kay, and son, Greg, for their sacrifices during my years of graduate study.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure | Page

CHAPTER I

INTRODUCTION

In hydraulic circuits, and indeed in any system where sharp edged disturbances occur, one becomes concerned with the transient behavior of the system. Often, as in the case of hydraulic devices, the main concern is not speed of response, but rather the ability of the system to respond in a smooth and regular fashion. The technique developed in this work provides a means for designing a system to such a criterion. This design approach is carried out in the time domain rather than the much used frequency domain. The time domain is particularly useful in studying the transient response, a time function, because all of the information one needs is available from the state model or one of its forms obtained through a linear transformation. Attempts have been made [11] to determine the character of the transient response through use of the location of the eigenvalues of the system transfer function on the complex plane. Although the eigenvalues, or modes, of the system are important to its transient behavior, this is not sufficient information for a complete description of the relationships governing the transient response. Chapter III will discuss these considerations in detail. The time domain provides some other advantages also. One of the most important of these is that the state model may be obtained directly from a operational block diagram of the system. The gains in each block become elements in the matrices making up the state model. For instance, consider Figure 1. A linearized hydraulic pressure control circuit is shown in block diagram form. This system is used in an example design in Chapter V. A schematic of the circuit is also shown. The state model may be obtained by defining the output of each integrator as a state variable. A series of first order differential equations is then written which relate the time derivative of each state variable to other pertinent elements of the system. These form a matrix differential equation. This form of the basic state model is convenient because each of the elements of the model can be readily interpreted with regard to its physical significance.

The concept of expressing the dynamic characteristics of the system in terms of matrix differential equations coupled with the use of a high speed digital computer provides an extremely powerful tool. The size of system that can be analyzed is not limited by the method but only by the memory capacity of the computer. This is a significant advantage over any of the presently used paper and pencil methods. In addition to its ability to accommodate large systems, however, the computer is capable of

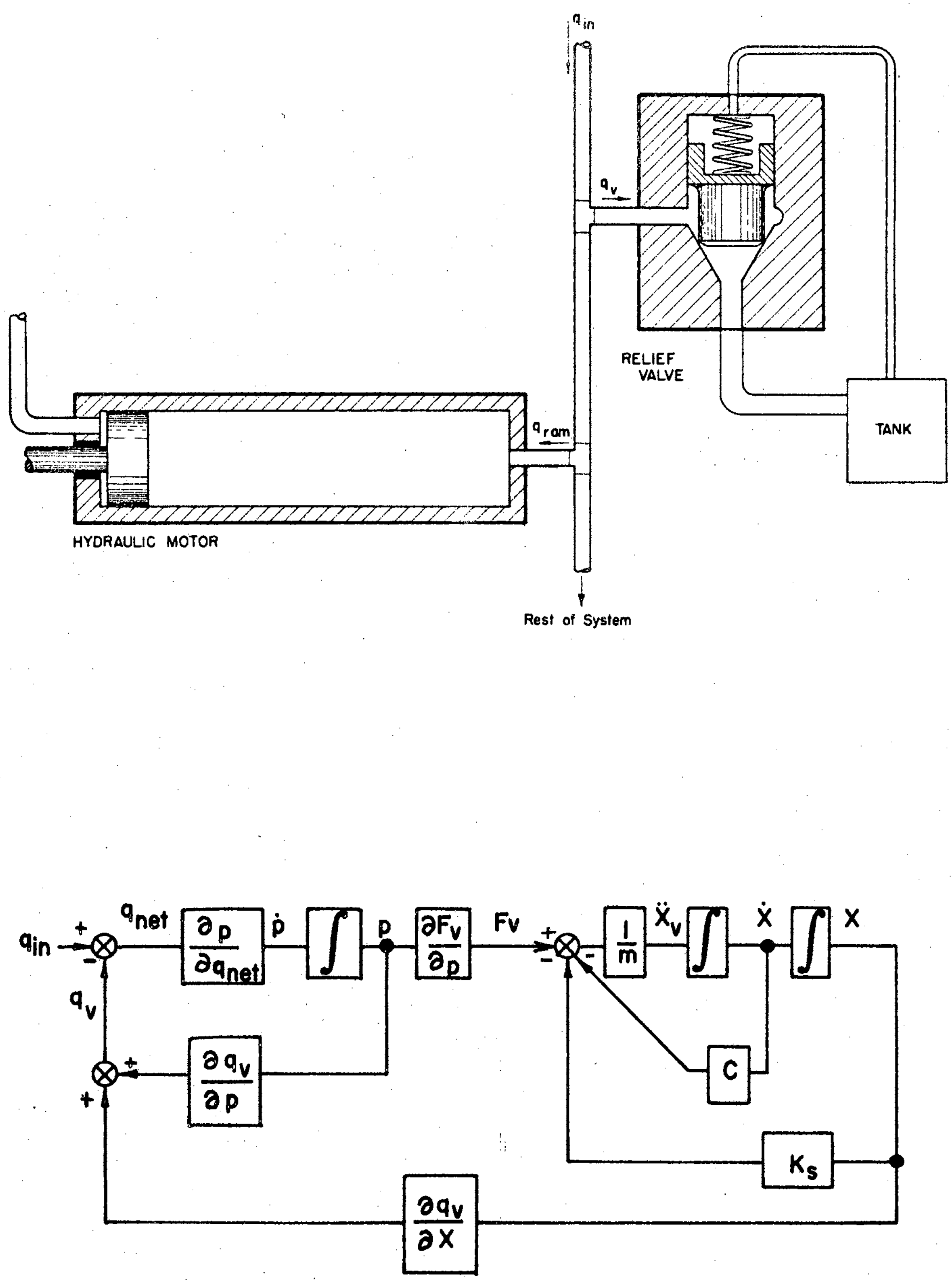


Figure 1. Typical Hydraulic Pressure Control Circuit

performing very quickly, operations that would be extremely laborious by hand. This thought suggests the possibility of a new class of analysis concepts [12]. Normally, troublesome procedures such as matrix inversion, polynomial factoring, matrix transformations, etc., are accomplished easily by the computer.

Some of the features of the approach presented in this thesis are listed below.

a. The mathematical description of the system may be taken directly from an operational block diagram.

b. Magnitude constraints may be placed on any number of elements in the state equations.

c. Additional state variable feedback may be added for system exploration.

d. The performance index used requires no norm to be calculated in order to evaluate the quality of the response.

e. Insight is provided into the influence of each mode of the system response on the total response by observing the components of the performance index attributable to each mode.

There are also some limitations of the method at this time. Only single input/output, linear time invariant systems have been investigated with this method. These systems are also assumed to possess distinct eigenvalues.

This last limitation is not very restrictive since most physical systems will possess distinct eigenvalues. A step is considered as the design input throughout this work. The systems used are to be controllable and observable. More discussion will be forthcoming relative to this in Chapter II.

The design approach presented in this work is particularly useful in industrial applications since it provides an all important link between hardware geometry and the mathematics required in the process of compensating the transient response of a system. The high speed capability of the digital computer permits parametric compensation studies to be made which would be impractical to do by hand. Accuracy of the computer, though sufficient for most design applications, may be increased by modifying parts of the program to double precision computation. The design procedure discussed in this thesis is considered, by the author, to be an early contribution to a new generation of design methods centering about the high speed digital computer.

CHAPTER II

RELATED WORK OF OTHER INVESTIGATORS

This chapter contains brief discussions of work accomplished by other investigators of interest to the work of this thesis. If the reader is interested in pursuing any of these papers in depth, they are all listed in the Selected Bibliography.

Evans, Vigour and Ellert [8]

This paper discusses a parameter optimization design method and its application to hydraulic servo design. The performance index used is composed of a velocity error and several sources of position error of a cutting tool relative to a desired contour. The index has the form

$$I = \int_0^{t_f} f_o(t)dt$$

where $f_o(t)$ is the sum of squared error functions that are designed to result in

1. small path errors,
2. smaller overshoots than undershoots at the corners of the path,
3. no oscillations due to backlash or inadequate

system damping, and

4. little or no flattening of the contour.

The designer must select values of numerous constants associated with the terms of $f_o(t)$. These determine the relative influence of each term and the regions of low and high penalty in the optimization procedure.

In this paper, five system parameters are selected as optimization variables. These include the gains of the position error and valve amplifiers, two time constants associated with a lag-lead network in the system and finally the leakage coefficient of the motor. A steepest descent technique is used to accomplish the minimization procedure. The criterion used in this paper has the advantage of encompassing several performance characteristics but it requires that the designer make numerous arbitrary selections of parameters.

Gall [9]

An optimization criterion, which is capable of simultaneously considering a number of different aspects of the performance of a control system, is described in this paper. This criterion is derived from a ranking array. The array is established by the designer by establishing the most desirable and maximum allowable values of the mean square magnitudes of a group of performance characteristics that are considered to be pertinent. In the second order example of this paper, these characteristics were the acceleration, velocity, displacement, and input force as shown in the array below.

TABLE I

RANKING ARRAY

Desirability	$J(i)$	(i=1) σ_A^2	(i=2) σ_V^2	(i=3) σ_D^2	(i=4) σ_F^2
Most Desirable	0	0	0	0	0
	1	1	8	0.1	10
	2	4	10	0.2	20
	3	9	12	0.3	40
Max. Allowable	4	16	16	0.4	80
	5	100	200	100	500

Mean square values of these characteristics for conditions between the most desirable and maximum allowable are selected arbitarily by the designer. This selection allows the designer to weigh the importance of each parameter. The last row of the array provides an indication of the field of acceptable values of each parameter. These are, by necessity, fairly large since the starting point of the optimization is taken at random. During the optimization, each of the mean square magnitudes discussed above are computed at each trial point in the search space. The ranking of the system is no better than the ranking of the worst parameter. The system is optimized by minimizing J(i), the ranking variable. This is referred to by Gall as a "Max-Ranking" criterion.

Since the partial derivatives of J are discontinuous, the steepest descent minimization approach may not be used. Gall employs a pseudo random search technique that employs a probability function. This technique is capable of including several characteristics in the performance criterion, but it proved to be very time consuming on the digital computer and was difficult to implement with high order systems.

Gustafson [11]

This paper describes an algebraic method of control system design using the system characteristic equation and the Routh array. The method is based on two concepts.

1. The first three time moments of the impulse response of a system may be expressed, using only the last three coefficients of the characteristic equation.
2. The first three frequency moments of the spectral density may be expressed using only the last three elements of the Routh array associated with the characteristic equation.

A truncated transfer function is formed by ignoring all terms in the denominator of the original function except the last three. An associated transfer function is formed by using the last three coefficients of the Routh array in the following way:

$$A(s) = \frac{b_o}{R_{n-1,1}s^2 + R_{n,1}s + R_{n+1,1}} \qquad (2\text{-}1)$$

where b_o is the same as the numerator of the original and the truncated transfer functions. The subscripts of the R's indicate the row and column of the Routh array. Gustafson shows that the integral square impulse response, ISIR of A(s) is identical to that of the original system. An energy ratio is formed that compares the ISIR's of the truncated transfer function to that of A(s). If this ratio is near 1.0, the truncated function is a good approximation to the transfer function of the original system. This approximation is the goal of the method so that the system can be designed as if it were a second order system. An example of a fifth order system indicated that the transient responses of the original system, the truncated function system and the system represented by A(s) are only generally comparable.

Morgan [14]

This paper presents a method for obtaining the transfer function matrix P(s). This represents an alternate approach for arriving at the system response. The development of the transfer function matrix is extended to a discussion of sensitivity. Specifically, this involves determining the sensitivity of P(s) to changes in the matrix elements of the state model. These considerations may be of use to future investigators in extending the work of this thesis. Chapter VI discusses two recommendations for future research that may depend upon a study of sensitivity.

Brockett [5]

Brockett states that if linear state variable feedback is applied to a system, then the system with feedback is controllable if and only if the original system is controllable. The proof of this theorem is based on the fact that if the original system is controllable, then for each initial state x_0 and each desired final state x_f, there is an input u' which drives the system from x_0 to x_f. If such a trajectory is called x', then for a system possessing feedback, the same trajectory can be obtained if an input $u = u' + kx'$ is applied.

It is shown in this paper that observability can be affected by state variable feedback. An example is given in which a system that is originally observable is modified by the addition of linear state variable feedback. A test of the necessary condition for observability ($[C, CA]$ be of rank n) shows that the modification results in a system that is not observable. The reason observability is lost is that a pole of the original system transfer function is moved by the addition of feedback such that a zero is canceled.

Brockett went on to show that through state variable feedback it is possible, ideally, to attain any desired set of eigenvalues. This is only of practical significance when all of the state variables are measurable.

The design procedure of this thesis depends upon the system's being both controllable and observable. Any modifications made to the system by way of state variable feedback must be made with these considerations in mind.

Bacon [1] and [2]

The first of these two papers is concerned with establishing the algebraic constraint equations that provide the necessary and sufficient conditions for a linear time invariant state model to have a given solution. To accomplish this, the state model is first written.

$$\dot{x}(t) = Ax(t) + Bv(t) \tag{2-2}$$

$$y(t) = Cx(t) + Dv(t) \tag{2-3}$$

The state vector x(t) and the system output y(t) are then written as the matrix equations

$$\bar{x}(t) = G_s F_s(t) + G_e F_e(t) \tag{2-4}$$

$$y(t) = N_s F_s(t) + N_e F_e(t) \tag{2-5}$$

where the bar over x(t) indicates that it is the desired response function. G_s, G_e, N_s, and N_e are constant coefficient matrices. $F_s(t)$ is the desired time function associated with the homogeneous solution of the state model. $F_e(t)$ is the time function associated with the input to the system. Differentiation of Equation (2-4) yields an equation that may be combined with Equation (2-2). This combination results in the algebraic constraint equations required such that Equation (2-3) will be equivalent to Equation (2-5). A matrix equation of the following form can now be written:

$$H K_1 = K_2 W \tag{2-6}$$

where H is a matrix containing all of the coefficient matrices of the state model as well as the initial conditions of the state vector x(t). K_1 and K_2 are matrices made up of constant matrices that relate the desired solution to the original state model. W is a matrix made up of constant matrices that associate the desired system output with the state vector and the initial conditions of time functions $F_s(t)$ and $F_e(t)$.

The second of the two papers places constraints on the entries of the matrices in the state model. Certain entries in the A matrix of Equation (2-2) are written as functions of several parameters p. This can be true, in general, of all the constant matrices of the state model. The form that results is

$$\dot{x}(t) = A(P)x(t) + B(P)v(t) \tag{2-7}$$

$$y(t) = C(P)x(t) + D(P)v(t). \tag{2-8}$$

From these equations and the development above, the following expression is obtained:

$$H(P)K_1(G) = K_2(G)W \tag{2-9}$$

where G is a new vector that contains all of the unknown entries of K_1 and K_2.

An error function is established since it will not be possible, in general, to satisfy Equation (2-9) exactly when constraints are imposed. This function has the form

$$Z = H(P)K_1(G) - K_2(G)W \tag{2-10}$$

and the performance index associated with Z is

$$u(P,G) = \sum_{i=1}^{r} \sum_{j=1}^{q} z_{ij}^2 \tag{2-11}$$

where Z is taken as an r x q matrix and z_{ij} is a typical element of Z. The optimum parameter vector P is defined as that vector which together with G minimizes the solution error index u(P,G).

This method has the following advantages:

1. It can accommodate multiple inputs/outputs.
2. It is not restricted to state models with distinct eigenvalues.
3. Specific parameters that make up the elements of the coefficient matrices in the

state model may be constrained.

Some of the disadvantages to using this approach as a general design technique are that one must specify a time response. No procedure is provided that might help the designer make this specification. The functions of p_1 that make up the elements of A, B, C, and D in the state model are likely peculiar to the particular design under investigation and cannot be programmed in a general sense. The computer program used to accomplish this task would require modification before each study.

CHAPTER III

A PERFORMANCE INDEX FOR TRANSIENT RESPONSE

In this discussion, the words "transient response" refer exclusively to the response of a system to a step input unless otherwise specified. The problem of developing a criterion upon which to judge the quality of the transient response is a complex one. This chapter presents an original step response criterion that allows the quality of the response to be expressed as a numerical value. This criterion emphasizes the tendency of a system to overshoot or oscillate. A value of one implies no oscillation or overshoot. A value of less than one implies the presence of oscillation or overshoot. The value of the criterion is referred to in this thesis as a steadiness factor. The design approach of this thesis is not totally dependent upon this particular performance criterion. Many other criteria could be used but the steadiness factor permits the quality of a "nice" response to be expressed quantitatively, a much sought after goal. This capability is of particular use when the physical parameters of a system are established primarily by steady-state rather than dynamic considerations.

It is interesting to compare the attributes of several of the common performance indices. Table II shows a group of these, along with the criterion of this thesis, with their associated individual advantages.

TABLE II

COMPARISON OF PERFORMANCE CRITERIA

Criterion	Reference	Primary Characteristics
$\int_0^\infty \varepsilon^2 \, dt$	16	large errors are penalized more heavily than small errors, although large errors may be tolerated for a short time
$\int_0^\infty \lvert\varepsilon\rvert \, dt$	16	gives more even penalty for large and small errors; easy to implement
$\int_0^\infty t^j \lvert\varepsilon\rvert \, dt$ $j \geq 1$	16	permits heavier weighting of sustained errors; easy to implement
$\int_0^{t_f} f_o(t) \, dt$	8	permits several response characteristics to be included; difficult to implement for a general procedure
Max-Ranking Array	9	permits random inputs; difficult to implement for large general systems
Steadiness Factor	this thesis	permits quantitative description of the quality of the transient response; is easily implemented

Most of these criteria were originally conceived for the purpose of optimizing a design through a minimization of the criterion. The steadiness factor is not used in quite this way. Although a steadiness factor of 1.0 indicates no tendency of the system to overshoot or oscillate, compensating a system to have this steadiness factor could result in sluggish operation. Experience has shown that a small tendency to overshoot can allow the system to respond more rapidly. Consequently, the design goal for steadiness factor is usually not 1.0 but rather a somewhat smaller value, say 0.95. The specific implication of values less than 1.0 will be explained later in this chapter.

Considerations Important to the Transient Response

Attempts have been made to design the response of systems through adjusting the eigenvalues alone. This probably stems from the overwhelming amount of experience and familiarity that most designers of dynamic systems have with first and second order differential equations and root locus techniques. This approach does not result in meaningful information for higher ordered systems or systems possessing zeros in the numerator of the system transfer function. Consider, for example, an open loop system transfer function of the form,

$$G_1(s) = \frac{k}{s(s + 5)} .$$

If unit feedback is applied to this system and the gain increased from zero, the locus of the roots will assume the form shown in Figure 2a. Assume the gain is raised until the eigenvalues are at $-2.5 \pm 3.12j$. This is equivalent to a damping ratio, ζ, of approximately 0.6. The transient response of such a system is shown in Figure 2b. Now, say that another system is designed which has an open loop transfer function of the following form:

$$G_2(s) = \frac{K(s+6)(s+2.7)}{s^2(s+5)} \qquad (3\text{-}1)$$

Now, with unit feedback and increasing the gain, the locus of roots appears as shown in Figure 3a. The previous root locus is shown also for comparison. At some gain the locus of the complex pair of roots arrives at the same end points as in the previous system. Since the time response of a system to a step input is made up of the sum of constants and exponentials of the eigenvalues, it is apparent that real eigenvalues make no direct contribution to oscillatory tendencies in the response. Consequently the complex pairs in both of the examples above make the only direct contributions to overshoot. Figure 3b illustrates that the transient response of the second system possesses significantly more overshoot than the first eventhough the complex eigenvalues in both systems are identical. It will be shown later in this chapter that this is caused by a difference in the phasing of the oscillatory modes.

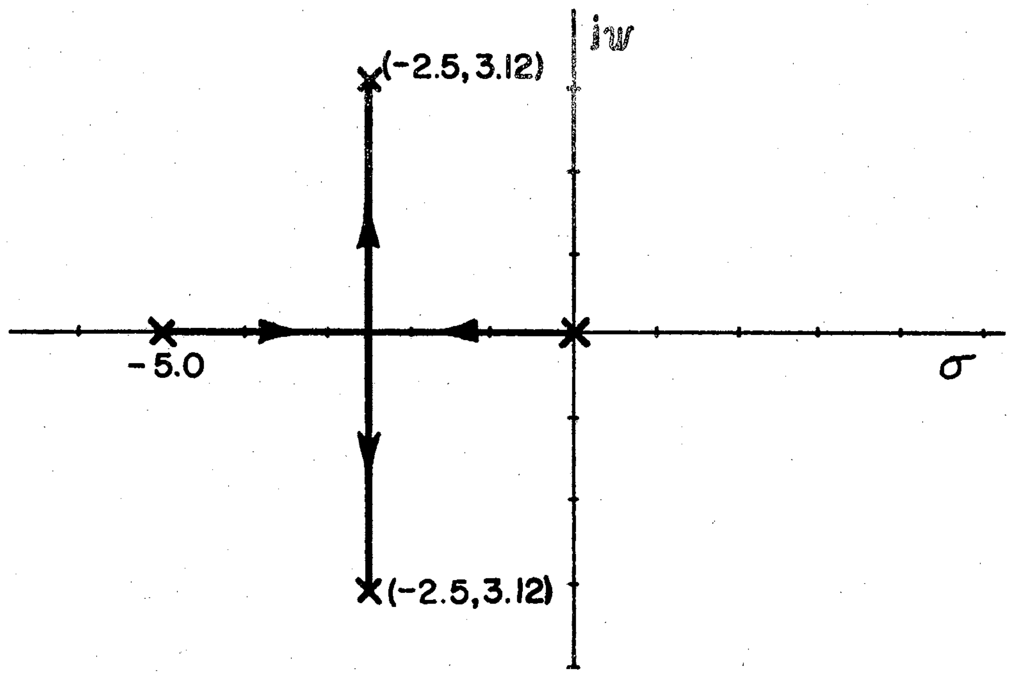


Figure 2a. Root Locus of a Simple Second Order System

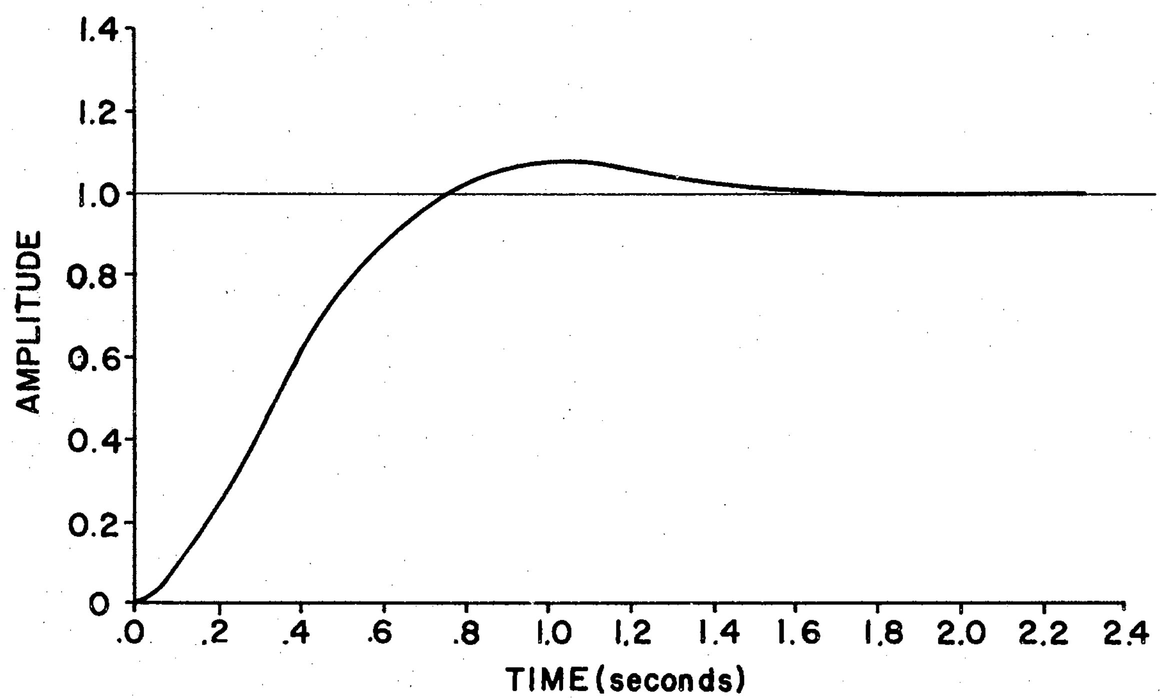


Figure 2b. Transient Response of a Simple Second Order System

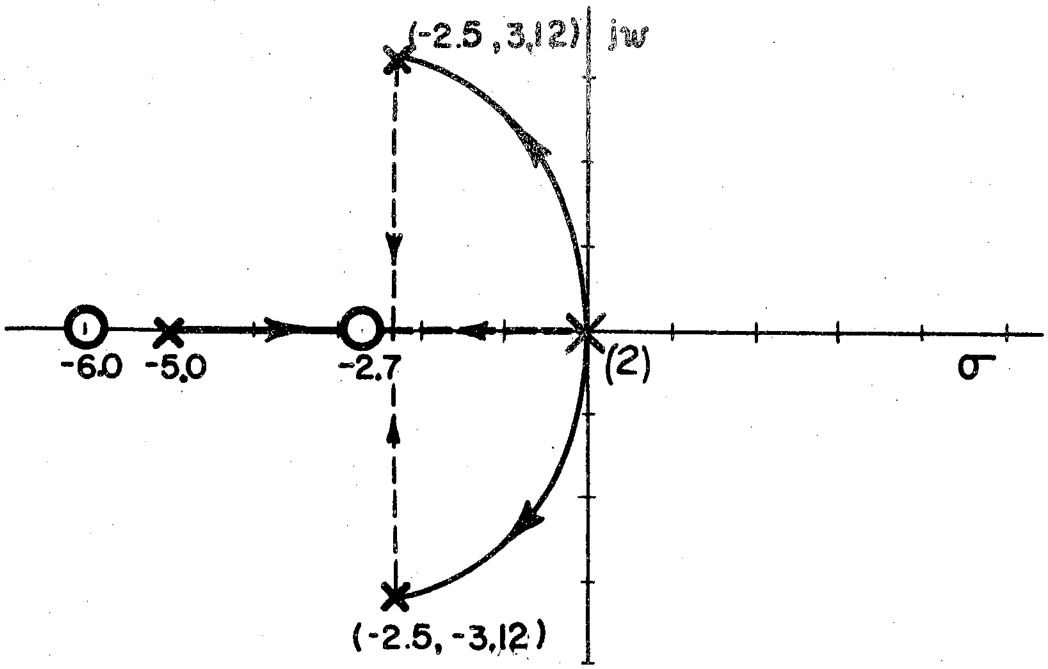


Figure 3a. Root Locus of a Second Order System With Zeros

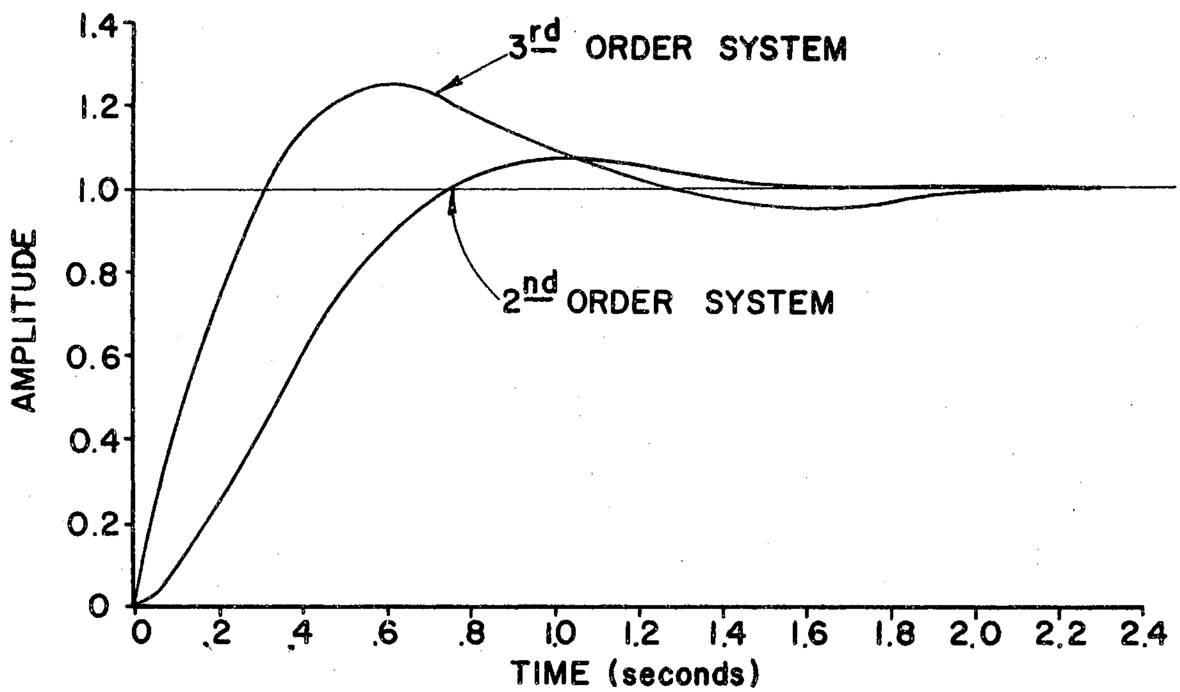


Figure 3b. Transient Response of a Second Order System With Zeros

Development of the Steadiness Factor Criterion

To begin this discussion consider a state model described by the following equations:

$$\dot{x} = Ax + Bv \tag{3-2}$$

$$y = Cx. \tag{3-3}$$

The solution of these equations may be written as follows for a step input:

$$x(t) = \Gamma(t)x(0) + \int_0^t \Gamma(t - \tau)Bvd\tau \tag{3-4}$$

$$y(t) = C\Gamma(t)x(0) + C\int_0^t \Gamma(t - \tau)Bvd\tau. \tag{3-5}$$

If, for simplicity $x(0)$ is allowed to be a zero vector and all coefficient matrices are constant, then Equation (3-5) may be written

$$y(t) = \int_0^t C\Gamma(t - \tau)Bvd\tau. \tag{3-6}$$

Again for the sake of simplicity, let C, Γ, and B all be in the normal form such that the modes of the system are uncoupled. Transformation of the coefficient matrices to normal form is discussed in Chapter IV. Now, $\Gamma(t - \tau)$ has the form,

$$\Gamma(t-\tau) = \begin{bmatrix} \Gamma_{11}(t-\tau) & & & & 0 \\ & \Gamma_{22}(t-\tau) & & & \\ & & \Gamma_{33}(t-\tau) & & \\ & & & \ddots & \\ 0 & & & & \Gamma_{nn}(t-\tau) \end{bmatrix}$$

Now, Equation (3-6) may be written,

$$y(t) = \int_0^t \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_n \end{bmatrix} \begin{bmatrix} \Gamma_{11}(t-\tau) & & & & 0 \\ & \Gamma_{22}(t-\tau) & & & \\ & & \Gamma_{33}(t-\tau) & & \\ & & & \ddots & \\ 0 & & & & \Gamma_{nn}(t-\tau) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} v\,d\tau \tag{3-7}$$

$$= \int_0^t \Big[c_1\Gamma_{11}(t-\tau)b_1 v + c_2\Gamma_{22}(t-\tau)b_2 v + c_3\Gamma_{33}(t-\tau)b_3 v + \dots c_n\Gamma_{nn}(t-\tau)b_n v \Big] d\tau$$

but, since $\Gamma_{ii}(t-\tau) = e^{\lambda_i(t-\tau)}$, one may write

$$y(t) = \int_0^t \Big[c_1 b_1 v e^{\lambda_1(t-\tau)} + c_2 b_2 v e^{\lambda_2(t-\tau)} + \dots c_n b_n v e^{\lambda_n(t-\tau)} \Big] d\tau$$

and this expression may be rewritten as follows:

$$y(t) = \int_0^t \Big[c_1 b_1 v e^{\lambda_1(t-\tau)} + c_2 b_2 v e^{\lambda_2(t-\tau)} + \dots c_n b_n v e^{\lambda_n(t-\tau)} \Big] d(t-\tau). \tag{3-8}$$

The integration may now be completed.

Assume for the moment that two of the eigenvalues in Equation (3-8) are complex conjugates, say λ_1 and λ_2. Now, let

$$\lambda_1 = \sigma + j\omega$$

$$\lambda_2 = \sigma - j\omega.$$

Making these substitutions and writing only the pertinent part of Equation (3-8) one obtains

$$y_{1,2}(t) = -\left[\frac{c_1 b_1 v}{\lambda_1} + \frac{c_2 b_2 v}{\lambda_2}\right] + \frac{c_1 b_1 v}{\lambda_1} e^{(\sigma+j\omega)t} + \frac{c_2 b_2 v}{\lambda_2} e^{(\sigma-j\omega)t}. \tag{3-9}$$

Let $\frac{c_1 b_1 v}{\lambda_1} = K_1$ and $\frac{c_2 b_2 v}{\lambda_2} = K_2$, then

$$y_{1,2}(t) = -\left[K_1 + K_2\right] + e^{\sigma t}\left[K_1 e^{j\omega t} + K_2 e^{-j\omega t}\right] \tag{3-10}$$

by applying the exponential definition of sine and cosine one obtains

$$= -\left[K_1 + K_2\right] + e^{\sigma t}\left[(K_1 + K_2)\cos\omega t + (K_1 - K_2)j \sin\omega t\right]. \tag{3-11}$$

Letting $K_1 + K_2 = K_3 \sin\Phi$ and $(K_1 - K_2)j = K_3 \cos\Phi$ for Φ equal to a constant, and applying a familiar trigonometric identity one finally obtains the following well known form for a second order system:

$$y_{1,2}(t) = -[K_1 + K_2] + K_3 e^{\sigma t}\sin(\omega t + \Phi). \tag{3-12}$$

This motion is such that at time, $t_0 = 0$, $\sin(\omega t + \Phi)$ is not necessarily zero but $\sin \Phi$. Since $\sin \Phi = \frac{K_1 + K_2}{K_3}$, it is apparent that at $t = 0$, $y_{1,2}(t) = 0$. The peak values of the second term of Equation (3-12) will occur just prior to the times at which $\omega t + \Phi = \frac{(2n-1)\pi}{2}$, ($n = 1, 2, 3, \ldots$). For a damped system, the first peak is the largest and, therefore, can be used as an indicator of the tendency of this mode to oscillate or overshoot. It can be seen from this equation that even though this mode may be damped the phase angle Φ can cause the modes of the system to be phased such that an overshoot occurs. Figure 4 illustrates this for two normalized cases. These two transient responses have the same damping envelops and the same damped natural frequency. This means that their eigenvalues are identical. Curve A overshoots considerably more than curve B, however. This difference is caused by the difference in phasing, Φ, of the two modes.

A performance index for transient response is now suggested based on the foregoing discussion. Let this index be called a <u>steadiness factor</u>. The steadiness factor for each mode containing a <u>real</u> eigenvalue is defined to be equal to 1.0. For the oscillatory modes, the steadiness factor is defined as follows:

$$SF_i = 1 - \frac{e^{\sigma_i t_p}}{\sin\Phi_i} \qquad (3\text{-}13)$$

where σ is the real part of the eigenvalue pair under consideration and t_p is the time to the first peak. Equation (3-13) comes from rewriting Equation (3-12) as

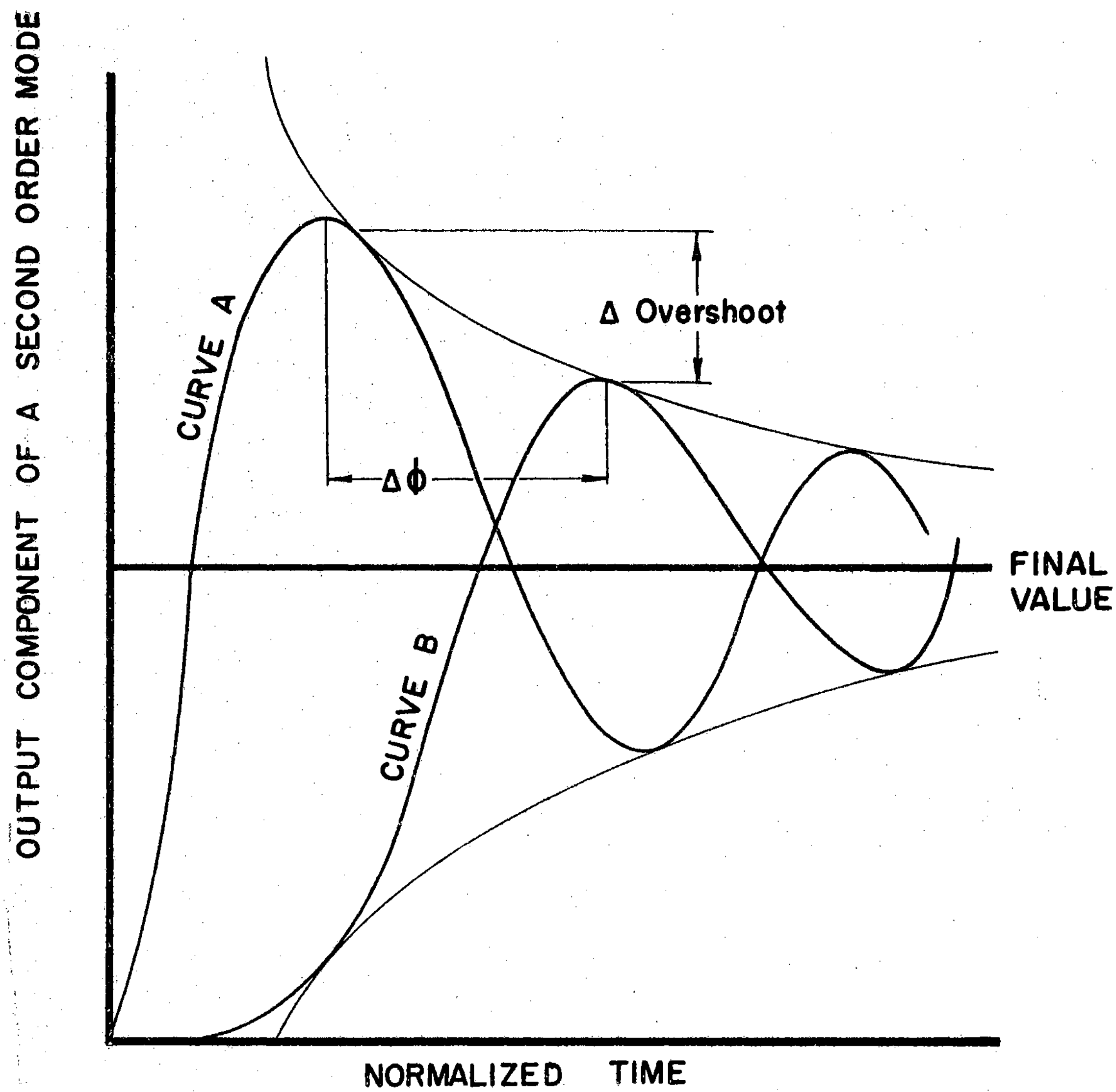


Figure 4. The Effect of Zeros on the Transient Response

$$y_i(t) = (K_1 + K_2)_i\left[1 - \left(\frac{K_3}{K_1 + K_2}\right)_i e^{\sigma_i t}\sin(\omega_i t + \Phi_i)\right] \qquad (3\text{-}14)$$

but $\frac{K_3}{K_1 + K_2} = \frac{1}{\sin\Phi}$, so Equation (3-14) may be written

$$y_i(t) = -(K_1 + K_2)\left[1 - \left(\frac{1}{\sin\Phi_i}\right)e^{\sigma_i t}\sin(\omega_i t + \Phi_i)\right]. \qquad (3\text{-}15)$$

The magnitude of the expression inside the square brackets of Equation (3-15) approaches 1.0 as t approaches infinity. If t is taken to correspond with the time at which $\sin(\omega_i t + \Phi_i) = -1.0$, then Equation (3-15) may be rewritten

$$y_i(t_p) = -(K_1 + K_2)\left[1 + \left(\frac{1}{\sin\Phi_i}\right)e^{\sigma_i t_p}\right] \qquad (3\text{-}16)$$

where t_p is the time discussed above. At t_p, $y_i(t_p)$ is approximately at a peak magnitude of y_i. This is actually the point of tangency of the sinusoid and the exponential of Equation (3-16). Since the steady state value of the expression in the brackets is 1.0, the overshoot of this expression can be expressed as

$$\text{overshoot} = \frac{e^{\sigma_i t_p}}{\sin\Phi_i}.$$

The steadiness factor is then expressed as the difference between the normalized steady state value of a mode's response and the overshoot. The smaller the overshoot becomes, then the closer SF_i approaches 1.0. SF_i will finally be weighted according to the influence of its associate mode in the response of the total system. This is shown in later paragraphs. The time to first peak, t_p, is calculated according to the following equation

$$t_p = \frac{3\pi/2 - \Phi}{\omega} \tag{3-17}$$

where ω is the magnitude of the imaginary part of the eigenvalue. The value, $3\pi/2$ is shown as a typical value in keeping with the requirement that $\sin(\omega t + \theta)$ be -1.0 discussed above. Actually, as the signs of K_1 and K_2 change and as Φ changes, adjustments are necessary to the multiple of $\pi/2$ or to the angle Φ in order to reflect the effect of the first peak. This adjustment is discussed in detail in Appendix C.

A steadiness factor for the complete system can be obtained by applying a weighting factor to each individual steadiness factor. In the interest of clarity let the following definitions hold for the constants associated with the weighting factors:

$$\frac{c_i b_i v}{\lambda_i} \text{ for real } \lambda\text{'s} = K_{i_r}$$

$$\frac{c_j b_j v}{\lambda_j} \text{ for complex } \lambda\text{'s} = K_{j_c}$$

$|K_{j_c} + K_{j_c}^*|$ = the absolute value of the sum of K_{j_c} and its conjugate.

It may be seen from Equation (3-8) that the terms

$\frac{c_i b_i v}{\lambda_i}$ establish the steady state magnitudes of each mode. For the real modes, these are expressed above as K_{ir}. For the complex modes that involve a pair of eigenvalues, the magnitude is given by the absolute value of the sum of the two corresponding K_{jc}'s defined above. The fraction of the total system steady state response that each mode represents can be determined by

$$\frac{K_{ir}}{\Sigma K_{ir} + \Sigma |K_{jc} + K_{jc}^*|} \text{ for real modes}$$

and

$$\frac{|K_{jc} + K_{jc}^*|}{\Sigma K_{ir} + \Sigma |K_{jc} + K_{jc}^*} \text{ for complex modes.}$$

Now the total system steadiness factor may be expressed as follows remembering that the basic steadiness factor of a real mode is 1.0 and that of a complex mode is computed by Equation (3-13):

$$SFT = \frac{\Sigma K_{ir} + \Sigma |K_{jc} + K_{jc}^*| SF_i}{\Sigma K_{ir} + \Sigma |K_{jc} + K_{jc}^*|}. \qquad (3\text{-}18)$$

Each mode is now weighted according to its influence on the total system response. SFT is equal to 1.0 minus the sum of the overshoots of each oscillatory mode taken at its first peak.

The value of SFT obtained from Equation (3-18) may then be compared to a desired value, say 0.95 to determine an acceptable response. Use of 0.95 as a criterion implies that the sum of the first peak overshoots of all the oscillatory modes totals to less than 5% of the total system steady state response. This is a direct measure of the deviation from nonoscillatory response. Chapter IV will discuss the details of adjusting the system in the event that SFT is less than the desired value.

The system associated with the transfer function of Equation (3-1) was used for a test case. The closed loop transfer function of this system may be written

$$\frac{C}{R}(s) = \frac{4.38s^2 + 38.2s + 71}{s^3 + 9.38s^2 + 38.2s + 71}$$

for a K of 4.38. The eigenvalues of this system are

$$\lambda_1 = -4.35$$

$$\lambda_2 = -2.5 + 3.162j$$

$$\lambda_3 = -2.5 - 3.162j.$$

The steadiness factor of the system was determined to be 0.824. The transient response for this system is shown in Figure 5. This implies that the sum of the first overshoots of all the oscillatory modes is approximately equal to 1.0 - 0.824 = 0.176. There is only one oscillatory mode in this case and it is interesting to note that $\frac{\zeta}{\sqrt{1-\zeta^2}} = .79$ from the quotient of the real part divided by the imaginary part of a complex eigenvalue. The above ratio corresponds to a value of ζ of .61. The overshoot associated with a simple (constant numerator) second order

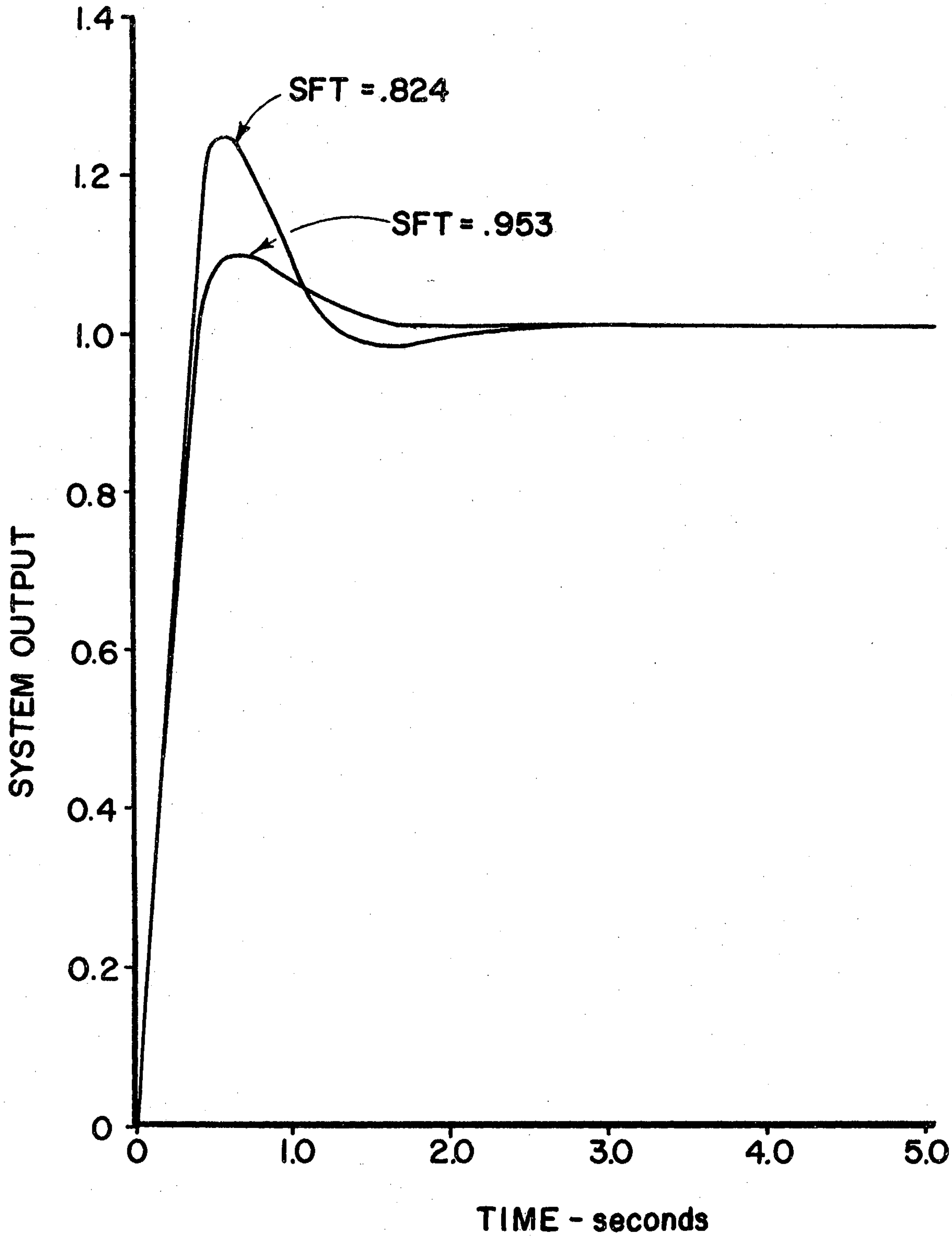


Figure 5. Transient Response of a Test Probelm Possessing Zeros

system with this damping ratio is about 10 percent. The increased overshoot, indicated by the steadiness factor above, is attributed to a difference in phasing of this mode from that of a simple second order system. The phase angle of the system in this test study is 120 degrees. In a simple second order system with the same complex eigenvalues the phase angle would be approximately 52 degrees. The time required for $\omega t + \Phi$ to reach 270 degrees is considerably less for the first case than for the second. Consequently, the multiplier $e^{\sigma t}$ is larger for the first than the second.

This system, compensated by the method of this thesis, yielded a steadiness factor of 0.953. It will be noticed from Figure 5 that although the steadiness factor indicates the overshoot to be approximately 5% of the steady state response, the transient appears to go up to nearly 1.1. This is due to the fact that the steadiness factor is computed relative to the point of tangency of the transient to the decay envelope. In this case, the real part of the complex eigenvalues, given below is large enough such that the tangent point is somewhat to the right of the transient peak.

$$\lambda_1 = -4.4$$

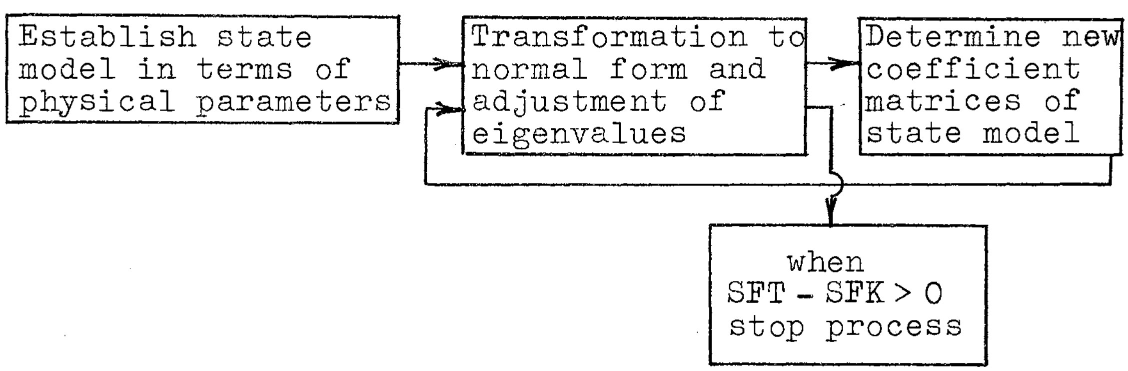
$$\lambda_2 = -3.2 + 2.46j$$

$$\lambda_3 = -3.2 - 2.46j.$$

CHAPTER IV

TIME DOMAIN COMPENSATION TECHNIQUE

This chapter presents a design approach that allows the designer to adjust the elements of his original design such that a transient response criterion is satisfied. A convenient and useful performance index, the steadiness factor, developed in Chapter III, is used in the discussion of this procedure. State space techniques are used in this approach. These techniques have been found to be well suited for implementation on the digital computer. The general operation of this procedure is depicted in the block diagram shown below.



Since this procedure depends upon the formation of a state model and several transformations it was deemed appropriate that the more significant of these operations be discussed. The next section of this chapter includes pertinent comments relative to these concepts. The subsequent section is devoted to a specific discussion of the design approach. Design examples, using this method are given in Chapter V.

Operations Useful to This Design Approach

The differential equations that govern the states of a dynamic system may be obtained from the equations of each component in the system. This approach permits the coefficients and the state variables to be readily interpreted into physical parameters. The differential equations of the components can be easily joined together through the use of an operational block diagram. The output of each integrator is then defined as a state variable of the system x_i. Together, all of the state variables form a state vector. The differential equations of the total system can be written as a matrix differential equation, called in this work "the state equation." The addition of an output equation describing the manner in which each state enters the system output completes the state model. Appendix A includes some detail discussion relative to the formation of a state model.

The state model may be written in the following form:

$$\dot{x} = Ax + Bv \tag{4-1}$$

$$y = Cx \tag{4-2}$$

where x is an n vector, A is an n x n square constant matrix, B is an $n \times 1$ constant matrix and C is a $1 \times n$ constant matrix.

Transformation to Normal Form

Although the form of the equations resulting from the method described above provide a much needed contact with the physical world, this form is not convenient for studying the parameters that more directly describe the transient characteristics of a system. In order to attain a "more convenient" form, a transformation is accomplished. In Chapter III, a performance criterion was discussed. It will be remembered that this criterion was made up of the eigenvalues and some multiplying factors that had the form $\frac{C_i B_i}{\lambda_i}$; $i = 1, 2, \ldots, n$. In the explanation of this criterion, it was assumed that the state model was in the normal form so that the dynamic modes of the system would be uncoupled.

The transformation of a system with distinct eigenvalues from a general form to the normal form is discussed in several texts on state variable analysis, e.g., [7] and [17]. A brief discussion is included here, however, to provide continuity. A transformation matrix must be obtained. In this case, since the transformation is to the normal form such a matrix is the modal matrix. This matrix may be determined by first forming the matrix $[\lambda I - A]$. The determinant of this matrix yields the characteristic polynomial, the zeros of which are the eigenvalues of the system. The following homogeneous equation may be formed:

$$[\lambda I - A]x = 0 \tag{4-3}$$

which comes from the basic concept of transforming a particular vector x into another vector y such that y is proportional to x, [7]. For each of the n eigenvalues $\lambda_i (i = 1, 2, \ldots, n)$ of A, a solution of Equation (4-3) for x can be obtained provided the λ_i's are distinct. The vectors x_i that are solutions of

$$[\lambda_i I - A]x_i = 0 \quad (i = 1, 2, \ldots, n) \tag{4-4}$$

are eigenvectors of A. Each of these eigenvectors makes up one column of the modal matrix. This matrix is not unique since any of the columns may be multiplied by a constant and the transformation is still valid.

The following describes the reasoning through the transformation. Let the original state variable, x, be described in the transformed space as Mq, where M is the modal matrix and q is the transformed state vector.

$$x = Mq. \tag{4-5}$$

Then since $\dot{x} = Ax + Bv$

and $y = Cx$

one obtains $$M\dot{q} = AMq + Bv \tag{4-6}$$

and $$y = CMq. \tag{4-7}$$

Now if both sides of Equation (4-6) are premultiplied by M^{-1}, the result is

$$\mathring{q} = M^{-1}AMq + M^{-1}Bv$$

and

$$y = CMq.$$

In the literature [7], $M^{-1}AM$ is called J, the Jordan canonical form of the coefficient matrix, A. $M^{-1}B$ and CM are often written B_N and C_N, respectively, to imply that they are in the normal form. At this point, it is interesting to note that although constant multipliers of any column of the modal matrix do not affect the transformation of A to J, they have a definite affect on B_N and C_N. The products of corresponding elements of B_N and C_N, however, remain fixed for a given system. The matrix J has the following form:

$$J = \begin{bmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ 0 & & & & \lambda_n \end{bmatrix},$$

a diagonal matrix made up of the eigenvalues. It can be seen that the state model in this form displays all of the parameters required to calculate the performance index, i.e., the steadiness factor. These are, it will be remembered, λ_i; i = 1, 2, ..., n, and $\frac{c_{Ni}b_{Ni}}{\lambda_i}$; i = 1, 2, ..., n. Now, the steadiness factor, SFT, may be calculated by Equation (3-18), repeated below for reference.

$$SFT = \frac{\Sigma K_{i\,r} + \Sigma |K_{j\,c} + K_{j\,c}^*| SF_i}{\Sigma K_{i\,r} + \Sigma |K_{j\,c} + K_{j\,c}^*|}. \qquad (4\text{-}8)$$

Transformation to the Phase Variable Form

Another transformation matrix called the Vandermonde matrix is formed during the design procedure. This matrix has the form shown below.

$$V = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \lambda_3^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

This matrix is a transformation matrix between the normal form of the state model and the phase variable or companion form. The transformation takes the following form:

$$P = VJV^{-1} \qquad (4\text{-}9)$$

$$C_p = C_N V^{-1} \qquad (4\text{-}10)$$

$$B_p = VB_N. \qquad (4\text{-}11)$$

Since the output of the original system expressed by

Equations (4-1) and (4-2) was taken as one of the state variables, x_1, it is possible to obtain a C_p with a similar form. This can be shown by the following argument. By normalizing the columns of the modal matrix M with respect to the first element in each column and expressing C as

$$C = [1 \quad 0 \quad 0 \quad 0 \quad \ldots \quad 0],$$

it follows that

$$CM = C_N = [1 \quad 0 \quad 0 \quad 0 \quad \ldots \quad 0]\begin{bmatrix} 1 & 1 & \ldots\ldots & 1 \\ m_{21} & m_{22} & \ldots\ldots & m_{2n} \\ m_{31} & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ m_{n1} & m_{n2} & \ldots\ldots & m_{nn} \end{bmatrix}$$

$$= [1 \quad 1 \quad 1 \quad 1 \quad . \; . \; . \quad 1].$$

A transformation also may be performed on C_p to obtain C_N.

$$C_pV = C_N = \begin{bmatrix} c_{p_1} & c_{p_2} & c_{p_3} & \ldots & c_{p_n} \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 & \ldots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \ldots & \lambda_n^2 \\ \cdot & & & & \\ \cdot & & & & \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \lambda_3^{n-1} & \ldots & \lambda_n^{n-1} \end{bmatrix}.$$

But $C_N = [1 \quad 1 \quad 1 \quad \ldots \quad 1]$, so if $C_p = [1 \quad 0 \quad 0 \quad \ldots \quad 0]$, the above product will yield the desired C_N. This development implies that one state variable of the companion form can always be made equivalent to one variable of the original system.

As the eigenvalues are changed during the compensation procedure, C_p will remain constant since the top row of the Vandermonde matrix is always made up of ones. B_p will vary in general since B_N will change and V^{-1} will change.

Coefficients of the Characteristic Polynomial

Another operation that is performed during the procedure of this thesis is that of determining the characteristic polynomial corresponding to a given A matrix. This is accomplished through the use of Bocher's formula [7].

For a characteristic equation of the form given by

$$s^n + a_1 s^{n-1} + a_2 s^{n-2} + \ldots a_{n-1} s + a_n = 0, \quad (4\text{-}12)$$

one may determine the coefficients a_i, $i = 1, 2, \ldots, n$ by first applying the following definitions. Let

$$T_1 = \text{trace of } A,$$
$$T_2 = \text{trace of } A^2, \text{ and}$$
$$T_3 = \text{trace of } A^3.$$

The trace of a matrix is the sum of its diagonal elements. Bocher's formula states that the coefficients of the characteristic equation may be computed in the following way:

$$
\begin{aligned}
a_1 &= -T_1 \\
a_2 &= -1/2\,(a_1 T_1 + T_2) \\
a_3 &= -1/3\,(a_2 T_1 + a_1 T_2 + T_3) \\
&\ \ \vdots \qquad\qquad\qquad \vdots \\
a_n &= -1/n\,(a_{n-1} T_1 + a_{n-2} T_2 + \ldots\ a_1 T_{n-1} + T_n).
\end{aligned}
\qquad (4\text{-}13)
$$

Note from Equation (4-13) that the calculation of the n^{th} coefficient may be accomplished progressively by starting with a_1 and working toward a_n, since in each equation for a_k only a's up through a_{k-1} are required in addition to the traces of various powers of the coefficient matrix, A.

The Time Domain Compensation Method

The discussion of this method will begin with the state model. The acquisition of the state model from component differential equations is covered in Appendix A. It is assumed that the designer has formed the state model in such a way that the interpretation of the matrix elements in terms of physical parameters is convenient. Any changes in the system due to compensation of the transient response will be reflected in these elements.

The first operation of the method is that of transforming the original state model to the normal form. This is accomplished through the useful procedures of the previous paragraphs. This form contains all of the necessary information for computing the transient response criterion, steadiness factor. SFT, the total system steadiness factor is computed according to Equation (3-18). This value is then compared with a specification or desired value, SFK, that is selected by the designer. If SFT is larger than SFK, then the specification is satisfied and no compensation is necessary. If SFT is less than SFK, the system must be compensated.

Adjustment of Eigenvalues

The compensation of the transient response of a system is accomplished by adjusting its eigenvalues. In general the system will possess both real and complex pairs of eigenvalues. The adjustment of the real eigenvalues is accomplished differently from that of the complex pairs. Both methods of adjustment are associated with the steepest descent optimization procedure discussed in Appendix B.

The adjustment of the real eigenvalues is accomplished according to the following steps:

1. One real eigenvalue is incremented by a percentage of its original magnitude.
2. A new Vandermonde, V matrix is formed.
3. Considering C_p and B_p to be fixed, V and V^{-1} are used with C_p and B_p to obtain a new B_N and C_N, using the transformations discussed in earlier paragraphs.
4. SFT is recalculated and this value is compared with the original value.

5. If $SFT_2 > SFT_1$, then the increment was in the proper direction and is stored for future use. The eigenvalue is then set back to its original value and another real eigenvalue is incremented. If $SFT_2 < SFT_1$, the sign of the increment is reversed and this increment is stored as before.
6. After the proper direction of each increment is established, a partial derivative of the form $\frac{\partial SFT}{\partial \lambda_i}$ is computed for future use.

These six steps are repeated until increments for all of the real eigenvalues have been determined.

Attention is now focused on adjustment of the complex eigenvalues. A typical pair of complex eigenvalues is shown on the complex plane in Figure 6. Instead of expressing these eigenvalues in terms of real and imaginary Cartesian coordinates, they may be expressed as a radius vector R and an angle θ. The adjustment of the complex eigenvalues is accomplished by incrementing θ by a percentage of its original value. This adjustment policy maintains the undamped natural frequency of each oscillatory mode. As this implies, the emphasis in this procedure is placed on modifying the damping rather than the gain. The sequence of events in establishing the signs of increments for the θ's associated with each oscillatory

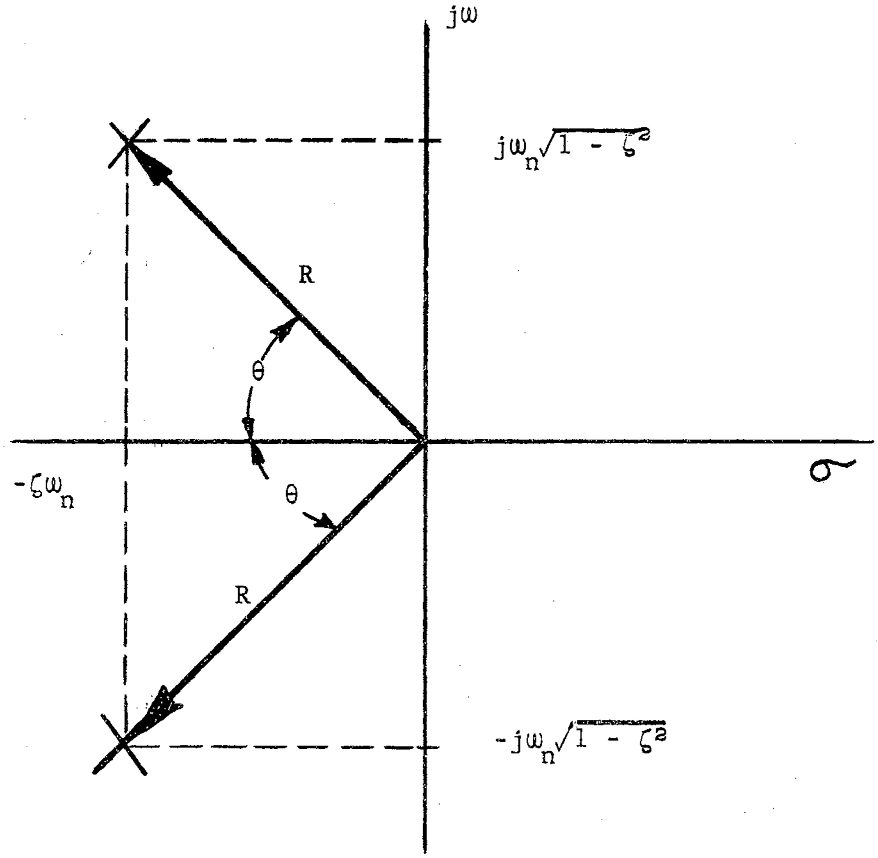


Figure 6. Vector Representation of Complex Eigenvalues

mode is identical to that for the real eigenvalues. The resulting partial derivatives of step 6 take the form $\frac{\partial SFT}{\partial \theta_i}$.

When all of the increment directions and partial derivatives have been established, a group of weighting functions are computed. These weighting functions are actually direction numbers in a space whose orthogonal coordinate axes correspond with each of the increment parameters, the λ_i's in the case of real eigenvalues and the θ_i's for the complex pairs. The weighting functions have the form

$$\frac{d\lambda_i}{ds} = \frac{\partial SFT}{\partial \lambda_i}\left[\sum_{i=1}^{n}\left(\frac{\partial SFT}{\partial \lambda_i}\right)^2\right]^{-\frac{1}{2}} \tag{4-14}$$

where ds is defined as

$$ds^2 = \sum_{i=1}^{n} d\lambda_i^2 .$$

In both of these expressions, let λ_i take on a general meaning including both the real eigenvalues and the θ_i's associated with the complex pairs.

Each of the increments used in the search procedure discussed above is now multiplied by its corresponding weighting factor, Equation (4-14). This results in the set of increments that is added to the original eigenvalues to obtain the adjusted eigenvalues. This completes the adjustment of the eigenvalues for the first cycle through the procedure. The next step is to determine how the A matrix of the original state model must be adjusted to result in these new eigenvalues.

Computation of New A Matrix

Obtaining the new A is not a simple matter of inverse transformation now because the modal matrix used before is no longer valid. The method of obtaining the form of Equations (4-1) and (4-2) reflecting the new eigenvalues involves the following steps.

The characteristic equation, being invariant with similarity transformations, provides a means of numerically determining the modifed A matrix. The coefficients of the characteristic equation of the system with adjusted eigenvalues are displayed in the last row of P, the state coefficient matrix in the companion or phase variable form. P is obtained using J, V and V^{-1} as shown in Equation (4- 9). The coefficients of the characteristic equation for the original system A matrix are determined through the use of Bocher's formula [7]. For simplicity, let the coefficients of the characteristic equation obtained from the A matrix of the original system through the above procedure be denoted by a_i, $i = 1, 2, \ldots, n$. Let the coefficients of desired system equation be denoted by p_i, $i = 1, 2, \ldots, n$.

Now, two characteristic equations are available, $\Delta(a)$ and $\Delta(p)$. The goal is to determine how to adjust the coefficient matrix A such that $\Delta(a)$ approaches $\Delta(p)$ within some tolerance. The following error function is established:

$$\psi = \sum_{i=1}^{n} (a_i - p_i)^2 . \qquad (4\text{-}15)$$

This function is made up of the sum of the squares of the differences between corresponding coefficients of the two characteristic equations. Since the argument, $(a_i - p_i)$ is squared and the a's and p's are real constants ψ will always be positive. If the A matrix is adjusted in exactly the correct manner, ψ will reduce to zero.

Minimization of the Difference Function ψ

The process of minimizing ψ is similar to that used to increase SFT. Let each variable entry of A be denoted by ax_i $(i = 1, 2, \ldots, k)$ where k is the total number of variable entries. The minimization of ψ is accomplished according to the following steps:

1. The first value of ax_i is incremented by a percentage of its original value.
2. A new characteristic polynomial is computed using Bocher's formula.
3. ψ is recomputed and compared with the original value.
4. If $\psi_2 > \psi_1$, the sign of the increment to ax_i is changed and the computation, steps

1, 2, and 3, is repeated. If $\psi_2 > \psi_1$ even after this, the increment is halved, the sign of the increment is returned to its original state and steps 1, 2, and 3 are repeated again. This continues until $\psi_2 < \psi_1$.

5. When $\psi_2 < \psi_1$, the increment in ax_i is stored and a partial derivative of the form $\frac{\partial\psi}{\partial ax_i}$ is computed and stored for future use. The next value of ax_i is then incremented and steps 1, 2, 3, 4, and 5 are repeated until all values of ax_i have been treated.

In step 5, when $\psi_2 < \psi_1$, the second time through the process this implies that the initial step size was too large for the shape of the contour of ψ in that direction. Figure 7 illustrates this condition and the effect of halving the increment.

When all of the increment directions and magnitudes have been established, a set of weighting functions are computed. These perform the same task as those associated with computing the eigenvalue adjustments to increase SFT. A typical weighting function has the form

$$\frac{dax_i}{ds} = \frac{\partial\psi}{\partial ax_i}\left[\sum_{i=1}^{k}\left(\frac{\partial\psi}{\partial ax_i}\right)^2\right]^{-\frac{1}{2}} \qquad (4\text{-}16)$$

where

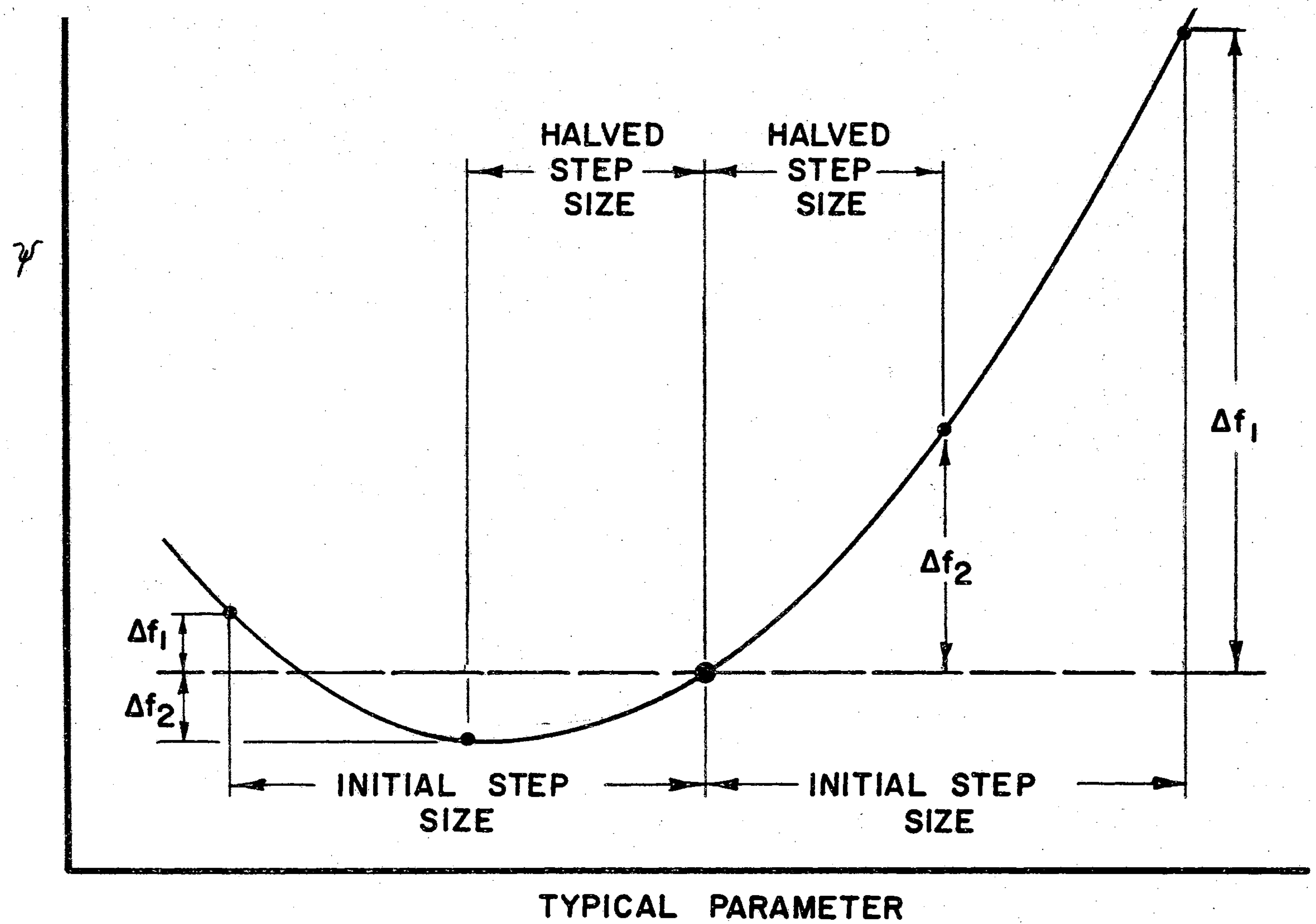


Figure 7. Step Size Adjustment in Descent Procedure

$$ds^2 = \sum_{i=1}^{k} (dax_i)^2 .$$

When the above procedure has been completed each ax_i is incremented, its increment being the increment resulting from steps 1 through 5, multiplied by its appropriate weighting function, Equation (4-16). After each system increment, Ψ is recalculated and if $\Delta\Psi$ is negative, the system is incremented again. This procedure is continued until $\Delta\Psi$ becomes positive or until Ψ is less than some acceptable value. Since this procedure is to be accomplished by a digital computer using finite differences, it is not likely that any criterion can be met exactly. It is necessary, therefore, to establish some criterion of acceptability. One such criterion is discussed later in this chapter. If $\Delta\Psi$ finally becomes positive, then the ax_i's are set back to the values they had just prior to the last increment and a new search for the direction of steepest descent may be started.

The criterion used to stop the descent process is not easily determined in terms of Ψ itself. In the case where constraints are imposed on the ax_i's, Ψ will probably have a minimum at some value other than $\Psi = 0$. A criterion which was used in this work and appears to have the desired characteristics is based on the size of $\Delta\Psi$ during a descent, compared to the original magnitude of Ψ at the beginning of the problem. The criterion states that when

$\Delta\Psi$ is less than the original Ψ divided by 10,000 then an effective minimum has been reached.

When the effective minimum value of Ψ is attained, then the new elements of A are joined with the invariant elements to obtain a new A matrix. This matrix is transformed by the same reasoning as that which was the basis for Equations (4-3), (4-4), and (4-5). The computation then proceeds through the recalculation of SFT as before, but now, since a new A is available, a new modal matrix M is also available. This completes one cycle of the process, the goal of which is to force the performance index, SFT, to satisfy a given criterion. The process is continued until (SFT - SFK) is positive.

A point of interest that deserves mention is that the original incremental values of the ax_i's are arbitrarily selected as a fractional portion of the original ax_i. If, during the descent, the contour of Ψ is well behaved and changes occur slowly, convergence of the procedure is accelerated by increasing the size of the increments. This acceleration is accomplished by first calculating a ficticious angle of descent, θ. θ is defined as follows:

$$\theta = \tan^{-1} \frac{\Delta\Psi}{\Delta s} \qquad (4\text{-}17)$$

where $$\Delta s = \left[\sum_{i=1}^{k} (\Delta ax_i)^2 \right]^{\frac{1}{2}}$$

after the first incremental step of the system $\Delta\theta$ is computed. If $\Delta\theta$ is less than some value, ξ_ℓ, then the increments are doubled. If $\Delta\theta$ is greater than some other value ξ_u, then the increments are halved. If $\xi_\ell \leq \Delta\theta \leq \xi_u$, then the increments are untouched and the computation proceeds.

Engineering Decisions in Adjusting the A Matrix

Some engineering judgment must be exercised in making the decision as to which elements of the A matrix should be variable. For example, a hydraulic circuit designer has little control over the bulk modulus of the oil to be used or even the volume in a power cylinder since this will probably be set by pressure - force - stroke steady state relationships. Adjustments can be made to the spring rates, the valve stem mass, and orifices or capillary tubes may be added to improve valve damping. More complex feedback relationships involving sensing a pressure rate of change may even be possible. Since the A matrix is made up of elements which have physical significance, the designer may decide which of these he wishes to allow to be variable and which should be held fixed. He probably would begin by holding the form of the system fixed and adjusting only those parameters considered to be variable. There also may be physical constraints or bounds on the variation of these elements. These bounds may be imposed by steady state operating requirements or by required manufacturing tolerances, etc. During the adjustment of the ax_i's, care must be taken that these constraints are not violated.

It is recommended when using this design procedure, even if constraints are to be placed on the variable elements of A, that all pertinent entries of the A matrix be allowed to vary at first. In this way, when constraints are imposed, one may judge to what extent each is depreciating the system performance.

An additional word is also pertinent regarding the constraints themselves. At times, the designer may be fortunate in having each variable element of A dependent upon only one physical parameter such as a spring constant. However, it also happens, as shown in Bacon's work [2], that these entries of A can be functions of several parameters. Bacon was able to solve the difficulty by defining the function in terms of variable parameters p_i and then using a vector P made up of all p_i's in an error index computation. In order that this design approach be generally usable for all types of linear systems, only the elements of A were constrained since there is no way of knowing beforehand what type of parametric function might make up these elements.

If such a system were studied with the approach discussed in this thesis, it would be necessary to perform a parametric study holding all but one parameter fixed at a time in each element. In this way, a change in the element could be traced directly to a change in a given parameter. Parametric relationships between elements are considered by holding the related elements fixed while one is varied. The process is then repeated after the related elements are adjusted to correspond parametrically to the variable element.

If constraints are imposed on the elements of A, then it may not be possible to make (SFT - SFK) positive. This situation implies that the resulting SFT is the best that can be expected from the system within the constraints imposed. This is a situation in which it is valuable to have the unconstrained solution at hand for comparison. Often, constraints are set somewhat arbitrarily and can be relaxed if significant good may result. These are design decisions that must be made by the engineer on the basis of information he has available in his mathematical and computer design tools.

There is another realm of investigation that could, in some cases, help the designer to hold his constraints and still gain some improvement in SFT. This involves changing the topology of his system. In most state models there will be zeros appearing in the coefficient matrices. These zeros indicate no dependence of the time derivative of a certain state variable to itself or some other state variable. A dependency may be added artificially by replacing one of these zeros with a nonzero entry and allowing this entry to be variable in the procedure. A parametric study involving all the original zero entries taken one at a time can, in some cases, uncover a relationship that will help the performance of the design. Interpretation of this relationship into hardware design may not always be readily apparent, but it would serve as a stimulant to find such a hardware arrangement if the need warranted.

Summary of the Design Method

The general block diagram shown early in this chapter is expanded in detail in Figure 8. The process begins with an initial state model provided by the designer. The transformation matrix M is formed and is used to transform the state model coefficient matrices to the normal form, J, B_N and C_N. The coefficients K_{r_i} and K_{j_c}, given by the expression $\frac{c_{n_i} b_{r_i}}{\lambda_i}$, are then formed. With these constants, the total system steadiness factor SFT is computed. This value is then compared to a specified goal for steadiness factor SFK. If SFT-SFK is positive, the system already satisfies the specification and there is no need for compensation. If SFT-SFK is negative, compensation is performed.

New eigenvalues are obtained after which a new Vandermonde and Jordon matrix are established. These are used to obtain P through a similarity transformation. This also results in a characteristic polynomial representing the new eigenvalues.

A characteristic polynomial for the original system

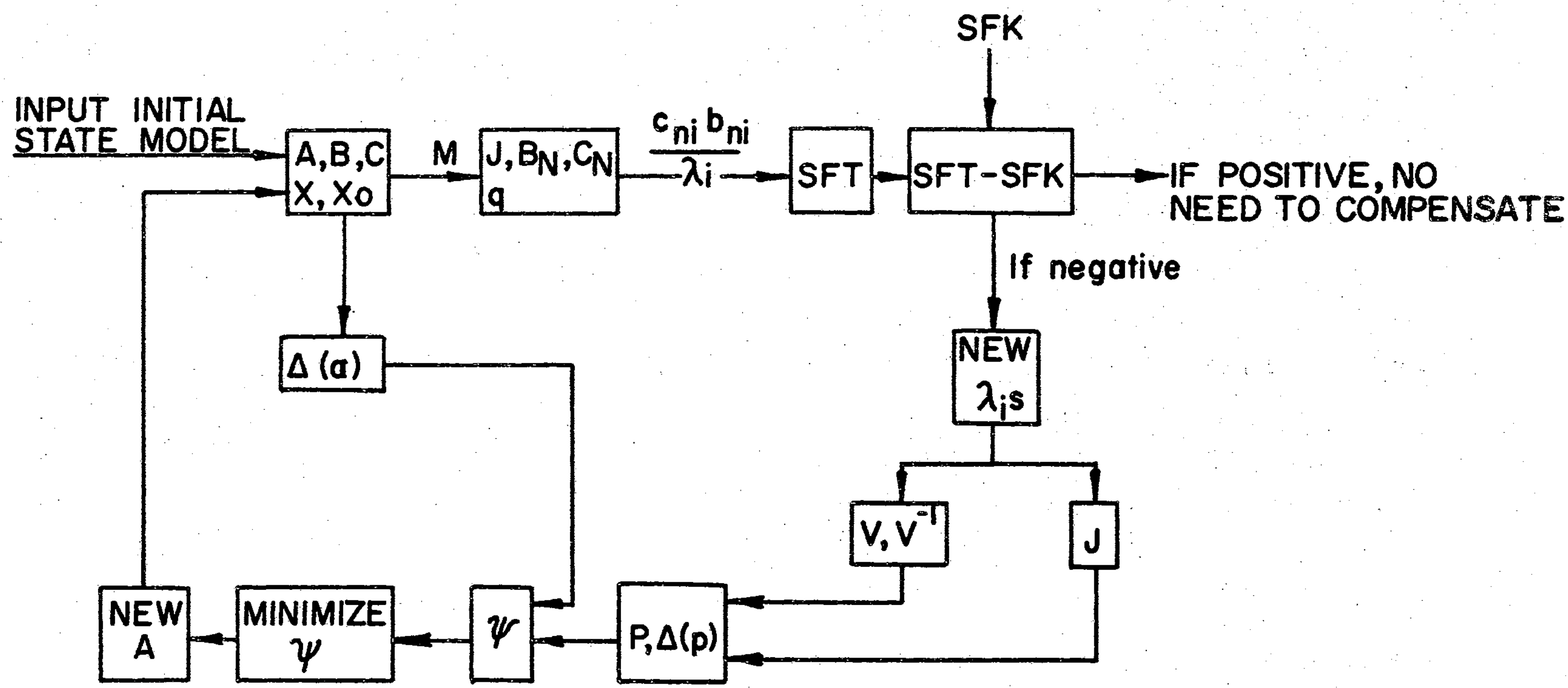


Figure 8. Block Diagram of Compensating Procedure

is obtained using Bocher's formula. The corresponding coefficients of the two characteristic polynomials are combined to form the difference function Ψ. Ψ is minimized through a steepest descents process and yields a new A matrix. This A matrix is entered back at the beginning of the process and SFT is recomputed. This cycle is repeated until SFT - SFK is positive.

The following chapter presents two sample design problems which will demonstrate the practical use of the design approach that has been discussed in this chapter.

CHAPTER V

EXAMPLE DESIGN PROBLEMS

In this chapter, two typical design problems are discussed in order to demonstrate the usefulness of this design approach. The first example is a circuit that appears in many hydraulic systems. This circuit is designed to protect the system from overpressurizing. The mathematical model of this system is developed in Appendix A. The rationale of the design process will be made clear as the procedure progresses.

The second example involves an electro-mechanical positioning system whose characteristics are expressed in a slightly different manner from those of the first example. This is a design problem in which it is desired to raise the system steady state gain in order to reduce the system's load sensitivity while maintaining an acceptable transient characteristic. In both of these examples, the systems are of low order in order to simplify the discussion of the procedure and results. The method is not limited to low order systems, however. The only limitation in this respect would depend upon the memory capacity of the digital computer being used.

Hydraulic Pressure Control Circuit

This circuit was used to illustrate some features of the design approach in Chapter IV. Since a poppet valve has a high flow (i.e., $\frac{\partial Q}{\partial x}$) pressure oscillations often occur. It is the purpose of this discussion to show how a valve, designed for steady state operation may be compensated such that it possesses acceptable transient characteristics.

The particular design considered is shown in Figure 9. The valve stem is retained on its seat by a coil spring. A secondary chamber is provided with a capillary tube outlet to tank to provide additional damping. The valve is opened by the system pressure acting on a differential area formed by stepping the stem down in diameter above the seat. The mathematical models of this valve and the associated ram chamber are discussed in Appendix A.

If the state variables are defined as

$$
\begin{aligned}
x_1 &= p & x_1(0) &= x_{10} \\
x_2 &= x_{valve} & x_2(0) &= x_{20} \\
x_3 &= \dot{x}_{valve} & x_3(0) &= x_{30} ,
\end{aligned}
$$

then the linearized state model for this system becomes

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -K_p & -K_x & 0 \\ 0 & 0 & 1 \\ \frac{A}{m} & -\frac{K_s}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{V_o}{\beta} \\ 0 \\ 0 \end{bmatrix} v \qquad (5\text{-}1)
$$

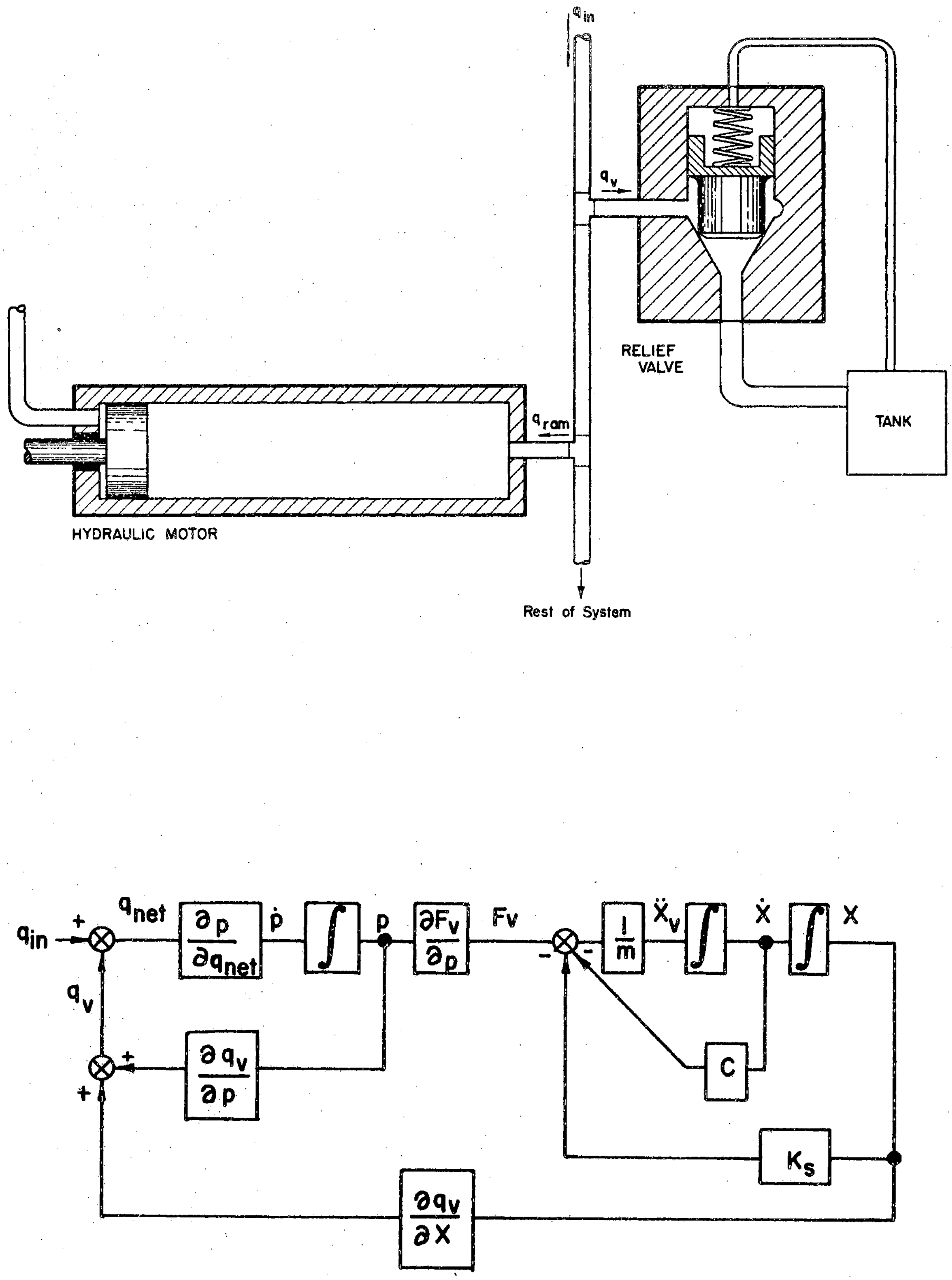


Figure 9. Hydraulic Pressure Control Circuit

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Initial values of the physical parameters resulted in elements of the A and B matrices of

$$K_p = 2.3 \qquad \frac{K_s}{m} = 288000$$

$$K_x = 657000 \qquad \frac{c}{m} = 150$$

$$\frac{A}{m} = 59 \qquad \frac{V_o}{\beta} = 312.5.$$

The following state model results:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -2.3 & -657000 & 0 \\ 0 & 0 & 1 \\ 59 & 288000 & 150 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 312.5 \\ 0 \\ 0 \end{bmatrix} v \tag{5-2}$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{5-3}$$

The eigenvalues of this system are

$$\lambda_1 = -137.69$$

$$\lambda_2 = -\ 7.30 + 535.1j$$

$$\lambda_3 = -\ 7.30 - 535.1j.$$

The steadiness factor for this system is .9396. Even though the eigenvalues indicate the presence of a very lightly damped mode, a relatively high steadiness factor results due to the small weighting coefficient for this mode. In this example

$$|A_1 + A_1^*| = 0.1402$$

and

$$\Sigma A_{r_i} + \Sigma(A_1 + A_1^*) = 2.2828,$$

such that

$$\frac{|A_1 + A_1^*|}{\Sigma A_{r_i} + \Sigma(A_1 + A_1^*)} = \frac{0.1402}{2.2828} = 0.0613.$$

Consequently, even though this lightly damped mode may overshoot its final value nearly 100%, the total system effective "unsteadiness" is only about 6%. The normalized transient response of this system is shown in Figure 10. It will be noticed that this response is quite oscillatory, although the amplitude of the oscillation is relatively small. The design method was applied, letting $\frac{A}{m}$, $\frac{K_s}{m}$, and $\frac{c}{m}$ be variable. The resulting transient response is shown in Figure 10. After compensation by means of adjusting the elements indicated above the steadiness factor became 0.9695 and the response became considerably smoother.

The history of ψ during the operation of this procedure is interesting. Figure 11 shows that after six gradient searches, ψ was very near zero. This figure is a plot of ψ versus the iteration number and gives an indication of the rapidity of convergence.

The original state equation and the revised state equation are presented in Figure 12 for comparison.

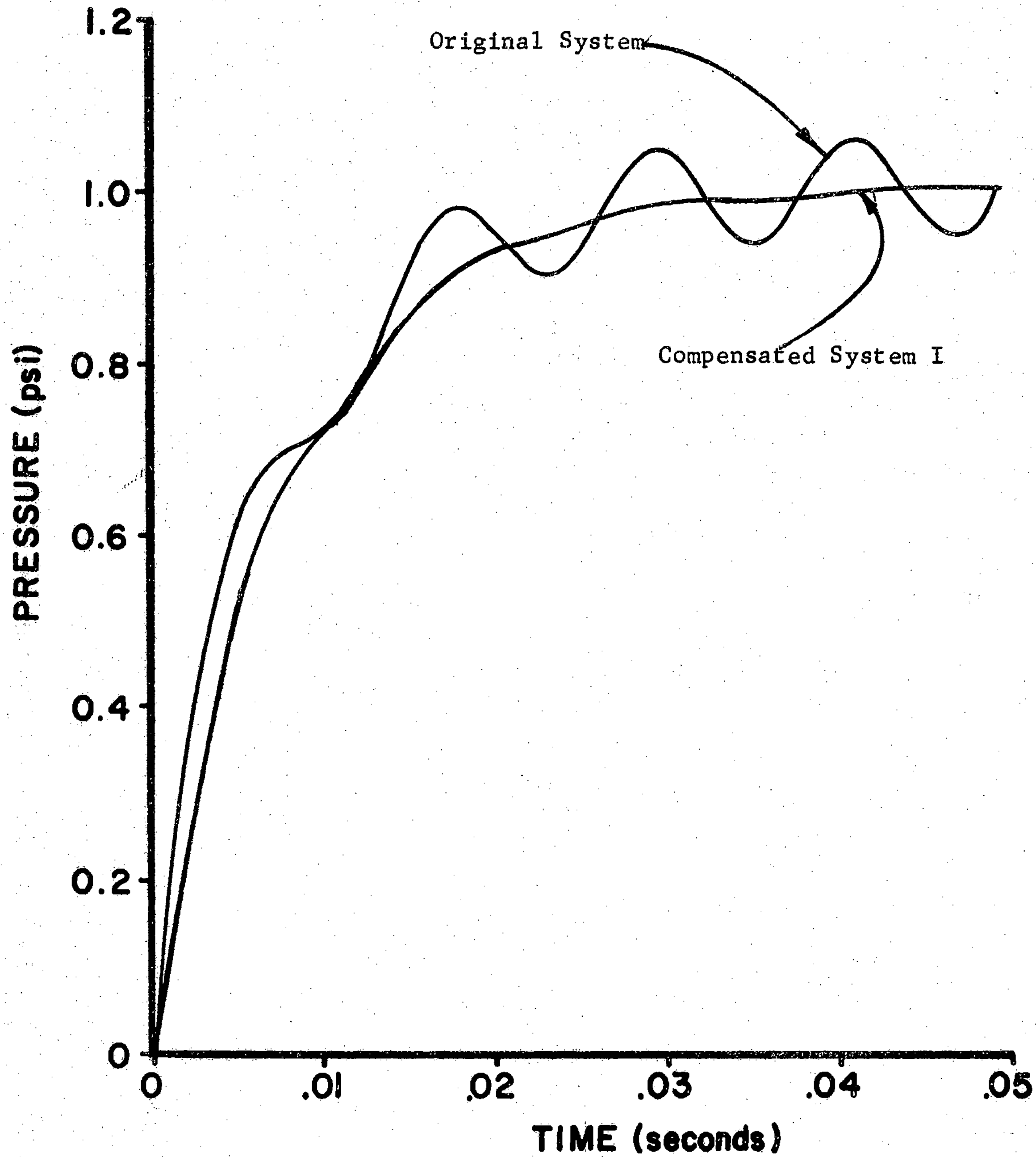


Figure 10. Comparison of Transients of Original and Modified Systems

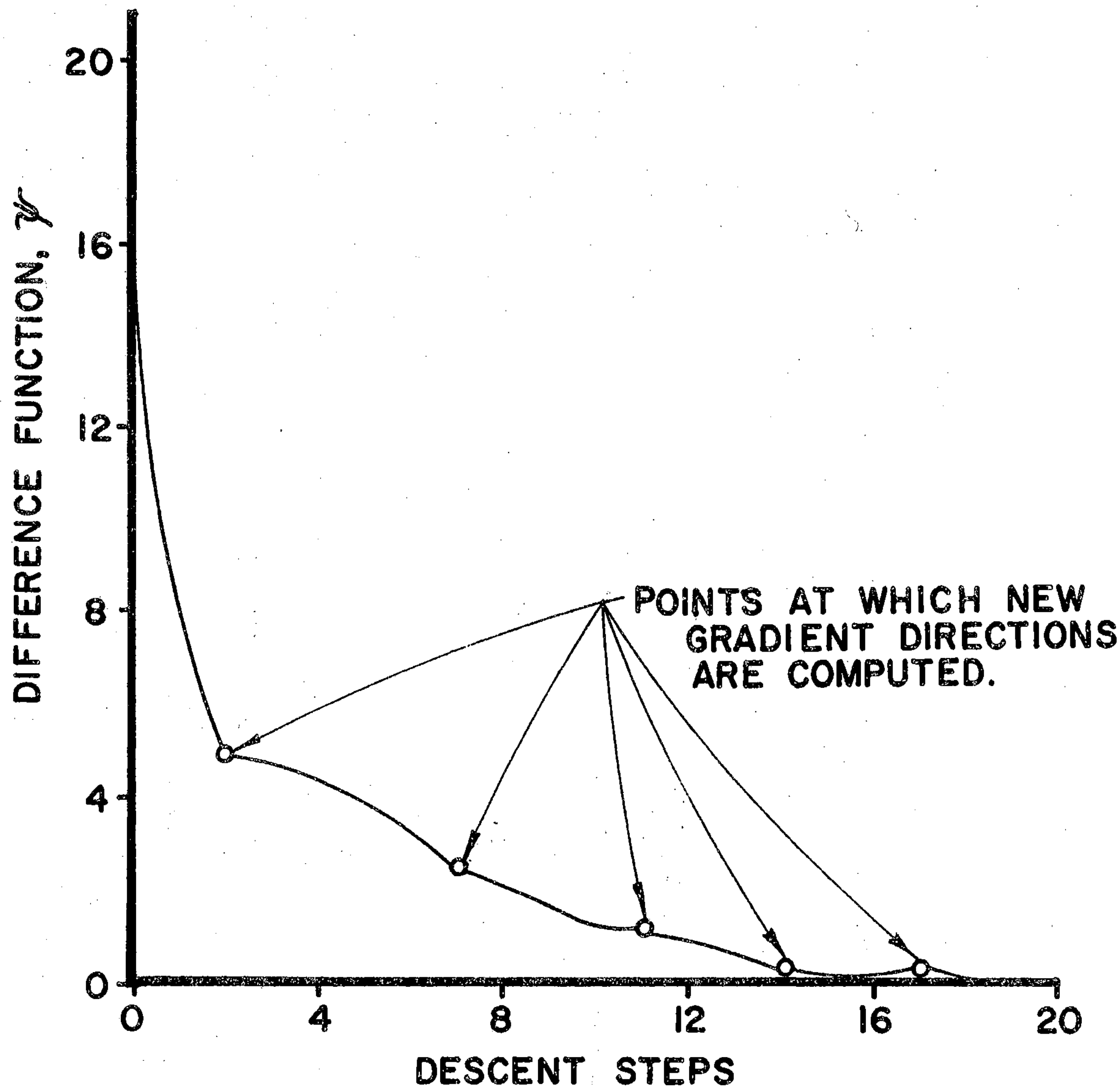


Figure 11. History of ψ During Compensation Procedure

State Equations for the Original System

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \left[\begin{array}{c|c|c} -2.3 & -6.57 \times 10^5 & 0 \\ \hline 0 & 0 & 1 \\ \hline 59 & -2.88 \times 10^5 & -150 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 312.5 \\ 0 \\ 0 \end{bmatrix} v$$

State Equations for the Compensated System I

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \left[\begin{array}{c|c|c} -2.3 & -6.57 \times 10^5 & 0 \\ \hline 0 & 0 & 1 \\ \hline 53.1 & -3.03 \times 10^5 & -282 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 312.5 \\ 0 \\ 0 \end{bmatrix} v$$

Figure 12. Comparison of State Equations of Original and Compensated Systems

Relatively small changes occurred, percentagewise, in some of the variable elements. $\frac{A}{m}$ changed by 10.2%, $\frac{K_s}{m}$ by 5%, but $\frac{c}{m}$ changed by 88%. Now, consider the implications of these changes in terms of hardware geometry. A certain amount of engineering judgment must be used in this part of the procedure. For instance a 10% change in $\frac{A}{m}$ can probably be more easily obtained by adjusting A than by adjusting m. In the case of $\frac{K_s}{m}$, K_s is more easily changed than m. There are several considerations that must be made in considering the change $\frac{c}{m}$. The radial clearance of the valve stem could be adjusted to obtain the larger viscous damping required. However, the radial clearance δ of the original system was .00045 inch. This is a close fit by most standards and to make δ smaller would probably be very expensive.

Another possibility is to change the unsteady flow force contribution. Even if a large enough benefit could be attained using this approach, it is likely that a new housing design would be required. Adjustment of the capillary tube damping augmenter appears to be the most promising approach. Since the diameter of the capillary tube appears in the expression for c to the fourth power, a simple change in the tube diameter results in adequate improvement in the effective damping. This parameter would probably be the easiest to adjust in most practical cases. The following table shows a comparison of the variable parameters which result for this example. Since other compensated systems will be discussed later, this system is identified as compensated system I.

TABLE III

COMPARISON OF VARIABLE PARAMETERS

Parameter	Original System	Compensated System I
A	.0419	.0377
K_s	205.2	195
capil. dia.	.163	.139
Eigenvalues		
λ_1	-137.69	-125.3
λ_2	-7.3 + 535.1j	-79.7 + 526.8j
λ_3	-7.3 - 535.1j	-79.7 - 526.8j
Weighting coefficients		
mode 1	.9395	.95
mode 2,3	.0605	.05

Next consider the effect of constraining some of the parameters in Table III to the extent that they cannot reach the values given. For instance, let $\frac{K_s}{m}$ be constrained to not less than -2.9×10^5, and $\frac{c}{m}$ to not less than -200. This constraint on $\frac{c}{m}$ is somewhat hypothetical but for the sake of illustration consider it to be true. This system is called compensated system II. Figure 13 shows the transient response of such a system with the response of the original system plotted also for comparison. Compensated system II shows an improvement over the original system, but it still displays a considerable tendency to oscillate even though the amplitude is small. The steadiness factor criterion is satisfied since compensated system II has a value of SFT equal to .9569 compared to the specified .95. Obviously, there is a lightly damped mode which has only a small influence on the total response. The result is a slowly decaying small amplitude (less than 5% of final value) oscillation. The lightly damped mode may be seen from the printout of the digital program.

If such a condition is considered to be unsatisfactory by the designer, as well may be the case, one of three attacks may be employed. The first and perhaps most obvious is to relieve the constraints. If this is impossible or impractical, SFK may be raised and an attempt may be made to improve the situation by adjusting free variables further. The third approach is to look for another state variable dependency that will help.

Element a_{13} was permitted to become non zero to test its effect on the steadiness of the response. The result was that this element had no significant effect on the response. The other two zero entries, a_{21} and a_{22}, were

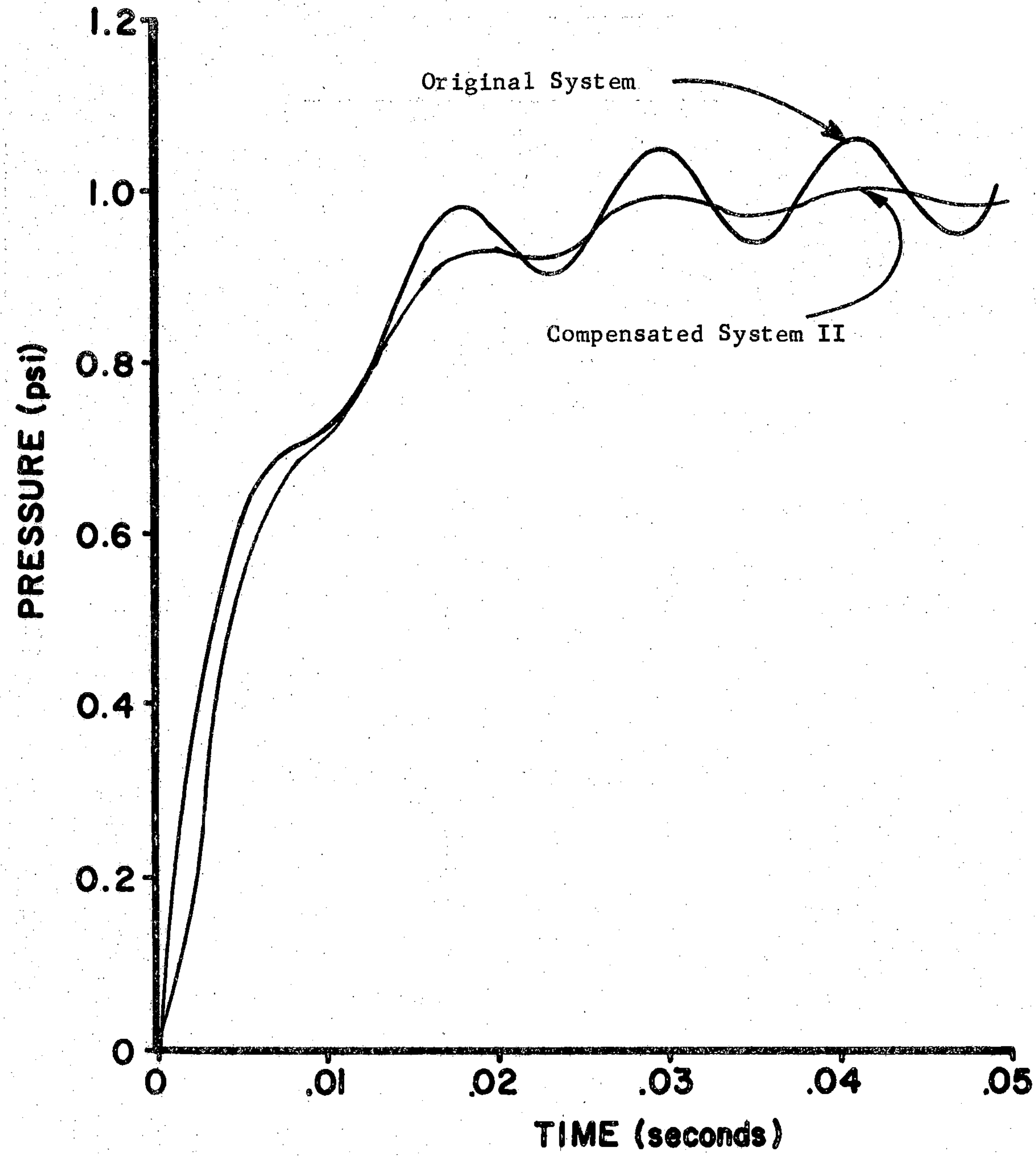


Figure 13. Comparison of Transients of Original and Constrained Systems

the allowed to be variable with similar results.

At this point, a different and more subtle type of constraint in the method should be mentioned. The method of adjusting the eigenvalues, especially in the case of the complex conjugates, is to fix ω_n and rotate the radius vector toward greater damping. This method of variation preserves the general rise time qualities of the system, but it will be recognized that this is not the only method for adjusting the eigenvalues that could be applied. The approach used here is to develop a characteristic equation that has some desired transient characteristic and then to adjust the original system to have the same or nearly equivalent characteristic equation. Adjusting certain elements of A may not have the desired influence on the characteristic equation. This peculiarity is discussed in more detail in Chapter VI in connection with recommendations for future research.

Position Control System

This example problem shows how the design technique may be used in a somewhat different way. Often a machine will be designed for good transient characteristics and later the designer will discover that due to gain level or distribution, the system is overly load sensitive. In order to correct this situation, the design must be re-evaluated. In this example, the original design is established by selecting initial values which appear reasonable and by allowing the design procedure to compute values that will meet the design criterion. A load disturbance is then imposed and the relation between the responses to the two inputs is discussed. The system is then adjusted to reduce load sensitivity and yet maintain the quality of transient behavior desired.

Figure 14 shows a schematic diagram of the system under consideration. A voltage amplifier, Amp, is provided to control an amplidyne circuit. The amplidyne provides power for the armature circuit of a d.c. electric motor. The motor uses a field winding with a constant current i_f to provide the flux field required for the motor operation. The shaft of the motor is attached to an inertia load and also to the wiper of a position sensing potentiometer. The wiper is connected electrically to one terminal of the amplifier. The winding of the pot is connected in parallel with the winding of another pot and with a d.c. voltage source. The wiper of the second pot provides the input position reference. The load disturbance is imposed at the shaft of the motor.

The amplifier is assumed to have a constant gain K_a and no significant dynamics. This gain includes the actual amplifier gain as well as the static gain of the amplidyne.

The purposes of this study are to determine which of the parameters in the system most affects its dynamic performance and to adjust all parameters such that a desired

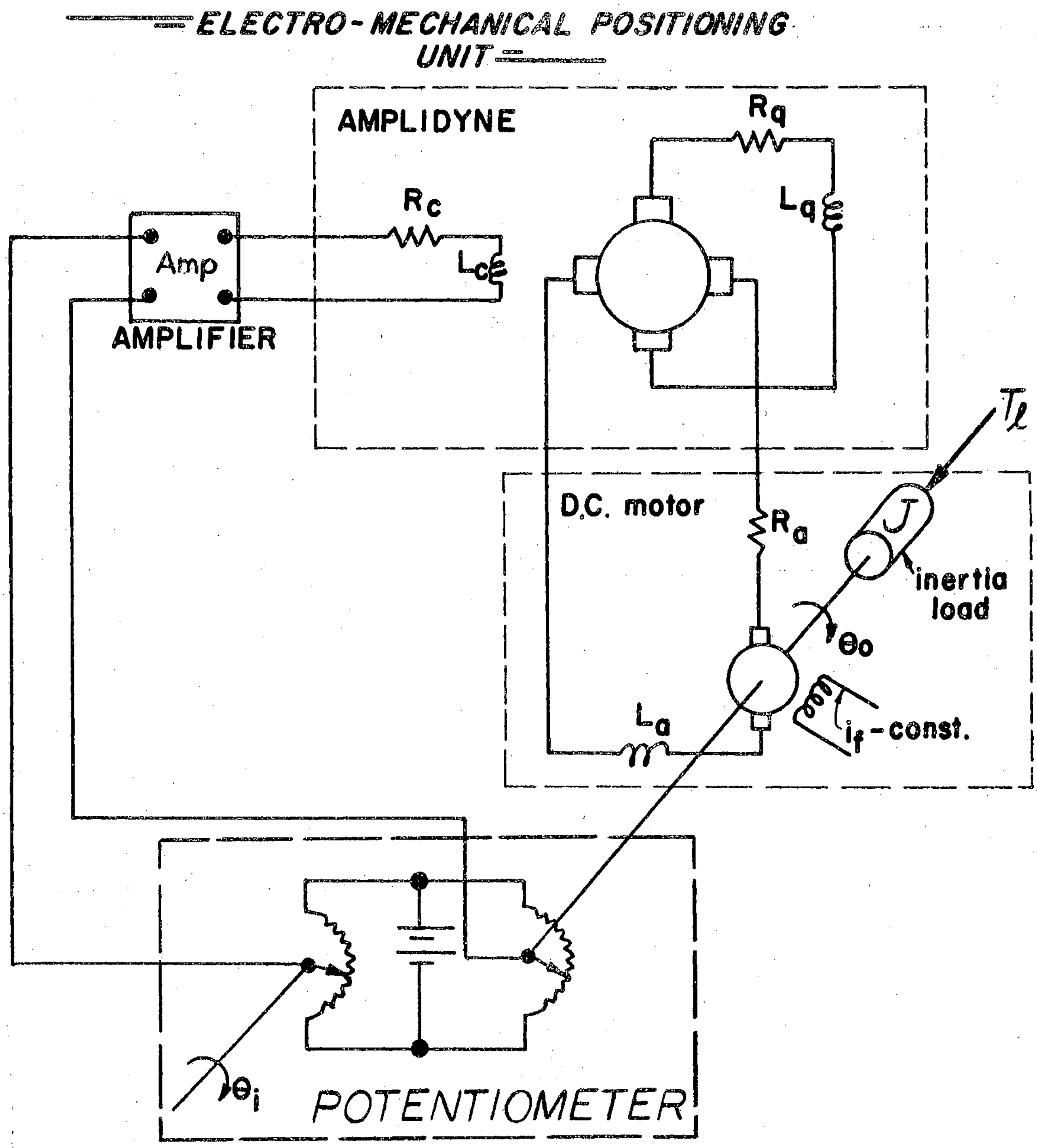


Figure 14. Schematic Diagram of Position Control System

characteristic is achieved. The state model for this system may be established from the basic relations governing the system as developed in Appendix A. The system differential equations become

$$\frac{di_c}{dt} = \frac{-R_c}{L_c} i_c + A(\theta_{in} - \theta_o) \tag{5-4}$$

$$\frac{di_q}{dt} = -\frac{R_q}{L_q} i_q + K_q i_c \tag{5-5}$$

$$\ddot{\theta} = \frac{K_T}{JR_a} (K_d i_q - K_m \dot{\theta}_o). \tag{5-6}$$

Defining

$$\begin{aligned} x_1 &= \theta_o \qquad v = \theta_{in} \\ x_2 &= \dot{\theta}_o \\ x_3 &= i_q \\ x_4 &= i_c, \end{aligned}$$

then the state model may be written

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-K_T K_m}{J\ R_a} & \frac{K_T K_d}{J\ R_a} & 0 \\ 0 & 0 & \frac{-\ R_q}{L_q} & K_q \\ -K_a & 0 & 0 & -\frac{R_c}{L_c} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ A \end{bmatrix} v$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{5-7}$$

The following initial values were used for this example:

$K_T = 0.75$ $\quad \dfrac{R_q}{L_q} = 5$ $\quad \dfrac{e}{\theta_o} = 1.0.$

$K_a = 250$ $\quad R_a = 5$

$\dfrac{R_c}{L_c} = 20$ $\quad K_d = 1$

$J = 0.1$ $\quad K_m = 1.33$

Using these values, the eigenvalues of the original system become

$$\lambda_1 = -\ 6.08$$

$$\lambda_2 = -19.930$$

$$\lambda_3 = -\ 0.493 + 1.69j$$

$$\lambda_4 = -\ 0.493 - 1.69j.$$

The steadiness factor is .6352. Figure 15 shows that the transient response for this system is quite oscillatory. Four iterations of the compensation procedure produced the following eigenvalues:

$$\lambda_1 = -\ 6.82$$

$$\lambda_2 = -20.43$$

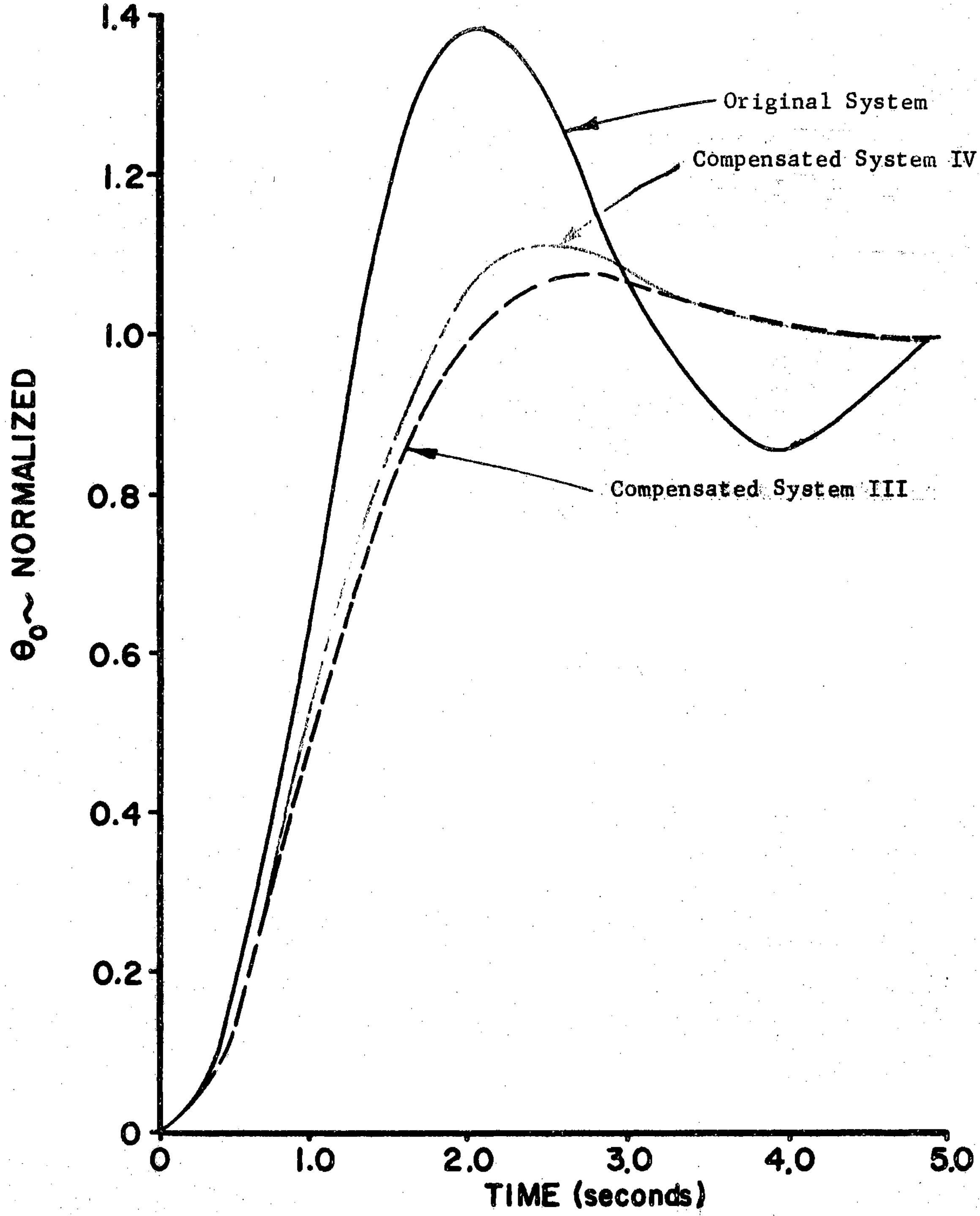


Figure 15. Comparison of Transient Responses

$$\lambda_3 = -1.157 + 1.159j$$

$$\lambda_4 = -1.157 - 1.159j.$$

The steadiness factor for this system called compensated system III is .9731. The transient response for compensated system III is shown in Figure 15.

Only one of the element, a_{22}, of the coefficient matrix, A, varied by a large amount from the original system to compensated system III even though all elements were considered to be variable. This element represents the damping term. Comparison of the two state equations in Figure 16 shows that a_{22} changed by 50% from its original value, but none of the other elements changed more than 15%. Compensated system IV, in Figure 15, shows the effect of varying elements a_{41} and a_{22} only. In this case, a_{41} has virtually no effect on the minimization of ψ. However, a change in element a_{22} from -2.0 to -3.35 accomplishes the necessary compensation.

Load Sensitivity Study

The load sensitivity of the system is investigated in the following manner. Consider Figure 17 showing a second input at the summing junction just ahead of the motor dynamics. This input represents an external torque disturbance to the shaft of the motor. Consider the effect of this input only on the total system response, i.e., let $v_1 = 0$ temporarily. Substituting the initial values

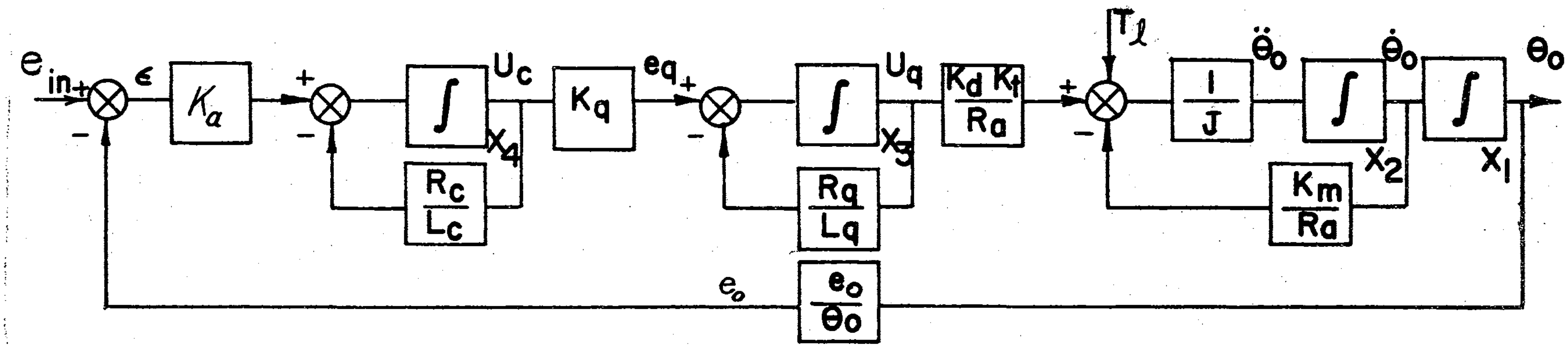
State Equation for Original System

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2 & 1.5 & 0 \\ 0 & 0 & -5 & 1 \\ -250 & 0 & 0 & -20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 250 \end{bmatrix} v$$

State Equation for Compensated System III

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -3.11 & 1.5 & 0 \\ 0 & 0 & -5.75 & 1 \\ -250 & 0 & 0 & -20.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 250 \end{bmatrix} v$$

Figure 16. Comparison of State Equations for Original and Compensated Systems



$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}
=
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & -\frac{K_m}{R_a J} & \frac{K_d K_T}{R_a J} & 0 \\
0 & 0 & -\frac{R_q}{L_q} & K_q \\
(\frac{e_o}{\theta_o})K_a & 0 & 0 & -\frac{R_c}{L_c}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
+
\begin{bmatrix}
0 & 0 \\
0 & \frac{1}{J} \\
0 & 0 \\
K_a & 0
\end{bmatrix}
\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}
$$

$$
y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
$$

Figure 17. Operational Block Diagram and State Model of System With Torque Disturbance

of all parameters except K_a into the state model shown in Figure 17 produces the following steady state relationships,

$$x_3 = -6.67v_2 \tag{5-8}$$

$$-K_a x_1 - 20x_4 = 0, \text{ and} \tag{5-9}$$

$$-5x_3 + x_4 = 0. \tag{5-10}$$

Combining Equations (5-8), (5-9), and (5-10)

$$x_1 = \frac{667}{K_a} v_2 . \tag{5-11}$$

Thus, reduction in the sensitivity of x_1 to v_2 requires an increase of K_a. If K_a has a value of 250, then $\frac{x_1}{v_2} = 2.6$. If K_a is set equal to 12500, then the ratio $\frac{x_1}{v_2}$ will equal .0533. Ideally, a designer wishes the system to be as load insensitive as possible. To accomplish this insensitivity, the value of K_a should be as high as possible within the capabilities of the equipment being used. For the purpose of illustration, a K_a equal to 12,500 is used in the following discussion.

Now that a value of K_a has been selected to reduce the load sensitivity of the system, v_2 is set equal to zero and the response of the system to v_1 is investigated. It might be expected that this system would become oscillatory or even unstable if K_a only were increased. In order to anticipate this difficulty and reduce design time element a_{22} (previously shown to be very influential) was increased by a similar ratio. The initial state model for this increased gain design is the same as that of the original system except for a_{41} and a_{22}. The initial values of these elements are 12,500 and -40, respectively. The resulting steadiness factor is .6947 and the transient response, resulting from an input, v_1, is shown in Figure 18. When the system is compensated, the resulting steadiness factor is .9584 and the transient response (compensated system V) is as shown in the figure. During this compensation, all pertinent entries in the A matrix were allowed to be variable. The resulting changes in these elements can be observed from the state equations shown in Figure 19. It is evident that the matrix element a_{22} is still the most important in the compensation process as might be expected from the previous case.

Since a_{22} represents a damping type term, there are several methods of acquiring the desired values of this parameter. Perhaps the most direct of these methods is the addition of a dashpot or fluid damper. Another possible method is the use of tachometer attached to the shaft of the motor with its output fed to a current amplifier. The current from this amplifier could then be used to supply the field current. Making the field current dependent upon the motor speed allows the torque to be modulated to create an effective damping influence. The question in such an approach would be whether sufficient

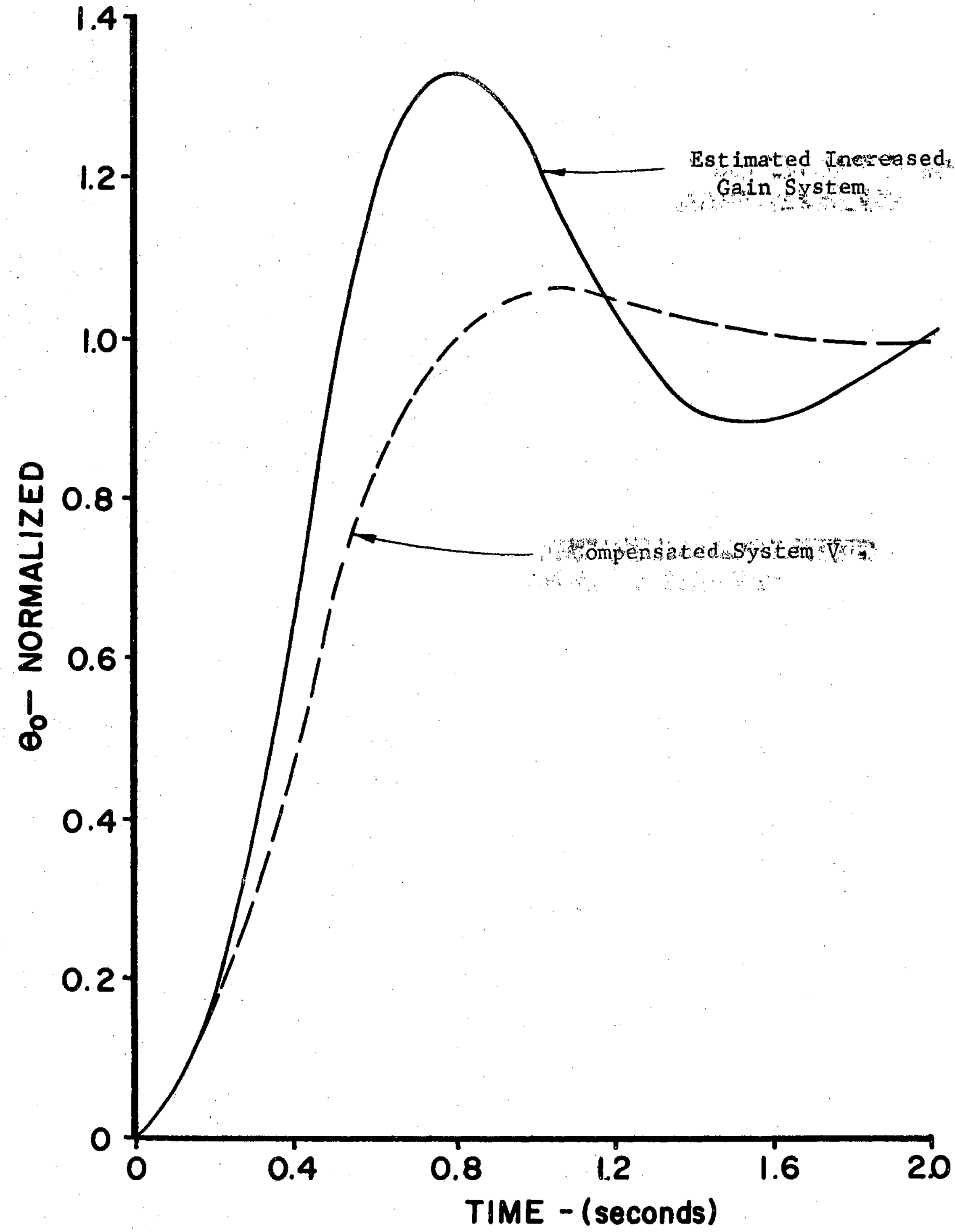


Figure 18. Comparison of Transient Responses for the Increased Gain System

State Equation for Estimated Increased
Gain System

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -40 & 1.5 & 0 \\ 0 & 0 & -5 & 1 \\ -12.5 \times 10^3 & 0 & 0 & -20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12.5 \times 10^3 \end{bmatrix} v_1$$

State Equations for Compensated System V

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -42.5 & 1.5 & 0 \\ 0 & 0 & -7.46 & 1 \\ -12.5 \times 10^3 & 0 & 0 & -22.3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12.5 \times 10^3 \end{bmatrix} v_1$$

Figure 19. Comparison of State Equations for the High Gain System

effect could be attained within the capabilities of the hardware available.

Summary

In both of the foregoing examples, the elements of the A matrices having the most influence in satisfying the response criterion, were apparent in the state models resulting from the compensation procedure. The compensated systems displayed much smoother transient characteristics than those of the original systems. In cases where the amplitude of a lightly damped mode is only a small percentage of the total system steady state response, it may be necessary to raise the steadiness factor specification to eliminate undesirable oscillations. Interpretation of the modified state models back into hardware changes was discussed along with the rationale of making the necessary engineering decisions. The ability to study load sensitivity of a linear system was also presented. The procedure, as presented, functions very well and provides needed information and insight into the transient dynamics of linear systems. There is room for expansion of the method which will be discussed in Chapter VI.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

A time domain technique has been presented which enables a designer to compensate the transient behavior of a linear model representing a physical system. Extensive use is made of the high speed digital computer in accomplishing the routine computation tasks. A performance criterion is developed to help the user assess the quality of the transient response of the system. The method requires the formation of a state model representing the dynamics of the system under study. The performance criterion is satisfied by adjusting the elements of the state model coefficient matrices.

The compensation method offers the following advantages over current methods:

1. The differential equations governing the system components may be used directly, in the form of a state model, rather than transforming the equations to the frequency domain in terms of transfer functions.
2. The transient response may be compensated directly by using a criterion that contains

all of the important factors governing the transient response.

3. The method does not limit the size of the system to be studied.
4. Constraints may be placed on the magnitudes of elements in the coefficient matrices of the state model.
5. The value of the performance index has direct physical significance. This is due to the fact that SFT indicates by its value, the amount of oscillation or overshoot that will be seen in the transient response.

There are also a number of limitations in the method which suggest natural extensions for recommended future work. These limitations are listed below.

1. The method presently is restricted to step inputs. Extension to include other types of inputs may require an adjustment to the performance index.
2. Only single input/output systems have been investigated. Extension of the method for use with multiple inputs and outputs may be useful in some applications.
3. The procedure requires that the system be linear and posssess distinct eigenvalues. Extension of the method to

include certain classes of nonlinearities will certainly be valuable.

4. Presently, there is no method of maintaining a functional correspondence between elements of the state model coefficient matrices. Extension of the procedure to include simple correspondence between elements, such as a proportionality, appears to be straightforward at this time, but more complex relationships will require further research.
5. There is probably a "best" way of adjusting the eigenvalues during the compensation, such that for the available variable elements of the A matrix, ψ is most effectively minimized. The method of adjusting the eigenvalues used in this investigation was selected in order to maintain the general rise time characteristics of the system, but this approach will not always allow the minimization of ψ if the number of variable elements in the A matrix is severely limited.

There is another area of investigation that is not associated directly with a limitation of the method. This suggested investigation involves the determination of a function that describes a line or region in space along or in which an index of performance is in an acceptable range of values. It is believed that the steadiness factor criterion introduced in this thesis could be adapted so that it defines the desired performance line or region for higher order systems. If such a line or region could be defined in hyperspace such that the desirable qualities of the system response are preserved, the procedure discussed in this thesis would be greatly enhanced.

The existence of such a line is suggested from a consideration of a second order system. If, for instance, a "good" response is considered to be one with a damping ratio of 0.7, then there exists a line in two dimensional space along which the damping ratio is constant at 0.7. Let a second order linear system be represented by the following equation:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{6-1}$$

or

$$s^2 + as + b = 0.$$

Let the coordinate directions of a two dimensional function space be defined by ai and bi, where i and j are unit orthogonal vectors. One can see that for ζ to equal 0.7 in Equation (6-1), the following relationship must exist between "a" and "b."

$$\frac{a}{\sqrt{b}} = 1.4$$

or

$$a = 1.4\sqrt{b}. \tag{6-2}$$

The locus of points in the a, b plane that satisfy Equation (6-2) form a parabola which is symmetric about the b axis and passes through the origin of the space. The only values of "a" that result in meaningful systems are those that are positive. Along this parabola, ζ remains fixed, but $_n$ varies, implying that the rise time characteristic of a transient for such a system will also vary. The damping ratio remaining fixed indicates that the transient response of this second order system will have similar characteristics all along the parabola.

SELECTED BIBLIOGRAPHY

(1) Bacon, C. M. "Linear State Space Models From Solution Specifications," Proceedings of the Eighth Midwest Symposium on Circuit Theory, Colorado State University (June, 1965).

(2) Bacon, C. M. "Parameter Optimization in Linear Systems - A State-Space Approach," Proceedings of the Third Annual Allerton Conference on Circuit and System Theory, Allerton House, Monticello, Illinois (October, 1965).

(3) Blackburn, J. F., G. Reethof, and J. L. Shearer. "Fluid Power Control" Cambridge, Massachusetts: The M.I.T. Press, 1960.

(4) Bose, J. E., S. Wendt, and J. G. Mrazek. "Digital Computer Design Techniques," Caterpillar Applied Control Series Volume 5, Fluid Power Control Laboratory, Oklahoma State University (August, 1967).

(5) Brockett, R. W. "Poles, Zeros, and Feedback: State Space Interpretation," IEEE Transactions on Automatic Control (April, 1965).

(6) D'Azzo, J. J., and C. H. Houpis. Feedback Control System Analysis and Synthesis. New York: McGraw-Hill Book Company, Inc., 1960.

(7) DeRusso, P. M., R. J. Roy, and C. M. Close. State Variables for Engineers. New York: John Wiley and Sons, Inc., 1965.

(8) Evans, G. G., H. E. Vigour, and F. J. Ellert. "An Application of Parameter Optimization to Hydraulic Servo Design," Joint Automatic Control Conference (August, 1966).

(9) Gall, D. A. "One Approach to the Practical Design of Automatic Controls," Joint Automatic Control Conference (August, 1966).

(10) Gantmacher, F. R. Theory of Matrices. New York: Chelsea Publishing Company, 1960.

(11) Gustafson, R. D. "A Paper and Pencil Control System Design Technique," Joint Automatic Control Conference (June, 1965).

(12) Kalman, R. E., and T. S. Englar. "A Users Manual for the Automatic Synthesis Program (Program C)," NASA Contractor Report Number CR-475, National Aeronautics and Space Administration, Washington, D. C. (June, 1966).

(13) Leithmann, G. Optimization Techniques. New York: Academic Press, 1963, pp. 205-210.

(14) Morgan, B. S. Jr. "Sensitivity Analysis and Synthesis of Multivariable Systems," IEEE Transactions on Automatic Control (April, 1965).

(15) Mrazek, J. G., J. E. Bose, and J. O. Matous. "Simulation of Hydrualic Controls," Caterpillar Applied Control Series Volume 2, Fluid Power Control Laboratory, Oklahoma State University (August, 1966).

(16) NORAIR. "Methods of Analysis and Synthesis of Piloted Aircraft Flight Control Systems," U. S. Navy Bureau of Aeronautics Report Number AE-61-41 (March, 1952).

(17) Tou, J. T. Modern Control Theory. New York: McGraw-Hill Book Company, 1964.

(18) Truxal, J. C. Control System Synthesis. New York: McGraw-Hill Book Company, 1955.

APPENDIX A

OBTAINING THE STATE MODEL

The design procedure presented in this thesis depends upon the engineer's ability to express the system mathematically in terms of a state model. One of the primary goals of the method of this thesis is to allow the designer to maintain contact with his original system. In the case of a hydraulic circuit or of a position control system such as were discussed in Chapter V, the recommended procedure initially involves drawing a functional block diagram of the system. This requires some knowledge of the differential equations that describe the dynamics of different components. Any design or analysis procedure requires that the components of the system be modeled mathematically. Modeling is still recognized as being difficult, especially in hydraulics research where there is a noticeable lack of test data to corroborate theoretical results. Once the mathematical descriptions of the hardware components are known or estimated, the design procedure may continue.

State Model for Hydraulic Circuit Example

The equations describing the dynamics of this system

stem from the following two primary sources:

1. Continuity of the fluid.

2. Equation of motion of the valve stem.

The pressure in the cylinder of Figure 9 is described by the following continuity relationships.

$$\frac{V_o}{\beta}\dot{P} = Q_{in} - Q_v - Q_{rest\ of\ system} \tag{A-1}$$

For simplicity, the last term on the right of Equation (A-1) will be assumed to be zero. Another assumption made in the following discussion is that the ram is either blocked by some outside force or that it is at the end of its travel. V_o, β, and Q_v are the chamber volume, the bulk modulus of the oil and the flow through the valve, respectively.

Equation (A-1) may be rewritten

$$\dot{P} = \frac{\beta}{V_o}\left(Q_{in} - Q_v\right) \tag{A-2}$$

This expression is now written in terms of perturbation variables. Taking the natural logarithm of both sides, Equation (A-2) becomes

$$\ln \dot{P} = \ln \frac{\beta}{V_o} + \ln(Q_{in} - Q_v).$$

Taking the differential of both sides yields

$$\frac{d\dot{P}}{\dot{P}} = \frac{d(Q_{in} - Q_v)}{(Q_{in} - Q_v}$$

or

$$d\dot{P} = \frac{\dot{P}}{(Q_{in} - Q_v)} \, d(Q_{in} - Q_v),$$

but from Equation (A-2), $\frac{\dot{P}}{(Q_{in} - Q_v)} = \frac{\beta}{V_o}$, so

$$d\dot{P} = \frac{\beta}{V_o} \, d(Q_{in} - Q_v)$$

and substituting lower case letters for the perturbation variables this yields

$$\dot{P} = \frac{\beta}{V_o} \, (q_{in} - q_v) \qquad \text{(A-3)}$$

A block diagram of this relationship would appear as shown in Figure 20a.

The equation governing flow through a valve is given by

$$Q_{valve} = C_d \pi d X \sqrt{\frac{2(P - P_t)}{\rho}} \qquad \text{(A-4)}$$

where C_d is the orifice coefficient, πd is the circumference of the valve stem, X_A is the valve displacement, ρ is the fluid density, P is the upstream pressure, and P_t is the tank pressure. Since P is usually very large compared to P_t, it is common to assume $P_t = 0$. Equation (A-4) becomes

$$Q_v = C_d \pi d X \sqrt{\frac{2P}{\rho}}. \qquad \text{(A-5)}$$

This nonlinear equation may be linearized for small perturbations about some steady state operating point.

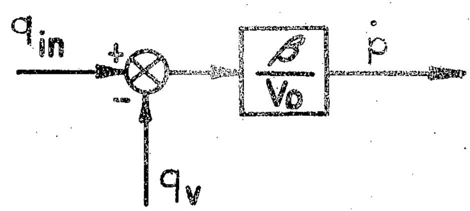


Figure 20a. Block Diagram of Continuity Equation

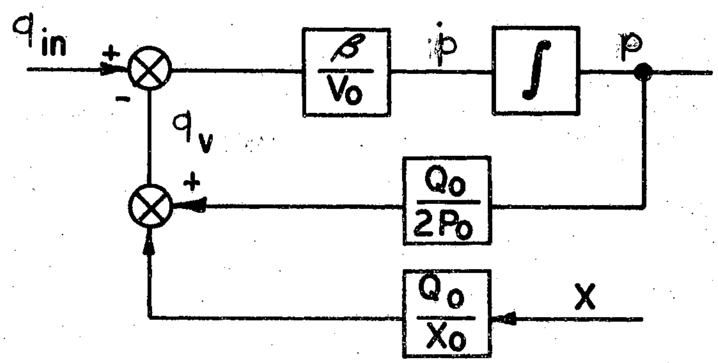


Figure 20b. Expanded Continuity Diagram

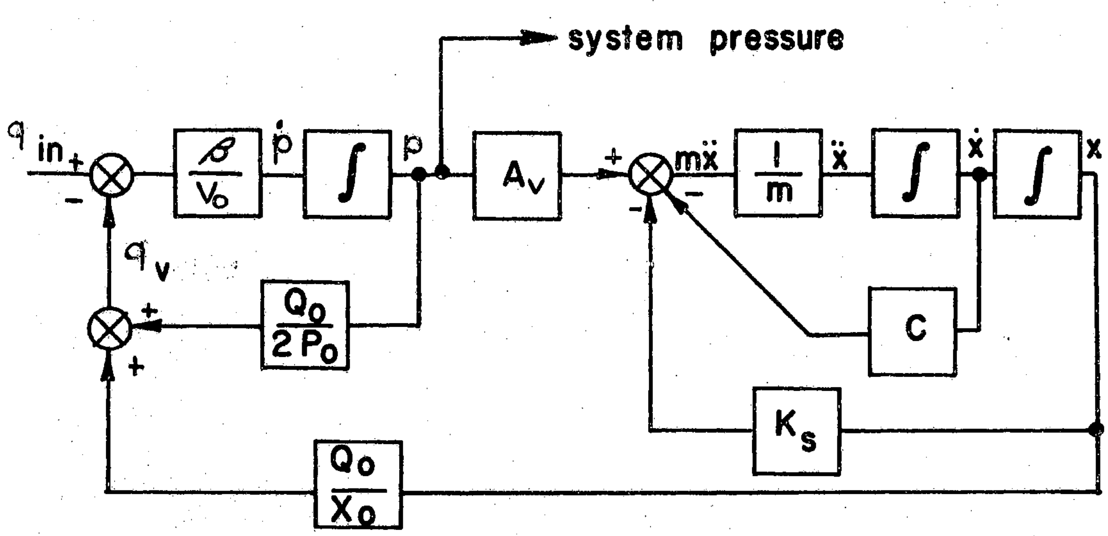


Figure 20c. Complete Pressure Control Circuit Diagram

Writing the natural logarithm of both sides of Equation (A-4) gives

$$\ln Q_v = \ln C_d \pi d\sqrt{\frac{2}{\rho}} + \ln X + \ln\sqrt{P}$$

$$+ \ln C_1 + \ln X + \tfrac{1}{2}\ln P. \qquad \text{(A-6)}$$

Taking the differential of both sides of (A-5) yields

$$\frac{dQ_v}{Q_v} = \frac{dX}{X} + \frac{dP}{2P}$$

or

$$dQ_{valve} = \frac{Q_v}{X}\, dX + \frac{Q_v}{2P}\, dP. \qquad \text{(A-7)}$$

If the perturbation variables are defined to be q_v, x, and p and the steady state operating point is denoted by Q_o, X_o, and P_o, Equation (A-6) becomes

$$q_v = \frac{Q_o}{X_o}\, x + \frac{Q_o}{2P_o} \quad . \qquad \text{(A-8)}$$

Figure 20a can now be expanded to 20b.

The equation of motion of the valve stem has the familiar form

$$m\ddot{X} + c\dot{X} + K_s X = PA_v. \qquad \text{(A-9)}$$

In this expression m, K_s, and A_v are constant coefficients representing, respectively, the valve stem mass, the spring constant of the retaining spring and the area upon which the pressure acts to open the valve. The damping coefficient c is made up of three components.

$$c = \frac{\mu\pi DL}{\delta} - \rho C_d \pi d_1 \cos 60^\circ \sqrt{\frac{2P}{\rho}}\,(2X + X_c) + \frac{128\mu A^2 \ell}{\pi\, d_2^4} \tag{A-10}$$

where the first term on the right of Equation (A-10) is the effect of viscous shear due the oil between the stem and the bore. The second term is due to the flow forces and is destabilizing. The cos60° factor in this term comes from the geometry of this particular example. The third term is from the capillary tube damping augmenter in the chamber over the valve stem. The development of the first and third terms of Equation (A-10) may be found in Reference [3]. The second term is developed in Reference [15].

The second term of Equation (A-9) can now be linearized in a manner similar to that used earlier. If the steady state operating point of $\dot{X}$ is taken as zero, then Equation (A-9) becomes, using perturbation variables as before,

$$m\ddot{x} + \left(\frac{\mu\pi DL}{\delta} - \rho C_d \pi d_1 \cos 60^\circ \sqrt{\frac{2P_o}{\rho}}\,(2X_o + X_c) + \frac{128\mu A^2 \ell}{\pi d_2^4} \right)\dot{x} + K_s x$$
$$= pA_1 .$$

This equation may be solved for $m\ddot{x}$, lumping the coefficient of $\dot{x}$ all into one constant c.

$$m\ddot{x} = pA_v - K_s x - c\dot{x}.$$

Figure 20c shows the complete operational block diagram of the system. The state model of the system may be established directly from this diagram by defining the outputs of each integrator as a state variable. A matrix differential equation may then be written which relates the inputs to each integrator to all of the outputs through the feedback constants. The literal form of the state model in this example is

$$\frac{d}{dt}\begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} \frac{Q_o \beta}{2P_o V_o} & \frac{Q_o \beta}{X_o V_o} & 0 \\ 0 & 0 & 1 \\ \frac{A_v}{m} & \frac{K_s}{m} & \frac{c}{m} \end{bmatrix} \begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} \frac{\beta}{V_o} \\ 0 \\ 0 \end{bmatrix} q$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix}.$$

The steady state operating point used for this example is defined by the parameters

$$P_o = 2000 \text{ psi}$$
$$X_o = .014 \text{ in}$$
$$\dot{X}_o = 0 .$$

Using these steady state conditions and some assumed geometric constants, the elements of the coefficient matrices in the state model as shown below.

$$\frac{d}{dt}\begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -2.3 & -657000 & 0 \\ 0 & 0 & 1 \\ 59 & -288000 & -200 \end{bmatrix} \begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 312.5 \\ 0 \\ 0 \end{bmatrix} q_{in} \tag{A-14}$$

and

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix}. \qquad \text{(A-15)}$$

Equations (A-13) and (A-14) together make up the state model of this example system.

State Model for Position Control System

A schematic diagram of the system under consideration is shown in Figure 14. A voltage amplifier, AMP is provided to control an amplidyne circuit. The amplidyne provides power for the armature circuit of a d.c. electric motor. The motor uses a field winding with a constant current i_f to provide the flux field required for the motor operation. The shaft of the motor is attached to an inertia load and also to the wiper of a position sensing potentiometer. The wiper is connected electrically to one terminal of the amplifier. The winding of the pot is connected in parallel with the winding of another pot and with a d.c. voltage source. The wiper of the second pot provides the input position reference. The load disturbance is input at the shaft of the motor.

The amplifier is assumed to have a constant gain K_a and no significant dynamics. This gain includes the actual amplifier gain as well as the static gain of the amplidyne.

The development of the differential equations describing the dynamics of the amplidyne follow the discussion of D'Azzo and Houpis [6]. Figure 21 shows an equivalent two stage representation of the amplidyne that will facilitate understanding of the equations. The input control voltage from Amp results in a control current i_c that must pass through a control winding with both inductance L_C and resistance, R_C. The resulting equation is given as follows

$$L_c \frac{di_c}{dt} + R_c i_c = e_c. \qquad \text{(A-16)}$$

A voltage e_q is induced by the current i_c in the first stage of the equivalent circuit such that

$$e_q = K_q i_c \qquad \text{(A-17)}$$

where K_q is a constant of proportionality. The quadrature winding also has inductance L_q and resistance R_q such that the current i_q resulting from e_q forms the relationship

$$L_q \frac{di_q}{dt} + R_q i_q = e_q. \qquad \text{(A-18)}$$

Also, i_q in turn induces a voltage e_d such that

$$e_d = K_d i_q \qquad \text{(A-19)}$$

where K_d is a constant of proportionality. The induced output voltage e_d is then fed to the armature of the d.c. motor. The torque produced by the motor may be written

$$T_Q = K_1 \Phi i_a \qquad \text{(A-20)}$$

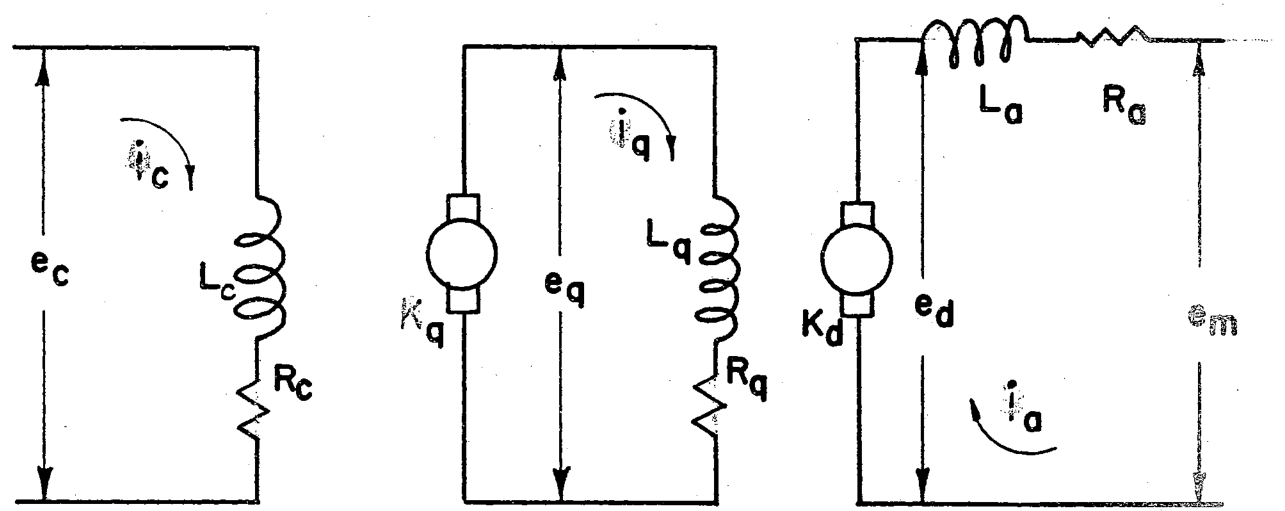


Figure 21. Two Stage Representation of Amplidyne Circuit

where K_1 is a constant of proportionality relating the field flux Φ and the armature current i_a to the torque. If a fixed voltage is applied to the field winding, then the flux is constant and the torque becomes proportional to only the armature current, i_a. The constant of proportionality is called K_T and now Equation (A-20) may be written

$$T_Q = K_T i_a.$$

When the armature is rotating, there is a back emf produced that is proportional to the motor speed $\dot{\theta}_o$. The voltage drop across the motor then becomes

$$e_m = K_m \dot{\theta}_o \tag{A-21}$$

where e_m is now given by

$$e_m = e_d - R_a i_a \tag{A-22}$$

where R_a is the armature resistance and the inductance is assumed to be negligible.

The state model for this system may also be established by forming the operational block diagram. Such a block diagram is shown in Figure 22. The output of each integrator is defined as a state variable. The state model of the system then becomes

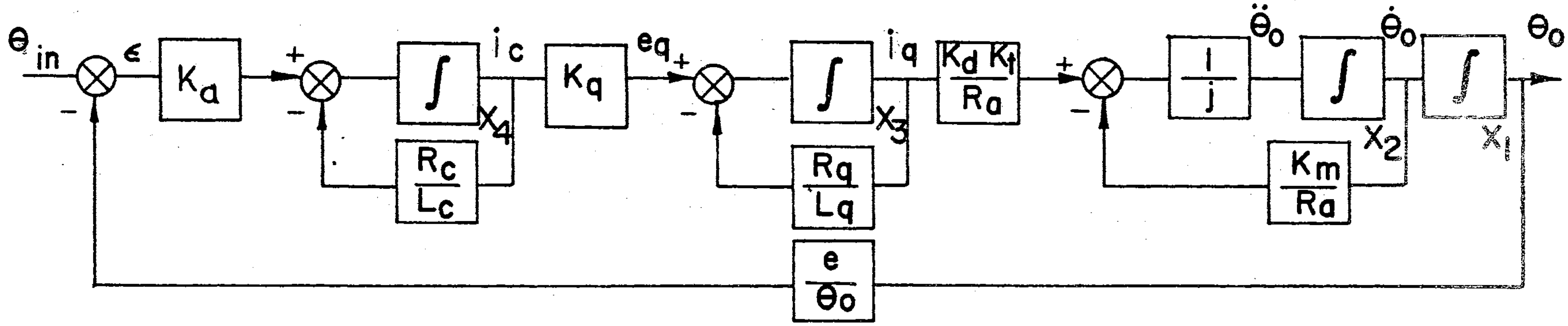


Figure 22. Operational Block Diagram of Position Control System

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-K_T K_m}{J\ R_a} & \frac{K_T K_d}{J\ R_a} & 0 \\ 0 & 0 & -\frac{R_q}{L_q} & K_q \\ -K_a & 0 & 0 & -\frac{R_c}{L_c} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_a \end{bmatrix} v \qquad \text{(A-23)}$$

$$y = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

where v is $_{in}$ and the output of the system x_1 is $_o$. When typical values are substituted in Equation (A-23), the result is

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2.0 & 1.5 & 0 \\ 0 & 0 & -5 & 0 \\ -250 & 0 & 0 & -20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 250 \end{bmatrix} v \qquad \text{(A-24)}$$

State Model of a Transfer Function With Zeros

It is often convenient to start with a transfer function in the formation of the time domain state model. The literature contains several methods of programming which result in different useful forms of the state model [7]. One very useful procedure will be presented in this appendix because it allows consideration of numerator dynamics as well as poles of the system.

Consider, for example, a transfer function which may be shown in the form

$$\frac{C}{R}(s) = \frac{K(s^2 + b_1 s + b_2)}{s^3 + a_1 s^2 + a_2 s + a_3} . \qquad \text{(A-25)}$$

This equation may be rewritten, by cross-multiplying, in the following form:

$$(s^3 + a_1 s^2 + a_2 s + a_3)C(s) = K(s^2 + b_1 s + b_2)R(s). \qquad \text{(A-26)}$$

Rearranging Equation (A-26) such that all terms containing zeroth powers of s are on the right side and all other terms are on the left, gives

$$(s^3 + a_1 s^2 + a_2 s)C(s) - K(s^2 + b_1 s)R(s) = Kb_2 R(s) - a_3 C(s). \qquad \text{(A-27)}$$

The right side of Equation (A-27) is defined as $x_1(s)$. Integrating $\dot{x}_1$ yields x_1 and the following equation:

$$(s^2 + a_1 s + a_2)C(s) - K(s + b_1)R(s) = x_1(s).$$

Transposing all zeroth power s terms from left to right results in

$$(s^2 + a_1 s)C(s) - KsR(s) = x_1 + Kb_1 R(s) - a_2 C(s).$$

The right side of this expression is defined as $\dot{x}_2(s)$. Integrating, yields

$$(s + a_1)C(s) - KR(s) = x_2(s).$$

Following the procedure one more time yields

$$sC(s) = x_2 + KR(s) - a_1 C(s)$$

where $$C(S) = x_3.$$

In summary, the following definitions have been made.

$$\dot{x}_1(s) = Kb_2 R(s) - a_3 C(s), \qquad (A\text{-}28)$$

$$\dot{x}_2(s) = x_1(s)\ Kb_1 R(s) - a_2 C(s), \qquad (A\text{-}29)$$

$$\dot{x}_3(s) = x_2(s) + Kb_2 R(s) - a_1 C(s). \qquad (A\text{-}30)$$

A block diagram may be constructed to assist in the visualization of these relationships as shown in Figure 23. Again, taking the outputs from the integrators and relating them to the inputs a state model of the form given below is obtained.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -a_3 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} Kb_2 \\ Kb_1 \\ K \end{bmatrix} R(t); \quad \begin{matrix} x_1(0) = x_{10} \\ x_2(0) = x_{20} \\ x_3(0) = x_{30} \end{matrix} \qquad (A\text{-}16)$$

$$y = [0 \quad 0 \quad 1] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad (A\text{-}17)$$

This technique for obtaining the state model from the transfer function has a particular usefulness. This form is called the Rational Canonical Form and is characterized

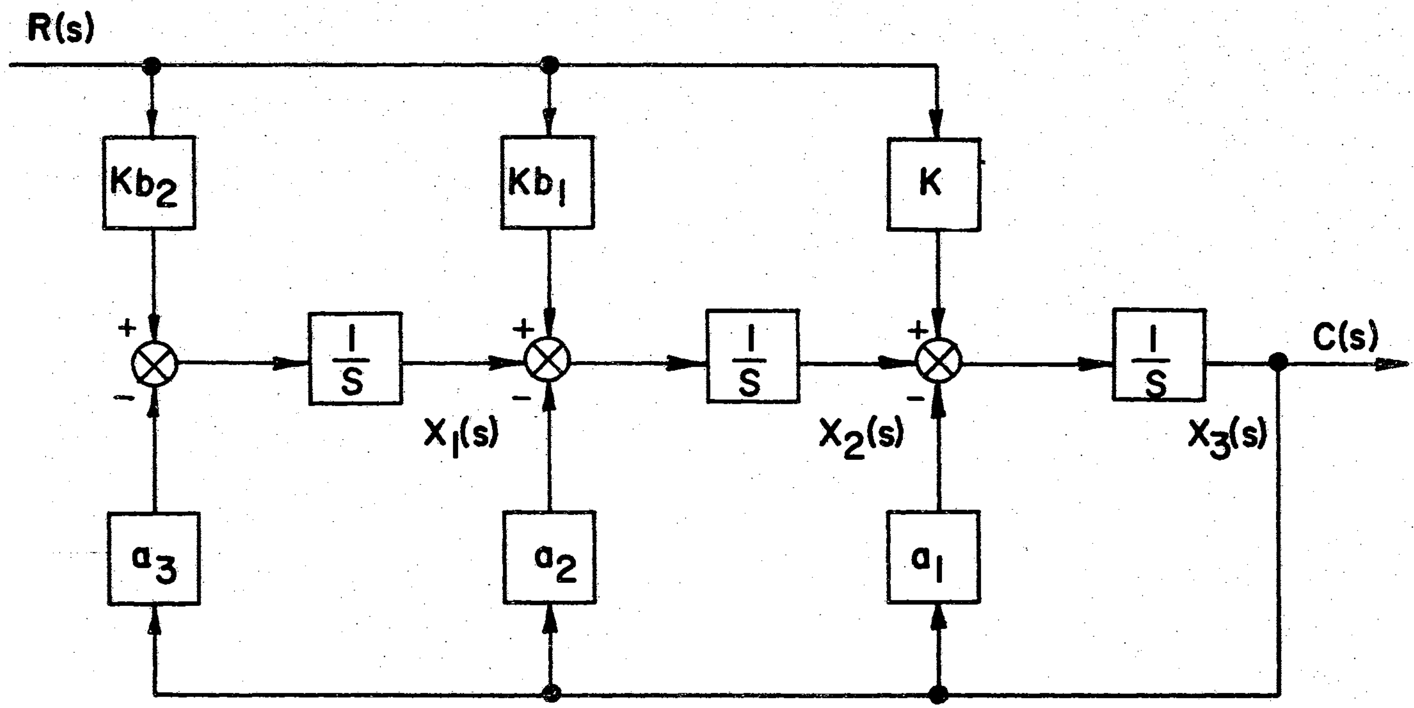


Figure 23. Programming Diagram for System With Zeros

by the display of the coefficients of the characteristic equation in the last column of the state coefficient matrix. It may also be noticed that the input coefficient matrix contains the static gain, K and the coefficients of the numerator polynomial. This convenient separation permits independent study of numerator and denominator effects. The change from s domain to T domain, above, is accomplished by considering appropriate initial conditions for the state variables.

APPENDIX B

METHOD OF STEEPEST DESCENT

The method of optimization through a steepest ascent/descent technique is particularly useful in numerical methods. The approach discussed in this thesis uses this concept twice in the determination of a compensated system. Since this technique is important to the design approach, it is discussed briefly in this appendix. This discussion follows that of Kelly [11].

Before considering the general gradient technique of optimization, first consider the continuous descent process. Let f be a function of several variables x_1, x_2, ..., x_n, defined in an open domain and possessing continuous partial derivatives with respect to the x_i; i = 1, 2, ..., n. Let a differential distance, ds, in this space, be defined in the following manner:

$$ds^2 = \sum_{i=1}^{n} dx_i^2 . \qquad \text{(B-1)}$$

Since the goal of this procedure is to move from some starting point $x_i = \bar{x}_i$, i = 1, 2, ..., n, toward a minimum f, first consider the directions in which the rate of change of f with respect to s, is negative.

$$\frac{df}{ds} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \left(\frac{dx_i}{ds}\right). \tag{B-2}$$

The direction of steepest descent is given by one of the directions that make Equation (B-2) stationary subject to (B-1).

Equation (B-1) may be rewritten in the following form:

$$1 - \sum_{i=1}^{n} \left(\frac{dx_i}{ds}\right)^2 = 0 \tag{B-3}$$

where $\frac{dx_i}{ds}$ may be considered to be direction cosines. This constraint may be adjoined to the right side of Equation (B-2) by means of a Lagrange multiplier λ_0 as follows:

$$\sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \frac{dx_i}{ds} + \lambda_0 \left[1 - \sum_{i=1}^{n} \left(\frac{dx_i}{ds}\right)^2\right]. \tag{B-4}$$

In order to find the value of λ_0 that extremizes Equation (B-2), the partial derivative of this expression is taken with respect to $\frac{dx_i}{ds}$. Setting the result equal to zero one obtains

$$\frac{\partial f}{\partial x_i} + \lambda_0 \left[-2 \frac{dx_i}{ds}\right] = 0 \qquad i = 1, 2, \ldots, n. \tag{B-5}$$

From (B-5) one may obtain

$$\frac{dx_i}{ds} = \frac{1}{2\lambda_0} \frac{\partial f}{\partial x_i} \qquad i = 1, 2, \ldots, n. \tag{B-6}$$

From (B-3) the following expression for λ_o results from combining Equations (B-6) and (B-3):

$$\lambda_o = \pm\tfrac{1}{2}\left[\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2\right]^{\frac{1}{2}}. \qquad \text{(B-7)}$$

If all the $\frac{\partial f}{\partial x_i}$ are not zero, the two distinct sets of direction numbers that make $\frac{df}{ds}$ stationary are

$$\frac{dX_i}{ds} = \pm\frac{\partial f}{\partial x_i}\left[\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2\right]^{-\frac{1}{2}}, \quad i = 1, 2, \ldots, n. \qquad \text{(B-8)}$$

It can be seen that the continuous case given above can be readily extended to use in a numerical technique. Since the determination of partial derivatives $\frac{\partial f}{\partial x_i}$ may be time consuming for high order systems, it is desirable to make the best use of each calculation of local gradient direction. The procedure used in this thesis is to follow the local gradient direction until f reaches a minimum. A new gradient direction is then calculated and the procedure repeated.

In this way, an n-dimensional minimum problem is reduced to a sequence of one-dimensional problems. The continuous and stepwise processes are contrasted in Figure 24 which shows the two types of motion as they occur in the vicinity of a minimum of a function of two variables $f(x_1, x_2)$. The gradient direction shown is normal to a contour while the local minimum in the gradient direction

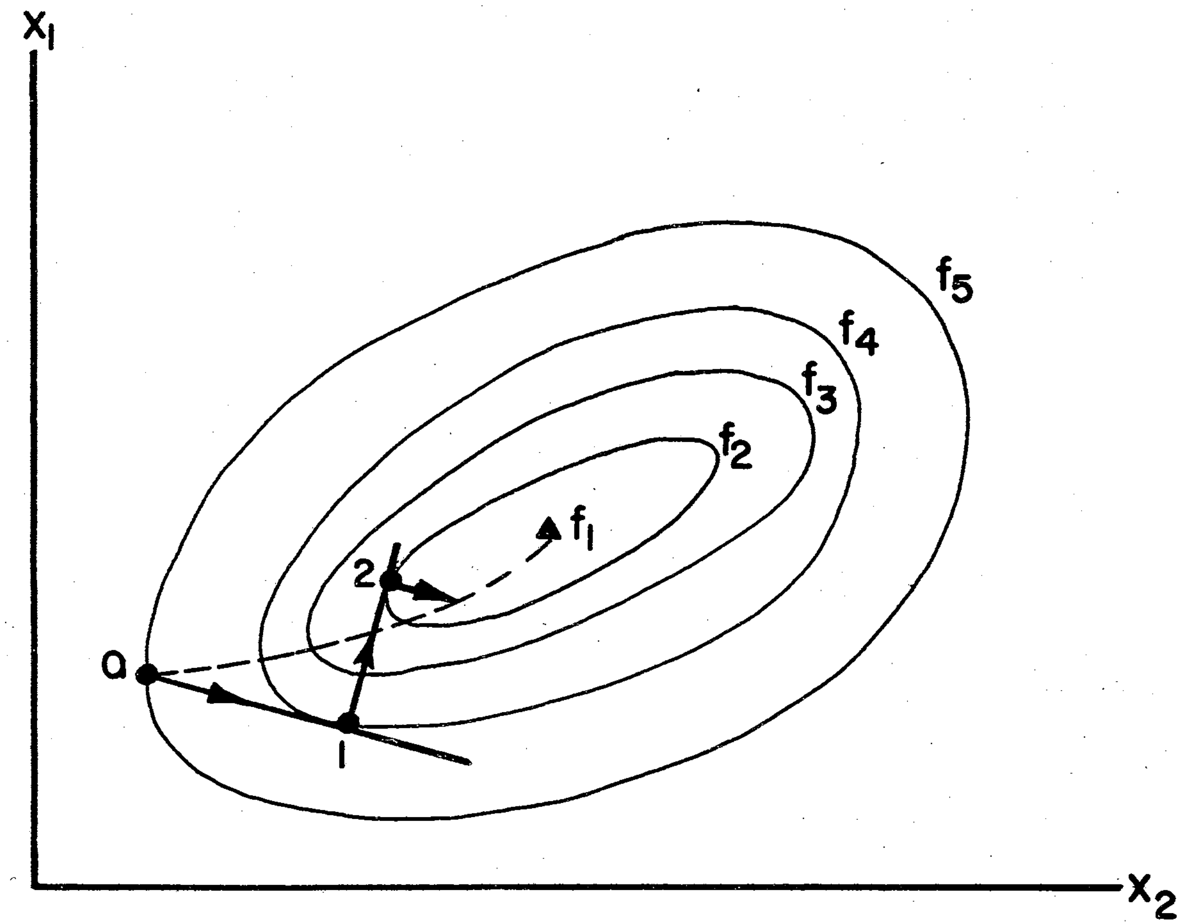


Figure 24. Comparison of Continuous and Numerical Descent Techniques

is found at a point of tangency to a contour. Higher ordered cases are not so easily displayed in a figure, but these same characteristics exist.

It is apparent that the stepwise path is not independent of the coordinate system selected. If a transformation of coordinates could be found that would result in circular contours, the first gradient direction would pass through minimum f. In the usual case, however, there is not sufficient information a priori to allow a sophisticated choice of coordinates. A normalization of parameters is possible, however, since generally a designer will have some idea of the practical range of each parameter. In this thesis, normalization is accomplished by letting an incremental step of a parameter be determined by a fractional multiple of the original value. In this way, the increments are percentage changes of the parameters.

APPENDIX C

DISCUSSION OF DIGITAL COMPUTER PROGRAM

The compensation procedure of this thesis is designed to make use of the extensive capabilities of the digital computer. A computer program which mechanizes the concepts of Chapter IV and was used to obtain the results of Chapter V, was written for the IBM 7040 in the Oklahoma State University Computing Center. The compiler language used is FORTRAN IV. The program consists of a main calling program and nine subroutines. The program is arranged to keep the number of input cards as low as possible to facilitate its use.

If the reader does not care to read the details of the program listings that follow, he may refer to Figure 25 for a flow diagram showing the major functions of the program. FORTRAN listings of all subprograms are included in later appendices, although only the significant subprograms are discussed in detail. Subroutines that invert a matrix or some other routine task are referred to in terms of the function they perform but are not discussed at length. The writer feels that eventually this sort of action will be accomplished by some FORTRAN statement much the same as multiplication and division is done today.

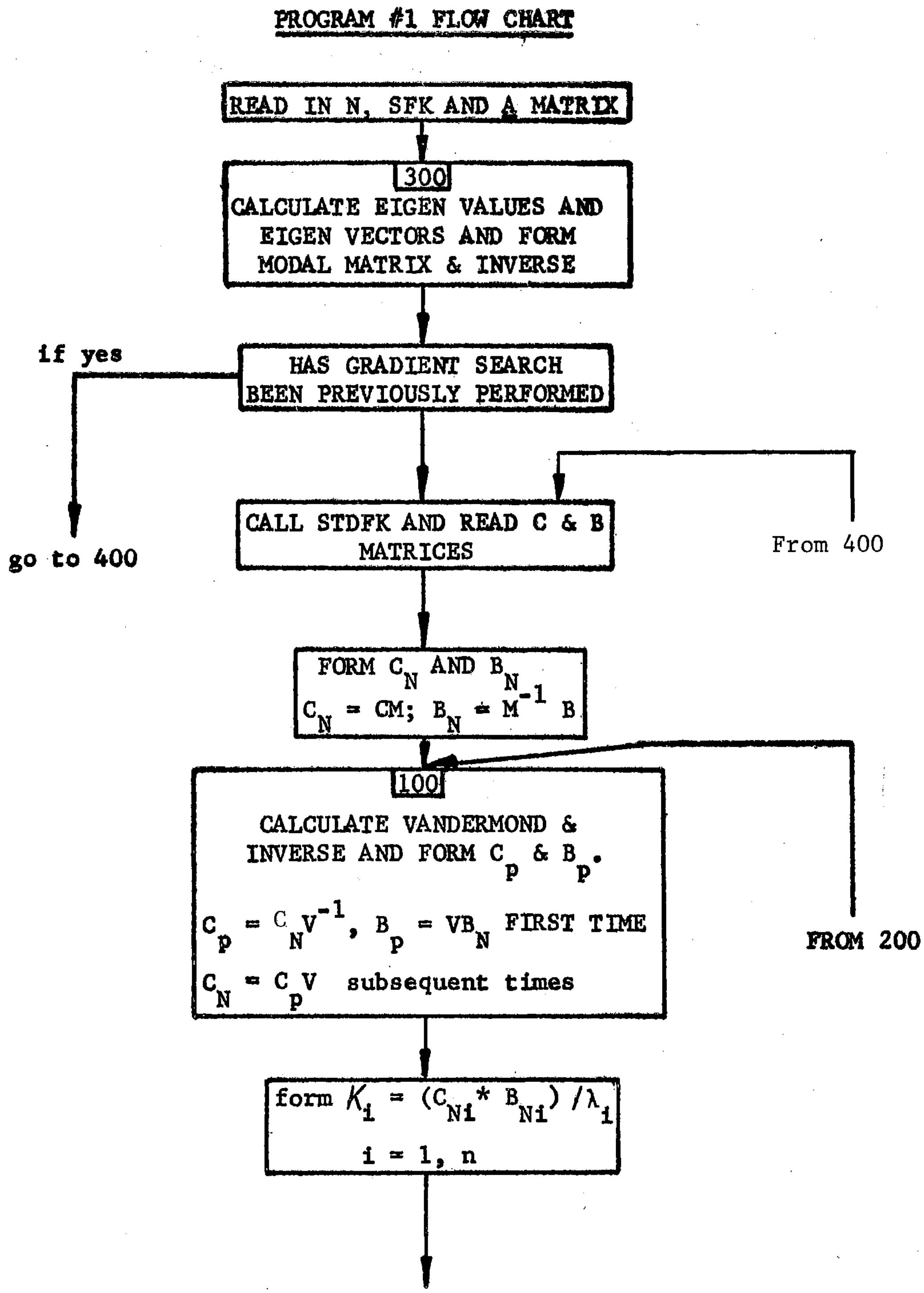


Figure 25. Digital Computer Flow Diagram

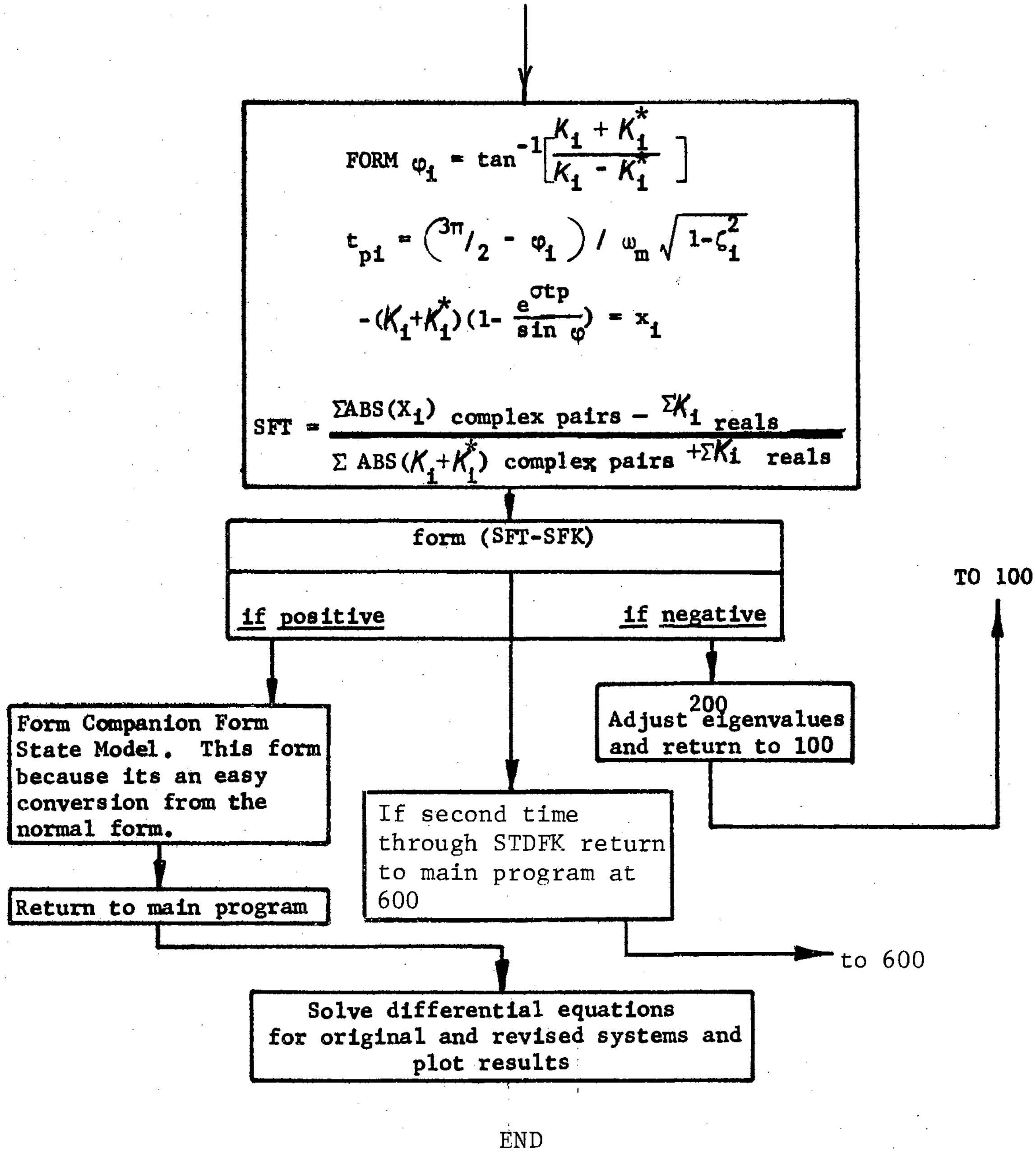


Figure 25. (Continued)

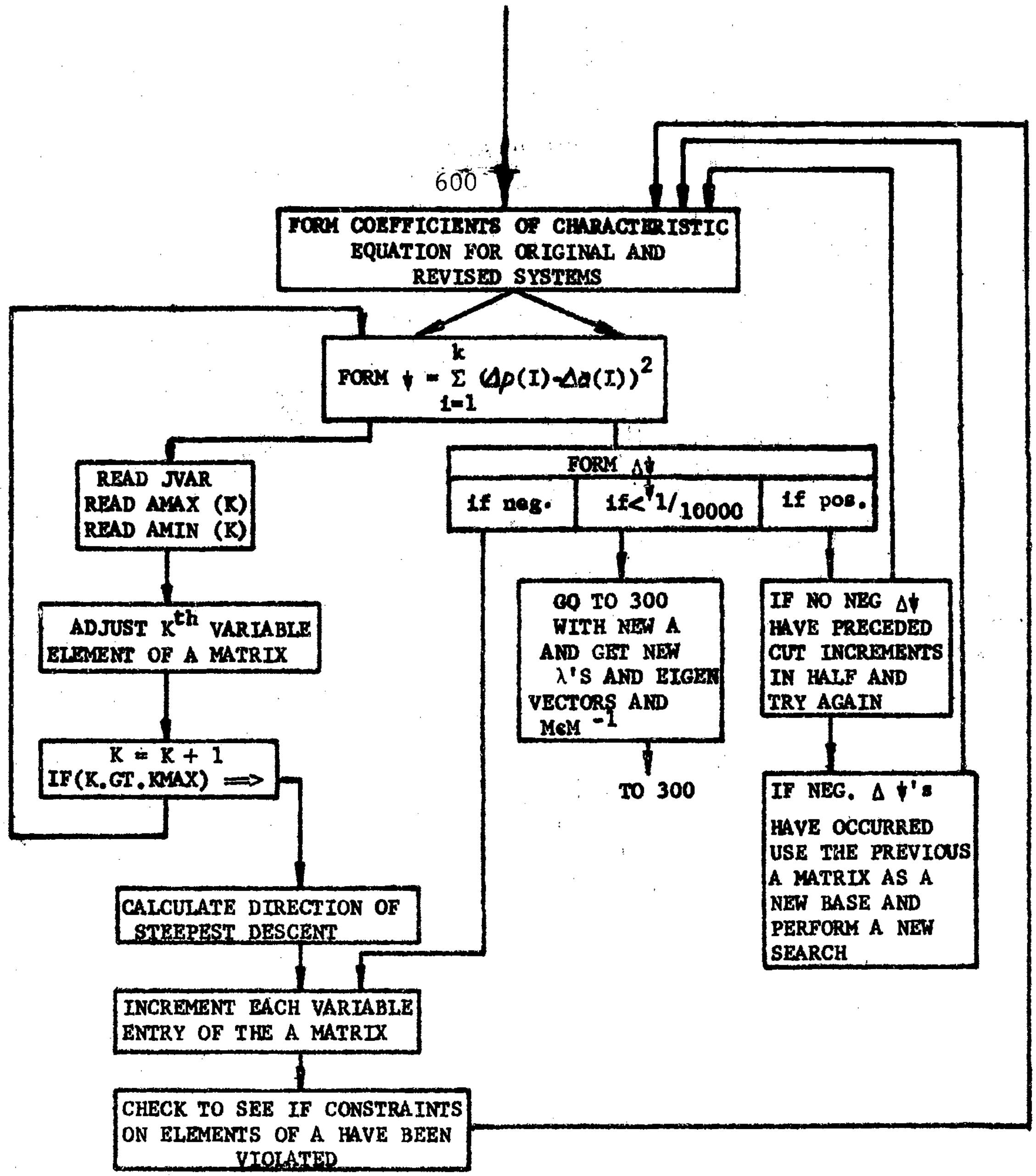


Figure 25. (Continued)

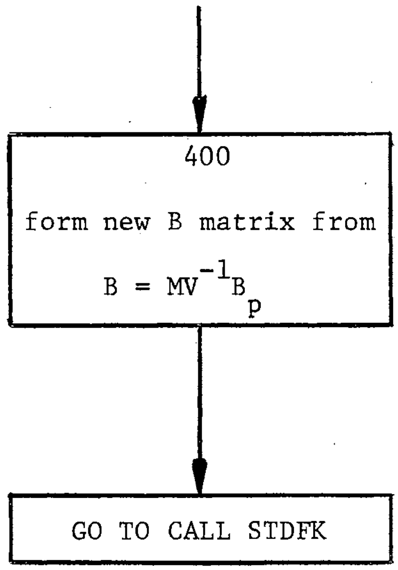


Figure 25. (Continued)

Main Calling Program

The main or calling program was developed from a library program from the SHARE series, known as DAL4. The original purpose of this program was to compute the eigenvalues and eigenvectors of an input A matrix. Significant additions were made to DAL4 in the process of developing a main program for this design procedure.

It will be noticed that a good part of DAL4 was written in double precision. This causes no difficulty if one is careful in preparing his input cards. The first action of the program is to read in N and SFK. N is the order of the system under study and SFK is the reference steadiness factor. Next, the positions of the variable elements of the A matrix are read in. These positions are then indexed for future reference by a numbering system that starts on the top row of A and increases from left to right. The maximum and minimum values of the variable elements of A are then read in. All of the entries of the B matrix are assumed to be variable so the maximum and minimum values of b_i are read in next. If one desires to hold an element of B fixed, then he must make the maximum and minimum values the same for that element. Some write statements occur next. None of the write statements will receive comment in the interest of expediting this discussion. The A matrix is then read in, in double precision.

Next DAL4 performs a normalization operation which is not particularly pertinent to the concepts of this work.

In the statements from 33 to 82, the traces of powers of A from 1 to N are calculated. In this process a subroutine SQMULT is called to obtain the various powers of A. Next, using the traces obtained above, the coefficients of the characteristic equation are obtained using Leverrier's algorithm. These coefficients are used to prepare inputs to another subroutine DNEWRA which factors the characteristic polynomial and produces the roots of the equation in the form RR(I) and RI(I). These are the real and imaginary parts, respectively, of the i^{th} eigenvalue. DAI4 then goes through some accuracy checks and then computes the eigenvectors. This point can be recognized by the notation EGVCR(III,JJJ) and EGVCI(III,JJJ) which are the real and imaginary parts of the elements of the JJJ^{th} eigenvector. These real numbers representing the real and imaginary parts of the eigenvectors are then formed into complex pairs and XMORIG(I,J) is formed. This is the first form of the system modal matrix, M. This matrix is then adjusted such that each column is normalized with respect to its top element. This results in a matrix, the top row of which is made up only of ones.

The next significant operation is the calling of subroutine STDFK. STDFK is significant to the procedure and will be discussed in detail later. At this point suffice it to say that STDFK uses the modal matrix, RR(I), RI(I), and SFK and returns an interim desired system in the phase variable form called PHSCOR and the proper size input, XKP, to result in unity output of the original system. At this point, if the program has cycled four times it branches to the section in which the differential equations are solved, otherwise it continues.

In the event that the program does not branch, it goes to 3333 where it begins the calculation of ψ, the minimization of which brings the characteristic equation of the original system, A, as close as possible to that of the desired system, PHYSCOR. The traces of the various powers of A are recomputed and Leverrier's algorithm is employed once again to obtain the coefficients of the characteristic equation. Since PHSCOR is in the phase variable form already, the desired coefficients for this system are displayed as a row of this matrix. ψ is then computed between 1553 and 1504.

At this point begins the operation that modifies the A matrix such that ψ is minimized. Since ψ is recalculated many times in the process of determining the proper gradient direction and then in the descent itself, a group of indices are set up to control traffic through this part of the program. These were all set equal to zero just below 3333.

An index for stopping the descent is established first by dividing the first value of ψ by 10000. It will be noticed that this part of the program only occurs if INDXIO is zero, or the first time through the routine. Next, using KMOR as an index, the program is sent to 1518 since initially KMOR is zero. In the range from 1518 to just above 1514, each variable entry of the A matrix is incremented. These increments are made one at a time and after each the program is sent back to 1509 and a new value of PSI is calculated. KCAL is increased by one every time through the loop until KCAL is greater than KMAX, the number of variable entries of A. KMOR is incremented by one every time through this loop also so the second time the IF statement just below 1520 is reached and all subsequent times during this loop the IF condition is false and the program proceeds through. A test is made on the direction of change of ψ. If the change is negative or zero, the program is sent to 1557; if positive to 1543.

At 1557 a partial derivative is calculated and the variable entry of A under consideration is set back to its original value so the next may be investigated. At 1543 a check is made to see if $\Delta\psi$ has been positive before when considering this element of A. If not, the sign of the increment to the element is changed and the element is set back to its original value, KSOK is incremented to record that the program has been through this branch and finally the program is sent to 1542 where the element is incremented in the new direction and the cycle repeated. When the program reaches 1511 again, if $\Delta\psi$ is negative or zero, the program is sent to 1557 as discussed above. If $\Delta\psi$ is still positive, however, the increment is halved and the cycle is repeated. This section of the program protects against taking too large a step and making an erroneous choice of gradient direction. This procedure continues until a $\Delta\psi$ which is negative or zero results or until the cycle has been traversed twenty times. This check is to protect the program from becoming locked in a loop.

After all of the increments, their directions, and their associated partial derivatives have been established, the gradient direction for steepest descent is determined. This is done in the range from 1578 to the statement just above 1557. The program is then sent to 1514 where the present entries of the A matrix are stored. Next, each variable entry of A is increment in a steepest descent sense. After these increments are made, a test is applied to see if any variable element of A exceeds its constraints. If so, the element is set equal to the constraining value. The traffic control indices, KMOR and KMOD, are reset to zero and KADJ is set equal to 1 to indicate that a descent is in progress. The program is then sent back to 1509 where ψ is recalculated for a new A with all variable entries adjusted. One can see by tracing the statements from 1509 that with KADJ = 1 there are some skips compared to previous cycles.

Finally, just below 1504 the program is sent to 1516 where ψ is written out and a check is made on the direction of $\Delta\psi$. This first time through, the change will probably be negative since the gradient search was just completed. If it should happen to be positive, however, the program is branched to 1545 where all the increments are halved and the program is sent to 1514 again. This will continue until $\Delta\psi$ is negative. In the usual case where $\Delta\psi$ is negative, the first time the value of $\Delta\psi$ is stored and the program is sent back to 1514 to increment A once more. Finally, during the descent a local minimum will be reached as was discussed in Appendix B. This is sensed by a positive $\Delta\psi$ after there has been at least one negative $\Delta\psi$. When this occurs, the program is branched to 1546 where A is set back to its form just prior to obtaining the positive $\Delta\psi$. Traffic control indices, KREP and KADJ are set back to zero and the program is sent back to 1509 where a new gradient search is initiated. The above procedure continues until $\Delta\psi$ is less than the first value of ψ divided by 10,000. This test is made just below 1516 every time a negative $\Delta\psi$ occurs. When this test is satisfied, the program is branched to 1521, toward the end of the listing of the main program.

At this point the new values of A are put back into double precision and control index JSKY is increased to 1. The program is then branched back to 1592 near the front of the listing where the eigenvalues and eigenvectors of the new A matrix are computed. The new modal matrix is also computed at this time. Just below 139, the program senses that JSKY is greater than zero and branches to

1591. In this part of the program, a new B matrix is computed by transforming the input coefficient matrix established in subroutine STDFK in the phase variable form to the form of the original state model. The resulting changes in the elements of B are then checked against the constraint limits of each entry. If any value b_i exceeds its constrained value, the entry is set at the value of the constraint.

At this point, the program is sent back to 1597 where subroutine STDFK is called again. This completes one major cycle of the progrm. This procedure is repeated until SFT exceeds the value of SFK the first time through subroutine STDFK or until a prescribed number of passes through STDFK have been made. This check is made by an IF statement just below the subroutine CALL statement using index KSICK. When either of the above criteria is satisfied, the program branches to 1598 where the matrix differential equations for the original and the revised systems are solved and a time trace comparison is plotted out on the output sheet.

Subroutine STDFK

This subroutine performs several important functions in the procedure. The inputs to STDFK include RR, RI, N, SFK, XMAJ, XMAJIN, and KSICK. This information is used to calculate the system steadiness factor, SFT and the matrices that make up the state model of the system expressed in the phase variable or companion form.

The first action of this subroutine is to read into memory CORIG and BORIG. These are the C matrix and the B matrix of the original system. It will be noticed from the IF statement just preceding the READ statements that these are read only the first time the program enters the subroutine since thereafter KSICK is greater than zero. BFIX is just the stored value of the input B matrix to be used when the differential equations of the original system are solved in the main program.

The matrices C_N and B_N are formed using CORIG, BORIG, and the modal matrix and its inverse. The next formation, starting at 108 through 8, is that of the Vandermonde matrix to be used in the transformation from normal form to phase variable form. After the Vandermonde matrix is formed, its inverse is obtained through subroutine CINV. This subroutine differs from that used in the main program in that this one allows complex entries where the other one does not. Once these matrices are available, the output and input coefficient matrices CP and BP of the phase variable form may be obtained. These are considered as a sort of pivot point for the approximate calculations to follow in that they are assumed not to change with changes in the eigenvalues later on.

The next significant computation occurs after 1011 where the products of corresponding entries of C_N and B_N are formed. This is in preparation for forming the K_i

coefficients discussed in Chapter III. Beginning at 1013 the eigenvalues are sorted out as to real and complex and the complex pairs are located. These locations are stored in arrays IJ(I) and IT(I) for the complex eigenvalues and KL(I) for the real eigenvalues. The eigenvalues are then formed into appropriate complex numbers and the coefficients K_1, mentioned above, are formed. Once these are available $\tan\Phi$ is calculated as discussed in Chapter III. The $\tan^{-1}$ of this quantity provides a value for Φ. It will be remembered that Φ is used to calculate t_p which is needed for the calculation of steadiness factor.

With Equation (3-13) it was discussed that the sign of $\sin\Phi$ would influence whether a reference of $\frac{3\pi}{2}$ is used in that equation or $\frac{\pi}{2}$. In order that the function be fixed in the digital program, a value of $\frac{3\pi}{2}$ is used for all cases. Φ is adjusted by π radians, however, whenever it was apparent from $(A_1 + A_2)$ and $-(A_1 - A_2)$ that Φ was in the third or fourth quadrant. This was necessitated in part by the fact that the inverse tangent subprogram only returns the principle values of the angle Φ. Once Φ has been established, TP is calculated in a straightforward manner. With this, then XSF is calculated for the pairs of complex eigenvalues. From 22 to just below 23 the contributions of both real and complex conjugate modes are added together to form the total system steadiness factor, SFT.

The program is branched at this point to 107 where

PHSCOR, the state coefficient matrix for this, the original system, is the phase variable form. After this, an input size for unity output is computed. More will be said of this computation later. The program is then branched back to 1021 where KSAP is increased from zero to one. At this point begins the adjustment of the system eigenvalues.

Each real eigenvalue is incremented individually from 1061 to just above 102. After each is incremented, the program branches back to 108 and SFT is recomputed, holding CP and BP constant. A partial derivative of SFT with respect to each real eigenvalue is computed, based on the change of SFT. After each partial derivative is calculated, the corresponding eigenvalue is set back to its original value. After all of the real eigenvalues have been investigated, the complex conjugate pairs are engaged. This is done below 2232 by forming each complex eigenvalue into a polar vector of the form $R\underline{/\theta}$. θ is then incremented and the program branches back to 108 and SFT is recalculated. Partial derivatives of the form $\frac{\partial SFT}{\partial \theta}$ are formed after each mode is studied.

Finally, when all increments have been made and adjusted as to sign such that SFT increases, the direction of steepest ascent is computed just below 2230 and each variable is incremented. The program then branches back to 108 where SFT is calculated with all eigenvalues adjusted by an appropriate amount. The program is then branched to 107 where the coefficient matrix of the phase variable or companion form is obtained. KSAP is greater than zero this time, so the program is returned to the main program. The main output of this subroutine is PHSCOR which is used in the calculation of ψ in the main program.

STDFK also serves another purpose. Since PHSCOR is obtained here and is a convenient form, the input size is computed that will result in unit output of the system. This input magnitude is called XKP in STDFK. The logic behind this computation may be seen from the block diagrams of Figure 26. This represents a system expressed in the phase variable form. Notice that in a steady state condition, the inputs to all integrators must be zero. This allows the following equations to be written for a unity output.

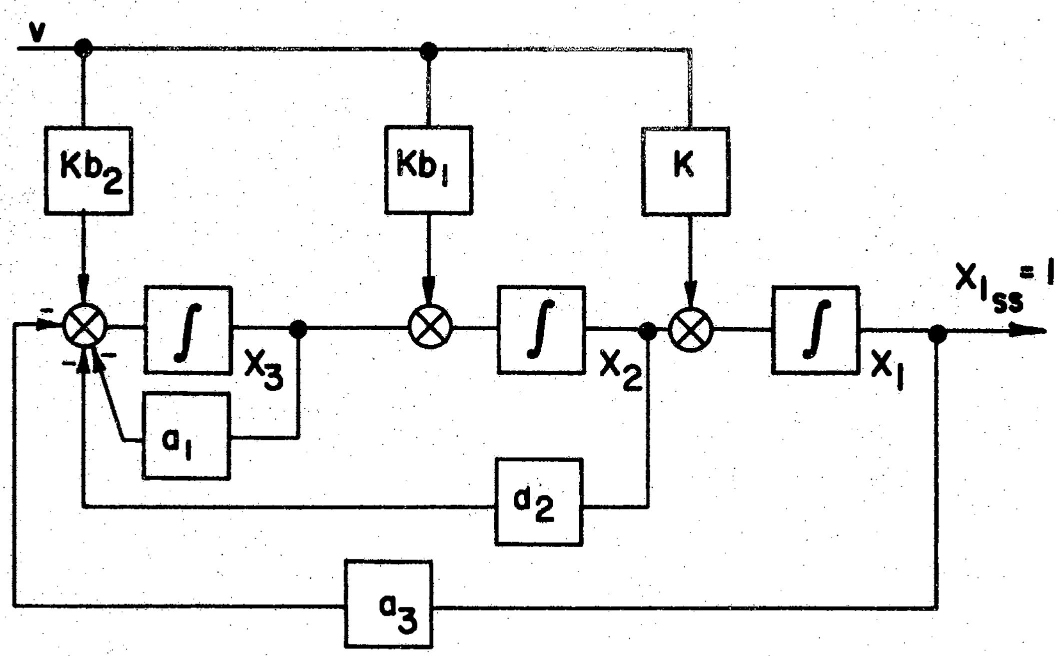
$$\text{If } \dot{x}_1 = 0 \text{ then } x_2 = -Kv$$
$$\text{If } \dot{x}_2 = 0 \text{ then } x_3 = -Kb_1 v$$
$$\text{If } \dot{x}_3 = 0 \text{ then } vKb_2 = a_1 x_3 + a_2 x_2 + a_3 x_1$$

but the desired value of x_1 is 1, so

$$vKb_2 = a_1(-Kb_1 v) + a_2(-Kv) + a_3$$
$$a_3 = v(Kb_2 + Kb_1 a_1 + Ka_2)$$

and
$$v = \frac{a_3}{Kb_2 + Kb_1 a_1 + Ka_2}.$$

This equation is easily mechanized on the digital computer as shown in STDFK from 2248 to just below 2250 near the



$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} X_{1_{ss}} \\ X_{2_{ss}} \\ X_{3_{ss}} \end{bmatrix} + \begin{bmatrix} K \\ Kb_1 \\ Kb_2 \end{bmatrix} v$$

$$1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{1_{ss}} \\ X_{2_{ss}} \\ X_{3_{ss}} \end{bmatrix}$$

Figure 26. Steady State Characteristic Diagram

end of the listing.

Subroutines DIFSOL and PLOT

These subroutines serve the purpose of solving the differential equations and plotting the results, respectively. Since these are only incidental to the procedure, they will not be discussed in detail. A discussion of the integration procedure may be found in Caterpillar Tractor Company Applied Control Series, Volume 5 published by the Oklahoma State University Fluid Power Control Laboratory in 1967 [4].

APPENDIX D

PREPARATION OF INPUT CARDS

The input data required for the use of this digital design program are read in on data cards. These cards form 13 groups. The groups are listed below with the required format information.

Group 1: One card that contains the order of the system under study and the specification value of steadiness factor, SFK. The format for this card is I2, F10.4.

Group 2: This group indicates which entries in the A matrix are to be variable, 1's for variable entries and zeros for fixed entries. These are arranged by rows in the format 5I2. Each card indicates the variable or fixed nature of the elements of one row of the A matrix. There will be as many cards in this group as there are rows in the A matrix.

Group 3: This group indicates the maximum magnitude allowed for each variable element of the A matrix. The values are arranged in a format of 7F10.4. The order of the data is by rows.

Group 4: This group indicates the minimum magnitude of the entries of the A matrix. The format and order are identical to those of Group 3.

Group 5: This group indicates the maximum values of the entries in the B matrix. The format is 10F7.4 and the order is from the top of the matrix.

Group 6: This group indicates the minimum values of the entries in the B matrix. The format and order are identical to those of Group 5. If it is desired to hold an entry of B fixed, make the maximum and minimum values the same.

Group 7: This group contains the starting values of the entries in the A matrix. The format is 5D14.7. The order is by <u>columns</u> of A.

Group 8: This group contains the entries of the C matrix in a format of (4F15.4, 4F15.4). The double format is because the entries of this matrix are input as complex numbers. The first part of the format is for the real part and the second is for the imaginary part. It is important to input zeros or leave the field blank corresponding to the second part if the entry is real. The order of the entries on the cards is the real part, then the imaginary part of each entry starting with the left most element of the C matrix.

Group 9: This group contains the elements of the B matrix where the format and order are identical to those of Group 8 except that the first entry is the top element of B.

Group 10: This group contains information needed for the numerical integration of the differential equations. The first entry is a value for the initial time step size. This should be some fraction of the shortest time constant of the system. The second entry is the quotient of the total time the designer wishes the solution to include divided by the initial time step size above. For instance, if the initial step size is .001 second and the desired final time is 7.5 seconds, then the second entry on the card would be $\frac{7.5}{.001} = 7500$. The format for this card is F10.5,I10.

Group 11: This group contains the initial conditions of the state variables starting with the first element of the state vector. The format for these entries is 8F10.5. If the order of the system is less than 8 and the initial conditions are to be zero then a single blank card will suffice.

Group 12: This group is made up of one card that contains the title of the plot of the step responses which the program provides. All 80 columns may be used.

Group 13: This group is also only one card. It contains the title of the vertical axis in columns 1 through 18, the symbols to be used on the plot in columns 19 through 58, the title of the horizontal axis in columns 59 through 76, and in columns 77, 78, and 79 are a decimal point, a plus sign, and a minus sign, respectively.

APPENDIX E

FORTRAN PROGRAM FOR TIME DOMAIN COMPENSATION OF LINEAR SYSTEMS

FORTRAN LISTING OF MAIN CALLING PROGRAM

```
      DIMENSION TRC(5),PQ(5),TD(5),ZQ(5),JVAR(5,5),AXE(25),IA(25),JA(25)
      DIMENSION A2(5,5),PA(5),ZA(5),AMAX(25),AMIN(25)
      DIMENSION DPSI(25),PDV(25),FAK(25),PSX(5)
      DIMENSION A( 5, 5),AA( 5, 5),TR( 5),AAMD( 5, 5),AAMT( 5, 5)        DAL40007
      DIMENSION DET(25),VMAX(25),VMIN(25),APWR(5,5),APWX(5,5)
      DIMENSION U( 40),RI( 5),RR( 5),IPIVOT(10),INDEX(10,2),PIVOT(10)
      DIMENSION RI1( 5),RR1( 5),C(10,10),COMM(12),A1( 5, 5),EGVCR( 5, 5)
      DIMENSION ZZI( 5, 5),DEF(10),TMRSP1(202),TMRSP2(202),SYSRSP(202)   DAL40009
      DIMENSION COLUMN( 5),Z( 5),P(16),X(30),G(30,1),TIME(202)           DAL40010
      DIMENSION EGVCI( 5, 5),E(5),B(5),QX(5,5)
      COMPLEX RMIINV(5,5),QSUM,TFX(5,5),BFIX(5),XB,YB,ZB,BMAX(5),BMIN(5)
      COMPLEX ZLMDA( 5),RM( 5, 5),D( 5, 5),CN( 5),F( 5),XMORIG( 5, 5)
      COMPLEX DINV( 5, 5),RMINV( 5, 5),ARGPHI( 5),XY,Z2,XMAJ( 5, 5)
      DIMENSION IJ( 5),IT( 5),PHI( 5),ARI( 5),TP( 5)
      COMPLEX XMINV( 5, 5),XMAJIN( 5, 5),XMARIG(5,5),BP(5),BORIG(5)
      DIMENSION SF( 5),RRGPHI( 5),TRR( 5),TRI( 5),ZI( 5),PHSCOR( 5, 5)
      COMMON PIVOT,INDEX,IPIVOT                                          DAL40011
      DOUBLE PRECISION RI1,RR1,C,A,AA,TR,AAMD,AAMT,U,RI,RR,ZZI,DEF,COLUMDAL40012
     1N,Z,P,X,G,FLON,H,SIGMA,DIFF,EPS,ZERO,FLOK,TREKR,TREKI,SUM,FDIV    DAL40013
      DOUBLE PRECISION PIVOT,DETERM                                      DAL40014
 1585 FORMAT(1H0,33X,4HPSI=,F15.4)
 1586 FORMAT(/1H0,11X,26HGRADIENT SEARCH COMMENCING)
 1575 FORMAT(7F10.4)
 1001 FORMAT(I2,I1)                                                      DAL40206
 1002 FORMAT(5D14.7)                                                     DAL40207
   10 FORMAT(12A6)                                                       DAL40213
   12 FORMAT(1H112A6)                                                    DAL40214
 3001 FORMAT(23H1          ERROR IN DNEWRA)                              DAL40204
 1108 FORMAT(28H1     THE DETERMINANT IS ZERO)                           DAL40205
 4102 FORMAT(8H          I2,2H  D16.9,6H        D16.9,20H               DAL40208
     1 I2,3H   D16.9,7H          D16.9)                                  DAL40209
 1581 FORMAT(///1H0,54X,22H*****FINAL SYSTEM*****)
 1112 FORMAT(68H1                                                        DAL40210
     1EIGENVECTOR I2//85H                                                DAL40211
     2REAL PART              IMAGINARY PART     /)                       DAL40212
 7101 FORMAT(////1H0,8X,40H INPUT MATRIX MULTIPLIED BY EIGENVECTOR I2,32DAL40215
     1H                      EIGENVECTORI2,27H  MULTIPLIED BY EIGENVALUE DAL40216
     2I2/113H                      REAL PART           IMAGINARY PART    DAL40217
     3                      REAL PART             IMAGINARY PART//)      DAL40218
 2004 FORMAT(/1H0,21X,18H NORMALIZED TRACES,13X,54H TRACE CHECKS(REAL PADAL40219
     1RT)  TRACE CHECKS(IMAGINARY PART))                                 DAL40220
 2001 FORMAT(1H1,19X,17HMATRIX DIMENSION=I2,12X,2HH=D16.9,11X,6HSIGMA=D1DAL40221
     16.9)                                                               DAL40222
 1609 FORMAT(//1H0,37X,59HTHE CYCLE COUNT HAS BEEN EXCEEDED FOR VARIABLE
     1 ENTRY NUMBER,I3)
   15 FORMAT(10F7.4)
    4 FORMAT(5F10.4)
 2013 FORMAT(1H ,22X,D16.9,23X,D16.9,2X,D16.9)                           DAL40223
 2023 FORMAT(1H ,22X,D16.9)                                              DAL40224
 2005 FORMAT(//1H0,38X,52H EIGENVALUES(REAL PART)  EIGENVALUES(IMAGINARY
     1 PART))                                                            DAL40226
 2015 FORMAT(1H ,41X,I2,2X,D16.9,2X,D16.9)
 1106 FORMAT(/1H0,   4(D12.4,2X,D12.4,4X))
    1 FORMAT(    1H1,54X,25H*****ORIGINAL SYSTEM*****)
    2 FORMAT(    1H1,54X,25H*****MODIFIED SYSTEM*****)
   20 FORMAT(/1H056X13H INPUT MATRIX/)                                   DAL40229
 6010 FORMAT(34X,5D15.7)
 2003 FORMAT(/1H0,18X,24H NORMALIZED COEFFICIENTS,7X,62H NORMALIZED ROOTDAL40231
```

```
     1S(REAL PART)  NORMALIZED ROOTS(IMAGINARY PART))                  DAL40232
 7111 FORMAT(1H1)                                                      DAL40233
 7113 FORMAT(1H0,10X,5HFIRST,25X,6HSECOND,26X,5HTHIRD,25X,6HFOURTH)
 7112 FORMAT(      8X,4(11HEIGENVECTOR,20X))
 1602 FORMAT(//1H0,47X,41H*****MODIFICATION OF SYSTEM A MATRIX*****)
 6000 FORMAT(1H0,32X,3HROW,I2)
    9 FORMAT (I2,F10.4)
   11 FORMAT (8F15.4)
   13 FORMAT (2F10.4)
   14 FORMAT (4F15.4,4F15.4)
 1583 FORMAT(     1H1,40X,55H***** TIME DOMAIN COMPENSATION OF A LINEAR SY
     1STEM *****)
 1584 FORMAT(1H0,56X,23HTHIS SYSTEM IS OF ORDER,I2)
   34 FORMAT(5I2)
C     INIZIALIZATION                                                   DAL40015
C                                                                      DAL40016
 3334 READ(5,9)N,SFK
      DO1505 I=1,N
      READ(5,34)(JVAR(I,J),J=1,N)
 1505 CONTINUE
      K=0
      DO1506 I=1,N
      DO1506 J=1,N
      KV=JVAR(I,J)
      IF(KV.LT.1)GO TO 1506
      K=K+1
      IA(K)=I
      JA(K)=J
 1506 CONTINUE
      KMAX=K
      READ(5,1575)(AMAX(I),I=1,KMAX)
      READ(5,1575)(AMIN(I),I=1,KMAX)
      READ(5,15   )(BMAX(I),I=1,N)
      READ(5,15   )(BMIN(I),I=1,N)
      IF(N)1111,3333,1111                                              DAL40020
 1111 WRITE(6,1583)
      WRITE(6,1584)N
      KSICK=0
      JSKY=0
      READ(5,1002)((A(I,J),I=1,N),J=1,N)                               DAL40022
 1592 FLON=N
      MAX=20                                                           DAL40023
      EPS=1.D-10                                                       DAL40024
      ZERO=1.D-10                                                      DAL40025
      WRITE (6,20)                                                     DAL40026
      DO 5000 J=1,N                                                    DAL40027
      WRITE(6,6000)J                                                   DAL40028
 5000 WRITE(6,6010)(A(J,I),I=1,N)                                      DAL40029
      H=.0D0                                                           DAL40030
      DO 50 K=1,N                                                      DAL40031
   50 H=H+A(K,K)                                                       DAL40032
      H=H/FLON                                                         DAL40033
      SIGMA=.0D0                                                       DAL40034
      DO 58 I=1,N                                                      DAL40035
      DO 58 J=1,N                                                      DAL40036
      IF(I-J)54,52,54                                                  DAL40037
   52 DIFF=(A(I,J)-H)**2                                               DAL40038
      GOTO 56                                                          DAL40039
   54 DIFF=(A(I,J))**2                                                 DAL40040
   56 SIGMA=SIGMA+DIFF                                                 DAL40041
   58 CONTINUE                                                         DAL40042
      SIGMA=SQRT(SIGMA/FLON)                                           DAL40043
```

```
      DO 70 I=1,N
      DO 70 J=1,N                                                       DAL40047
      IF(I-J)62,64,62                                                   DAL40048
   62 AA(I,J)=A(I,J)                                                    DAL40049
      GOTO 70                                                           DAL40050
   64 AA(I,J)=A(I,J)-H                                                  DAL40051
   70 CONTINUE                                                          DAL40052
   33 TR(1)=.0D0                                                        DAL40053
      DO 72 L=1,N                                                       DAL40054
   72 TR(1)=TR(1)+AA(L,L)                                               DAL40055
      TR(1)=TR(1)/SIGMA                                                 DAL40056
      DO 76 I=1,N                                                       DAL40058
      DO 76 J=1,N                                                       DAL40059
   76 AAMD(I,J)=AA(I,J)                                                 DAL40060
      K=2                                                               DAL40061
   88 CALL SQMULT(AA,AAMD,AAMT,N)                                       DAL40062
      TR(K)=.0D0                                                        DAL40063
      DO 78 L=1,N                                                       DAL40064
   78 TR(K)=TR(K)+AAMT(L,L)                                             DAL40065
      TR(K)=TR(K)/(SIGMA**K)                                            DAL40066
      IF(K-N)82,84,84                                                   DAL40067
   82 K=K+1                                                             DAL40068
      DO 86 I=1,N                                                       DAL40069
      DO 86 J=1,N                                                       DAL40070
   86 AAMD(I,J)=AAMT(I,J)                                               DAL40071
      GOTO 88                                                           DAL40072
   84 P(1)=TR(1)                                                        DAL40073
      DO 92 K=2,N                                                       DAL40074
      FLOK=K                                                            DAL40075
      Z(K)=.0D0                                                         DAL40076
      LIM=K-1                                                           DAL40077
      DO 94 I=1,LIM                                                     DAL40078
      LIMM=LIM+1-I                                                      DAL40079
   94 Z(K)=Z(K)+P(I)*TR(LIMM)                                           DAL40080
   92 P(K)=(TR(K)-Z(K))/FLOK                                            DAL40081
      U(1)=1.D0                                                         DAL40082
      U(2)=0.D0                                                         DAL40083
      DO 96 I=1,N                                                       DAL40084
      U(2*I+1)=-P(I)                                                    DAL40085
   96 U(2*I+2)=0.D0                                                     DAL40086
      CALL DNEWRA(N,U,X,MAX,EPS,ZERO,G,KE)                              DAL40087
      IF(KE.NE.0)GO TO3333
   95 DO 100I=1,N                                                       DAL40089
      RR(I)=X(2*I-1)                                                    DAL40090
  100 RI(I)=X(2*I)                                                      DAL40091
      DO 120 I=1,N                                                      DAL40092
      RI1(I)=RI(I)                                                      DAL40093
  120 RR1(I)=RR(I)                                                      DAL40094
      TREKR=.0D0                                                        DAL40097
      TREKI=.0D0                                                        DAL40098
      DO 122 I=1,N                                                      DAL40099
      TREKR=TREKR+RR(I)                                                 DAL40100
  122 TREKI=TREKI+RI(I)                                                 DAL40101
      DO 136 K=2,N                                                      DAL40104
      TREKR=.0D0                                                        DAL40105
      TREKI=.0D0                                                        DAL40106
      DO 130 I=1,N                                                      DAL40107
      Z(I)=RR(I)*RI1(I)+RI(I)*RR1(I)                                    DAL40108
      RR(I)=RR(I)*RR1(I)-RI(I)*RI1(I)                                   DAL40109
      TREKR=TREKR+RR(I)                                                 DAL40110
      RI(I)=Z(I)                                                        DAL40111
  130 TREKI=TREKI+RI(I)                                                 DAL40112
```

```
  136 CONTINUE
      DO 140 I=1,N                                      DAL40114
      RI(I)=RI1(I)*SIGMA                                DAL40115
  140 RR(I)=RR1(I)*SIGMA+H                              DAL40116
  209 NM=N*2                                            DAL40120
      DO 777 K=1,N                                      DAL40122
      DO 208 I=1,N                                      DAL40123
      X(I)=A(I,I)-RR(K)                                 DAL40124
      NNN=I+N                                           DAL40125
  208 X(NNN)=X(I)                                       DAL40126
      DO 213 I=1,NM                                     DAL40127
      DO 213 J=1,NM                                     DAL40128
  213 C(I,J)=.0                                         DAL40129
      DO 219 I=1,N                                      DAL40130
      INN=I+N                                           DAL40131
      C(INN,I)=-RI(K)                                   DAL40132
      C(I,INN)=RI(K)                                    DAL40133
      DO 219 J=1,N                                      DAL40134
      C(I,J)=A(I,J)                                     DAL40135
      JNN=J+N                                           DAL40136
  219 C(INN,JNN)=A(I,J)                                 DAL40137
      DO 400 I=1,NM                                     DAL40138
  400 C(I,I)=X(I)                                       DAL40139
  403 MM=NM-1                                           DAL40140
  402 DO 405 I=1,MM                                     DAL40141
  404 X(I)=C(I,N)                                       DAL40142
      DO 405 J=N,MM                                     DAL40143
  405 C(I,J)=C(I,J+1)                                   DAL40144
      DO 408 J=1,MM                                     DAL40145
      DO 408 I=N,MM                                     DAL40146
  408 C(I,J)=C(I+1,J)                                   DAL40147
      DO 410 I=N,MM                                     DAL40148
  410 X(I)=X(I+1)                                       DAL40149
      DO 411 I=1,MM                                     DAL40150
  411 G(I,1)=-X(I)                                      DAL40151
      LH=MM-1                                           DAL40152
      CALL MATINV(C,LH,G,1,DETERM)                      DAL40153
      IF(DETERM)233,231,233                             DAL40154
  231 WRITE(6,1108)                                     DAL40155
      GO TO 11                                          DAL40156
  233 DO 235 I=N,MM                                     DAL40157
      IZZ=MM+N-I                                        DAL40158
  235 G(IZZ+1,1)=G(IZZ,1)                               DAL40159
      G(N,1)=1.D0                                       DAL40160
      G(NM,1)=.0D0                                      DAL40161
      SUM=.0D0                                          DAL40162
      DO 240 I=1,NM                                     DAL40163
  240 SUM=SUM+G(I,1)**2                                 DAL40164
      FDIV=SQRT(SUM)                                    DAL40165
      DO 243 I=1,NM                                     DAL40166
  243 G(I,1)=G(I,1)/FDIV                                DAL40167
  703 DO 246 I=1,N                                      DAL40169
      KK=I+N                                            DAL40170
  246 TR(I)=G(KK,1)                                     DAL40171
      DO25 III=1,N
      JJJ=K
      EGVCR(III,JJJ)=G(III,1)
      EGVCI(III,JJJ)=TR(III)
   25 CONTINUE
      DO 4000 I=1,N                                     DAL40173
      KK=I+N                                            DAL40174
      DEF(I)=.0D0                                       DAL40175
```

```
      DEF(KK)=.0D0
      DO 4000 J=1,N
      KKKK=J+N
      DEF(I)=DEF(I)+A(I,J)*G(J,1)
      DEF(KK)=DEF(KK)+A(I,J)*G(KKKK,1)
 4000 CONTINUE
      DO 4005 I=1,N
      KK=I+N
      AAMD(I,K)=DEF(I)
 4005 AAMT(I,K)=DEF(KK)
      DO 4010 J=1,N
      KK=J+N
      DEF(J)=RR(K)*G(J,1)-RI(K)*G(KK,1)
      DEF(KK)=RR(K)*G(KK,1)+RI(K)*G(J,1)
 4010 CONTINUE
      DO 4020 I=1,N
      KK=I+N
      AA(I,K)=DEF(I)
 4020 ZZI(I,K)=DEF(KK)
  777 CONTINUE
      WRITE(6,2005)
      WRITE(6,2015)(I,RR(I),RI(I),I=1,N)
      WRITE(6,7113)
      WRITE(6,7112)
      DO 3 III=1,N
      WRITE(6,1106)(EGVCR(III,JJJ),EGVCI(III,JJJ),JJJ=1,N)
    3 CONTINUE
      DO137 I=1,N
      DO137 J=1,N
      XXX=EGVCR(I,J)
      YYY=EGVCI(I,J)
      XMORIG(I,J)=CMPLX(XXX,YYY)
  137 CONTINUE
      DO 138 J=1,N
      DO138 I=1,N
      XMARIG(I,J)=XMORIG(I,J)/XMORIG(1,J)
  138 CONTINUE
      DO1385 I=1,N
      DO1385 J=1,N
      XMORIG(I,J)=XMARIG(I,J)
 1385 CONTINUE
      CALL CINV(XMORIG,N,XMINV,KKK)
      DO139 J=1,N
      DO139 I=1,N
      XMAJ(I,J)=XMORIG(I,J)*XMINV(J,N)
  139 CONTINUE
      CALL CINV(XMAJ,N,XMAJIN,KKK)
      IF(JSKY.GT.0)GO TO 1591
 1597 CALL STDFK(RR,RI,N,SFK,SFT,XMAJ,XMAJIN,BORIG,RMIINV,CN,KSICK,BFIX,
     1PHSCOR,KSAP,XKP)
      IF(KSICK.GT.3)GO TO 1598
      DO1416 I=1,N
      XY=BFIX(I)
      E(I)=REAL(XY)
 1416 CONTINUE
      DO141 I=1,N
      DO141 J=1,N
      A1(I,J)=SNGL(A(I,J))
      IF(KSICK.GT.0)GO TO 141
      BETO=XKP
      QX(I,J)=A1(I,J)
  141 CONTINUE
```

```
      IF(KSAP.LT.1)GO TO 1598
      GO TO 3333
 1598 NZZZ=1
      WRITE(6,1)
      CALL DIFSOL(QX,N,NZZZ,TMRSP1,TMRSP2,IJK,E,BETO)
      DO142 I=1,IJK
      SYSRSP(I)=TMRSP1(I)
      TIME(I)=TMRSP2(I)
  142 CONTINUE
      DO1415 I=1,N
      XY=BORIG(I)
      B(I)=REAL(XY)
 1415 CONTINUE
      MJK=IJK
      WRITE(6,2)
      CALL DIFSOL(A1     ,N,NZZZ,TMRSP1,TMRSP2,IJK,B,XKP)
      DO143 I=1,IJK
      I2=I+MJK
      SYSRSP(I2)=TMRSP1(I)
      TIME(I2)=TMRSP2(I)
  143 CONTINUE
      TMAX=TIME(I2)
      RMAX=1.5
      CALL PLOT(TIME,0.0,TMAX,0,SYSRSP,0.0,1.5,0,0.0,0.0,0.0,0,I2,2,1,3,
     1 2)
      GO TO 3335
 3333 CONTINUE
      WRITE(6,1602)
      KMOD=0
      KADJ=0
      KMOR=0
      KREP=0
      KCYC=0
      IND10=0
      KSOK=0
 1509 TRC(1)=0.0
      DO1531 L=1,N
 1531 TRC(1)=TRC(1)+A1(L,L)
      DO1534 I=1,N
      DO1534 J=1,N
 1534 APWR(I,J)=A1(I,J)
      LX=2
 1537 DO1535 K=1,N
      DO1533 I=1,N
      SUM=0.0
      DO1532 J=1,N
      SUM=SUM+A1(I,J)*APWR(J,K)
 1532 CONTINUE
 1533 APWX(I,K)=SUM
 1535 CONTINUE
      DO1540 I=1,N
      DO1540 J=1,N
 1540 APWR(I,J)=APWX(I,J)
      TRC(LX)=0.0
      DO1536 L=1,N
 1536 TRC(LX)=TRC(LX)+APWR(L,L)
      LX=LX+1
      IF(LX.GT.N)GO TO 1538
      GO TO 1537
 1538 CONTINUE
      IF(KADJ.GT.0)GO TO 1510
      IF(KMOD.GT.0)GO TO 1510
```

```
      IF(KCYC.GT.0)GO TO 1510
      DO1500 I=1,N
      K=N-I+1
 1500 PQ(I)=-PHSCOR(N,K)
      TD(1)=-PQ(1)
      DO1502 K=2,N
      ZQ(K)=0.0
      FLOK=K
      LIX=K-1
      DO1501 I=1,LIX
      LIXX=LIX+1-I
 1501 ZQ(K)=ZQ(K)+PQ(I)*TD(LIXX)
 1502 TD(K)=-PQ(I)*FLOK-ZQ(K)
 1510 PA(1)=-TRC(1)
      DO1553 K=2,N
      FLOK=K
      ZA(K)=0.0
      LIM=K-1
      DO1552 I=1,LIM
      LIMM=LIM+1-I
 1552 ZA(K)=ZA(K)+PA(I)*TRC(LIMM)
 1553 PA(K)=-(TRC(K)+ZA(K))/FLOK
      SUM=0.0
      DO1504 I=1,N
      PSI2=PQ(I)-PA(I)
      PSI1=ABS(PSI2)**2.0
      SUM=SUM+PSI1
      PSI=SUM
 1504 CONTINUE
      IF(KADJ.GT.0)GO TO 1516
      IF(KMOD.GT.0)GO TO 1511
      PSO=PSI
      WRITE(6,1585)PSI
      IF(IND10.GT.0)GO TO 1578
      IND10=IND10+1
      PSIX=PSI/10000.
 1578 CONTINUE
 1520 KCAL=0
      IF(KMOR.LT.1)GO TO 1518
 1511 IF(PSI-PSO)1557,1557,1543
 1543 KSOR=KSOR+1
      IF(KSOR.GT.20)GO TO 1573
      IF(KSOK.GT.0)GO TO 1572
      DET(KCAL)=-DET(KCAL)
      A1(IX,JX)=A1HLD
      KSOK=KSOK+1
      GO TO 1542
 1572 DET(KCAL)=-0.5*DET(KCAL)
      A1(IX,JX)=A1HLD
      KSOK=0
      GO TO 1542
 1573 WRITE(6,1609)KCAL
      GO TO 1557
 1558 SUM=0.0
      DO1559 I=1,KMAX
 1559 SUM=SUM+(PDV(I)**2)
      PDVS=SUM
      DO1560 I=1,KMAX
      FAK(I)=PDV(I)/(PDVS**0.5)
 1560 FAK(I)=ABS(FAK(I))
      DO1561 I=1,KMAX
 1561 DET(I)=DET(I)*FAK(I)
```

```
      XDET=1.0
      YDET=1.0
      GO TO 1514
 1557 DPSI(KCAL)=PSI-PSO
      KSOR=0
      PDV(KCAL)=DPSI(KCAL)/ABS(DET(KCAL))
 1513 A1(IX,JX)=A1HLD
 1518 KCAL=KCAL+1
      KSOK=0
      KSOR=0
      KCYE=0
      KCYF=0
      KMOR=KMOR+1
      IF(KCAL.GT.KMAX)GO TO 1558
      IX=IA(KCAL)
      JX=JA(KCAL)
      CAL=A1(IX,JX)
      CAL=ABS(CAL)
      IF(CAL.GT.0.0)GO TO 1522
      DET(KCAL)=1.0
      GO TO 1523
 1522 DET(KCAL)=0.05*A1(IX,JX)
 1523 A1HLD=A1(IX,JX)
 1542 A1(IX,JX)=A1(IX,JX)+DET(KCAL)
      KMOD=KMOD+1
      GO TO 1509
 1514 KREP=KREP+1
      IF(KREP.GT.15)GO TO 1521
      DO1551 I=1,N
      DO1551J=1,N
 1551 A2(I,J)=A1(I,J)
      DO1515 K=1,KMAX
      IX=IA(K)
      JX=JA(K)
      A1(IX,JX)=A1(IX,JX)+DET(K)
      XNEW=A1(IX,JX)
      AMAK=AMAX(K)
      AMIX=AMIN(K)
      IF(XNEW-AMAK)1570,1570,1569
 1569 A1(IX,JX)=AMAX(K)
      GO TO 1515
 1570 IF(XNEW-AMIX)1571,1515,1515
 1571 A1(IX,JX)=AMIN(K)
 1515 CONTINUE
      KMOD=0
      KADJ=1
      K=0
      KMOR=0
      GO TO 1509
 1516 WRITE(6,1585)PSI
      IF(PSI-PSO)1567,1517,1545
 1567 PSXX=PSI-PSO
      PSXX=ABS(PSXX)
      IF(PSXX.LT.PSIX)GO TO 1521
      IF(KCYF.GT.0)GO TO 1562
      KCYF=1
 1568 CHG=PSI-PSO
      PSO=PSI
      KCYE=1
      GO TO 1514
 1562 THET1=CHG/XDET
      THET1=ATAN(THET1)
```

```
      THET2=(PSI-PSO)/YDET
      THET2=ATAN(THET2)
      DELTH=THET1-THET2
      DELTH=ABS(DELTH)
      IF(DELTH.GT.0.8)GO TO 1563
      IF(DELTH.LT.0.5)GO TO 1564
      XDET=YDET
      GO TO 1568
 1563 DO1565 I=1,KMAX
 1565 DET(I)=0.5*DET(I)
      XDET=YDET
      YDET=YDET*(0.5**0.5)
      GO TO 1568
 1564 DO1566 I=1,KMAX
 1566 DET(I)=2.0*DET(I)
      XDET=YDET
      YDET=YDET*(2.0**0.5)
      GO TO 1568
 1545 IF(KCYE.GT.0)GO TO 1546
      DO 1548 I=1,N
      DO 1548 J=1,N
 1548 A1(I,J)=A2(I,J)
      DO 1549 I=1,KMAX
 1549 DET(I)=DET(I)*0.5
      XDET=XDET*(0.5**0.5)
      GO TO 1514
 1546 DO 1550 I=1,N
      DO 1550 J=1,N
 1550 A1(I,J)=A2(I,J)
      KCYE=0
      GO TO 1517
 1517 CONTINUE
      KREP=0
      KADJ=0
      KCYC=KCYC+1
      IF(KCYC.GT.10)GO TO 1521
      WRITE(6,1586)
      GO TO 1509
 1521 DO1590 I=1,N
      DO1590 J=1,N
      XERX=A1(I,J)
 1590 A(I,J)=DBLE(XERX)
      JSKY=JSKY+1
      GO TO 1592
 1591 DO1596 J=1,N
      DO1596 I=1,N
 1596 XMORIG(I,J)=XMORIG(I,J)*CN(J)
      DO1594 K=1,N
      DO1594 I=1,N
      QSUM=(0.0,0.0)
      DO1593 J=1,N
 1593 QSUM=QSUM+XMORIG(I,J)*RMIINV(J,K)
 1594 TFX(I,K)=QSUM
      DO1595 I=1,N
      QSUM=(0.0,0.0)
      DO1595 J=1,N
      QSUM=QSUM+TFX(I,J)*BP(J)
 1595 BORIG(I)=QSUM
      DO1599 I=1,N
      XB=BMAX(I)
      ZB=BMIN(I)
      YB=BORIG(I)
```

```
      YBB=REAL(YB)
      XBB=REAL(XB)
      ZBB=REAL(ZB)
      IF(YBB.LT.XBB)GO TO 1601
      BORIG(I)=BMAX(I)
      GO TO 1599
 1601 IF(YBB.GT.ZBB)GO TO 1599
      BORIG(I)=BMIN(I)
 1599 CONTINUE
      KSICK=KSICK+1
      IF(KSAP-1)1598,1597,1597
 3335 CONTINUE
      STOP
      END
```

```
$IBFTC SQMULT
C       SQUARE MATRIX MULTIPLICATION                                      DAL40324
C                                                                         DAL40325
        SUBROUTINE SQMULT(A,B,C,N)                                        DAL40326
        DIMENSION A( 5, 5), B( 5, 5), C( 5, 5), COLUMN( 5)                DAL40327
        DOUBLE PRECISION A,B,C,COLUMN                                     DAL40328
    5 M=N                                                                 DAL40329
   10 DO 50 J=1,M                                                         DAL40330
   20 DO 25 K=1,M                                                         DAL40331
   25 COLUMN(K)=B(K,J)                                                    DAL40332
   30 DO 50 I=1,M                                                         DAL40333
   35 C(I,J)=0.0                                                          DAL40334
   40 DO 50 K=1,M                                                         DAL40335
   50 C(I,J)=C(I,J)+A(I,K)*COLUMN(K)                                      DAL40336
   60 RETURN                                                              DAL40337
        END                                                               DAL40338
```

```
$IBFTC DNEWR
      SUBROUTINE DNEWRA(M,C,R,MAX,EPS,ZERO,GG,KE)                       DAL40340
      DIMENSION C(1),R(1),GG(2),G(4),S(2),Y(2),X(2)                     DAL40341
      DOUBLE PRECISION C,R,S,T,X,P,Q,G                                  DAL40342
      DATA G(3),G(4)/.1D0,.1D0/                                         DAL40343
      KE=0                                                              DAL40344
      N=M                                                               DAL40345
      S(1)=C(1)                                                         DAL40346
      S(2)=C(2)                                                         DAL40347
      Q=S(1)**2+S(2)**2                                                 DAL40348
      IF(N.NE.1) GO TO 5                                                DAL40349
      R(1)=-(S(1)*C(3)+S(2)*C(4))/Q                                     DAL40350
      R(2)=(S(2)*C(3)-S(1)*C(4))/Q                                      DAL40351
      GO TO 200                                                         DAL40352
    5 L=0                                                               DAL40353
      KR=0                                                              DAL40354
      KT=0                                                              DAL40355
      EN=N                                                              DAL40356
      G(1)=GG(1)                                                        DAL40357
      G(2)=GG(2)                                                        DAL40358
      CM=(C(2*N+1)**2+C(2*N+2)**2)/Q                                    DAL40359
      IF(CM.NE.0.) GO TO 10                                             DAL40360
      CM=1.                                                             DAL40361
      R(1)=0.D0                                                         DAL40362
      R(2)=0.D0                                                         DAL40363
      G(1)=.1D0                                                         DAL40364
      L=1                                                               DAL40365
   10 DO 15 I=1,N                                                       DAL40366
      S(1)=S(1)+C(2*I+1)                                                DAL40367
   15 S(2)=S(2)+C(2*I+2)                                                DAL40368
      IF(SQRT((S(1)**2+S(2)**2)/CM)-ZERO)20,20,25                       DAL40369
   20 R(2*L+1)=1.D0                                                     DAL40370
      R(2*L+2)=0.D0                                                     DAL40371
      G(1)=1.1D0                                                        DAL40372
      G(2)=0.D0                                                         DAL40373
      L=L+1                                                             DAL40374
      IF(L-N)25,200,200                                                 DAL40375
   25 IF(MOD(N,2))30,35,30                                              DAL40376
   30 S(1)=C(3)-C(1)                                                    DAL40377
      S(2)=C(4)-C(2)                                                    DAL40378
      K=3                                                               DAL40379
      GO TO 40                                                          DAL40380
   35 S(1)=C(1)                                                         DAL40381
      S(2)=C(2)                                                         DAL40382
      K=2                                                               DAL40383
   40 DO 45 I=K,N,2                                                     DAL40384
      S(1)=S(1)-C(2*I-1)+C(2*I+1)                                       DAL40385
   45 S(2)=S(2)-C(2*I)+C(2*I+2)                                         DAL40386
      IF(SQRT((S(1)**2+S(2)**2)/CM)-ZERO)50,50,55                       DAL40387
   50 R(2*L+1)=-1.D0                                                    DAL40388
      R(2*L+2)=0.D0                                                     DAL40389
      G(1)=-1.1D0                                                       DAL40390
      G(2)=0.D0                                                         DAL40391
      L=L+1                                                             DAL40392
      IF(L-N)55,200,200                                                 DAL40393
   55 Q=G(1)**2+G(2)**2                                                 DAL40394
      IF(Q-1.)75,75,60                                                  DAL40395
   60 G(1)=G(1)/Q                                                       DAL40396
      G(2)=(-G(2))/Q                                                    DAL40397
      KR=-1                                                             DAL40398
   65 CM=1./CM                                                          DAL40399
      K=(N+1)/2                                                         DAL40400
```

```
      DO 70 I=1,K
      J=N-I
      T(1)=C(2*I-1)
      T(2)=C(2*I)
      C(2*I-1)=C(2*J+3)
      C(2*I)=C(2*J+4)
      C(2*J+3)=T(1)
   70 C(2*J+4)=T(2)
   75 IG=1
   80 X(1)=G(2*IG-1)
      X(2)=G(2*IG)
      DO 110 J=1,MAX
      P=EN
      S(1)=C(1)*X(1)-C(2)*X(2)+C(3)
      S(2)=C(2)*X(1)+C(1)*X(2)+C(4)
      T(1)=P*C(1)
      T(2)=P*C(2)
      DO 85 I=2,N
      Q=S(1)*X(1)-S(2)*X(2)+C(2*I+1)
      S(2)=S(2)*X(1)+S(1)*X(2)+C(2*I+2)
      S(1)=Q
      P=P-1.D0
      Q=T(1)*X(1)-T(2)*X(2)+P*C(2*I-1)
      T(2)=T(2)*X(1)+T(1)*X(2)+P*C(2*I)
   85 T(1)=Q
      P=S(1)**2+S(2)**2
      Q=(T(1)*S(1)+T(2)*S(2))/P
      T(2)=(T(2)*S(1)-T(1)*S(2))/P
      T(1)=Q
      IF(P.EQ.0.) T(1)=1.D16
      IF(L)90,100,90
   90 DO 95 I=1,L
      S(1)=X(1)-R(2*I-1)
      S(2)=X(2)-R(2*I)
      Q=S(1)**2+S(2)**2
      T(1)=T(1)-S(1)/Q
   95 T(2)=T(2)+S(2)/Q
  100 Q=T(1)**2+T(2)**2
      X(1)=X(1)-T(1)/Q
      X(2)=X(2)+T(2)/Q
      IF(1./SQRT(Q*(X(1)**2+X(2)**2))-EPS)105,105,110
  105 IF(SQRT(P/CM)-ZERO)160,160,110
  110 CONTINUE
      IG=IG+1
      IF(IG-2)115,80,115
  115 IF(KT)120,155,120
  120 IF(L)125,175,125
  125 KE=L
  130 IF(KR)135,200,135
  135 KT=0
  140 DO 145 I=1,L
      Q=R(2*I-1)**2+R(2*I)**2
      R(2*I-1)=R(2*I-1)/Q
  145 R(2*I)=(-R(2*I))/Q
  150 IF(KT)65,200,65
  155 KT=1
      KR=KR+1
      Q=X(1)**2+X(2)**2
      G(1)=X(1)/Q
      G(2)=(-X(2))/Q
      IF(L)140,65,140
  160 IF(ABS(X(2)/X(1))-ZERO)165,165,170
```

```
165 X(2)=0.                                                    DAL40463
170 R(2*L+1)=X(1)                                              DAL40464
    R(2*L+2)=X(2)                                              DAL40465
    G(1)=1.1D0*X(1)                                            DAL40466
    G(2)=1.1D0*X(2)                                            DAL40467
    L=L+1                                                      DAL40468
    IF(L-N)75,130,130                                          DAL40469
175 KE=-1                                                      DAL40470
200 RETURN                                                     DAL40471
     END                                                       DAL40472
```

```
$IBFTC MATINV
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS   DAL40238
C                                                                       DAL40239
      SUBROUTINE MATINV(A,N,B,M,DETERM)                                 DAL40240
C                                                                       DAL40241
      DIMENSION IPIVOT(10), A(10,10), B(30,1), INDEX(10,2), PIVOT(10)   DAL40242
      COMMON PIVOT,INDEX,IPIVOT                                         DAL40243
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)         DAL40244
      DOUBLE PRECISION DETERM,AMAX,SWAP,T,A,B,PIVOT                     DAL40245
C     INITIALIZATION                                                    DAL40246
C                                                                       DAL40247
   10 DETERM=1.D0                                                       DAL40248
   15 DO 20 J=1,N                                                       DAL40249
   20 IPIVOT(J)=0                                                       DAL40250
   30 DO 550 I=1,N                                                      DAL40251
C                                                                       DAL40252
C     SEARCH FOR PIVOT ELEMENT                                          DAL40253
C                                                                       DAL40254
   40 AMAX=0.D0                                                         DAL40255
   45 DO 105 J=1,N                                                      DAL40256
   50 IF (IPIVOT(J)-1) 60, 105, 60                                      DAL40257
   60 DO 100 K=1,N                                                      DAL40258
   70 IF (IPIVOT(K)-1) 80, 100, 740                                     DAL40259
   80 IF (ABS(AMAX)-ABS(A(J,K))) 85, 100, 100                           DAL40260
   85 IROW=J                                                            DAL40261
   90 ICOLUM=K                                                          DAL40262
   95 AMAX=A(J,K)                                                       DAL40263
  100 CONTINUE                                                          DAL40264
  105 CONTINUE                                                          DAL40265
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1                                   DAL40266
C                                                                       DAL40267
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                 DAL40268
C                                                                       DAL40269
  130 IF (IROW-ICOLUM) 140, 260, 140                                    DAL40270
  140 DETERM=-DETERM                                                    DAL40271
  150 DO 200 L=1,N                                                      DAL40272
  160 SWAP=A(IROW,L)                                                    DAL40273
  170 A(IROW,L)=A(ICOLUM,L)                                             DAL40274
  200 A(ICOLUM,L)=SWAP                                                  DAL40275
  205 IF(M) 260, 260, 210                                               DAL40276
  210 DO 250 L=1, M                                                     DAL40277
  220 SWAP=B(IROW,L)                                                    DAL40278
  230 B(IROW,L)=B(ICOLUM,L)                                             DAL40279
  250 B(ICOLUM,L)=SWAP                                                  DAL40280
  260 INDEX(I,1)=IROW                                                   DAL40281
  270 INDEX(I,2)=ICOLUM                                                 DAL40282
  310 PIVOT(I)=A(ICOLUM,ICOLUM)                                         DAL40283
  320 DETERM=DETERM*PIVOT(I)                                            DAL40284
C                                                                       DAL40285
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                                 DAL40286
C                                                                       DAL40287
  330 A(ICOLUM,ICOLUM)=1.0                                              DAL40288
  340 DO 350 L=1,N                                                      DAL40289
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)                                  DAL40290
  355 IF(M) 380, 380, 360                                               DAL40291
  360 DO 370 L=1,M                                                      DAL40292
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)                                  DAL40293
C                                                                       DAL40294
C     REDUCE NON-PIVOT ROWS                                             DAL40295
C                                                                       DAL40296
  380 DO 550 L1=1,N                                                     DAL40297
  390 IF(L1-ICOLUM) 400, 550, 400                                       DAL40298
```

```
  400 T=A(L1,ICOLUM)                                                DAL40299
  420 A(L1,ICOLUM)=0.0                                              DAL40300
  430 DO 450 L=1,N                                                  DAL40301
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                 DAL40302
  455 IF(M) 550, 550, 460                                           DAL40303
  460 DO 500 L=1,M                                                  DAL40304
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T                                 DAL40305
  550 CONTINUE                                                      DAL40306
C                                                                   DAL40307
C     INTERCHANGE COLUMNS                                           DAL40308
C                                                                   DAL40309
  600 DO 710 I=1,N                                                  DAL40310
  610 L=N+1-I                                                       DAL40311
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                      DAL40312
  630 JROW=INDEX(L,1)                                               DAL40313
  640 JCOLUM=INDEX(L,2)                                             DAL40314
  650 DO 705 K=1,N                                                  DAL40315
  660 SWAP=A(K,JROW)                                                DAL40316
  670 A(K,JROW)=A(K,JCOLUM)                                         DAL40317
  700 A(K,JCOLUM)=SWAP                                              DAL40318
  705 CONTINUE                                                      DAL40319
  710 CONTINUE                                                      DAL40320
  740 RETURN                                                        DAL40321
      END                                                           DAL40322
```

```
$IBFTC STDFK
      SUBROUTINE STDFK(RR,RI,N,SFK,SFT,XMAJ,XMAJIN,BORIG,RMIINV,CN,KSICK
     1,BFIX,PHSCOR,KSAP,XKP)
      COMPLEX XMAJ( 5, 5),XMAJIN( 5, 5),CORIG( 5),BORIG( 5),RMI( 5, 5)
      COMPLEX XLAM1,XLAM2,A1,A2,PROD,SUM6,DENX1,DENX2,CMD,RLC,RLCOF
      COMPLEX RMIINV( 5, 5),BN( 5),CP( 5),BP( 5),RLRT(5),DEL(5),XSF
      COMPLEX ZLMDA( 5),RM( 5, 5),D( 5, 5),CN( 5),F( 5),YX,SUM2,BFIX(5)
      COMPLEX DINV( 5, 5),RMINV( 5, 5),ARGPHI( 5),XY,Z2,XX,XSFT,RLRT1
      COMPLEX FIXER,BPFIX(5),BNFIX(5)
      COMPLEX SUM,ASUM,AX1(5,5),BSUM,PHSCON(5,5),XJ(5,5),X7,XDEL,X10
      DIMENSION IJ( 5),IT( 5),PHI( 5),ARI( 5),TP( 5),RR( 5),RI( 5),KL(5)
      DIMENSION SF( 5),RRGPHI( 5),TRR( 5),TRI( 5),ZI( 5),PHSCOR( 5, 5)
      DIMENSION DELZ2(5),PSFI(5),R(5),PSFT(5)
      DIMENSION CMHLD(5),BPFIR(5)
      DOUBLE PRECISION RR,RI
 2218 FORMAT(//1H0,50X,35HOUTPUT COEF. MATRIX OF ORIG. SYSTEM)
 2219 FORMAT(/1H0,50X,34HINPUT COEF. MATRIX OF ORIG. SYSTEM)
 2220 FORMAT(/1H0,53X,28HORIGINAL SYSTEM MODAL MATRIX)
 2221 FORMAT(/1H0,49X,36HINVERSE OF ORIG. SYSTEM MODAL MATRIX)
 2222 FORMAT(/1H0,49X,37HNORMAL FORM OUTPUT COEFFICIENT MATRIX)
 2223 FORMAT(1H0,19X,13HCOEFFICIENT A,I1,1H=,F10.4,F10.4)
 2224 FORMAT(1H0,18X,13HCOEFFICIENT A,I1,2H*=,F10.4,F10.4)
 2225 FORMAT(1H0,27X,7HARGPHI=,F10.4,F10.4)
    1 FORMAT(I2)
    9 FORMAT (I2,F10.4)
   11 FORMAT (8F15.4)
   13 FORMAT (2F10.4)
   14 FORMAT (4F15.4,4F15.4)
   15 FORMAT(10F7.4)
 2201 FORMAT(/1H0,9X,15HCALCULATION NO.,I2,12H  **********)
 2202 FORMAT(/1H0,60X,17HVANDERMOND MATRIX)
 2203 FORMAT(1H0, 8X,6H(REAL),8X,11H(IMAGINARY),5X,6H(REAL)8X,11H(IMAGIN
     1ARY),5X,6H(REAL),8X,11H(IMAGINARY),5X,6H(REAL),8X,11H(IMAGINARY))
 2204 FORMAT(/1H0,56X,25HINVERSE VANDERMOND MATRIX)
 2205 FORMAT(/1H0,49X,38HOUTPUT COEFFICIENT MATRIX(NORMAL FORM))
 2206 FORMAT(/1H0,30X, 4HPHI=,F7.4,8H RADIANS)
 2207 FORMAT(/1H0,15X,19HTIME TO FIRST PEAK=,F10.4,8H SECONDS)
 2208 FORMAT(/1H0,31X, 3HX1=,F10.4)
 2209 FORMAT(/1H0,16X,18HSTEADINESS FACTOR=,F7.4)
 2270 FORMAT(1H0,41HSUMMATION OF COEFFICIENTS OF REAL MODES =,F10.4,F10.
     14)
 2269 FORMAT(1H0,44HSUMMATION OF COEFFICIENTS OF COMPLEX MODES =,F10.4,F
     110.4)
 2268 FORMAT(1H0,    32HTOTAL SYSTEM STEADINESS FACTOR =,F7.4)
 2210 FORMAT(/1H0,48X,6H(REAL),8X,11H(IMAGINARY))
 2211 FORMAT(1H0,40X,F15.4,F15.4)
 2212 FORMAT(//1H0,50X,21HRESULTING EIGENVALUES)
 2213 FORMAT(//1H0,52X,30HRESULTING PHASE CANONICAL FORM)
 2214 FORMAT(1H0,52X,28HPHASE CANONICAL MODAL MATRIX)
 2215 FORMAT(1H0,47X,39HINVERSE OF PHASE CANONICAL MODAL MATRIX)
 2216 FORMAT(1H0,48X,36HPHASE CANONICAL OUTPUT COEFF. MATRIX)
 2217 FORMAT(1H0,49X,35HPHASE CANONICAL INPUT COEFF. MATRIX)
      IF(KSICK.GT.0)GO TO 2227
      READ(5,15)CORIG
      READ(5,15)BORIG
      DO2228 I=1,N
 2228 BFIX(I)=BORIG(I)
 2227 WRITE(6,2218)
      WRITE(6,14)(CORIG(I),I=1,N)
      WRITE(6,2219)
      WRITE(6,14)(BORIG(I),I=1,N)
      DO 103 I=1,N
```

```
      TRR(I)=RR(I)
      TRI(I)=RI(I)
  103 CONTINUE
      INDX3=0
      KAY=0
      JERRY=1
      DO1074 J=1,N
      INDX1=0
      INDX2=0
      SUM=(0.0,0.0)
      DO1074 I=1,N
      ASUM=SUM+CORIG(I)*XMAJ(I,J)
      SUM=ASUM
      CN(J)=SUM
 1074 CONTINUE
      DO1075 I=1,N
      SUM=(0.0,0.0)
      DO1075 J=1,N
      ASUM=SUM+XMAJIN(I,J)*BORIG(J)
      SUM=ASUM
      BN(I)=SUM
 1075 CONTINUE
      KSAK=0
      KSAP=0
      KBO=0
  108 DO 6 I=1,N
      X=TRR(I)
      Y=TRI(I)
      ZLMDA(I)=CMPLX(X,Y)
    6 CONTINUE
      DO7 J=1,N
      DO7 I=1,N
      K=I-1
      RM(I,J)=ZLMDA(J)**K
    7 CONTINUE
      DO8 I=1,N
      DO8 J=1,N
      D(I,J)=RM(I,J)
    8 CONTINUE
      IF(KSAK.GT.0)GO TO 2251
      KDS=KSICK+1
      WRITE(6,2201)KDS
      WRITE(6,2202)
      WRITE(6,2203)
      DO71 I=1,N
      WRITE(6,14)(D(I,J),J=1,N)
   71 CONTINUE
 2251 CALL CINV(D,N,DINV,KKK)
      DO91 I=1,N
      DO91 J=1,N
      RMINV(I,J)=DINV(I,J)
   91 CONTINUE
      IF(KSAK.GT.0)GO TO 2252
      WRITE(6,2204)
      WRITE(6,2203)
      DO 81 I=1,N
      WRITE(6,14)(RMINV(I,J),J=1,N)
   81 CONTINUE
 2252 DO79 J=1,N
      DO79 I=1,N
      RMI(I,J)=CN(J)*RM(I,J)
   79 CONTINUE
```

```
2253 CALL CINV(RMI,N,RMIINV,KKK)
2254 IF(KBO.GT.0)GO TO 1015
     DO82 I=1,N
     SUM=(0.0,0.0)
     DO82 J=1,N
     ASUM=SUM+CN(J)*RMIINV(J,I)
     SUM=ASUM
     CP(I)=SUM
  82 CONTINUE
     IF(KSAK.GT.0)GO TO 2255
     WRITE(6,2216)
     WRITE(6,2203)
     WRITE(6,14)(CP(I),I=1,N)
2255 DO83 I=1,N
     SUM=(0.0,0.0)
     DO83 J=1,N
     ASUM=SUM+RMI(I,J)*BN(J)
     SUM=ASUM
     BP(I)=SUM
  83 CONTINUE
1015 IF(KSAK.GT.0)GO TO 2256
     WRITE(6,2217)
     WRITE(6,2203)
     WRITE(6,14)(BP(I),I=1,N)
2256 IF(KBO.LT.1)GO TO 1011
     DO101 J=1,N
     SUM=(0.0,0.0)
     DO101 I=1,N
     ASUM=SUM+CP(I)*RM(I,J)
     SUM=ASUM
     CN(J)=SUM
 101 CONTINUE
     DO1011 I=1,N
     SUM=(0.0,0.0)
     DO1011 J=1,N
     ASUM=SUM+RMINV(I,J)*BP(J)
     SUM=ASUM
     BN(I)=SUM
1011 CONTINUE
     DO1012 I=1,N
     CN(I)=CN(I)*BN(I)
     BN(I)=(1.0,0.0)
1012 CONTINUE
     IF(KSAK.GT.0)GO TO 1013
     WRITE(6,2205)
     WRITE(6,2203)
     WRITE(6,14)(CN(I),I=1,N)
1013 K=0
     KR=0
     KBO=KBO+1
     DO 201 I=1,N
     X=TRI(I)
     IF(X)19,202,19
  19 DO 20 J=1,N
     Y=TRI(J)
     IF(X+Y)20,21,20
  21 K=K+1
     IJ(K)=J
     IT(K)=I
  20 CONTINUE
     GO TO 201
 202 KR=KR+1
```

```
      RLRT(KR)=TRR(I)
      KL(KR)=I
  201 CONTINUE
      K=K/2
      KIM=K
      SUM3=0.0
      DO 22 I=1,K
      IS=I+1
      N1=IJ(I)
      N2=IT(I)
      X=TRR(N1)
      Y=TRI(N1)
      XLAM2=CMPLX(X,Y)
      X=TRR(N2)
      Y=TRI(N2)
      XLAM1=CMPLX(X,Y)
      A1=CN(N2)/(-XLAM1)
      A2=CN(N1)/(-XLAM2)
      IF(KSAK.GT.0)GO TO 2257
      WRITE(6,2223)I,A1
      WRITE(6,2224)I,A2
 2257 IF(Y.GT.0.0)GO TO 203
      ARGPHI(I)=(A2+A1)/(A2-A1)
      GO TO 204
  203 ARGPHI(I)=(A1+A2)/(A1-A2)
  204 IF(KSAK.GT.0)GO TO 2258
      WRITE(6,2225)ARGPHI(I)
 2258 XY=ARGPHI(I)
      RRGPHI(I)=AIMAG(XY)
      PHI(I)=ATAN(RRGPHI(I))
      X7=A1+A2
      X8=REAL(X7)
      X10=-A1+A2
      X9=AIMAG(X10)
      IF(X8.LT.0.0)GO TO 2239
      IF(X9.LT.0.0)GO TO 2238
      GO TO 205
 2238 PHI(I)=PHI(I)+3.1416
      GO TO 205
 2239 IF(X9.LT.0.0)GO TO 205
      PHI(I)=PHI(I)+3.1416
  205 IF(KSAK.GT.0)GO TO 2259
      WRITE(6,2206)PHI(I)
 2259 ARI(I)=ABS(TRI(N2))
      TP(I)=(9.4251/2.0-PHI(I))/ARI(I)
      IF(KSAK.GT.0)GO TO 2260
      WRITE(6,2207)TP(I)
 2260 X1=TRR(N2)*TP(I)
      IF(KSAK.GT.0)GO TO 2261
      WRITE(6,2208)X1
 2261 X6=ABS(PHI(I))
      XSF=(A1+A2)*(1.0-EXP(X1)/SIN(X6))
      SF(I)=REAL(XSF)
      SUM3=SUM3+SF(I)
      IF(KSAK.GT.0)GO TO 2262
      WRITE(6,2209)SF(I)
 2262 CMD=A1+A2
      CMHLD(I)=CABS(CMD)
   22 CONTINUE
 2263 SUM2=(0.0,0.0)
      DO220 I=1,KR
      N3=KL(I)
```

```
      RLC=CN(N3)/(-TRR(N3))
      RLCOF=SUM2+RLC
      SUM2=RLCOF
  220 CONTINUE
      IF(KSAK.GT.0)GO TO 2264
      WRITE(6,2270)RLCOF
 2264 XLCOF=REAL(RLCOF)
      SFT=SUM3+XLCOF
      SUM6=(0.0,0.0)
      DO23 I=1,K
      DENX1=SUM6+CMHLD(I)
      SUM6=DENX1
   23 CONTINUE
      IF(KSAK.GT.0)GO TO 2265
      WRITE(6,2269)DENX1
 2265 DENX2=DENX1+RLCOF
 2266 DENR2=REAL(DENX2)
      SFT=SFT/DENR2
      IF(KSAK.GT.0)GO TO 2267
      WRITE(6,2268)SFT
 2267 IF(KAY.GT.0)GO TO 102
      IF(INDX1.GT.0)GO TO 1065
      GO TO 107
 1021 SFOR=SFT
      KSAK=KSAK+1
      KSAP=KSAP+1
      IF(KSAP.GT.1)GO TO 107
 1065 IF(SFT-SFOR)1061,1069,1063
 1061 DEL(INDX2)=-DEL(INDX2)
 1063 TRR(IA)=XHOLD
      XDEL=DEL(INDX2)
      YDEL=REAL(XDEL)
      PSFT(INDX2)=(SFT-SFOR)/YDEL
      PSFT(INDX2)=ABS(PSFT(INDX2))
 1069 INDX2=INDX2+1
      IF(INDX2-KR)1066,1066,2232
 1066 IA=KL(INDX2)
      DEL(INDX2)=RLRT(INDX2)*0.2
      RLRT1=RLRT(INDX2)+DEL(INDX2)
      XHOLD=TRR(IA)
      TRR(IA)=REAL(RLRT1)
      INDX1=INDX1+1
      KR=0
      GO TO 108
  102 IF(SFT-SFOR)2231,2232,2233
 2231 DELZ2(INDX3)=-DELZ2(INDX3)
 2233 PSFI(INDX3)=(SFT-SFOR)/DELZ2(INDX3)
      PSFI(INDX3)=ABS(PSFI(INDX3))
      N1=IJ(INDX3)
      N2=IT(INDX3)
      TRI(N2)=TI2HLD
      TRR(N2)=TR2HLD
      TRI(N1)=-TRI(N2)
       TRR(N1)=TRR(N2)
      ZI(INDX3)=ZIHLD
 2232 INDX3=INDX3+1
      IF(INDX3.GT.KIM)GO TO 2?30
      N1=IJ(INDX3)
      N2=IT(INDX3)
      X=TRR(N2)
      Y=TRI(N2)
      Z2=CMPLX(X,Y)
```

```
      R(INDX3)=CABS(Z2)
      TR1=ABS(TRR(N2))
      ZI(INDX3)=ATAN(TRI(N2)/TR1)
      DELZ2(INDX3)=0.1*ZI(INDX3)
      ZIHLD=ZI(INDX3)
      ZI(INDX3)=ZI(INDX3)-DELZ2(INDX3)
      TR2HLD=TRR(N2)
      TI2HLD=TRI(N2)
      TRI(N2)=R(INDX3)*SIN(ZI(INDX3))
      TRR(N2)=-R(INDX3)*COS(ZI(INDX3))
      TRI(N1)=-TRI(N2)
      TRR(N1)=TRR(N2)
      KAY=KAY+1
      GO TO 108
 2230 SUM7=0.0
      DO2234 I=1,KR
 2234 SUM7=SUM7+PSFT(I)**2.0
      SUM8=0.0
      DO2235 I=1,KIM
 2235 SUM8=SUM8+PSFI(I)**2.0
      SUMPT=SUM7+SUM8
      SUMPT=SUMPT**0.5
      DO2236 I=1,KR
 2236 DEL(I)=DEL(I)*(PSFT(I)/SUMPT)
      DO2237 I=1,KIM
      DELZ2(I)=DELZ2(I)*(PSFI(I)/SUMPT)
      ZI(I)=ZI(I)-DELZ2(I)
      N1=IJ(I)
      N2=IT(I)
      TRI(N2)=R(I)*SIN(ZI(I))
      TRR(N2)=-R(I)*COS(ZI(I))
      TRI(N1)=-TRI(N2)
 2237 TRR(N1)=TRR(N2)
      DO1068 I=1,KR
      IA=KL(I)
      RLRT(I)=TRR(IA)+DEL(I)
      YX=RLRT(I)
      TRR(IA)=REAL(YX)
 1068 CONTINUE
      KAY=0
      INDX1=0
      KSAK=0
      GO TO 108
  107 CONTINUE
      WRITE(6,2212)
      WRITE(6,2210)
      WRITE(6,2211)(TRR(I),TRI(I),I=1,N)
      DO1006 I=1,N
      DO1006 J=1,N
      IF(I-J)1007,1008,1007
 1007 XJ(I,J)=(0.0,0.0)
      GO TO 1006
 1008 XJ(I,J)=ZLMDA(I)
 1006 CONTINUE
      DO1002 K=1,N
      DO1002 I=1,N
      SUM=(0.0,0.0)
      DO1001 J=1,N
      ASUM=SUM+D(I,J)*XJ(J,K)
      SUM=ASUM
 1001 CONTINUE
      AX1(I,K)=SUM
```

```
1002 CONTINUE
     DO1003 K=1,N
     DO1003 I=1,N
     SUM=(0.0,0.0)
     DO1005 J=1,N
     BSUM=SUM+AX1(I,J)*RMINV(J,K)
     SUM=BSUM
1005 CONTINUE
     PHSCON(I,K)=SUM
1003 CONTINUE
     WRITE(6,2213)
     WRITE(6,2203)
     DO1009 I=1,N
1009 WRITE(6,14)(PHSCON(I,J),J=1,N)
     DO1004 I=1,N
     DO1004 J=1,N
     XX=PHSCON(I,J)
     PHSCOR(I,J)=REAL(XX)
1004 CONTINUE
     IF(KSAP.GT.0)GO TO 2226
     DO2245 I=1,N
2245 BNFIX(I)=CN(I)
     DO2247 J=1,N
     SUM=(0.0,0.0)
     DO2246 I=1,N
2246 SUM=SUM+RM(J,I)*BNFIX(I)
2247 BPFIX(J)=SUM
     DO2248 I=1,N
     FIXER=BPFIX(I)
2248 BPFIR(I)=REAL(FIXER)
     XSUM=BPFIR(N)
     J=N-1
     DO2250 I=1,J
     K=I+1
2250 XSUM=XSUM-BPFIR(I)*PHSCOR(N,K)
     XINPT=-PHSCOR(N,1)/XSUM
     XKP=XINPT
     IF(SFT-SFK)1021,2226,2226
2226 CONTINUE
     RETURN
     END
```

```
$IBFTC INV
      SUBROUTINE CINV(D,N,DINV,KKK)
      COMPLEX D( 5, 5),DINV( 5, 5),S(10),B( 5,10),T
1000  FORMAT(6E15.4)
      MM=2*N
      KKK=0
      DO 23 I=1,N
      DO 23 J=1,N
23    B(I,J)=D(I,J)
      K=N+1
      DO 24 I=1,N
      DO 24 J=K,MM
24    B(I,J)=(0.0,0.0)
      DO 25 I=1,N
      K=I+N
25    B(I,K)=(1.0,0.0)
      DO 33 I=1,N
      J=I
      IF(CABS(B(I,J)).GT.1.0E-08) GO TO26
      L=I+1
      DO 12 K=L,N
      IF(CABS(B(K,J)).GT.1.0E-08) GO TO27
12    CONTINUE
      KKK=1
      RETURN
27    DO 13 M=1,MM
13    S(M)=B(K,M)
      DO 14 M=1,MM
14    B(K,M)=B(I,M)
      DO 16 M=1,MM
16    B(I,M)=S(M)
26    T=B(I,J)
      DO 17 K=1,MM
17    B(I,K)=B(I,K)/T
      L=I+1
      IF(L.GT.N) GO TO 33
      DO 10 K=L,N
      T=B(K,J)
      DO 10 NN=1,MM
10    B(K,NN)=B(K,NN)-T*B(I,NN)
33    CONTINUE
      IF(N.EQ.1) GO TO 50
      DO 20 I=2,N
      J=I
      L=I-1
29    T=B(L,J)
      DO 21 K=J,MM
21    B(L,K)=B(L,K)-T*B(I,K)
      IF(L.EQ.1) GO TO 20
      L=L-1
      GO TO 29
20    CONTINUE
50    DO 30 I=1,N
      DO 30 J=1,N
      NN=J+N
30    DINV(I,J)=B(I,NN)
      RETURN
      END
```

```
$IBFTC DIFSOL
      SUBROUTINE DIFSOL(Q,N,NZZZ,TMRSP1,TMRSP2,IJK,E,ABE)
      DIMENSION  Q( 5, 5),B( 5, 5),V( 5),XIC( 5),S( 5, 5),TMRSP1(202),TM
     1RSP2(202),E(5)
      COMMON Y(202)
      COMMON/DATA/NXK,S,B,V
1     FORMAT (F10.5,I10)
   14 FORMAT(///1H0,44X,44HTABULATED SOLUTION OF DIFFERENTIAL EQUATIONS)
   15 FORMAT(/1H0,9X,5HFIRST,10X,6HSECOND,9X,5HTHIRD,10X,6HFOURTH,9X,5HF
     1IFTH,10X,5HSIXTH)
   16 FORMAT(1H0,8X,8HVARIABLE,7X,8HVARIABLE,6X,8HVARIABLE,7X,8HVARIABLE
     1,6X,8HVARIABLE,7X,8HVARIABLE)
2     FORMAT(8F10.5)
    4 FORMAT(1H0,30X,4F15.5)
5     FORMAT(8F15.5)
   10 FORMAT(//1H0,54X,24HINPUT COEFFICIENT MATRIX)
   11 FORMAT(//1H0,54X,24HSTATE COEFFICIENT MATRIX)
   13 FORMAT(//1H0,60X,12HINPUT MATRIX)
    6 FORMAT(///1H0,40X,51H*****SOLUTION OF MATRIX DIFFERENTIAL EQUATION
     1S*****)
    9 FORMAT(//1H0,57X,18HINITIAL CONDITIONS)
    8 FORMAT(/1H0,30X,4F15.5)
      IF(NZZZ-1)102,102,1025
  102 READ(5,1)A,NST
      READ(5,2)(XIC(I),I=1,N)
      NZZZ=NZZZ+1
 1025 CONTINUE
      NXK=N
      DO103 I=1,N
      DO103 J=1,N
      S(I,J)=Q(I,J)
  103 CONTINUE
      DO1026 I=1,N
      B(I,1)=E(I)
 1026 CONTINUE
      DO1027 J=2,N
      DO1027 I=1,N
      B(I,J)=0.0
 1027 CONTINUE
      V(1)=ABE
      DO105 I=2,N
      V(I)=0.0
  105 CONTINUE
      WRITE(6,6)
      WRITE(6,9)
      WRITE(6,8)(XIC(I),I=1,N)
      WRITE(6,11)
      DO106 I=1,N
      WRITE(6,4)(Q(I,J),J=1,N)
  106 CONTINUE
      WRITE(6,10)
      DO107 I=1,N
      WRITE(6,4)(B(I,J),J=1,N)
  107 CONTINUE
      WRITE(6,13)
      WRITE(6,4)        (V(I),I=1,N)
      DO 7 I=1,N
7     Y(I)=XIC(I)
      Y(N+1)=0.0
      NT=N+1
      FNST=NST
      TF=FNST*A
```

```
      NT=NT*2
      CALL START(N,3,1,A,.01 ,0.0,0.1,.000001,0.5                    )
      WRITE(6,14)
      WRITE(6,15)
      WRITE(6,16)
      KNT=N+2
      WRITE(6,5)(Y(I),I=1,KNT)
      IJK=0
   12 IJK=IJK+1
      CALL KAMSUB(1)
      WRITE(6,5)(Y(I),I=1,KNT)
      KXT=N+1
      TMRSP1(IJK)=Y(1)
      TMRSP2(IJK)=Y(KXT)
      IF(Y(N+1)-TF)12,3,3
3     CONTINUE
      RETURN
      END
$IBFTC DERFUN
```

```
$IBFTC START
      SUBROUTINE START(M1,M2,M3,A1,A2,A3,A4,A5,A6)
      COMMON/SHARE/NN,SPACE,MODE,KKA,E1MAX,E1MIN,E2MAX,E2MIN,FACT
C
C    NO INFORMATION IS REQUIRED IN THIS SUBROUTINE
C
      NN=M1
      MODE=M2
      KKA=M3
      E1MAX=A2
      E1MIN=A3
      E2MAX=A4
      E2MIN=A5
      FACT=A6
      SPACE=A1
      CALL KAMSUB(0)
      END
C
$IBFTC KAMSUB
      SUBROUTINE KAMSUB(NSTART)
      DIMENSION DELY(4,100),BET(4),XV(5),FV(4,100),YU(5,100)
      COMMON /SHARE/NN,SPACE,MODE,KKA,E1MAX,E1MIN,E2MAX,E2MIN,FACT
      COMMON Y(202)
      COMMON/INTDAT/Z(5,202),IERR
      DOUBLE PRECISION YU
C
C    NO INFORMATION IS REQUIRED IN THIS SUBROUTINE
C
      IF(NSTART.LE.0)GO TO 9977
      GO TO (1001,2000,2000),MODE
C      RUNGE-KUTTA
1000  LL=1
1001  DO 1034 K=1,4
      DO 1350 I=1,NN
      DELY(K,I)=Y(N2)*FV(MM,I)
      Q=YU(MM,I)
1350  Y(I)=Q+BET(K)*DELY(K,I)
      Y(NP1)=BET(K)*Y(N2)+XV(MM)
      CALL DERFUN
      DO 1100 I=1,NN
      IPN2=I+N2
1100  FV(MM,I)=Y(IPN2)
1034  CONTINUE
      DO 1039 I=1,NN
      DEL=(DELO(1,I)+2.0*DELY(2,I)+2.0*DELY(3,I)+DELY(4,I))/6.0
      YU(MM+1,I)=YU(MM,I)+DEL
1039  CONTINUE
      MM=MM+1
      XV(MM)=XV(MM-1)+Y(N2)
      DO 1400 I=1,NN
1400  Y(I)=YU(MM,I)
      Y(NP1)=XV(MM)
      CALL DERFUN
      GO TO (42,100,100),MODE
100   DO 150 I=1,NN
      IPN2=I+N2
150   FV(MM,I)=Y(IPN2)
      GO TO (1001,1001,1001,2000),MM
C     ADAMS-MOULTON
2000  DO 2048 I=1,NN
      DEL=Y(N2)*(55.0*FV(4,I)-59.0*FV(3,I)
     1+37.0*FV(2,I)-9.0*FV(1,I))/24.0
```

```
      Y(I)=YU(4,I)+DEL
2048  DELY(1,I)=Y(I)
      Y(NP1)=XV(4)+Y(N2)
      CALL DERFUN
      XV(5)=Y(NP1)
      DO 2051 I=1,NN
      IPN2=I+N2
      DEL=Y(N2)*(9.0*Y(IPN2)+19.0*FV(4,I)
     1-5.0*FV(3,I)+FV(2,I))/24.0
      YU(5,I)=YU(4,I)+DEL
2051  Y(I)=YU(5,I)
      CALL DERFUN
      GO TO (42,42,3000),MODE
C     ERROR ANALYSIS
3000  SSE=0.0
      DO 3033 I=1,NN
      EPSIL=R*ABS(Y(I)-DELY(1,I))
      GO TO (3301,3307),KKA
3301  IF(Y(I))3650,3307,3650
3650  EPSIL=EPSIL/ABS(Y(I))
3307  IF(SSE-EPSIL)3032,3033,3033
3032  SSE=EPSIL
3033  CONTINUE
      IF(E1MAX-SSE)3034,3034,3035
3034  IF(ABS(Y(N2))-E2MIN)42,42,4340
3035  IF(SSE-E1MIN)3036,42,42
3036  IF(E2MAX-ABS(Y(N2)))42,42,5360
4340  LL=1
      IERR = 1
      MM=1
      Y(N2)=Y(N2)*FACT
      GO TO 1001
5360  GO TO (42,5361),LL
5361  XV(2)=XV(3)
      XV(3)=XV(5)
      DO 5363 I=1,NN
      FV(2,I)=FV(3,I)
      IPN2=I+N2
      FV(3,I)=Y(IPN2)
      YU(2,I)=YU(3,I)
5363  YU(3,I)=YU(5,I)
      Y(N2)=2.0*Y(N2)
      IERR = 2
      LL=2
      MM=3
      GO TO 1001
C     EXIT ROUTINE
   42 GO TO(43,44,44),MODE
   44 DO 707 K=1,4
      Z(K,NP1)= XV(K)
      Z(K,N2)=XV(K+1)-XV(K)
      DO 707 I=1,NN
      Z(K,I)= YU(K,I)
      IPN2 = N2 + I
  707 Z(K,IPN2)= FV(K,I)
   43 Z(5,NP1) = XV(5)
      DO 708 I= 1,NN
      Z(5,I)= YU(5,I)
      IPN2  = N2 + I
  708  Z(5,IPN2)= Y(IPN2)
      Z(5,N2)= Y(N2)
      DO 12 K=1,3
```

```
      XV(K)=XV(K+1)
      DO 12 I=1,NN
      FV(K,I)=FV(K+1,I)
12    YU(K,I)=YU(K+1,I)
      LL=2
      MM=4
      XV(4)=XV(5)
      DO 52 I=1,NN
      IPN2=I+N2
      FV(4,I)=Y(IPN2)
52    YU(4,I)=YU(5,I)
      GO TO (70,70,73),MODE
9977  CONTINUE
      IERR = 3
      ALPHA=Y(NN+1)
      EPM=0.0
      GO TO (7,9,9),MODE
7     MM=4
      GO TO 8
9     MM=1
8     BET(1)=0.5
      BET(2)=0.5
      BET(3)=1.0
      BET(4)=0.0
5     N2=NN+2
      Y(N2)=SPACE
      NP1=NN+1
      R=19.0/270.0
      XV(MM)=Y(NP1)
      IF(E1MIN)2,2,1
2     E1MIN=E1MAX/55.0
1     IF(FACT)4,4,3
4     FACT=1.0/2.0
3     CALL DERFUN
      DO 320 I=1,NN
      IPN2=I+N2
      FV(MM,I)=Y(IPN2)
320   YU(MM,I)=Y(I)
      GO TO 1000
73    E=ABS(XV(4)-ALPHA)
      IF(E-EPM)2000,2000,71
71    EPM=E
70    RETURN
      END
```

```
      SUBROUTINE DERFUN
      DIMENSION  S( 5, 5),B( 5, 5),V( 5),XIC( 5)
      COMMON Y(202)
      COMMON/DATA/NXK,S,B,V
    5 FORMAT(4F15.5,I10)
      N=NXK
      DO 7 I=1,N
      SUM=0.
      DO 8 J=1,N
8     SUM=SUM+S(I,J)*Y(J)+B(I,J)*V(J)
      KI=N+2+I
      Y(KI)=SUM
    7 CONTINUE
      RETURN
      END
```

```
$IBFTC PLOT
      SUBROUTINEPLOT(X,XMIN,XMAX,LX,Y,YMIN,YMAX,LY,Z,ZMIN,ZMAX,LZ,NPT,
     1NPLOT,NCOPY,NCD,NDIM)
      DIMENSION X(1),Y(1),Z(1),SX(7),TITLE(12),L(134),NCH(41),MOP(18)
1     FORMAT(12A6)
    2 FORMAT(58A1,3A6,4A1)
3     FORMAT(1H1,26X,12A6)
4     FORMAT(1H ,A1,1PE9.2,121A1)
5     FORMAT(132A1)
6     FORMAT(1PE17.2,5E20.2,E15.2)
    7 FORMAT(1PE17.2,E116.2)
    8 FORMAT(1PE17.2,E61.2,E55.2)
    9 FORMAT(1PE17.2,2E40.2,E36.2)
   10 FORMAT(1PE17.2,3E30.2,E26.2)
   11 FORMAT(1PE17.2,4E24.2,E20.2)
   12 FORMAT(1HK,62X,3A6)
      SLOG(F)=ALOG(F)/2.302585
      LLX=LX+1
      NDD=NCD+1
        GOTO(15,13,14,13),NDD
13     READ(5,1)(TITLE(I),I=1,12)
   14 IF(NDD.GE.3)READ(5,2)(MOP(I),I=1,18),(NCH(I),I=1,40),TAB1,TAB2,TAB
     13,ND,NP,NM,NB
   15 NCH(41)=NB
      NPN=NPT/NPLOT
      IF(LX.GT.0)GOTO17
      CX=120./(XMAX-XMIN)
      SX(1)=XMIN
      SX(7)=XMAX
      U=XMIN
      DO16K=2,6
      U=(XMAX-XMIN)/6.+U
   16 SX(K)=U
      GOTO19
   17 XLX=LX
      CX=120./XLX
      NX=SLOG(XMIN)
      DO18K=1,LLX
   18 SX(K)=10.**(NX+K-1)
   19 CALLPOT(X,XMIN,LX,NPT,0,120.,CX)
      IF(LY.GT.0)GOTO20
      CY=50./(YMAX-YMIN)
      GOTO21
   20 YLY=LY
      CY=50./YLY
      KY=CY
      NY=SLOG(YMIN)
   21 CALLPOT(Y,YMIN,LY,NPT,1,50.,CY)
      IF(NDIM.LT.3)GOTO24
      IF(LZ.GT.0)GOTO22
      CZ=40./(ZMAX-ZMIN)
      GOTO23
   22 ZLZ=LZ
      CZ=40./ZLZ
   23 CALLPOT(Z,ZMIN,LZ,NPT,0,40.,CZ)
   24 DO50NN=1,NCOPY
      M1=1
      T1=33.
      LYY=LY
      TT=50.
      WRITE(6,3)(TITLE(I),I=1,12)
      DO43KK=1,51
```

```
      N=1
      NNN=NPN
      JED=1
      T=51-KK
      DO25J=1,133
   25 L(J)=NB
      L(133)=ND
      IF(LY.GT.0)GOTO26
      L(13)=NP
      IF(T.GT.TT)GOTO30
      SCALE=T/CY+YMIN
      L(133)=NP
      N=0
      TT=TT-5.
      IF(T.LE.0.)SCALE=YMIN
      GOTO30
   26 GOTO(27,27,28,28,27,28),LY
   27 SS=KY*LYY
      GOTO29
   28 SS=KY*LYY+1
   29 L(13)=ND
      IF(T.GT.SS)GOTO30
      SCALE=10.**(NY+LYY)
      N=0
      LYY=LYY-1
      L(13)=NP
      L(133)=NP
   30 IF(50..EQ.T)GOTO31
      IF(0..NE.T)GOTO37
   31 DO32J=14,133
   32 L(J)=NM
      IF(LX.GT.0)GOTO34
      DO33J=13,133,10
   33 L(J)=NP
      GOTO36
   34 KX=120/LX
      DO35J=13,133,KX
   35 L(J)=NP
   36 IF(50..EQ.T)L(133)=ND
   37 DO40LM=1,NPLOT
      DO39I=JED,NNN
      IF(Y(I).NE.T)GOTO39
      J=X(I)
      IF(NDIM.NE.3)GOTO38
      IZ=Z(I)
      L(J+13)=NCH(IZ+1)
      GOTO39
   38 L(J+13)=NCH(LM)
   39 CONTINUE
      JED=NNN+1
      NNN=NNN+NPN
   40 CONTINUE
      IF(T1.NE.T)GOTO41
      IF(15..GE.T)GOTO41
      L(2)=MOP(M1)
      M1=M1+1
      T1=T1-1.
   41 IF(N.EQ.1)GOTO42
      WRITE(6,4)L(2),SCALE,(L(J),J=12,132)
      GOTO43
42    WRITE(6,5)(L(J),J=1,132)
   43 CONTINUE
```

```
      GOTO(44,45,46,47,48,49,44),LLX
   44 WRITE(6,6)(SX(K),K=1,7)
      GOTO50
   45 WRITE(6,7)(SX(K),K=1,LLX)
      GOTO50
   46 WRITE(6,8)(SX(K),K=1,LLX)
      GOTO50
   47 WRITE(6,9)(SX(K),K=1,LLX)
      GOTO50
   48 WRITE(6,10)(SX(K),K=1,LLX)
      GOTO50
   49 WRITE(6,11)(SX(K),K=1,LLX)
   50 WRITE(6,12)TAB1,TAB2,TAB3
      RETURN
      END
$IBFTC POT
      SUBROUTINEPOT(V,VMIN,LV,NP,J,VC,C)
      DIMENSIONV(1)
      IF(LV.GT.0)GOTO2
      DO1I=1,NP
    1 V(I)=FLOAT(IFIX(C*(V(I)-VMIN)+.5))
       GOTO4
    2 DO3I=1,NP
    3 V(I)=FLOAT(IFIX(C*(ALOG(V(I)/VMIN)/2.302585)+.5))
    4 IF(J.GT.0)GOTO7
      DO6I=1,NP
      IF(V(I).LT.0.)GOTO5
      IF(V(I).LE.VC)GOTO6
    5 V(I)=VC+1.
    6 CONTINUE
    7 RETURN
      END
```

APPENDIX F

SAMPLE PROGRAM OUTPUT

***** TIME DOMAIN COMPENSATION OF A LINEAR SYSTEM *****

THIS SYSTEM IS OF ORDER 3

INPUT MATRIX

ROW 1
-0.2300000D-01 -0.6570000D 04 0.

ROW 2
0. 0. 0.1000000D-01

ROW 3
0.5900000D 00 -0.2880000D 04 -0.1500000D 01

	EIGENVALUES(REAL PART)	EIGENVALUES(IMAGINARY PART)
1	-0.137690531D 01	0.
2	-0.730473430D-01	0.535051451D 01
3	-0.730473430D-01	-0.535051451D 01

FIRST EIGENVECTOR		SECOND EIGENVECTOR		THIRD EIGENVECTOR		FOURTH EIGENVECTOR
-0.9996D 00	-0.	0.9165D 00	-0.2109D-01	0.9165D 00	0.2109D-01	
-0.2060D-03	-0.	-0.1019D-04	-0.7465D-03	-0.1019D-04	0.7465D-03	
0.2836D-01	0.	0.3995D 00	0.	0.3995D 00	0.	

OUTPUT COEF. MATRIX OF ORIG. SYSTEM

1.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

INPUT COEF. MATRIX OF ORIG. SYSTEM

3.1250	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

CALCULATION NO. 1 **********

VANDERMOND MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
-1.3769	0.0000	-0.0730	5.3505	-0.0730	-5.3505
1.8959	-0.0000	-28.6227	-0.7817	-28.6227	0.7817

INVERSE VANDERMOND MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
0.9441	0.0000	0.0048	0.0000	0.0330	0.0000
0.0279	-0.1219	-0.0024	-0.0940	-0.0165	-0.0040
0.0279	0.1219	-0.0024	0.0940	-0.0165	0.0040

PHASE CANONICAL OUTPUT COEFF. MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
1.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000

PHASE CANONICAL INPUT COEFF. MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
3.1250	0.0000	-0.0719	-0.0000	0.0017	0.0000

OUTPUT COEFFICIENT MATRIX(NORMAL FORM)

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
2.9501	0.0000	0.0875	-0.3741	0.0875	0.3741

COEFFICIENT A1= 0.0701 0.0154

COEFFICIENT A1*= 0.0701 -0.0154

ARGPHI= 0.0000 -4.5569

PHI= 1.7868 RADIANS

TIME TO FIRST PEAK= 0.5468 SECONDS

X1= -0.0399

STEADINESS FACTOR= 0.0023

SUMMATION OF COEFFICIENTS OF REAL MODES = 2.1425 0.0000

SUMMATION OF COEFFICIENTS OF COMPLEX MODES = 0.1402 0.0000

TOTAL SYSTEM STEADINESS FACTOR = 0.9396

RESULTING EIGENVALUES

(REAL)	(IMAGINARY)
-1.3769	0.0000
-0.0730	5.3505
-0.0730	-5.3505

RESULTING PHASE CANONICAL FORM

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
-0.0000	-0.0000	1.0000	-0.0000	-0.0000	0.0000
-0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
-39.4254	-0.0000	-28.8345	0.0000	-1.5230	-0.0000

CALCULATION NO. 1 **********

VANDERMOND MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
-1.2425	0.0000	-0.7973	5.2913	-0.7973	-5.2913
1.5438	-0.0000	-27.3618	-8.4379	-27.3618	8.4379

INVERSE VANDERMOND MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
1.0155	0.0000	0.0566	0.0000	0.0355	0.0000
-0.0078	-0.1181	-0.0283	-0.0969	-0.0177	-0.0015
-0.0078	0.1181	-0.0283	0.0969	-0.0177	0.0015

PHASE CANONICAL INPUT COEFF. MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
3.1250	0.0000	-0.0719	-0.0000	0.0017	0.0000

OUTPUT COEFFICIENT MATRIX(NORMAL FORM)

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
3.1695	0.0000	-0.0222	-0.3620	-0.0222	0.3620

COEFFICIENT A1= 0.0663 -0.0142

COEFFICIENT A1*= 0.0663 0.0142

ARGPHI= 0.0000 4.6702

PHI= 1.3599 RADIANS

TIME TO FIRST PEAK= 0.6336 SECONDS

X1= -0.5052

STEADINESS FACTOR= 0.0508

SUMMATION OF COEFFICIENTS OF REAL MODES = 2.5509 0.0000

SUMMATION OF COEFFICIENTS OF COMPLEX MODES = 0.1325 0.0000

TOTAL SYSTEM STEADINESS FACTOR = 0.9695

RESULTING EIGENVALUES

(REAL)	(IMAGINARY)
-1.2425	0.0000
-0.7973	5.2913
-0.7973	-5.2913

RESULTING PHASE CANONICAL FORM

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
0.0000	-0.0000	1.0000	0.0000	-0.0000	0.0000
-0.0000	0.0000	-0.0000	0.0000	1.0000	0.0000
-35.5773	-0.0000	-30.6147	-0.0000	-2.8372	-0.0000

*****MODIFICATION OF SYSTEM A MATRIX*****

PSI= 19.7041

PSI= 9.6891

PSI= 5.0931

PSI= 13.7578

GRADIENT SEARCH COMMENCING

PSI= 5.0931

PSI= 4.8887

PSI= 5.0011

GRADIENT SEARCH COMMENCING

PSI= 4.8887

PSI= 4.7719

PSI= 4.6579

PSI= 4.4377

PSI= 4.0298

PSI= 3.3441

PSI= 2.4999

PSI= 2.9811

GRADIENT SEARCH COMMENCING

PSI= 2.4999

PSI= 2.3235

PSI= 2.4948

GRADIENT SEARCH COMMENCING

PSI= 2.3235

PSI= 2.2931

PSI= 2.2728

PSI= 2.2623

PSI= 2.3625

GRADIENT SEARCH COMMENCING

PSI= 2.2623

PSI= 2.1982

PSI= 2.4657

GRADIENT SEARCH COMMENCING

PSI= 2.1982

PSI= 2.1551

PSI= 2.1361

PSI= 2.1705

GRADIENT SEARCH COMMENCING

PSI= 2.1361

PSI= 2.1125

PSI= 2.1086

PSI= 2.1603

GRADIENT SEARCH COMMENCING

PSI= 2.1086

PSI= 2.0722

PSI= 2.0704

INPUT MATRIX

ROW 1
-0.6904737D-01 -0.6569923D 04 0.

ROW 2
0. 0. 0.1000000D-01

ROW 3
0.5135435D 00 -0.2900392D 04 -0.2672995D 01

	EIGENVALUES(REAL PART)	EIGENVALUES(IMAGINARY PART)
1	-0.716708529D 00	0.517677602D 01
2	-0.130862484D 01	0.
3	-0.716708529D 00	-0.517677602D 01

FIRST EIGENVECTOR		SECOND EIGENVECTOR		THIRD EIGENVECTOR		FOURTH EIGENVECTOR
0.8921D 00	-0.2393D 00	-0.9997D 00	-0.	0.8921D 00	0.2393D 00	
-0.1006D-03	-0.7265D-03	-0.1886D-03	-0.	-0.1006D-03	0.7265D-03	
0.3833D 00	0.	0.2468D-01	0.	0.3833D 00	0.	

OUTPUT COEF. MATRIX OF ORIG. SYSTEM

1.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

INPUT COEF. MATRIX OF ORIG. SYSTEM

3.1250	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

CALCULATION NO. 2 **********

VANDERMOND MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
-0.7167	5.1768	-1.3086	0.0000	-0.7167	-5.1768
-26.2853	-7.4205	1.7125	-0.0000	-26.2853	7.4205

INVERSE VANDERMOND MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
-0.0030	-0.1267	-0.0264	-0.0996	-0.0184	-0.0021
1.0060	0.0000	0.0528	0.0000	0.0368	0.0000
-0.0030	0.1267	-0.0264	0.0996	-0.0184	0.0021

PHASE CANONICAL OUTPUT COEFF. MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
1.0000	0.0000	-0.0000	0.0000	0.0000	0.0000

PHASE CANONICAL INPUT COEFF. MATRIX

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
3.1250	0.0000	-0.2158	0.0000	0.0149	-0.0000

OUTPUT COEFFICIENT MATRIX(NORMAL FORM)

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
-0.0040	-0.3746	3.1330	0.0000	-0.0040	0.3746

COEFFICIENT A1= 0.0709 -0.0106

COEFFICIENT A1*= 0.0709 0.0106

ARGPHI= -0.0000 6.6987

PHI= 1.4226 RADIANS

TIME TO FIRST PEAK= 0.6355 SECONDS

X1= -0.4555

STEADINESS FACTOR= 0.0509

SUMMATION OF COEFFICIENTS OF REAL MODES = 2.3941 0.0000

SUMMATION OF COEFFICIENTS OF COMPLEX MODES = 0.1418 0.0000

TOTAL SYSTEM STEADINESS FACTOR = 0.9641

RESULTING EIGENVALUES

(REAL)	(IMAGINARY)
-0.7167	5.1768
-1.3086	0.0000
-0.7167	-5.1768

RESULTING PHASE CANONICAL FORM

(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)	(REAL)	(IMAGINARY)
-0.0000	-0.0000	1.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
-35.7421	0.0000	-29.1885	-0.0000	-2.7420	-0.0000

*****ORIGINAL SYSTEM******

*****SOLUTION OF MATRIX DIFFERENTIAL EQUATIONS*****

INITIAL CONDITIONS

-0.00000	-0.00000	-0.00000

STATE COEFFICIENT MATRIX

-0.02300	-6570.00000	0.00000
0.00000	0.00000	0.01000
0.59000	-2880.00000	-1.50000

INPUT COEFFICIENT MATRIX

3.12500	0.00000	0.00000
-0.00000	0.00000	0.00000
-0.00000	0.00000	0.00000

INPUT MATRIX

0.43806	0.00000	0.00000

TABULATED SOLUTION OF DIFFERENTIAL EQUATIONS

FIRST VARIABLE	SECOND VARIABLE	THIRD VARIABLE	FOURTH VARIABLE	FIFTH VARIABLE	SIXTH VARIABLE
0.00548	0.00000	0.00001	0.00400	0.00100	
0.34241	0.00002	0.01986	0.25700	0.06400	
0.63592	0.00013	0.03807	0.57700	0.12800	
0.68544	0.00017	0.02749	0.70500	0.12800	
0.70589	0.00019	0.00959	0.83300	0.12800	
0.71568	0.00019	-0.00839	0.96100	0.12800	
0.73294	0.00017	-0.01929	1.08900	0.12800	
0.76911	0.00015	-0.01885	1.21700	0.12800	
0.82399	0.00013	-0.00782	1.34500	0.12800	
0.88673	0.00013	0.00854	1.47300	0.12800	
0.94089	0.00015	0.02268	1.60100	0.12800	
0.97212	0.00019	0.02816	1.72900	0.12800	
0.97472	0.00022	0.02241	1.85700	0.12800	
0.95429	0.00024	0.00783	1.98500	0.12800	
0.92546	0.00024	-0.00928	2.11300	0.12800	
0.90554	0.00022	-0.02152	2.24100	0.12800	
0.90709	0.00019	-0.02368	2.36900	0.12800	
0.93253	0.00016	-0.01496	2.49700	0.12800	
0.97329	0.00015	0.00067	2.62500	0.12800	
1.01365	0.00016	0.01624	2.75300	0.12800	
1.03784	0.00019	0.02487	2.88100	0.12800	
1.03695	0.00022	0.02277	3.00900	0.12800	
1.01288	0.00024	0.01090	3.13700	0.12800	
0.97747	0.00025	-0.00551	3.26500	0.12800	
0.94726	0.00023	-0.01926	3.39300	0.12800	
0.93626	0.00020	-0.02436	3.52100	0.12800	
0.94985	0.00017	-0.01864	3.64900	0.12800	
0.98245	0.00016	-0.00470	3.77700	0.12800	
1.02012	0.00016	0.01128	3.90500	0.12800	
1.04669	0.00019	0.02227	4.03300	0.12800	
1.05091	0.00022	0.02346	4.16100	0.12800	
1.03134	0.00024	0.01440	4.28900	0.12800	
0.99695	0.00025	-0.00089	4.41700	0.12800	
0.96312	0.00024	-0.01566	4.54500	0.12800	
0.94484	0.00021	-0.02344	4.67300	0.12800	
0.95017	0.00018	-0.02088	4.80100	0.128C0	
0.97673	0.00016	-0.00915	4.92900	0.12800	
1.01283	0.00016	0.00653	5.05700	0.12800	
1.04266	0.00018	0.01925	5.18500	0.12800	
1.05323	0.00021	0.02346	5.31300	0.12800	
1.04010	0.00023	0.01736	5.44100	0.12800	
1.00922	0.00025	0.00369	5.56900	0.12800	
0.97428	0.00024	-0.01150	5.6970C	0.12800	
0.95068	0.00022	-0.02155	5.82500	0.12800	
0.94872	0.00019	-0.02208	5.95300	0.12800	
0.96914	0.00017	-0.01291	6.08100	0.12800	
1.00287	0.00016	0.00186	6.20900	0.12800	
1.03503	0.00017	0.01573	6.33700	0.12800	
1.05156	0.00020	0.02260	6.46500	0.12800	
1.04530	0.00023	0.01953	6.59300	0.12800	
1.01916	0.00024	0.00790	6.72100	0.12800	
0.98474	0.00025	-0.00711	6.84900	0.12800	
0.95721	0.00023	-0.01891	6.97700	0.12800	
0.94859	0.00020	-0.02234	7.10500	0.12800	
0.96255	0.00018	-0.01594	7.23300	0.12800	
0.99283	0.00016	-0.00260	7.36100	0.12800	
1.02604	0.00017	0.01180	7.48900	0.12800	
1.04760	0.00019	0.02093	7.61700	0.12800	

*****MODIFIED SYSTEM*****

*****SOLUTION OF MATRIX DIFFERENTIAL EQUATIONS*****

INITIAL CONDITIONS

-0.00000	-0.00000	-0.00000

STATE COEFFICIENT MATRIX

-0.06905	-6569.92279	0.00000
0.00000	0.00000	0.01000
0.51354	-2900.39233	-2.67299

INPUT COEFFICIENT MATRIX

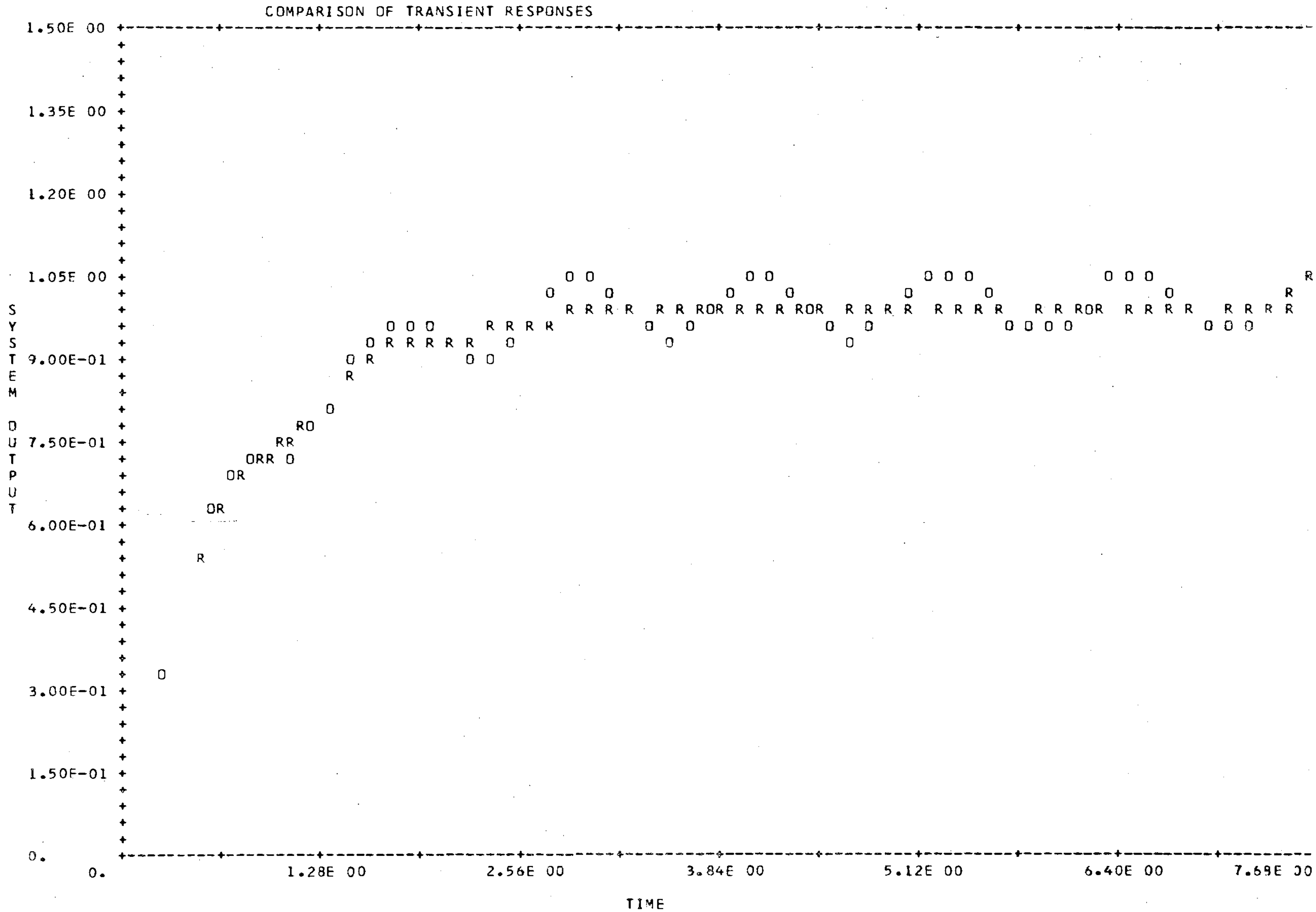
3.12500	0.00000	0.00000
-0.00000	0.00000	0.00000
-0.00000	0.00000	0.00000

INPUT MATRIX

0.39434	0.00000	0.00000

TABULATED SOLUTION OF DIFFERENTIAL EQUATIONS

FIRST VARIABLE	SECOND VARIABLE	THIRD VARIABLE	FOURTH VARIABLE	FIFTH VARIABLE	SIXTH VARIABLE
0.00493	0.00000	0.00001	0.00400	0.00100	
0.54938	0.00007	0.02721	0.51300	0.12800	
0.62850	0.00010	0.02463	0.64100	0.12800	
0.68102	0.00013	0.01671	0.76900	0.12800	
0.71515	0.00015	0.00677	0.89700	0.12800	
0.72839	0.00015	0.00228	0.96100	0.06400	
0.74138	0.00015	-0.00129	1.02500	0.06400	
0.75462	0.00015	-0.00376	1.08900	0.06400	
0.76881	0.00014	-0.00503	1.15300	0.06400	
0.85506	0.00014	0.00237	1.47300	0.12800	
0.88893	0.00014	0.00684	1.60100	0.12800	
0.91556	0.00015	0.00906	1.72900	0.12800	
0.93274	0.00016	0.00836	1.85700	0.12800	
0.94102	0.00017	0.00527	1.98500	0.12800	
0.94357	0.00018	0.00126	2.11300	0.12800	
0.94448	0.00018	-0.00207	2.24100	0.12800	
0.94728	0.00017	-0.00360	2.36900	0.12800	
0.95363	0.00017	-0.00308	2.49700	0.12800	
0.96309	0.00016	-0.00107	2.62500	0.12800	
0.97367	0.00017	0.00136	2.75300	0.12800	
0.98289	0.00017	0.00314	2.88100	0.12800	
0.98889	0.00017	0.00363	3.00900	0.12800	
0.99111	0.00018	0.00280	3.13700	0.12800	
0.99036	0.00018	0.00114	3.26500	0.12800	
0.98744	0.00018	-0.00122	3.45700	0.12800	
0.98678	0.00018	-0.00187	3.58500	0.12800	
0.98801	0.00017	-0.00160	3.71300	0.12800	
0.99088	0.00017	-0.00067	3.84100	0.12800	
0.99448	0.00017	0.00045	3.96900	0.12800	
0.99764	0.00017	0.00127	4.09700	0.12800	
0.99951	0.00018	0.00152	4.22500	0.12800	
0.99982	0.00018	0.00116	4.35300	0.12800	
0.99891	0.00018	0.00044	4.48100	0.12800	
0.99689	0.00018	-0.00061	4.67300	0.12800	
0.99620	0.00018	-0.00089	4.80100	0.12800	
0.99642	0.00018	-0.00076	4.92900	0.12800	
0.99742	0.00018	-0.00034	5.05700	0.12800	
0.99945	0.00018	0.00037	5.24900	0.12800	
1.00040	0.00018	0.00061	5.37700	0.12800	
1.00072	0.00018	0.00059	5.50500	0.12800	
1.00043	0.00018	0.00034	5.63300	0.12800	
0.99913	0.00018	-0.00027	5.88900	0.12800	
0.99876	0.00018	-0.00039	6.01700	0.12800	
0.99880	0.00018	-0.00033	6.14500	0.12800	
0.99918	0.00018	-0.00015	6.27300	0.12800	
0.99999	0.00018	0.00015	6.46500	0.12800	
1.00036	0.00018	0.00026	6.59300	0.12800	
1.00047	0.00018	0.00025	6.72100	0.12800	
1.00032	0.00018	0.00014	6.84900	0.12800	
0.99972	0.00018	-0.00012	7.10500	0.12800	
0.99955	0.00018	-0.00017	7.23300	0.12800	
0.99955	0.00018	-0.00014	7.36100	0.12800	
0.99971	0.00018	-0.00007	7.48900	0.12800	
1.00004	0.00018	0.00007	7.68100	0.12800	


COMPARISON OF TRANSIENT RESPONSES
SYSTEM OUTPUT
TIME
1.50E 00
1.35E 00
1.20E 00
1.05E 00
9.00E-01
7.50E-01
6.00E-01
4.50E-01
3.00E-01
1.50E-01
0.
0.
1.28E 00
2.56E 00
3.84E 00
5.12E 00
6.40E 00
7.68E 00

VITA

Jerry Graham Mrazek

Candidate for the Degree of

Doctor of Philosophy

Thesis: TIME DOMAIN COMPENSATION OF LINEAR SYSTEMS

Major Field: Engineering

Biographical:

Personal Data: Born May 10, 1933, in Ponca City, Oklahoma, the son of Oldrich and Marvel Mrazek.

Education: Graduated from West High School, Denver, Colorado, in June, 1952; received the degree of Bachelor of Science in Aeronautical Engineering from Oklahoma University, June, 1956; received the degree of Master of Science in Aeronautical Engineering from Wichita State University, June, 1964; completed the requirements for the Degree of Doctor of Philosophy, July, 1967.

Professional Experience: Employed by General Dynamics Corporation, Fort Worth division from June, 1956, to April, 1957, as an associate aerophysics engineer; from April, 1957 to April, 1959, in the United States Air Force, as a maintenance officer; from April, 1959, to December, 1959, by General Dynamics, Fort Worth Division, as an aerophysics engineer; from December, 1959, to April, 1965, by Beech Aircraft Corporation, as an aerodynamics engineer, Chief of Aerotechnology; from April, 1965, to September, 1965, as a technical consultant to Ryan Aeronautical Corporation; from September, 1965, to September, 1967, at Oklahoma State University, as a graduate research assistant.

Professional Organizations: Member of Sigma Gamma Tau, and Pi Mu Epsilon.