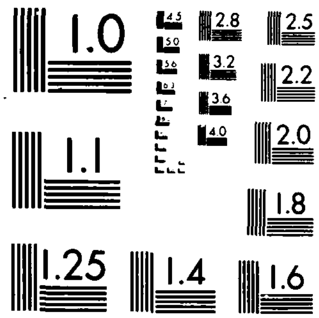
AD-A173 956 FEEDBACK CONTROL ANALYSIS USING PARAMETER PLANE TECHNIQUES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA D M POTTER JUN 86 1/2

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

FEEDBACK CONTROL ANALYSIS USING
PARAMETER PLANE TECHNIQUES

by

DANIEL MICHAEL POTTER

June 1986

Thesis Advisor: George J. Thaler

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS AI73956
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 32	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943 - 5100	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943 - 5100
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO / TASK NO. / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)

FEEDBACK CONTROL ANALYSIS USING PARAMETER PLANE TECHNIQUES

12 PERSONAL AUTHOR(S) Potter, Daniel M.

13a TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1986 June	117

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Parameter Plane, Feedback Compensation, Dominancy of Roots, Subroutine LPARAM

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The immediate attention of the control systems engineer is directed to the dynamic behavior of the system under study. It is important to study the effects on overall system performance of varying one or more parameters (mass, inertia, gain, resistance, etc.). It is equally important to determine whether a desired dynamic behavior can be achieved with any set of values for the parameters--if not, redesign is indicated.

In this thesis a control systems analysis package is developed using parameter plane methods. It is an interactive, user-friendly computer aid. Given a characteristic equation containing two variable parameters, the output of the analysis may be either tabular or graphical, with plots of any of the following types:

1. Constant damping curves as a function of frequency,
2. Constant frequency curves as a function of damping,
3. Constant sigma lines (real root lines),
4. Constant zeta-omega (damping-frequency) curves.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL George J. Thaler	22b TELEPHONE (Include Area Code) (408) 242-2134 — 22c OFFICE SYMBOL 86Tr

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited

Feedback Control Analysis Using
Parameter Plane Techniques

by

Daniel M. Potter
Lieutenant, United States Navy
B.S., Syracuse University, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1986

Author: Daniel M. Potter

Approved by: G.J. Thaler, Thesis Advisor

H. Titus, Second Reader

H. Rigas, Chairman, Department of Electrical
and Computer Engineering

J.N. Dyer, Dean of Science and Engineering

ABSTRACT

The immediate attention of the control systems engineer is directed to the dynamic behavior of the system under study. It is important to study the effects on overall system performance ~~of~~ by varying one or more parameters (mass, inertia, gain, resistance, etc.). It is equally important to determine whether a desired dynamic behavior can be achieved with any set of values for the parameters--if not, redesign is indicated.

In this thesis a control systems analysis package is developed using parameter plane methods. It is an interactive, user-friendly computer aid. Given a characteristic equation containing two variable parameters, the output of the analysis may be either tabular or graphical, with plots of any of the following types:

1) Constant damping curves as a function of frequency,
2) Constant frequency curves as a function of damping,
3) Constant sigma lines (real root lines),
4) Constant zeta-omega (damping-frequency) curves.

TABLE OF CONTENTS

I. INTRODUCTION

The analysis and synthesis of linear feedback control systems, or the compensation of same, can be realized by three general methods. The first of these can be called the integral method. Given a control system, described by a set of differential equations, one selects a cost function to be minimized with respect to certain variable system parameters. The major drawback with this method is the difficulty of varying more than one parameter at a time. The second method is the Bode frequency response technique whereby the system's open loop transfer function is manipulated to obtain the desired system response. This method also has its inherent weaknesses: difficulty of application to non-unity feedback control systems, difficulty in interpreting the closed loop transient response in terms of the open loop frequency response, and difficulty of varying more than one parameter. Third are the algebraic methods. Within this category can be included the familiar root locus method. Here, a graphical technique is provided by which the set of all points which could potentially be made roots are plotted in the S-plane. The root locus method is a valuable and powerful tool when only one parameter is varied; results are less satisfactory for two parameters and of little use when three or more parameters are involved.

Methods for studying the parameter-root relationship when two or more parameters are variable are clearly of considerable value. For a linear system, the set of differential equations that describe that system can be transformed into algebraic equations and manipulated to provide a characteristic polynomial. Since the coefficients of the characteristic polynomial are deterimined by the system parameters, it follows that some relationship exists between the value of any parameter and the value of the characteristic roots. In reference (1), Mitrovic developed an algebraic/graphical method for obtaining the roots of a polynomial in terms of two variable parameters. In references (2), (3), and (4), Choe, Hyon, and Nutting, respectively developed and extended the Mitrovic method to the compensation of linear continuous feedback control systems. The disadvantage of the Mitrovic method is that the variable parameters may appear in no more than two coefficients of the characteristic equation, which limits the flexibility of the technique. In reference (5), Siljak introduced a method for obtaining the roots of a polynomial in terms of two variable parameters that may appear in any and all of the coefficients of the polynomial. Later, Thaler and Towill [Ref. 6] extended this method to the compensation of linear continuous feedback control systems. It is from the latter work that the ensuing parameter plane equations were developed. General methods of compensation will be presented, and an attempt will be made to relate the root locus and the parameter plane methods as a set of complementary techniques which, when applied in tandem, represent the most satisfactory tool to date for designing linear feedback control systems.

The parameter plane method, which works well for two variable parameters and which may be extended to three or more parameters, is purely algebraic, and the resulting plots are valuable aids to analysis. The term parameter plane comes from the plot for two parameters--in a rectangular coordinate space one parameter will define the abscissa while the second parameter defines the ordinate (the S-plane is inconvenient for presenting the desired results). Three parameters define a 3-dimensional parameter space, etc. For design problems it is convenient to think of the algebraic calculations as a mapping procedure. By choosing a point on the S-plane, the characteristic polynomial acts as a mapping function whereby the point may be "mapped" onto the alpha-beta plane (alpha and beta are the two variable parameters to be used throughout the remainder of this text). The relationship between being able to place the roots of a polynomial at specific locations in the S-plane and the compensation of linear feedback control systems is as follows. A feedback control system, including any added compensators which may contain variables, can be reduced to a ratio of two polynomials (the closed loop transfer function). A specified system response in terms of overshoot, bandwidth, settling time, steady-state accuracy, etc., can theoretically be obtained by placing a pair of complex conjugate roots of the characteristic equation at a specific location in the S-plane, while ensuring that the real part of this complex root pair (the dominant roots) is smaller in magnitude than the real parts of the remaining roots of the characteristic equation. The problem of compensation, thus of feedback control system design, reduces to one of placing the dominant roots of the characteristic equation at the desired location. The ability of the parameter plane method to achieve this goal will become obvious.

II. DERIVATION OF PARAMETER PLANE EQUATIONS

A linear feedback control system's characteristic equation can be expressed as a polynomial of the following form:

$$f(s) = \sum_{k=0}^{m} a_k S^k = 0, \text{ where} \tag{2-1}$$

a_k (k=0,1,...,m) are real coefficients

$S = -\sigma \pm j\omega = -\xi\omega \pm j\omega\sqrt{1-\xi^2}$

ω is the undamped natural frequency and

$\xi\omega$ is the relative damping coefficient

In reference (5) it is noted that S^k may be represented by the following:

$$S^k = \omega^k(T_k(-\xi) \pm j\sqrt{1-\xi^2}\, U_k(-\xi)) \tag{2-2}$$

where

$T_k(-\xi)=(-1)^k T_k(\xi)$ and $U_k(-\xi)=(-1)^{k+1}U_k(\xi)$.

$T_k(\xi)$ and $U_k(\xi)$ are Chebishev functions of the first and second kind respectively. Values of zeta and omega will be considered such that $0 \le \xi \le 1$ and $0 \le \omega \le \infty$. Values of T_k and U_k are tabulated in various appendixes. More important to digital computer analysis, they can be obtained from the following recursive relations:

$$T_{k+1}(\xi) - 2T_k(\xi) + T_{k-1}(\xi) = 0$$

$$U_{k+1}(\xi) - 2U_k(\xi) + U_{k-1}(\xi) = 0 \qquad (2\text{-}3)$$

Here, $T_0(\xi)=1$, $T_1(\xi)=\xi$, $U_0(\xi)=0$, $U_1(\xi)=1$. Substituting equation (2-2) into (2-1) and setting the real and imaginary parts to zero independently, one obtains:

$$\sum_{k=0}^{m} a_k\omega^k T_k(-\xi)=0$$

$$\sum_{k=0}^{m} a_k\omega^k U_k(-\xi)=0 \qquad (2\text{-}4)$$

Employing equations (2-3), one obtains from equation (2-4):

$$\sum_{k=0}^{m} (-1)^k a_k\omega^k U_{k-1}(\xi) = 0$$

$$\sum_{k=0}^{m} (-1)^k a_k\omega^k U_k(\xi) = 0 \qquad (2\text{-}5)$$

Now consider the coefficients a_k of the characteristic equation (2-1) as linear functions of the variable system parameters, α and β, as follows:

$$a_k = b_k\alpha + c_k\beta + d_k \qquad (2\text{-}6)$$

Using this relation for a_k, equations (2-5) become:

$$\alpha B_1 + \beta C_1 + D_1 = 0 \qquad (2\text{-}7)$$

$$\alpha B_2 + \beta C_2 + D_2 = 0$$

where

$$B_1 = \sum_{k=0}^{m} (-1)^k b_k \omega^k U_{k-1} \qquad B_2 = \sum_{k=0}^{m} (-1)^k b_k \omega^k U_k$$

$$C_1 = \sum_{k=0}^{m} (-1)^k c_k \omega^k U_{k-1} \qquad C_2 = \sum_{k=0}^{m} (-1)^k c_k \omega^k U_k \qquad (2\text{-}8)$$

$$D_1 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_{k-1} \qquad D_2 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_k$$

Since equations (2-7) are linear in the two unknowns alpha and beta, Cramer's rule may be applied to obtain:

$$\alpha = \frac{C_1 D_2 - C_2 D_1}{B_1 C_2 - B_2 C_1} \qquad \beta = \frac{B_2 D_1 - B_1 D_2}{B_1 C_2 - B_2 C_1} \qquad (2\text{-}9)$$

Equations (2-9) are now functions of zeta and omega. Hence, by fixing either zeta or omega and varying the remaining parameter, the constant omega or constant zeta S-plane contours respectively can be mapped into the real domain of the alpha-beta or parameter plane.

In reference (5) the following relationships are noted:

$$S^k = P_k + j\omega\sqrt{1-\xi^2}\; Q_k$$

$$P_{k+1} + 2\omega P_k + \omega^2 P_{k-1} = 0$$

$$Q_{k+1} + 2\omega Q_k + \omega^2 Q_{k-1} = 0 \qquad (2\text{-}10)$$

$$P_k = -\omega\xi Q_k - \omega^2 Q_{k-1}$$

$$P_o = Q_o = 0,\; P_1 = -\xi\omega,\; Q_1 = 1$$

P_k and Q_k are related to the Chebishev functions by:

$$P_k = \omega^k T_k(-\xi) = (-1)^k \omega^k T_k(\xi)$$

$$(2\text{-}11)$$

$$Q_k = \omega^{k-1} U_k(-\xi) = (-1)^{k+1} \omega^{k-1} U_k(\xi)$$

By employing equations (2-10) and (2-11), one obtains (proceeding as before);

$$\sum_{k=0}^{m} a_k Q_{k-1} = 0 \qquad \sum_{k=0}^{m} a_k Q_k = 0 \tag{2-12}$$

Combining equations (2-6) and (2-12) with Cramer's rule, one again arrives at equations (2-9) where the following expressions now apply:

$$B_1 = \sum_{k=0}^{m} b_k Q_{k-1} = 0 \qquad B_2 = \sum_{k=0}^{m} b_k Q_k = 0$$

$$C_1 = \sum_{k=0}^{m} c_k Q_{k-1} = 0 \qquad C_2 = \sum_{k=0}^{m} c_k Q_k = 0 \tag{2-13}$$

$$D_1 = \sum_{k=0}^{m} d_k Q_{k-1} = 0 \qquad D_2 = \sum_{k=0}^{m} d_k Q_k = 0$$

Equations (2-9) and (2-13) are useful for mapping constant zeta-omega curves from the S-plane into the parameter plane. As will be demonstrated later, these curves play an important role in dominance considerations.

If the complex variable S is substituted in equation (2-1) by letting $S = -\sigma$, where sigma corresponds to values of S along the real axis, then according to equation (2-6) the characteristic equation (2-1) becomes:

$$\alpha \sum_{k=0}^{m} (-1)^k b_k \sigma^k + \beta \sum_{k=0}^{m} (-1)^k c_k \sigma^k + \sum_{k=0}^{m} (-1)^k d_k \sigma^k = 0 \qquad (2\text{-}14)$$

The above expression represents a straight line in the alpha-beta plane for a given value of sigma. Hence a point on the real axis in the S-plane maps into a straight line in the alpha-beta plane. In addition, for given values of alpha, beta, and sigma which satisfy equation (2-14), the characteristic equation (2-1) must have a real root at minus sigma. On the constant zeta and omega curves previously defined, for certain values of alpha and beta (say, for values obtained from equations (2-9) for given values of zeta and omega) the characteristic equation will have a pair of complex conjugate roots at $S = -\xi\omega \pm j\omega\sqrt{1-\xi^2}$.

The significance of the above discussion is that by applying equations (2-9) and (2-14) one can, for a specified value of zeta, omega, and sigma, compute the value of alpha and beta such that the characteristic equation will have a pair of roots at $S = -\xi_1\omega_1 \pm j\omega_1\sqrt{1-\xi_1^2}$. The m-2 remaining roots of the characteristic equation can then be calculated by dividing out the two known or specified roots. This method, where zeta, omega and sigma, or simply zeta and omega are specified, and where the computations for alpha and beta are done algebraically, will be referred to as the algebraic parameter plane solution.

To solve the problem in general for all values of zeta, omega, and sigma, it becomes necessary to plot a family of parameter plane curves for various values of zeta, omega, sigma, and if desired, zeta-omega. On the resulting parameter plane plot one can, by choosing an operating point, graphically read from the curves the values of alpha and beta and, hence, the values of the m roots of the corresponding m^{th} order characteristic equation. This latter method will be referred to as the graphical parameter plane solution.

The algebraic solution has the advantage that the labor of plotting the curves can be avoided, but the disadvantage remains that without the curves it is difficult to pick the optimum values of zeta and omega so as to ensure dominance while still meeting the system specifications. The graphical solution has the advantage that one has a "picture" of the way the characteristic roots move about in the S-plane as alpha and beta are varied. This enables one to choose the values of alpha and beta corresponding to the best values of zeta, omega, sigma, and zeta-omega for all roots of the characteristic equation. This feature of the parameter plane points out a strong justification for attempting to obtain the parameter plane curves. And with the employment of a digital computer and an appropriate algorithm to realize the parameter plane curves, the advantage of the algebraic method becomes muted.

Recursion methods (equation (2-3)) are by no means the only methods of producing algebraic and graphical parameter plane data. Thaler and Karmarkar [Ref. 7] describe a matrix solution to the parameter plane problem. Essentially, a matrix of coefficients may be manipulated to obtain the following general form:

$$\begin{bmatrix} b_1 & c_1 & -\omega^2 & 0 & 0 & \cdot\cdot & 0 & d_1 & e_1 \\ b_2 & c_2 & -2\xi\omega & -\omega^2 & 0 & \cdot\cdot & 0 & d_2 & e_2 \\ \cdot & \cdot & -1 & -2\xi\omega & -\omega^2 & \cdot\cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & -1 & -2\xi\omega & \cdot\cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\cdot & -\omega^2 & d_{m-2} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\cdot & -2\xi\omega & d_{m-1}-\omega^2 & \cdot \\ b_m & c_m & 0 & 0 & 0 & \cdot\cdot & -1 & d_m-2\xi\omega & e_m \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ {}^{m-2}R_{m-2} \\ \cdot \\ {}^{m-2}R_1 \\ 1 \\ \alpha\beta \end{bmatrix} = 0$$

where b_k, c_k, d_k, ω, and $\xi\omega$ are as described before, and:

e_k (k=1, ..., m) are the coefficients associated with the non-linear alpha-beta product terms

$\overset{m-2}{R_1}$ is the sum of the m-2 roots of the polynomial characteristic equation taken one at a time

.
.
.

$\overset{m-2}{R_{m-2}}$ is the sum of the m-2 roots taken m-2 at a time

Further, by the application of appropriate row operations, this matrix may be reduced to the following row-echelon form:

$$\begin{bmatrix} 1 & 0 & 0 & . & . & 0 & k_{11} & k_{12} \\ 0 & 1 & 0 & . & . & 0 & k_{21} & k_{22} \\ 0 & 0 & 1 & . & . & 0 & k_{31} & k_{32} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & 1 & k_{m1} & k_{m2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \overset{m-2}{R_{m-2}} \\ . \\ . \\ 1 \\ \alpha\beta \end{bmatrix} = 0 \qquad (2\text{-}15)$$

For the case when all coefficients of the characteristic equation are linear, i.e. $e_k(k=1,\ldots,m) = 0$, then K_{k2} $(k=1,\ldots,m) = 0$, and

$$\alpha = -K_{11}$$

$$\beta = -K_{21}$$

as obtained from the first two rows of the matrix equation. One should note that in arriving at equations (2-15), approximately m^2 row operations are required for the row-echelon matrix formulation for <u>each</u> point of the parameter plane curves (e.g., each time either zeta or sigma are varied). Compare this with the approximate m calculations required to obtain the recursion equations of the previous chapter, and the matrix method becomes relatively inefficient for larger order systems.

One should not, however, discard the matrix approach entirely. For small order characteristic equations, this technique compares favorably with the recursion method. And when the variable parameters are non-linear--when one must deal with alpha-beta product terms--the matrix approach affords a more direct method of obtaining the alpha-beta pairs.[1] Whether the recursion or matrix method is utilized

---

[1] From equations (2-15), one obtains the two quadratic forms

$$K_{22}\alpha^2-(K_{21}\ K_{12}-K_{11}\cdot K_{22}-1)\alpha+K_{11}=0$$
$$K_{12}\beta^2+(K_{21}\ K_{12}-K_{11}\ K_{22}+1)\beta+K_{21}=0$$

from which alpha and beta are easily derived.

should depend on the inclusion of alpha-beta product terms; ultimately, it is a matter of personal preference.

III. APPLICATION OF THE PARAMETER PLANE METHOD

A. ALGEBRAIC SOLUTION

In this section it will be assumed that the system performance specifications can be met by placing a pair of complex conjugate roots at a specific location (i.e., by choosing appropriate values for zeta and omega). If after computation of the necessary values of alpha and beta to locate the roots as desired it is found that these specified roots are not dominant, then either a different value of zeta and/or omega must be used (possibly at the sacrifice of some performance measure), or a different method of compensation will have to be attempted. In a later section a method will be addressed whereby the dominancy requirement may be achieved.

1. Feedback Compensation

For a unity feedback control system, let

$$G = \frac{K}{e(S)} = \frac{K}{S^m + e_{m-1}S^{m-1} + \ldots + e_L S^L} \tag{3-1}$$

where K is the forward path gain (a variable) and e(S) is a polynomial in S representing the poles of the open loop transfer function of the uncompensated system. In equation

(3-1), L corresponds to the system type--for a type 0 system, L=0, for type 1, L=1, etc. The system's error coefficient is defined as:

$$K_e = \lim_{S \to 0} S^L G_{cc} \tag{3-2}$$

where G_{cc} is the open loop transfer function of the compensated system.

a. Tachometer Plus Acceleration Feedback

In order to achieve the system performance specifications, a feedback compensator must be introduced. Let

$$H = K_t S + K_a S^2$$

The resulting compensated system's characteristic equation becomes:

$$e(S) + K(K_t S + K_a S^2) = 0 \tag{3-3}$$

and by expanding e(S), equation (3-3) becomes:

$$S^m + e_{m-1} S^{m-1} + \ldots + (e_2 + KK_a) S^2 + (e_1 + KK_t) S + e_o + K = 0 \tag{3-4}$$

where L is zero for a type 0 system (the most general case). The following results also apply to a type 1 system if e_o is set to zero, and, similarly, for a type 2 system if both e_o and e_1 are set to zero, etc. Combining equations (3-2), (3-3), and (3-4) the error coefficient becomes:

$$K_e = \lim_{S\to 0} \frac{S^0 K}{e(S)+K(K_t S+K_a S^2)} = \frac{K}{e_o+K} \qquad (3\text{-}5)$$

or for a type 1 uncompensated system:

$$K_e = \frac{K}{e_1+KK_t} \qquad (3\text{-}6)$$

or for a type 2 uncompensated system:

$$K_e = \frac{K}{KK_t} \qquad (3\text{-}7)$$

Note that if the uncompensated system is type 2, the compensated system would be type 1 if tachometer feedback or tachometer plus acceleration feedback is used.

In the compensated system's characteristic equation (3-4) let alpha = KK_a and beta = KK_t. Equation (3-4) then becomes:

$$S^m + e_{m-1}S^{m-1} + \ldots + (e_2+\alpha)S^2 + (e_1+\beta)S + e_o + K = 0$$

Recalling equation (2-6) where in general the coefficients f the characteristic equation are of the form:

$$a_k = b_k\alpha + c_k\beta + d_k$$

and letting m=k, then from equations (2-8) one obtains:

$$B_1 = (-1)\omega^2 U_1 = \omega^2 \qquad B_2 = \omega^2 U_2$$

$$C_1 = -\omega U_o = 0 \qquad C_2 = -\omega U_1 = -\omega \tag{3-8}$$

$$D_1 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_{k-1} \qquad D_2 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_k$$

since $U_o=0$ and $U_1=1$. From equations (2-9) one derives:

$$\alpha = \frac{C_1D_2-C_2D_1}{B_1C_2-B_2C_1} = \frac{\sum_{k=0}^{m} (-1)^k d_k U_{k-1}}{-\omega^3} = -\sum_{k=0}^{m} (-1)^k d_k \omega^{k-2} U_{k-1}$$

$$(3-9)$$

$$\beta = \sum_{k=0}^{m} (-1)^k d_k \omega^{k-1} (U_k - U_2 U_{k-1})$$

If alpha and beta are linear functions of K, the forward path gain, one can use the steady-state error specification to define K in terms of alpha and/or beta. Since zeta and omega were assumed to be specified, then from equations (3-9) one may solve for alpha and beta. From this, K_o and K_t are readily determined.

Example 3-1

The system of Figure (3-1) is to be compensated by using tachometer plus acceleration feedback. The system specifications are as follows:

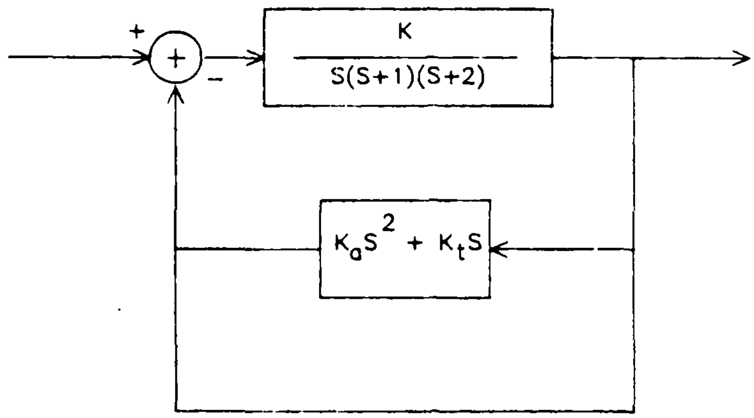


Figure 3-1

1. Complex roots corresponding to $\xi=0.7$ and $\omega=10$.
2. $K_e \geq 6$

From equation (3-2):

$$K_e = \frac{K}{2+KK_t} \geq 6$$

From which $K \geq 12+6KK_t$. The compensated characteristic equation is

$$S^3+S^2(3+KK_a)+S(2+KK_t)+K = 0 \qquad (3\text{-}10)$$

Letting alpha = KK_a and beta = KK_t, equation (3-10) becomes:

$$S^3+S^2(3+\alpha)+S(2+\beta)+K = 0 \qquad (3\text{-}10a)$$

From equations (3-8):

$B_1 = 100$ $\qquad$ $B_2 = 140$

$C_1 = 0$ $\qquad$ $C_2 = -10$

$D_1 = -1100-K$ $\qquad$ $D_1 = -1120$

and from equations (3-9):

$$\alpha = \frac{10(-1100-K)}{-1000}, \quad \beta = \frac{140(-1100-K)+56000}{-1000} \qquad (3\text{-}11)$$

From the steady-state accuracy specifications, then, it is necessary that $K \geq 12+6\beta$; let $K=12+6\beta$. From equations (3-11) it is found that $\beta=623$, hence $K=3750$. Therefore, $\alpha=48.5$, and since $\alpha=KK_a$ and $\beta=KK_t$:

$$K_a = \frac{48.5}{3750} = 0.0129$$

$$K_t = \frac{623}{3750} = 0.1661$$

The compensated system's characteristic equation becomes

$$S^3+51.5S^2+625S+3750=0 \qquad (3\text{-}12)$$

Now zeta=0.7 and ω=10 corresponds to $S^2+14S+100=0$. Dividing equation (3-12) by this quadratic, the remainder is S+37.5. Since zeta·omega of the desired roots = 7<<37.5, the complex roots are dominant and the problem is solved.

b. Tachometer Feedback Only

Let $H = K_t S$. The characteristic equation of the compensated system becomes:

$$S^m+e_{m-1}S^{m-1}+\ldots+e_2S^2+(e_1+\alpha)S+e_0+\beta=0$$

Proceeding as in the previous example, one obtains:

$B_1 = 0$ $\qquad$ $B_2 = -\omega$

$C_1 = -1$ $\qquad$ $C_2 = 0$

$D_1 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_{k-1}$ $\qquad$ $D_2 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_k$

and

$$\alpha = \sum_{k=0}^{m} (-1)^k d_k \omega^{k-1} U_k \; , \qquad \beta = \sum_{k=0}^{m} (-1)^k d_k \omega^{k} U_{k-1}$$

(3-13)

For a specified value of zeta and omega, alpha and beta can be obtained from equations (3-13). The error coefficient is then determined directly from equations (3-5), (3-6), or (3-7). Thus the error coefficient is fixed for a given value of zeta and omega, and if this parameter is to be met, the values of zeta and omega may require adjustment. One possible approach might be to fix zeta at some value, whereby from the given K_e and equations (3-5), (3-6), or (3-7) alpha could be computed. Equations (3-13) could then be solved for, first, omega and then beta. The calculations would prove tedious, however.

Example 3-2

The same system as used in example (3-1) will be studied here, this time with tachometer feedback alone. The same system performance specifications are to be met, namely, $K_e \geq 6$, zeta = 0.7, and omega = 10. The compensated system's characteristic equation becomes:

$$S^3+3S^2+(2+KK_t)S+K = 0 \qquad (3\text{-}14)$$

Letting $\alpha=KK_t$ and $\beta=$ K here, equation (3-14) becomes:

$$S^3+3S^2(2+\alpha)S+\beta = 0$$

From equation (3-13) it is found that:

$$\alpha = -2+30(1.4)-100(0.96) = -56$$

Since alpha is negative, it is seen that positive tachometer feedback is required. Further, it is found that the remaining root (when equation (3-14) is divided by $S^2+14S+100$) is positive; the system is unstable. Hence the desired system specifications cannot be met with tachometer feedback alone.

c. Acceleration Feedback Only

Let $H = K_a S^2$. The characteristic equation of the compensated system becomes:

$$S^m+e_{m-1}S^{m-1}+\ldots+(e_2+KK_a)S^2+e_1S+e_o+K = 0$$

Proceeding as before, where now $\alpha=KK_a$ and $\beta=K$:

$$B_1 = \omega^2U_1=\omega^2 \qquad B_2 = \omega^2U_2$$

$$C_1 = U_{-1} = -1 \qquad C_2 = 0$$

$$D_1 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_{k-1} \qquad D_2 = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_k$$

Solving for alpha and beta yields:

$$\alpha = \frac{-D_2}{\omega^2U_2} = -\frac{1}{U_2}(-1)^k d_k \omega^{k-2} U_k$$

$$\beta = \sum_{k=0}^{m} (-1)^k d_k \omega^k U_{k-1} - \frac{1}{U_2}\sum_{k=0}^{m} (-1)^k d_k \omega^k U_k \tag{3-15}$$

Calculations for alpha, beta, and K_e are performed in the same manner as with the preceding tachometer feedback example.

Example 3-3

The same system of examples (3-1) and (3-2) will now be compensated using acceleration feedback alone. As before, $K_e \geq 6$, zeta=0.7, and omega=10. Therefore $K_e=\frac{K}{2}$ and the error coefficient is unaffected by the acceleration feedback. Hence one can conveniently choose K=12 to meet the specifications. The compensated system's characteristic equation becomes:

$$S^3+(3+KK_a)S^2+2S+K = 0 \tag{3-16}$$

If K in equation (3-16) is set equal to 12 as prescribed, only one parameter remains and the parameter plane equations produce an indeterminate solution. If K is left as the variable beta, then equation (3-16) becomes (after the usual substitutions):

$$S^3+(3+\alpha)S^2+2S+\beta = 0$$

By employing equations (3-15), one obtains α=4 and β= -700. Since beta is negative, it is concluded that the desired roots (i.e., desired values of zeta and omega) cannot be realized using acceleration feedback alone, and of course neither can the desired error specification be obtained. One would therefore choose an alternate method of compensation.

If one chooses to use feedback compensation then perhaps tachometer plus acceleration feedback might be attempted first using equations (3-9) and the appropriate steady-state error specification. If the specifications cannot be met in this manner, then it follows that neither tachometer nor acceleration feedback alone will suffice. In this case either the system's specifications must be eased or another type of compensation must be utilized. If it is found that the specifications are achievable with the combined tachometer and acceleration feedback, then, if desired, equations (3-13) and (3-15) can be employed to investigate the feasibility of tachometer or acceleration feedback alone, respectively.

d. Case For Which Feedback Is Not Available Near The Forward Path Amplifier

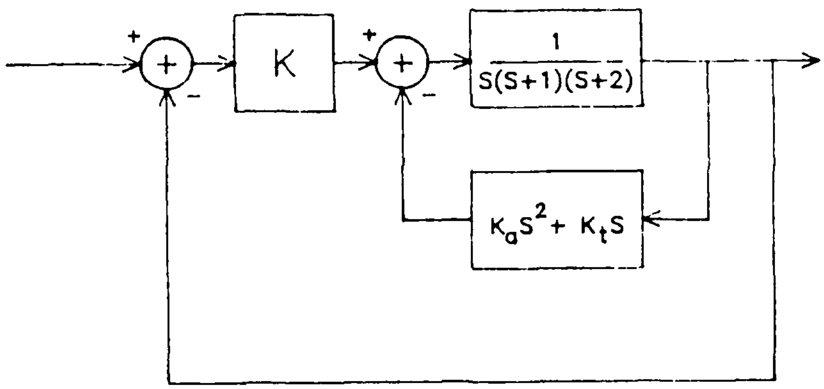


Figure 3-2

Figure 3-2 shows a system similar to that used in example (3-1) except that now the feedback is inserted at the output terminals of the amplifier represented by gain K. This illustrates a system for which it may not be possible or practical to access the input terminals of the error detector. This problem will be solved by means of an example.

Example 3-4

As before, the same system specifications are to be met, i.e., $K_e \geq 6$, zeta=0.7, and omega=10. The characteristic equation becomes:

$$S^3+(3+K_a)S^2+(2+K_t)S+K = 0$$

Letting $\alpha=K_a$ and $\beta=K_t$:

$$S^3+(3+\alpha)S^2+(2+\beta)S+K = 0 \qquad (3\text{-}17)$$

Comparison of equations (3-17) and (3-10a) show that they are identical, that is, the solution obtained for alpha and beta in example (3-1) applies. There, alpha and beta were found to be 48.5 and 623, respectively, while K=3750. For the present example no further computations are necessary to find K_a and K_t, since they are now the parameters alpha and beta. This points out an important advantage of the parameter plane method, namely, that the solutions depend only on the characteristic equation and not on the system from which the characteristic equation was formed. This principle can similarly be applied to control problems involving tachometer or acceleration feedback alone.

2. Cascade Compensation

For a unity feedback control system let G have the form of equation (3-1):

$$G = \frac{K}{e(s)} = \frac{K}{S^m + e_{m-1}S^{m-1} + \ldots + e_L S^L} \qquad (3\text{-}1)$$

where again K is the forward path gain (a variable) and e(s) is a polynomial in S representing the poles of the open loop transfer function of the uncompensated system. The letter L again indicates the system type. If in order to satisfy the system's requirements a cascade compensator G_C is required, then let:

$$G_C = \frac{P(S+Z)}{Z(S+P)}$$

With a d.c. gain of unity, placement of this compensator in the forward path will not affect steady-state accuracy. With G_C as indicated here, the values of P and Z are computed to obtain the desired system response. If P is less than Z, a lag network is required and the factor of $\frac{P}{Z}$ of the compensator is inherently present due to the physical nature of the compensator (usually an R-C network). In this case all forward path amplifier gains can remain unaltered to meet the specified accuracy demands. If, however, the computed value of P is greater than Z, a lead network is required and the compensated system's forward path gain must be raised by the factor of $\frac{P}{Z}$ to meet the accuracy specifications. As the physical nature of the lead network is such that the factor $\frac{P}{Z}$ is not inherently present, this factor must be provided either by adding an amplifier in cascade with the lead network or by raising the gain K of the existing amplifier as required to achieve steady-state accuracy.

Continuing, the compensated system's forward path transfer function is:

$$G_{CC} = G_C G = \frac{K}{e(s)} \cdot \frac{P}{Z} \cdot \frac{S+Z}{S+P} = \frac{\gamma(S+\frac{P}{\gamma})}{S+P} \cdot \frac{K}{e(s)}$$

Applying the definition of the error coefficient one obtains:

$$K_e = \lim_{s \to 0} S^{L} \left[\frac{K}{e(s)} \cdot \frac{\gamma(s+\frac{P}{\gamma})}{(S+P)}\right] = \frac{K}{e_L}$$

and again assuming a type 0 system where L=0, the compensated system's characteristic equation becomes:

$$e(s)(S+P)+K\gamma(S+\frac{P}{\gamma}) = 0$$

or after expansion:

$$S^{m+1}+(P+e_{m-1})S^{m}+(Pe_{m-1}+e_{m-2})S^{m-1}+\ldots$$

$$+(Pe_2+e_1)S^2+(K\gamma+Pe_1+e_o)S+P(e_o+K) = 0 \qquad (3\text{-}18)$$

Letting α=P and $\beta=\gamma$, equation (3-18) becomes:

$$S^{m+1}+(\alpha+e_{m-1})S^m+(e_{m-1}\alpha+e_{m-2})S^{m-1}+\ldots$$

$$+(e_2\alpha+e_1)S^2+(K\beta+e_1\alpha+e_0)S+\alpha(e_0+K) = 0 \qquad (3\text{-}19)$$

Comparison of equation (3-19) with the general form of the characteristic equation as specified in equations (2-1) and (2-6), it is apparent that K=m+1, $b_0=e_0+K$, $c_0=d_0=0$, $b_1=e_1$, $c_1=K$, $d_1=e_0$, $b_2=e_2$, $c_2=0$, $d_2=e_1$, etc.

It is important to note that the parameter plane variable beta represents the pole-to-zero ratio of the cascade compensator. The S-plane can be divided into regions where lag compensation or lead compensation is needed. By mapping of variables in the above manner, the parameter plane can effectively be divided into corresponding regions above and below the straight line β=1. Then, for values of beta less than one a lag network is required and for beta greater than one a lead network is needed. In addition, if beta is less than 0.1 or greater than 10, a multiple lag or multiple lead network, respectively, is required.

Based on equations (3-19) and (2-8) it is found that:

$$B_1 = -(e_0+K)+\omega^2 e_2+\ldots+(-1)^{k-2}\omega^{k-2}U_{k-3}+(-1)^{k-1}\omega^{k-1}U_{k-2}$$

$$C_1 = 0$$

$$D_1 = \omega^2 e_1 + \ldots + (-1)^{k-2} e_{m-2}\omega^{k-2}U_{k-3} + (-1)^{k-1} e_{m-1}\omega^{k-1}U_{k-2}$$

$$+ (-1)^{k}\omega^{k}U_{k-1}$$

$$B_2 = -\omega e_1 + \omega^2 e_2 U_2 + \ldots + (-1)^{k-2}\omega^{k-2}U_{k-2} + (-1)^{k-1}\omega^{k-1}U_{k-1}$$

$$C_2 = -\omega K$$

$$D_2 = -\omega e_o + \omega^2 e_1 U_2 + \ldots + (-1)^{k-2} e_{m-2}\omega^{k-2}U_{k-2}$$

$$+(-1)^{k-1} e_{m-1}\omega^{k-1}U_{k-1} + (-1)^{k}\omega^{k}U_k \qquad (3\text{-}20)$$

and from equations (2-9) it is found that:

$$\alpha = \frac{-D_1}{B_1} \quad , \quad \beta = \frac{B_2D_1 - B_1D_2}{C_2B_1} \qquad (3\text{-}21)$$

For a type 1 system, e_o in equations (3-20) is set equal to zero, for a type 2 system $e_o = e_1 = 0$, etc. On the basis of equations (3-20) and (3-21) a cascade compensator can be designed.

Example 3-5

For the system of Figure (3-3) it is desired to design a cascade compensator which places a pair of roots at zeta=0.5 and omega=1. The error coefficient K_e should be 50.

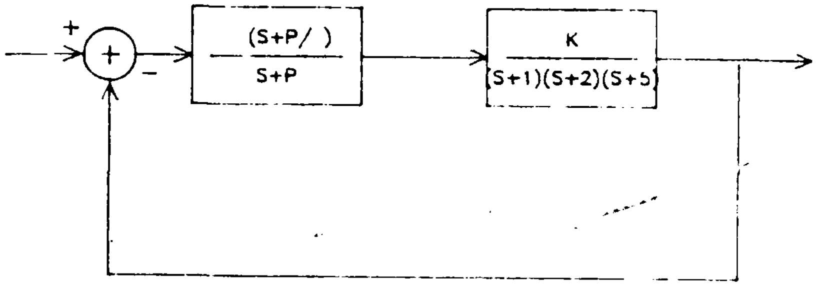


Figure 3-3

It is apparent from Figure (3-3) that $K_e = \frac{K}{10} = 50$, or $K=500$. The characteristic equation is:

$$S^4+(8+P)S^3+(17+8P)S^2+(10+17P+K\gamma)S+P(10+K)=0$$

and by the substitutions of $\alpha=P$ and $\beta=\gamma$:

$$S^4+(8+\alpha)S^3+(17+8\alpha)S^2+(10+17\alpha+500\beta)S+510\alpha = 0$$

Applying equations (3-20) and (3-21) one obtains:

$B_1 = -503$	$B_2 = -9$
$C_1 = 0$	$C_2 = -500$
$D_1 = 9$	$D_2 = 6$
$\alpha = 0.0179 = P$	$\beta = 0.0117 = \gamma$

and since $\gamma=\frac{P}{Z}$, Z=1.529. This is a lag network for which the factor 0.0117 is inherent in the R-C filter design.

Although treatment will not be presented here, the algebraic application of the parameter plane technique can be readily applied to combination cascade and feedback compensation.

B. DOMINANCY OF THE SPECIFIED ROOTS

In the preceding section nothing was done in the calculations to make the specified roots a dominant pair. As mentioned earlier, the ability to predict a system's response on the basis of the location of a pair of complex conjugate roots was based on the assumption that the magnitude of the real part of the specified roots was much less than that of the remaining roots of the characteristic equation. In practice, if the real part of the specified or primary roots is one half to one fifth or less of the real parts of all secondary roots, the system is said to be dominant in the primary roots. In many cases the system will still meet the specifications even if two pairs of complex roots have the same real part, provided the zetas for both pairs of roots meet the specifications, and the undamped natural frequencies are such that the component time responses are not highly additive. Further, even if there exists a characteristic root whose real part is closer to the origin than that of the primary pair, the presence of a closed-loop zero could make the residue of the close-in root negligible as compared to the primary root. If possible, however, one usually attempts to make the real parts of all secondary roots large in magnitude.

In the preceding examples it should be pointed out that in many cases there were actually three and possibly four variable parameters. For instance, the forward path gain was usually a fixed value in the computations so as to meet the minimum steady-state accuracy requirements. There is, however, no reason why the gain cannot be raised above the minimum value, thus permitting a third degree of freedom. When cascade and feedback compensation are used simultaneously, the forward path gain and tachometer gain become the third and fourth parameters.

Recall that the system characteristic equation has the form $f(S) = \sum_{k=0}^{m} a_k S^k = 0$, where $a_k = b_k\alpha + c_k\beta + d_k$. To realize the system specifications, one places a pair of complex roots at $S = -\xi_1\omega_1 \pm j\omega_1\sqrt{1-\xi_1^2}$, which implies that $S^2 + 2\xi_1\omega_1 S + \omega_1^2 = 0$. Since ξ_1 and ω_1 are known, the quadratic can be divided out of the characteristic equation, leaving a polynomial which contains the secondary roots of the characteristic equation. Since only two of the variable parameters were used in fixing the primary roots, the remaining parameters will appear in the coefficients of the quotient polynomial, and it is these parameters that can be varied to achieve dominance.

Instead of division to obtain the quotient polynomial, coefficients of like powers will be equated to achieve a system of equations. Let the quotient polynomial be given by:

$$f_1(s) = \sum_{k=0}^{n} f_k S^k = 0 \qquad (3\text{-}22)$$

where n=m-2, i.e., equation (3-22) is of order two less than the characteristic equation. Applying equations (2-1), (3-22), and the quadratic it follows that:

$$(S^2+2\xi_1\omega_1 S+\omega_1^2)\left(\sum_{k=0}^{n} f_k S^k\right) = \sum_{k=0}^{m} a_k S^k \qquad (3\text{-}23)$$

Taking $a_m=1$ and equating coefficients of like power:

$$\begin{aligned}
a_m &= f_n = 1 \\
a_{m-1} &= f_{n-1}+2\xi_1\omega_1 \\
a_{m-2} &= f_{n-2}+2\xi_1\omega_1 f_{n-1}+\omega_1^2 \\
&\vdots \\
a_2 &= f_o+2\xi_1\omega_1 f_1+f_2\omega_1^2 \\
a_1 &= 2\xi_1\omega_1 f_o+f_1\omega_1^2 \\
a_0 &= f_0\omega_1^2
\end{aligned} \qquad (3\text{-}24)$$

Equations (3-23) and (3-24) can be solved for the coefficients f in terms of the coefficients a. The results will be applied to the following cases:

Case of k=3, n=1

Equation (3-23) becomes:

$$(S^2+2\xi_1\omega_1 s+\omega_1^2)(f_1 S+f_0)=S^3+a_2 S^2+a_1 S+a_0$$

Equating coefficients of like power one obtains:

$$a_3 = f_1 = 1$$

$$a_2 = f_0+2\xi_1\omega_1 f_1$$

$$a_1 = f_1\omega_1^2+2\xi_1\omega_1 f_0$$

$$a_0 = f_0\omega_1^2$$

Solving for the coefficients f results in:

$$f_1 = 1$$

$$f_0 = \frac{a_0}{\omega_1^2} = \frac{(a_1-\omega_1^2)}{2\xi_1\omega_1} = a_2-2\xi_1\omega_1 \qquad (3\text{-}25)$$

Case of k=4, n=2

Proceeding as before, the a coefficients become:

$$
\begin{aligned}
a_4 &= f_2 = 1 \\
a_3 &= 2\xi_1\omega_1 + f_1 \\
a_2 &= \omega_1^2 + 2\xi_1\omega_1 f_1 + f_o \\
a_1 &= f_1\omega_1^2 + 2\xi_1\omega_1 f_0 \\
a_0 &= f_o\omega_1^2
\end{aligned}
\tag{3-26}
$$

When solved for the coefficients f, equations (3-26) yield:

$$
\begin{aligned}
f_2 &= 1 \\
f_1 &= a_3 - 2\xi_1\omega_1 = \frac{a_2}{2\xi_1\omega_1} - \frac{\omega_1}{2\xi_1} - \frac{a_0}{2\xi_1\omega_1{}^3} = \frac{1}{\omega_1^2}\left(a_1 - \frac{2\xi_1 a_0}{\omega_1}\right) \\
f_0 &= \frac{a_0}{\omega_1^2} = \frac{a_1}{2\xi_1\omega_1} - \frac{a_3\omega_1}{2\xi_1} + \omega_1^2 = a_2 - 2\xi_1\omega_1 a_3 - \omega_1^2 + 4\xi_1^2\omega_1^2
\end{aligned}
\tag{3-27}
$$

Case of k=5, n=3

Similarly:

$$a_5 = f_3 = 1$$

$$a_4 = 2\xi_1\omega_1 + f_2$$

$$a_3 = \omega_1^2 + 2\xi_1\omega_1 f_2 + f_1$$

$$a_2 = \omega_1^2 f_2 + 2\xi_1\omega_1 f_1 + f_0$$

$$a_1 = \omega_1^2 f_1 + 2\xi_1\omega_1 f_0$$

$$a_0 = \omega_1^2 f_0$$

and:

$$f_3 = 1$$

$$f_2 = a_4 - 2\xi_1\omega_1 = \frac{a_2}{\omega_1^2} + a_0\left(\frac{4\xi_1^2}{\omega_1^4} - \frac{1}{\omega_1^4}\right) - \frac{2\xi_1 a_1}{\omega_1^3}$$

$$= \frac{a_3}{2\xi_1\omega_1} - \frac{a_1}{2\xi_1\omega_1^3} + \frac{a_0}{\omega_1^4} - \frac{\omega_1}{2\xi_1}$$

$$f_1 = \frac{a_1}{\omega_1} - \frac{2\xi_1 a_0}{\omega_1^3} = a_3 - 2\xi_1\omega_1 a_4 + 4\xi_1^2\omega_1^2 - \omega_1^2$$

$$f_0 = \frac{a_0}{\omega_1^2}$$

Although the coefficients of f have been derived for only up to the fifth order case, they can easily be obtained for higher order cases if necessary.

Example 3-6 (Third Order Characteristic Equation)

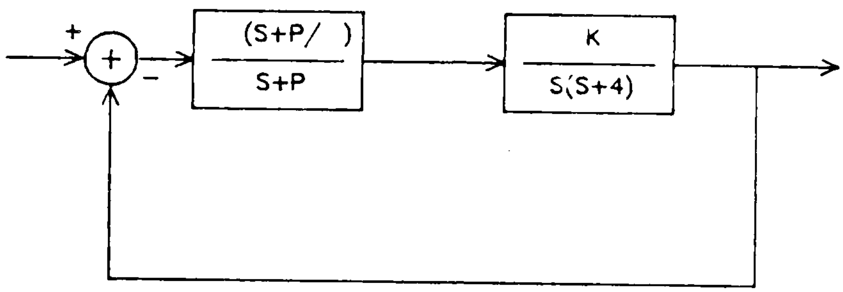


Figure 3-4

Design a cascade compensator for the system of Figure (3-4) to obtain:

1. Characteristic roots at zeta=0.5 and omega=40.
2. $K_e \geq 250$.
3. The specified roots are to be dominant.

The characteristic equation of Figure (3-4) is:

$$S^3 + (4+P)S^2 + (4P+K\gamma)S + KP = 0$$

or

$$S^3 + (4+\alpha)S^2 + (4\alpha+K\beta)S + K\alpha = 0$$

where $\alpha = P$ and $\beta = \gamma$.

Here $G = \frac{K}{e(s)} = \frac{K}{s^2+4s}$, so $e_0=0$, $e_1=4$, $e_2=1$, $U_2=1$, $U_3=0$.

Application of equations (3-20) yields:

$$B_1 = -K+\omega^2 = -K+1600 \qquad B_2 = -4\omega + \omega^2 U_2 = 1440$$

$$C_1 = 0 \qquad C_2 = -\omega K = -40K$$

$$D_1 = 4\omega^2 - \omega^3 U_2 = -57600 \qquad D_2 = 4\omega^2 U_2 - \omega^3 U_3 = 6400$$

From equations (3-21) are obtained:

$$\alpha = \frac{57600}{-K+1600}$$

$$\beta = \frac{1440(-57600)-(-K+1600)(6400)}{-40K(-K+1600)} \tag{3-28}$$

For any value of K, equations (3-28) will produce values of alpha and beta that provide characteristic roots at zeta=0.5 and omega=40. However, only certain ranges of K will meet the steady-state error and dominance requirements. To satisfy the error consideration it is necessary that $K \geq 1000$. Since $f_1(s)$ is of order one, equations (3-25) apply and:

$$f_1 = 1$$

$$f_0 = \frac{a_0}{\omega_1^2} = \frac{(a_1-\omega_1^2)}{2\xi_1\omega_1} = a_2 - 2\xi_1\omega_1$$

From the characteristic equation it is seen that

$$a_2 = 4+\alpha$$

$$a_1 = 4\alpha + K\beta$$

$$a_0 = K\alpha$$

The real part of the specified roots is $\xi\omega=20$. Arbitrarily choosing a dominance factor of five, the dominance criterion becomes: $f_0 > 5\xi\omega = 100$. To satisfy this requirement, the simplest form of f_0 will be chosen, namely $f_0=\frac{a_0}{\omega^2}$. It is then seen that $f_0 = \frac{K\alpha}{1600} = \frac{57600K}{1600(-K+1600)} = \frac{36K}{1600-K}>100$. This requires that K>1176.5. Since K>1176.5 also satisfies the error specification, a value of K=1180 is arbitrarily chosen. Using this value of K, it is found that: $\alpha=137$, $\beta=4.3$ and $f_0=101$. As a check, from the expression $f_0=a_2-2\xi\omega$:

$$f_0 = (4+137)-2(0.5)(40) = 101$$

Example 3-7 (Fourth Order Characteristic Equation)

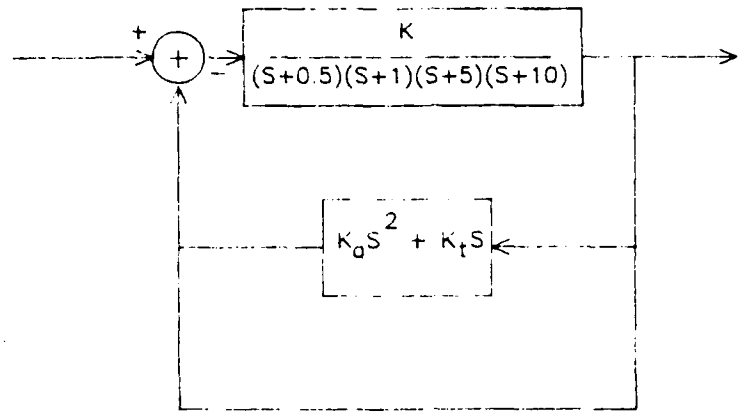


Figure 3-5

Compensate the system of Figure (3-5) using tachometer plus acceleration feedback to obtain:

1. Characteristic roots at zeta = 0.5 and omega = 2.
2. $K_e \geq 12$.
3. Dominance of specified roots.

The characteristic equation of Figure (3-5) is:

$$S^4+16.5S^3+(73+\alpha)S^2+(82.5+\beta)S+25+K = 0$$

where $\alpha = KK_a$ and $\beta = KK_t$.

$$G = \frac{K}{e(s)} = \frac{K}{S^4+16.5S^3+73S^2+82.5S+25}$$

By inspection $e_0=25$, $e_1=82.5$, $e_2=73$, $e_3=16.5$, $e_4=1$, $U_2=1$, $U_3=0$, and $U_4=-1$. When equations (3-9) are employed one finds:

$$\alpha = \frac{K-135}{4}, \qquad \beta = \frac{K-24}{2}$$

From the characteristic equation it is seen that $a_4=1$, $a_3=16.5$, $a_2=73+\alpha$, $a_1=82.5+\beta$, $a_0=25+K$. Since the quotient polynomial $f_1(s)$ is a quadratic, i.e., $S^2+f_1S+f_2=0$, equations (3-27) apply. It would be desirable that, from a dominancy viewpoint, $f_1>5\xi_1\omega_1 = 5$. However, looking at the dominancy equations for this case (equations (3-27)), it is seen that one of the several expressions for f_1 is $f_1=a_3-2\xi_1\omega_1$, which is a fixed constant even though the remaining expressions for f_1 involve one or more variables. Thus, $f_1=14.5$. Noting the most simple expression for f_0 in equations (3-27), $f_0= \frac{a_0}{\omega^2}$. Now, since $f_1=14.5>5$, a dominant situation already exists. However, the system's performance can be further improved by choosing appropriate values of zeta and omega for the secondary roots. From the error specification it is necessary that $\frac{K}{25}\geq 12$ or $K\geq 300$. Now $f_1(s)=S^2+14.5S+\frac{a_0}{\omega_2^2}$ or $f_1(s)=S^2+14.5S+6.25+0.25K$. For K=300, $f_1(s)= S^2+14.5S+81.25$. Therefore, $2\xi_2\omega_2=14.5$, $\omega_2^2=81.25$ implying $\omega_2=9$. Then zeta=0.806. These appear to be reasonable values for ξ_2 and ω_2 since the secondary roots taken alone would produce less overshoot and a smaller settling time than the primary roots. Using this value of K one obtains for alpha and beta:

$\alpha = 41.25$, $\beta = 138$

and since $\alpha = KK_t$ and $\beta = KK_a$, K_t is found to be 0.1375 while $K_a = 0.46$.

As an added bonus of this method, all the roots of the characteristic equation are now known and the system's time response could be computed if desired.

IV. PROGRAM DESCRIPTION

The goal in developing any computer aided design program should be twofold: (1) provide the user with a single, easily understandable, easily usable and comprehensive engineering tool, and (2) dramatize its efficiency above that of other currently available methods. Through the examples of the following chapter the second goal will be demonstrated. It is first desirable to reveal the methodology and internal structure of the parameter plane curve program--as a consequence it is hoped that the first goal will be affirmed.

A. THE PROGRAM

The parameter plane curve--generating program, or "program" as it will be called henceforth, consists of a large driving routine which includes all necessary calculations with which to generate the curve data, and several supporting subroutines (i.e., curve plotting, root solving, data saving, etc.). This entire package is included as a user-selected option within another controls system computer aided design package. Among other options, the latter CAD program includes a root-locus analysis--as mentioned earlier, the usefulness of either the parameter plane or root locus technique for design of a controls system is somewhat limited, but in combination their effectiveness is synergistic (Chapter V will assert the dual roles of the root-locus and parameter plane methods). Facilities available within the program are many; the major options are:

1. Plotting of constant zeta curves, with zeta as a function of omega.

2. Plotting of constant omega curves, with omega as a function of zeta.

3. Plotting of constant sigma (real root) curves.

4. Plotting of constant zeta-omega curves.

5. Tabular output.

6. Rescaling of the plots.

Input of certain data is required to enable the program. These inputs include:

1. Starting value of ω_n.

2. Decades of ω_n to be considered.

3. Number (and values) of constant zeta, omega, sigma, and zeta-omega curves.

4. Coefficients associated with the constant, alpha, and beta terms of the characteristic equation.

Each of the basic program option areas will be described in appropriate detail, as well as their interaction with the input data.

1. Constant Zeta Contours

In practice design specifications for control systems are given in terms of percent overshoot, settling time, error constraints, etc. A value of zeta can be associated with the first of these specifications--that is, a given percent overshoot requirement can be related to a specific value of zeta. Given a specific zeta value, the program calculates the alpha and beta coefficients of the characteristic equation by holding the zeta value constant and varying the value of omega. The limits within which omega is varied are defined by the user's choice of the initial ω_n value, and the number of decades of omega to be considered.

From the nature of the mapping process, it is clear that when the contour of the coefficient plane passes through a designated point (M-point), the original mapping contour on the S-plane passes through a point which is a root of the characteristic equation. The zeta value chosen for the contour is then the zeta for the root. The value of omega associated with the M-point is the radial distance from the origin of the S-plane to the root. Thus, a complex root is determined when the M-point lies on a constant zeta curve of the parameter plane. The value of this root and its complex conjugate is:

$$S = -\xi\omega \pm j\omega\sqrt{1-\xi^2}$$

If the characteristic equation is such that several complex roots exist, then the parameter plane curves required to realize these roots must all pass through the M-point. If the complex roots have the same zeta but different omega values, then the constant zeta curve must pass through the M-point more than once. In fact, once any point on the zeta contour is defined, omega, alpha, and beta are also defined, and all roots of the characteristic equation are thus fixed.

2. Constant Omega Contours

Within the program, for a given value of omega, zeta is varied between zero and one inclusively while omega is held constant.

As with the constant zeta curves, any point on a constant omega contour is the omega for a complex root of the characteristic equation. By selecting an operating point, zeta is also defined whereby a pair of complex conjugate roots is established. Again, once any point is chosen on either a constant zeta or a constant omega curve, all roots of the characteristic equation are established.

3. Constant Sigma Contours

When real roots are to be evaluated, it is convenient to return to the characteristic equation:

$$\sum_{k=0}^{m} a_k S^k = 0 \qquad (2\text{-}1)$$

where, again, a_k represents a linear combination of constant, alpha, and beta terms. $S=-\sigma$ (a real number) is then an equation of a straight line on the parameter plane. If any line of constant sigma value passes through the M-point, then the alpha and beta coordinates of the M-point satisfy the characteristic equation for a real root located at $-\sigma$.

For program considerations, one enters a positive value for sigma (corresponding to a real root at $-\sigma$) and the constant sigma contour (straight line) is developed. The coordinates of any point on this curve produce a real root at $-\sigma$. That this is a useful tool, consider that system specifications can be achieved by placing a pair of complex conjugate roots of the characteristic equation at a specific location. To ensure dominance of this root pair, the real part of the complex roots so placed should be smaller in magnitude than that of the remaining system roots. Roots placed at a specified sigma value can thus ensure at least one real root whose magnitude is greater than the real part of the intended complex conjugate pair.

4. Constant Zeta-Omega Curves

For a fixed value of zeta and omega a pair of complex conjugate roots is defined in terms of the expression:

$$S = -\xi\omega \pm j\omega\sqrt{1-\xi^2}$$

The real part of these roots is, thus, defined by the zeta-omega product. Note that settling time is defined as $T_s = \frac{4}{\xi\omega}$. If the $\xi\omega$ product is known, so, too, is the duration of the transient response. Thus, by specifying a constant value for the zeta-omega product, any point on the contour generated by this value will produce a given settling time.

For any of the parameter plane contours, it is desirable to ascertain the values of the characteristic roots for every few values of alpha and beta. This feature has been incorporated within the tabular output facility as described below.

5. Tabular Output

For each of the zeta, omega, and zeta-omega parameter plane curves, an arbitrary though reasonable 300 points are calculated with which to plot the contours. For the constant sigma curves, only two points are needed to define the required straight lines (in practice, 4 points are generated to ensure that the sigma contours can be plotted within the user-defined axes limits). Because of the bulk of data points so generated, tabular output is offered as an option (as is graphical output), and all points so generated are listed for the user. In addition, it is worthwhile to calculate system roots for given values of alpha and beta. However, computation of roots for each alpha and beta pair would cost unnecessary computer time and will likely tax the user's patience with the bulk of output so generated. Thus the characteristic roots are generated for every tenth[1] pair of alpha and beta values.

6. Plot Rescaling

Regardless of the plot scale selected by the user, the program generates the full 300 data points for each curve requested (4 points for constant sigma lines). When the graphical output option is requested, the first family of curves is automatically scaled to encompass each and every data point. The disadvantage of this technique is that, because most activity for the vast majority of systems occurs near the physical origin (i.e., alpha=beta=0), the curves may at first appear within only a very small sector of the entire plot area, and often they are indistinguishable from one another. The advantages of automatic scaling for the first set of parameter plane curves far outweigh this disadvantage. First, by plotting all available data points, the possible limits for alpha and beta are exposed--this is important if very large values of alpha and/or beta are required to meet the design specifications. Second, for some systems the area of activity may not occur near the origin, and automatic scaling spares the user the task of selecting a sector and possibly missing a sector of interest.

---

[1] Although a seemingly arbitrary choice, the generation of characteristic roots for every tenth alpha, beta pair produces a very tidy output on the commonly-used IBM-3278 computer terminal.

Once the user is able to view the panoramic alpha, beta parameter curves, it becomes obvious which sector(s) are of interest. The user then has the option to rescale the set of curves by selecting upper and lower limits for the alpha and beta axes. He may continue to rescale the family of parameter curves as often as is desirable, and at any time the autoscaling option may be recalled.

The curves generated by the above program are sufficient to explore most control system engineering problems. The use of these parameter plane contours, and their interaction with one another, will be evidenced in the next chapter. The source code listing of the program is included as Appendix B.

B. INSTRUCTION TO THE USER

The parameter plane program is highly interactive--the user is prompted for each required input. A brief description of all but the most trivial input items follows.

- Starting value of ω_n: For most control systems the initial value of ω_n is chosen to be zero. Because ω_n is used in the denominator of certain of the parameter plane equations, ω_n must be greater than zero. However, the user may choose ω_n arbitrarily close to zero if desired.

- Number of decades: For the majority of control system problems a suitable number of decades to be considered might be two or three. For higher order systems, it would be advisable to start with a slightly larger number of decades, especially if the initial ω_n value is small. For subsequent families of curves, the number of decades can easily be changed.

- Constant, alpha, and beta coefficient values: Characteristic coefficients are requested from the highest to lowest order term. By way of an example, a third order characteristic equation might be:

$$S^3 + (3\alpha + \beta + 10)S^2 + \alpha S + (\beta+5) = 0$$

  Here, the constant coefficients would be entered in the following sequence: 1,10,0,5 while alpha and beta coefficients would be entered as 0,3,1,0 and 0,1,0,1 respectively.

- Zeta values: By convention, values are restricted to between zero and one, inclusively.

- Sigma values: Positive values of sigma correspond to negative real roots. Since few, if any, practical engineering applications exist for designing a positive real root into a system, negative values for sigma are disallowed.

- Omega values: Values for constant omega curves are restricted in the lower limit to the starting ω_n value, and in the upper bound by $\omega_n x10^{decades}$.

- Zeta-omega values: As with the constant sigma curves, values for constant zeta-omega contours must be greater than or equal to zero.

The user then has the following options:

1. Review entries.
2. Change any entry.
3. Tabular output.
4. Graphical output.

Remember that tabular output includes 300 data points for all but the sigma contours. Characteristic roots are displayed for every tenth alpha, beta pair. Because of the bulk of output for this option, use it only when necessary. If a printed copy of the tabular output is desired, type "record on" before invoking the program. Upon exiting the program, type "record off", after which the user may save the preceding terminal session in a listing file designated by a name of his choosing. Simply print the listing file, which will include all output which has transpired on the terminal between the two calls to "record".

When graphical output is requested, all curves are superimposed on the same plot. The first set of curves is produced with an autoscaling feature, which plots all points calculated (the range of points depends on your choice of initial ω_n, number of decades, zeta values, omega values, etc.). For most characteristic equations only the first quadrant (i.e., positive alpha and beta values) will be of interest, since negative values usually (but not necessarily) imply negative characteristic coefficients and, thus, positive roots leading to system instability. The nature of the first (autoscaled) set of curves will reveal the actual areas of interest for subsequent plots for the same system.

Finally, after each family of curves is plotted, the user has six additional options:

1. New problem.
2. Same problem.
3. Root finder.
4. Save problem.

5. Create "DISSPLA metafile".

6. Return to main menu.

Items 1, 4, and 6 are self-explanatory. The remaining options deserve some additional explanation.

- Option 2: This option can be used to re-enter the problem at a point prior to actual plotting. Then, specific input values can be added or modified, tabular and/or graphical output can be requested, and entries can be reviewed. It is within this option that the graph coordinate axes can be rescaled. If user-defined scaling is desired, the minimum and maximum values for the axes are requested.

- Option 3: Although within the tabular output feature a set of characteristic roots is produced for every tenth pair of alpha and beta values, the bulk of output using that option may prove excessive for some applications. Here, the user has the option of choosing specific values of alpha and beta (e.g. extracted from the family of parameter plane curves) and obtaining the system roots.

- Option 5: The program provides a choice from among four graphic output devices. Usually the user will nominate the TEK618 graphics terminal due to its relatively high quality plot resolution. Once the user has produced a parameter plane plot to his liking, he may wish a final plot of very high resolution. By selecting this option, the plot is stored as a DISSPLA metafile, and the program is terminated (termination of the program is necessary at this point due to an anomaly of the DISSPLA graphics package). Simply type "DISSPOP" and follow the simple instructions, choosing the default options as they are presented. Within the "DISSPOP" routine, any of several output devices can be called, including the high resolution Versatec plotter and the 3800 laser printer.

V. PARAMETER PLANE CURVES-GRAPHICAL METHOD

A. GRAPHICAL SOLUTION

The algebraic solutions discussed in Chapter III have the disadvantage that a fixed value of zeta and omega must first be chosen to compute the alpha and beta terms. In some instances it is possible to modify the remainder polynomial so as to ensure that the specified roots are dominant. However, it is not always possible to guarantee that roots placed at a specified location can be made dominant. Thus, an exhaustive trial-and-error procedure may be required to achieve the best values for the various parameters. Trial-and-error may also be required in the design of cascade compensators where a specific root location may require parameter values that are not physically realizable. In these cases, the calculation must be repeated in terms of slightly modified specifications; possibly a different means of compensation must be used.

To avoid this trial-and-error analytical approach, one can employ the graphical solution. Once a family of curves is generated by the program one can, by choosing an M-point in the parameter plane, obtain from the curves the n roots of the nth order characteristic equation. The trial-and-error procedure can then be done visually to reveal an operating point which best meets the given specifications.

Example 5.1 (An Attitude Control System for Large Launch Vehicles)

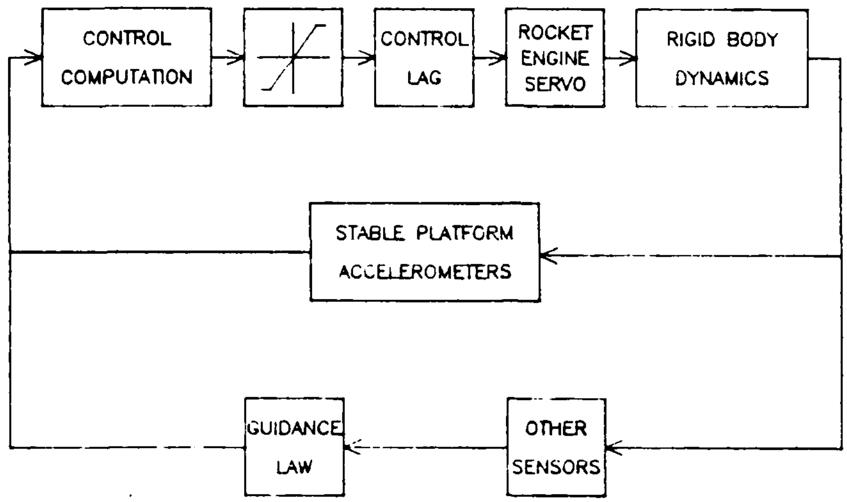


Figure 5-1

Figure 5-1 shows the mechanization for a control system for a large launch vehicle; Figure 5-2 shows the equivalent block diagram

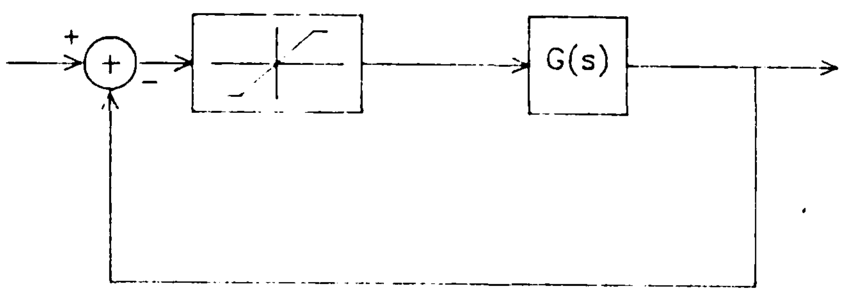


Figure 5-2

where the equivalent G(s) is:

$$G(s) = \frac{(a_0 S^2 + K_1 S + K_2)(S^2 - C)}{S^3\left(\frac{S^3}{\omega_e^2} + \frac{2\xi_e S^2}{\omega_e} + S + a_1 C\right)}$$

From G(s) one obtains the system's characteristic equation:

$$\frac{S^6}{\omega_e^2} + \frac{2\xi_e}{\omega_e} S^5 + (1+a_0)S^4 + (a_1 C + K_1)S^3 + (K_2 - a_0 C)S^2$$

$$- CK_1 S - CK_2 = 0$$

where a_0, a_1, K_1, K_2 = control system gains

ξ_e = damping ratio of control servo

ω_e = natural frequency of control servo

Gains K_1 and K_2 are chosen as the system parameters α and β, respectively, to be portrayed in the parameter plane. One must then find suitable values for these parameters to yield a desired stability margin and a satisfactory transient response. For a typical choice of system parameters (such as those describing Saturn V), it is assumed that:

$\xi_e = 0.717$

$\omega_e = 4.71$ Hz = 29.594 radians

$a_0 = -0.5$

$a_1 = 1.0$

$C = 0.7$

Then the system's characteristic equation becomes:

$$0.0011S^6 + 0.0485S^5 + 0.5S^4 + (0.7+\alpha)S^3 + (0.35 + \beta)S^2$$

$$- 0.7\alpha S - 0.7\beta = 0$$

Various values of a_0 and a_1 were used to deduce their effect on the stability regions, which is indicated in Figure 5-3. The numbers of stable and unstable roots, respectively, are portrayed in parentheses for each region of the parameter plane.

The analysis procedes as follows: the ξ=0 contour represents boundaries of stability (or relative stability when ξ>0) associated with pairs of complex conjugate roots. The S=0 (sigma=0) curve represents real root stability boundaries. The region of stability is thus that area bounded by these two contours. See Figures 5-4a and 5-4b for a magnified view of this area. Any negative alpha-beta pair from within this region will exhibit six stable (i.e., all within left-half S-plane) roots and system stability will be assured. Note that from the form of the characteristic equation, both alpha and beta must be negative to obtain a stable system. To illustrate, let us select an arbitrary operating point constrained to lie within the lower loop of Figure 5-4b

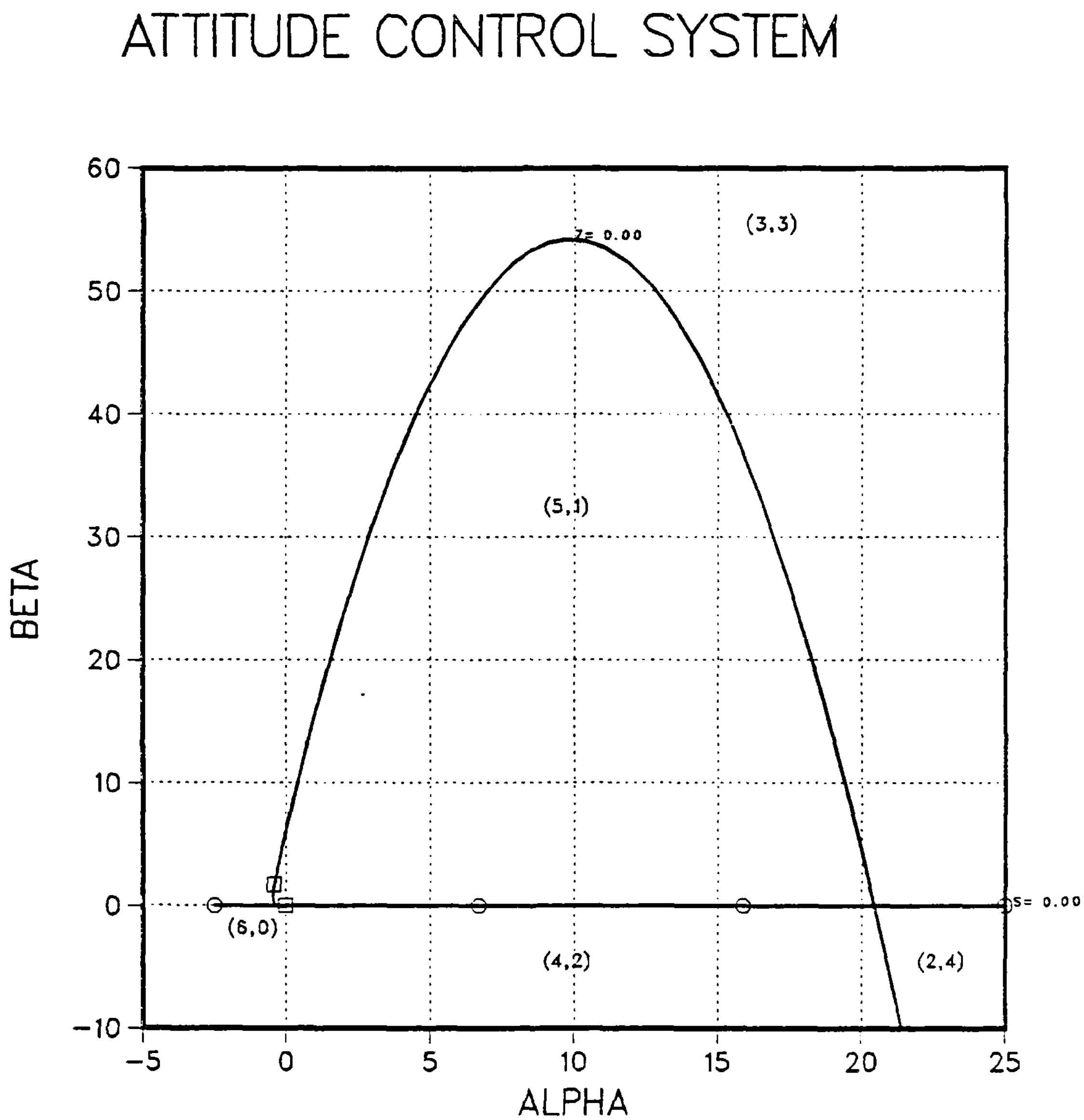


Figure 5-3
Parameter Plane Curves for

$$0.0011\ S^6 + 0.0485S^5 + 0.5S^4 + (0.7+\alpha)S^3 + (0.35+\beta)S^2 - 0.7\beta\, S - 0.7\alpha = 0$$

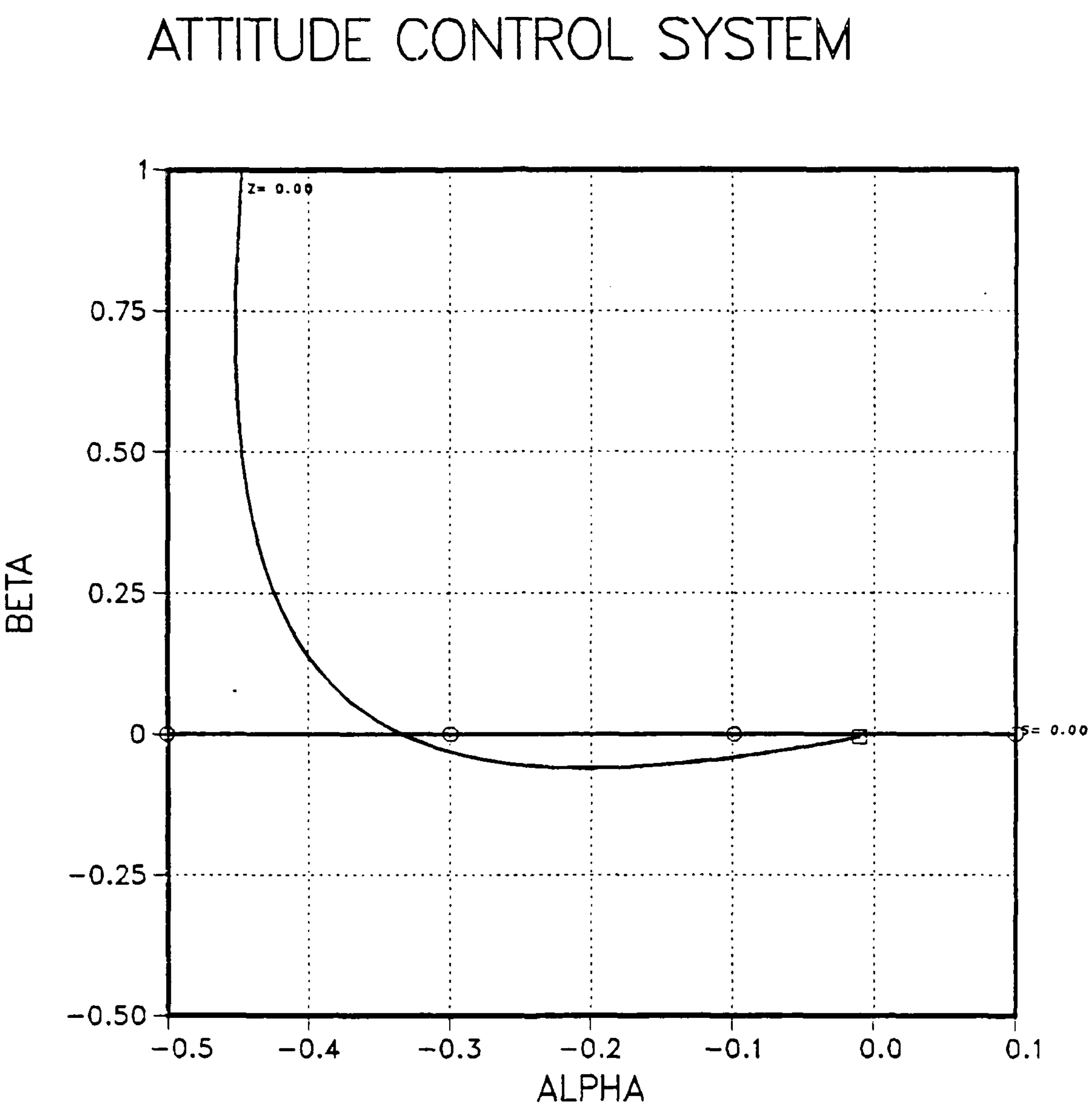


Figure 5-4a
Parameter Plane Curves for

$$0.0011S^6 + 0.0485S^5 + 0.5S^4 + (0.7+\alpha)S^3 + (0.35+\beta)S^2 - 0.7\beta S - 0.7\alpha = 0$$

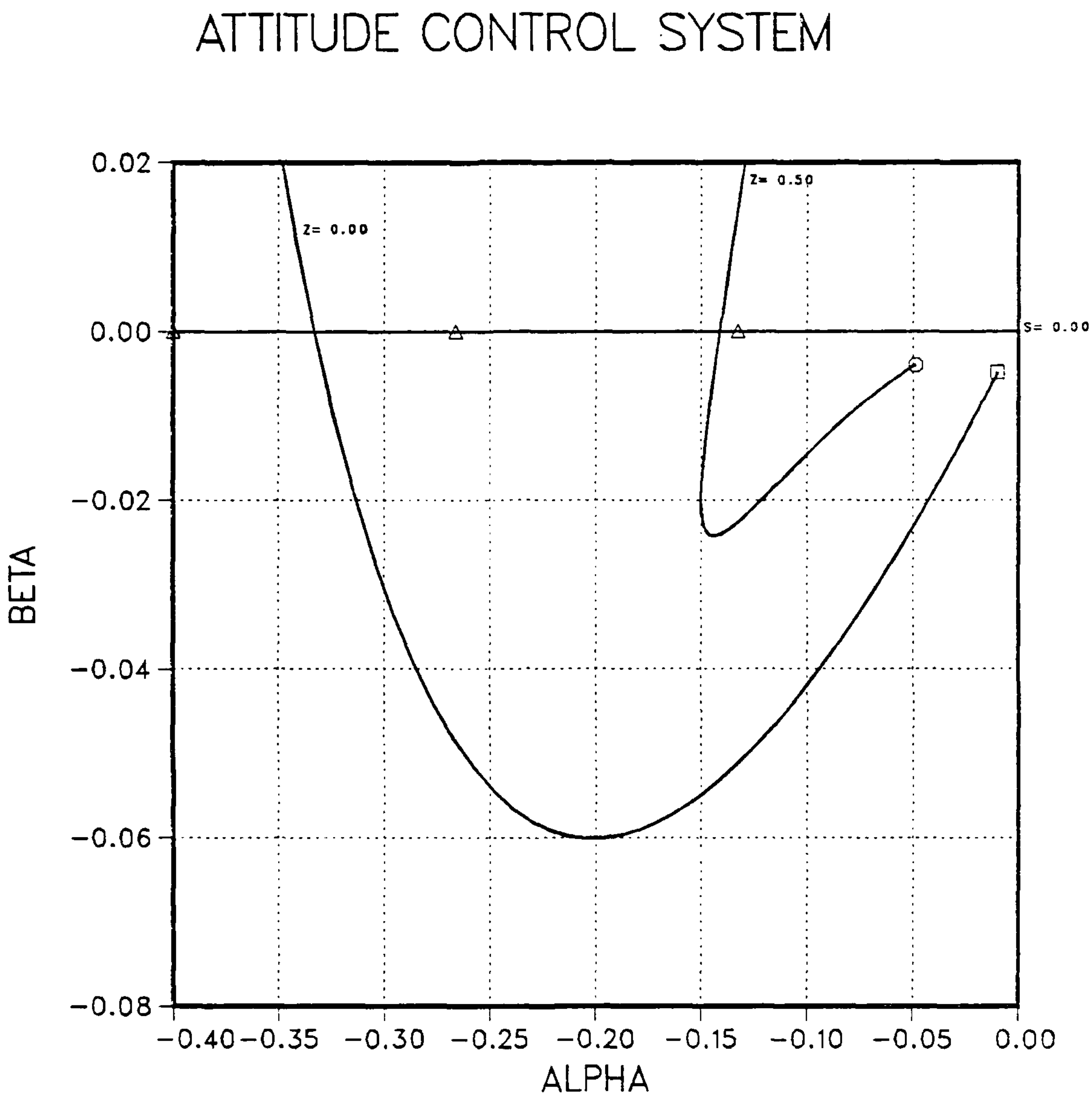


Figure 5-4b
Parameter Plane Curves for
$$0.0011S^6 + 0.0485S^5 + 0.5S^4 + (0.7+\alpha)S^3 + (0.35+\beta)S^2 - 0.7\beta S - 0.7\alpha = 0$$

and also satisfying the requirement that both alpha and beta be negative. Superimposed on Figure 5-4b is the constant $\xi=0.5$ contour, upon which our M-point might be chosen. If we select, say, $\alpha=-0.15$ and $\beta=-0.02$ (corresponding to $\xi=0.5$ and $\omega=0.5$), the system roots are:

$-0.26 + j0.45$	-dominant roots
$-0.26 - j0.45$	
$-0.32 + j0.12$	
$-0.32 - j0.12$	
$-14.6 + j0$	
$-26.7 + j0$	

Coincidentally, the roots associated with our choice of zeta and omega are seen to be dominant. The actual choice of an operating point may depend on other criteria not discussed here.

Since K_1 and K_2 (α and β, respectively) are functions of ω, ξ, a_0, a_1, C, ω_e, and ξ_e, they may be determined for various instances of flight by plotting several constant zeta and constant omega curves. Actually, K_1 and K_2 vary so little within the range of values used for C, for specified values of ξ and ω, that it becomes possible to choose constant values for K_1 and K_2.

This has been a relatively simplistic treatment of a complicated control system problem, but it demonstrates the power of the parameter plane graphical te nique. Knowing nothing more than the system's characteristic equation, the static and dynamic stability boundaries may be obtained to define the area of overall stability. Of course, other system constraints may exist to further limit this area. As a note, the root-finding option available within the program was used to confirm the numbers of stable and unstable roots lying within each region of Figures 5-3 and 5-4.

Example 5-2 (Alternator Voltage Regulator)

In designing a voltage regulator of the type shown in Figure 5-5, we must find values for K_0, K_1, and K_2 that provide stability and good transient performance.

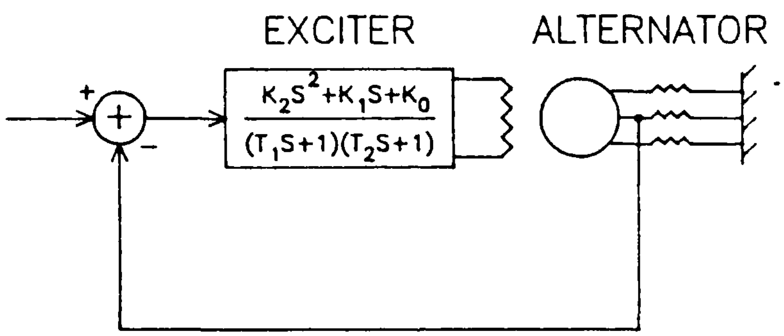


Figure 5-5

The characteristic equation of the system, including alternator, tie-line, etc. is (see reference 8 for details):

$$0.0095S^5 + 0.1325S^4 + 1.72S^3 + (K_2+7.55)S^2 + (K_1+9.1)S + K_0-2.5 = 0$$

We will consider $K_o=0$ as fixed and α and β as variable parameters corresponding to $K_2+7.55$ and $K_1+9.1$, respectively. Curves of α versus β are plotted in Figure 5-6 with ω_n as the variable parameter (for this system, K_1 and K_2 are known to be functions of ω_n alone). Since the program plots constant zeta curves as a function of varying omega, these curves were selected as logical candidates to study the problem.

The beta axis corresponds to the zero value of damping constant (sigma) since below the zeta=1.0 curve, the real roots (Thaler and Brown 1960) are the negative slopes of the tangents drawn from the point in question to the $\xi=1$ curve. The machine is then stable for any (α, β) pair between the $\xi=0$ and S=0 curves. The greatest stability of the machine is then possible when both zeta and sigma are largest. Further, the best stability can be expected in the region bounded by the $\xi=0.3$ and $\xi=1.0$ contours (Kabriel 1967). Similar stability limits of α and β can be investigated by taking K_0 as 10, 20, 30 and so on.

Consider now $K_1=0$ as fixed. Then α and β will represent the variable parameters $K_2+7.55$ and $K_0-2.5$, respectively. The characteristic equation then becomes:

$$0.0095S^5 + 0.1325S^4 + 1.72S^3 + \alpha S^2 + 9.1S + \beta = 0$$

Again, K_0 and K_2 are known to be functions of ω_n alone. Here, however, ω_n can be solved explicitly [Ref. 8] and one obtains three straight-line equations:

$$\beta = 0$$

$$5.421\alpha - \beta = 3.894$$

$$175.63\alpha - \beta = 4085.0$$

or in terms of system parameters:

$$K_0 = 2.5$$

$$5.421(K_2+7.55)-K_o = 3.894 - 2.5$$

$$175.63(K_2+7.55)-K_0 = 4085.0 - 2.5$$

These are plotted in Figure 5-7. The triangular region bounded by these three lines represents a stable region. The values of (α, β) within the triangle and hence corresponding values of K_0 and K_2 can be predicted for stable operation. The procedure can be repeated for further investigation by taking K_1=15, 30, 45, etc.

The analytical parameter plane technique can be used to determine the stability limits of K_0 and K_1 by choosing several constant values of K_2. But since K_2 has to be selected arbitrarily for this purpose, this method becomes cumbersome and time-consuming. The method presented here, on the other hand, can be used to determine the stable range of either

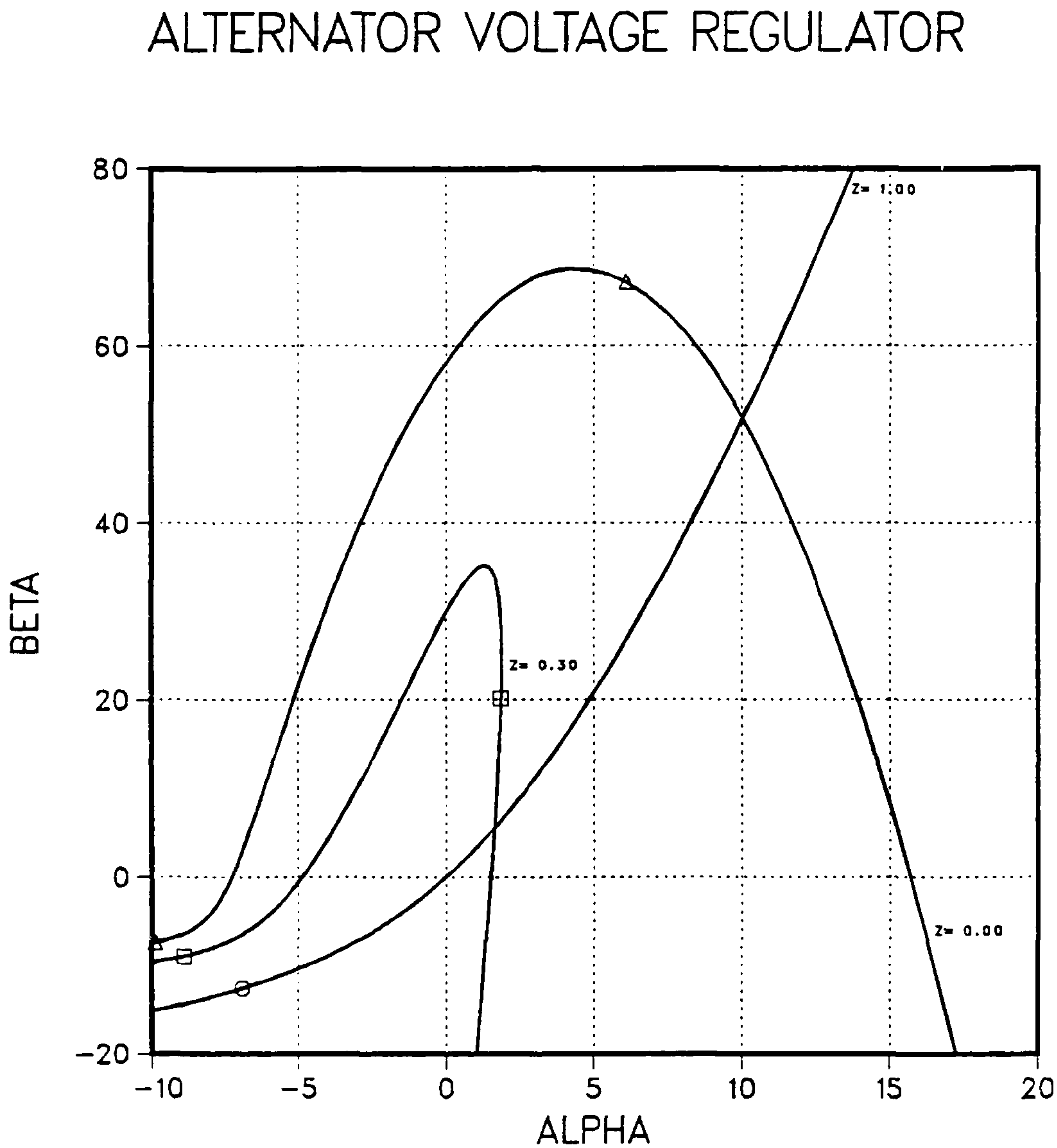


Figure 5-6
Parameter Plane Curves for
$0.0095S^5 + 0.1325S^4 + 1.72S^3 + \alpha S^2 + \beta S - 2.5 = 0$

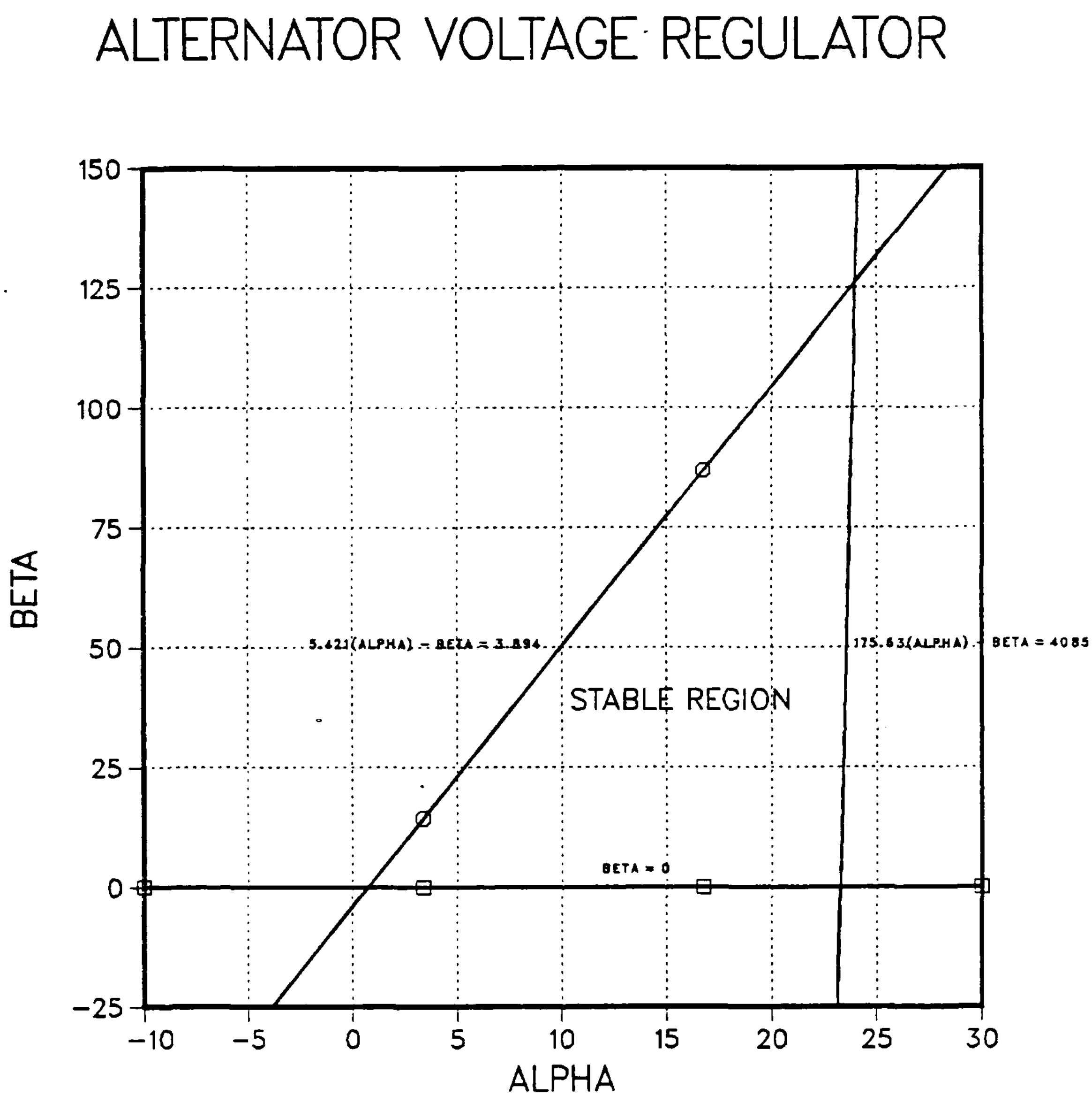
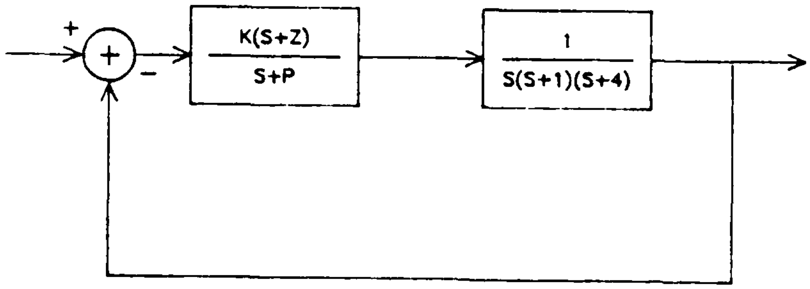


Figure 5-7
Parameter Plane Curves for
$0.0095S^5 + 0.1325S^4 + 1.72S^3 + \alpha S^2 + 9.1S + \beta = 0$

K_0 and K_2 or K_1 and K_2 fixing the third parameter. Over and above the ability to predict stable operation, the method povides a direct measure of the damping at and around a chosen operating point.

Example 5-3 (Lead Compensator)

Consider the system:



The characteristic equation is:

$$S^4 + (5+P)S^3 + (4+5P)S^2 + (4P+K)S + KZ = 0$$

We choose to cancel the pole at S=-1 with the zero; thus Z=1.0 and the characteristic equation becomes:

$$S^4 + (5+P)S^3 + (4+5P)S^2 + (4P+K)S + K = 0$$

Let $P = \alpha$ and $K = \beta$. Then the parameter plane curves are as shown in Figure 5.8. For the coefficients of the characteristic equation to remain positive (and thus ensure stability), it is convenient to consider only positive values for alpha and beta. By inspection we choose $\zeta=0.5$ and $\omega_n=2.0$ as a

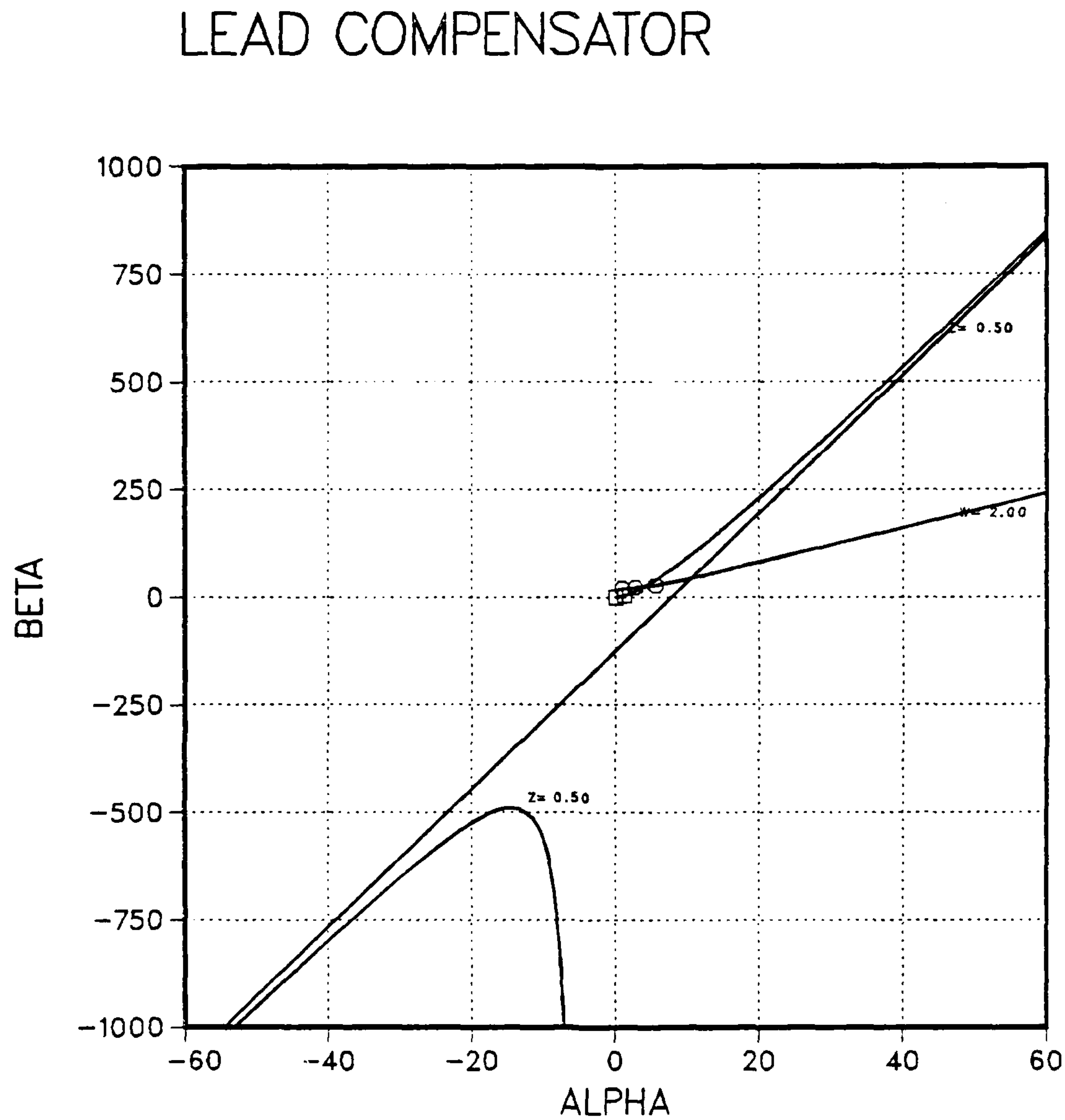


Figure 5-8
Parameter Plane Curves for
$$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+\beta)S + \beta = 0$$

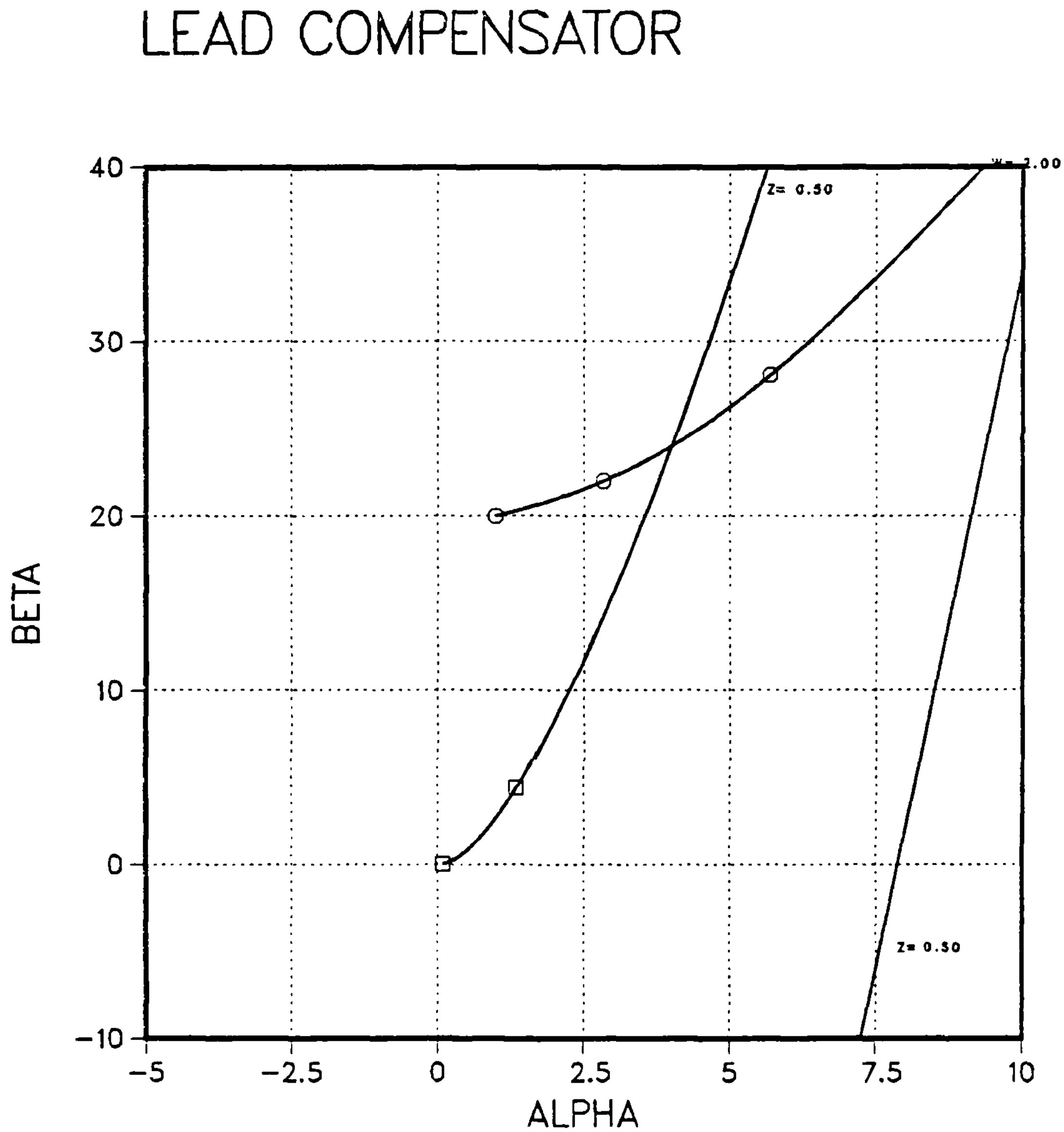


Figure 5-9
Parameter Plane Curves for
$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+\beta)S + \beta = 0$

good operating point, for which $\alpha = 4.0$ and $\beta = 24.0$. The corresponding vicinity of Figure 5-8 is re-scaled in Figure 5-9. Again, using the root-finding option, the roots associated with this (ξ, ω_n) pair are shown to be dominant.

Example 5-4 (Lag Compensator)

If we are especially concerned with steady-state accuracy for a ramp input, it may be advisable to design a lag compensator. The parameter plane permits us to consider steady-state accuracy while designing the transient response. If our system is the same as that considered for the lead compensator, the characteristic equation remains:

$$S^4 + (5+P)S^3 + (4+5P)S^2 + (4P+K)S + KZ = 0$$

But now the error coefficient is $K_e = \frac{KZ}{4P}$. Having three unknown parameters, K, Z, and P, two must be selected (or some combination of two) to be α and β while a numerical value is assigned to the third. Once this choice has been made th^ parameter plane curves can be calculated and plotted, and the loci of constant K_e can be superimposed. Let Z=0.1, $P=\alpha$, and $K=\beta$. The characteristic equation becomes:

$$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+\beta)S + 0.1\beta = 0$$

Parameter plane curves are shown in Figures 5-10 and 5-11. Lines of $K_e = \frac{0.1\beta}{4\alpha} = 0.1, 0.2, 0.3,$ etc. may be superimposed. If we select $K_e = 0.15$ and (similarly to the lead compensator

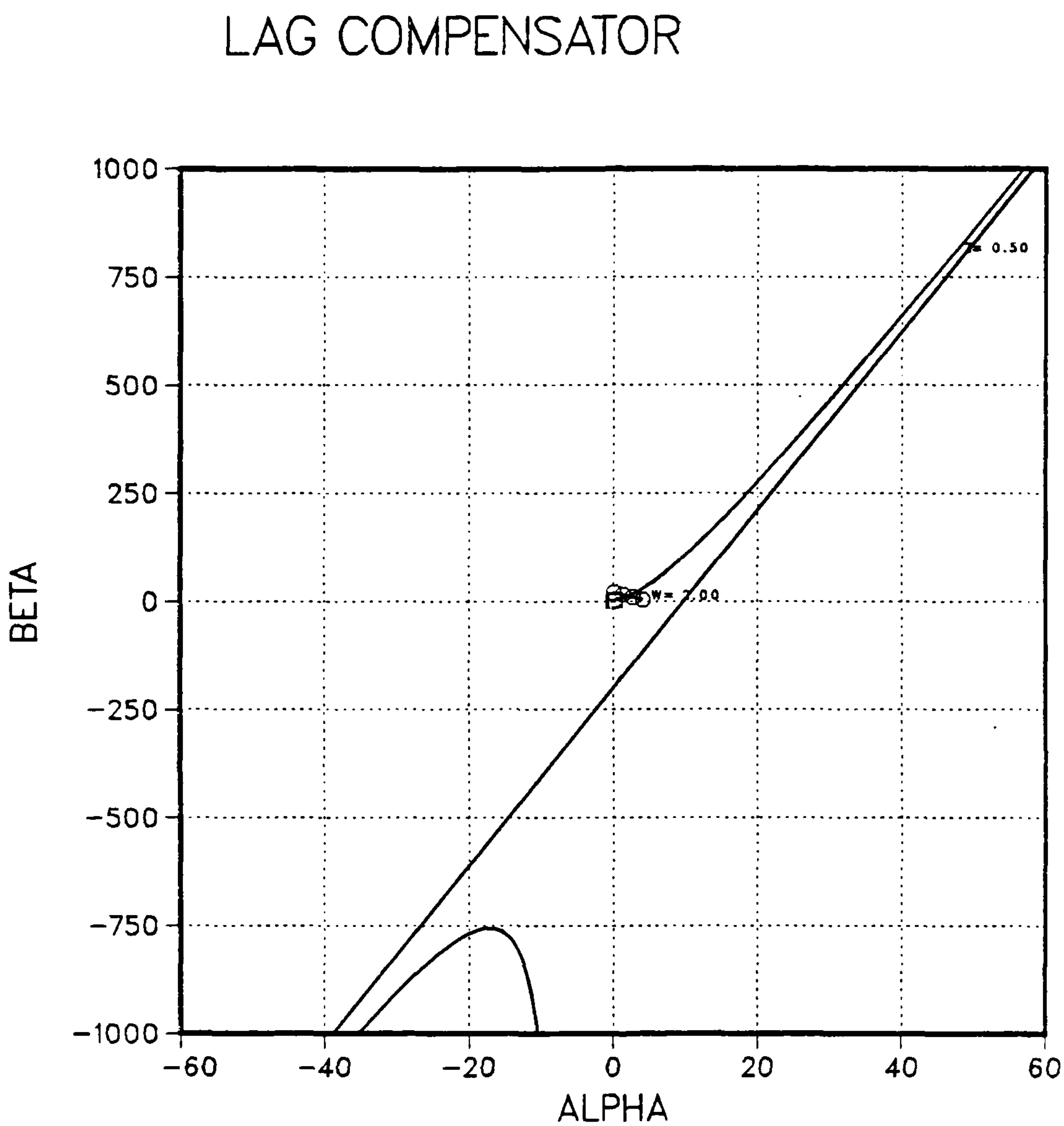


Figure 5-10
Parameter Plane Curves for
$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+\beta)S + 0.1\beta = 0$

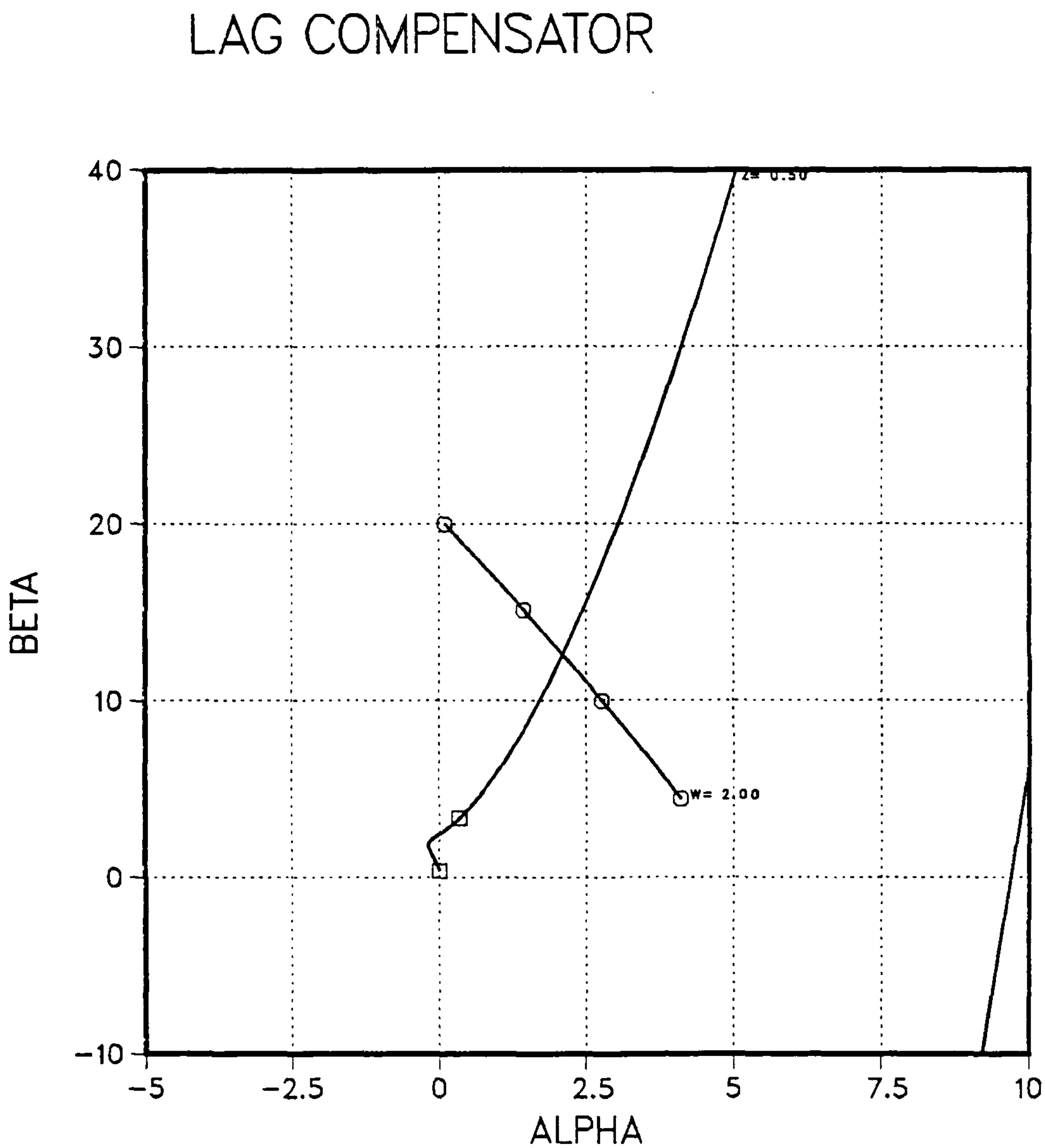


Figure 5-11
Parameter Plane Curves for
$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+\beta)S + 0.1\beta = 0$

example) $\xi=0.5$ and $\omega_n=2.0$, then $\alpha=2.14$ while $\beta=12.89$. This produces dominant roots at $-1.0 \pm j1.7$.

Let us reconsider the systems for which a lead compensator and a lag compensator were designed. The characteristic equation as well as the error coefficient each contain three unknowns (if we assume a numerical value for K_e). We can imbed the error coefficient in the characteristic equation by direct substitution, thereby eliminating one of the unknowns. Let us eliminate the gain parameter K - note that $K = \frac{4PK_e}{Z}$.

Returning to the characteristic equation:

$$S^4 + (5+P)S^3 + (4+5P)S^2 + (4P+\frac{4PK_e}{Z})S + 4PK_e = 0$$

Letting $P = \alpha$ and $\frac{P}{Z} = \beta$, the characteristic equation becomes:

$$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+4K_e\beta)S + 4K_e\alpha = 0$$

If a constant value is now chosen for K_e, say $K_e = 2.0$, the characteristic equation becomes:

$$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+8\beta)S + 8\alpha = 0$$

for which the parameter plane contours are first shown in Figure 5-12 and are further magnified in Figure 5-13. Now when any operating point is chosen on the parameter plane curve(s), the selected (α,β) pair generates $K_e=2.0$. Of course, this procedure can be repeated for any choice of K_e.

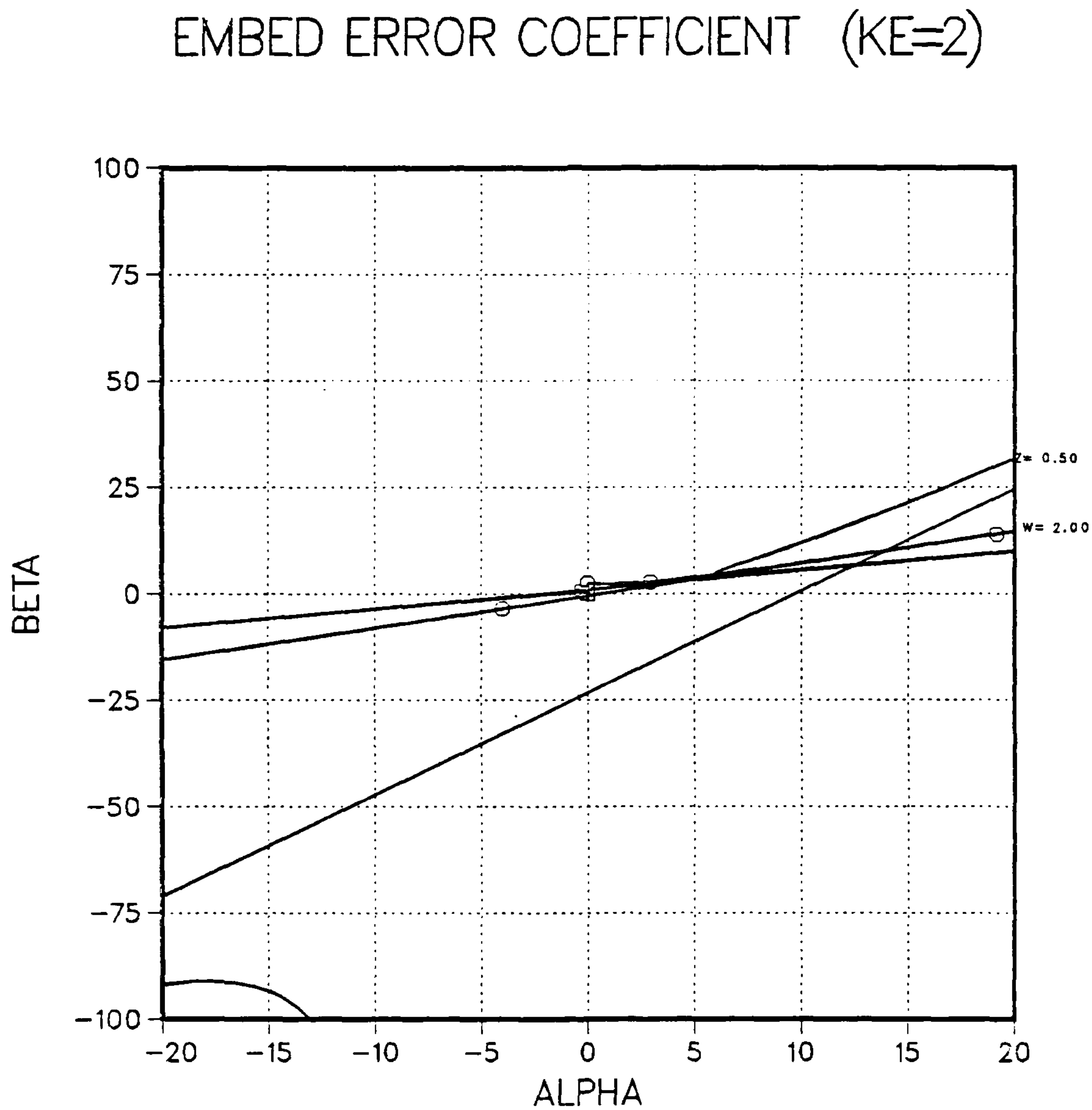


Figure 5-12
Parameter Plane Curves for
$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+8\beta)S + 8\alpha = 0$

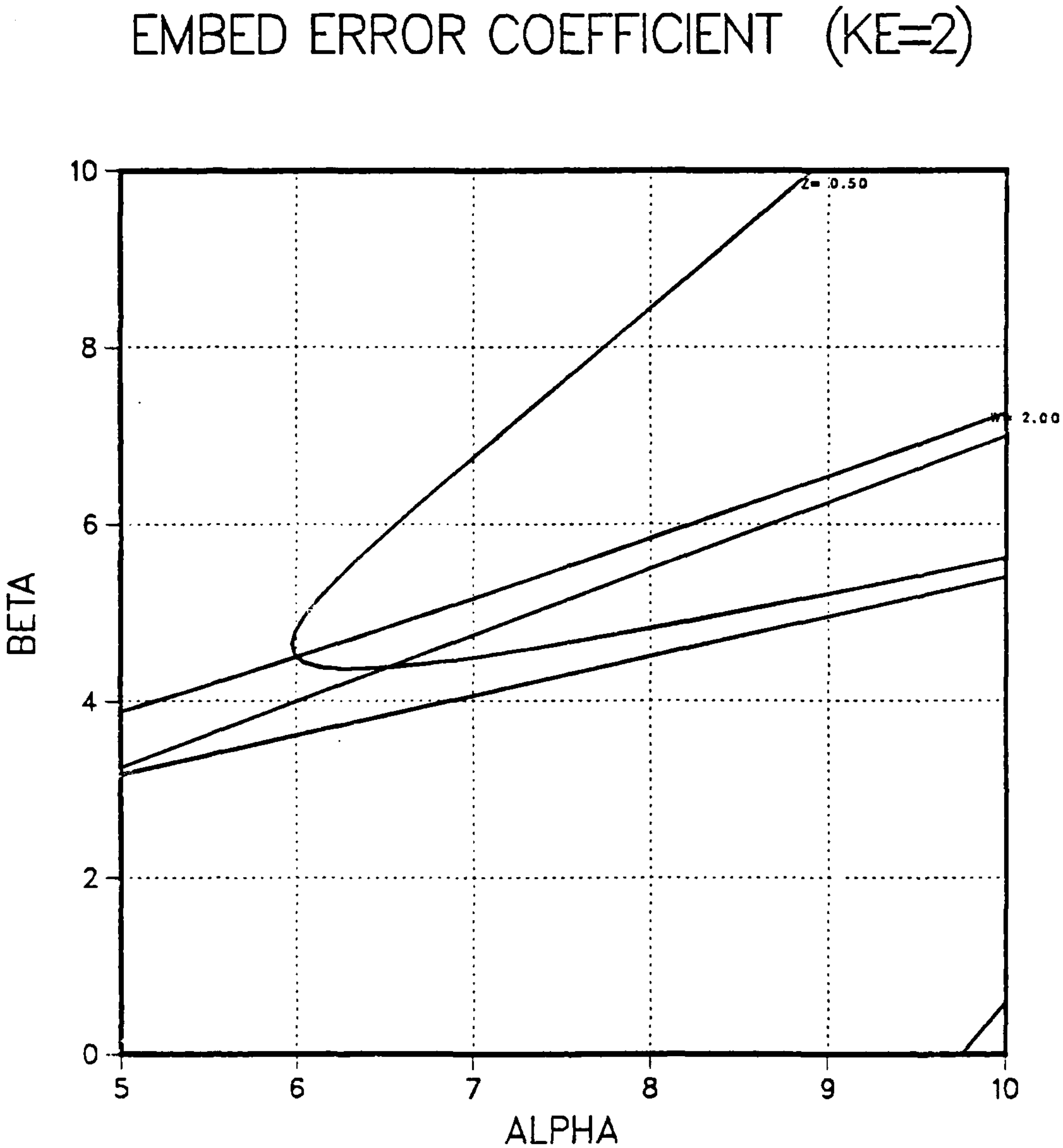


Figure 5-13
Parameter Plane Curves for
$S^4 + (5+\alpha)S^3 + (4+5\alpha)S^2 + (4\alpha+8\beta)S + 8\alpha = 0$

On the parameter plane curves let us choose $\zeta = 0.5$ and $\omega_n = 2.0$ for which:

$$\alpha = 6.00 = P$$

$$\beta = 4.50 = \frac{P}{Z}$$

$$\frac{\alpha}{\beta} = 1.33 = Z$$

$$K = 83 \quad = 36$$

As a check, $K_e = \frac{KZ}{4P} = 2.0$. The roots associated with the selected zeta and omega values are shown to be dominant using the root-finding algorithm and are:

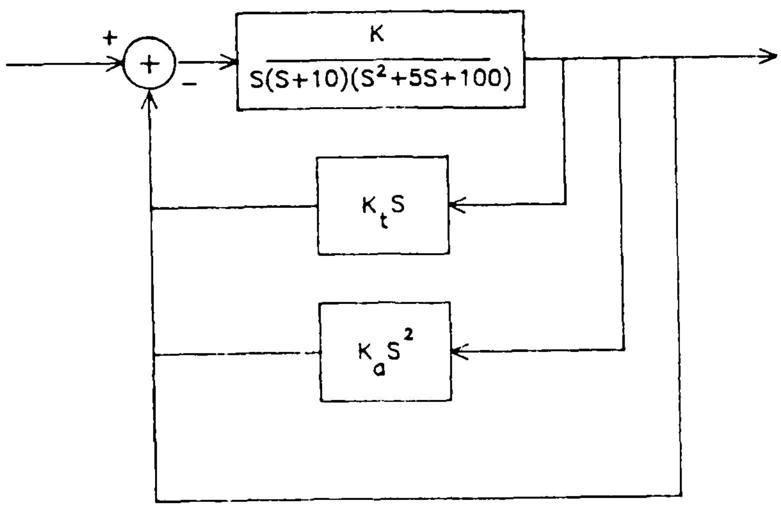
-1.00 + j1.73
- Dominant Pair
-1.00 - j1.72

-1.63 + j0

-7.37 + j0

<u>Example 5-5</u>

As a final engineering example, consider the system below:



For this system, the following specifications are to be met:

1. Set K to the stability limit.

2. Place a dominant root pair within the following region: $0.4 \leq \xi \leq 0.7$, and $2 \leq \omega \leq 6$.

3. Both tachometer and acceleration feedback may be used, but if possible choose only one.

From the figure the uncompensated system's open loop transfer function is:

$$GH = -1 = \frac{K}{S(S+10)(S^2+5S+100)}$$

From which, when expanded, the characteristic equation becomes:

$$S^4 + 15S^3 + 150S^2 + 1000S + K = 0$$

To determine the value of K at the stability limit the Routh array is employed:

1	150	K
15	1000	0
1250	15K	0
$1.25x10^6$-225K	0	0
15K	0	0

Here, the stability limit is seen to be K=5555.5. If both tachometer and acceleration feedback are used the compensated system's characteristic equation becomes:

$$S^4 + 15S^3 + (150+5555.5\alpha)S^2 + (1000+5555.5\beta)S + 5555.5=0$$

where $\alpha=K_a$, $\beta=K_t$, and K has been set to the stability limit. Parameter plane curves for this system are shown in Figure 5-14.

From these curves, the following analysis can be made. The origin of the parameter plane corresponds to the roots of the uncompensated system. Since the $\xi=0$ curve passes through the origin, two roots are located on the $j\omega$ axis of the S-plane as was to be expected from the Routh array. The remaining two roots are also complex and correspond to $\xi=0.8$ and $\omega=5.0$. It is important to note that when an operating point involves two different pairs of complex roots, then the curves for two different values of omega and two different values of zeta must pass through the point. To determine which value of omega corresponds to which value of zeta, it becomes necessary to refer to the program's tabular data output, which is not included here due to lack of space.

With $K_a=0$, the effect of tachometer feedback alone corresponds to movement of the M point along the β axis. In Figure 5-14 the unstable region is determined by an inspection of the way the constant zeta curves tend

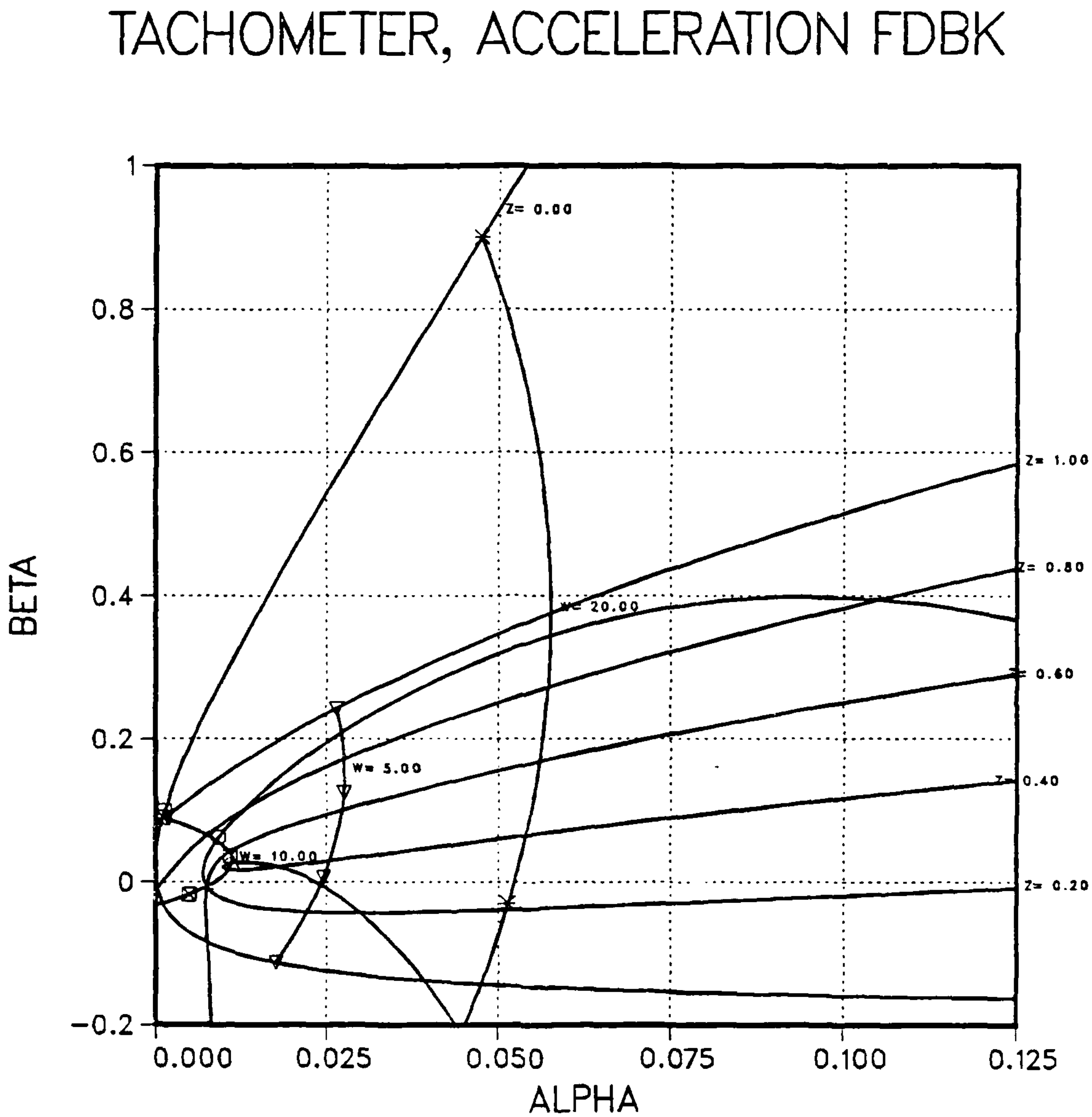


Figure 5-14
Parameter Plane Curves for
$S^4 + 15S^3 + (150+5555.5\alpha)S^2 + (1000+5555.5\beta)S + 5555.5 = 0$

as zeta increases. Since the β axis is always in the unstable region, it is concluded that tachometer feedback alone cannot stabilize the system.

The effect of acceleration feedback alone can be observed by traveling along the α-axis of Figure 5-14. If K_a is varied between 0.01 and 0.06, the system will exhibit two pairs of complex roots with the following ranges of values for zeta:

$$0.3<\xi<0.5 \text{ and } 0.25<\xi<0.32$$

If both tachometer and acceleration feedback are used, it is seen that tachometer feedback will in general cause the zeta of one pair of roots to increase while the zeta of the other pair decreases. Acceleration feedback alone would appear to be the better choice.

From the set of curves it is determined that with $K_t=0$ and $K_a=0.012$, four complex roots are located with associated values at $\xi=0.45$, $\omega=4.0$, and $\xi=0.32$, $\omega=13.0$. Since $(0.45)(4)=1.8<<(0.32)(13)=4.15$, it is apparent that the roots at $\xi=0.45$ and $\omega=4.0$ are dominant. The specifications have been met and the problem is solved.

B. MISCELLANEOUS ASPECTS OF THE PARAMETER PLANE

It can be demonstrated that constant zeta parameter plane curves of order two through five originate at a point where alpha $= \frac{M}{K}$ and beta $= \frac{N}{K}$, where M, N, and K are determined by the zero and first power coefficients only. Intuition can be used to conclude that constant zeta curves of any order originate at this common point which is determined only by the zero and first power coefficients. An exception is when K=0. In this case the origin point depends on higher order coefficients and its location will be obvious given a specific problem. If K is not zero, the origin is independent of the order of the characteristic equation.

Inspection of the expressions for alpha and beta indicates that the shape of the constant zeta curves as omega becomes larger is primarily determined by the coefficients of higher power, and in general the curves become more complex and less well behaved as the order of the characteristic equation increases. For a given characteristic equation, an increase in complexity can be observed as alpha and beta appear in more coefficients.

All constant zeta curves tend to plus or minus infinity. The relative magnitudes of the coefficients determine whether the limit is plus or minus infinity. It is therefore necessary to choose a frequency range of interest before plotting the curves, thus limiting the analysis to one "window" of the infinite plane.

Since no stability criteria, either relative or absolute, has been established for the parameter plane, it is necessary to base the stability analysis on observing which way the curves tend as omega and zeta are varied. For this reason it is worthwhile to plot curves for as many values of zeta, omega, sigma, and if desired, zeta-omega, as are necessary to ascertain the pattern.

VI. CONCLUSIONS

Parameter plane techniques have been applied to the compensation of linear control systems. General equations have been derived for the cases of feedback, cascade, and combination feedback-cascade compensation, to enable one to place a pair of complex conjugate roots at a specific location in the S-plane, while simultaneously satisfying the steady-state accuracy requirements. A dominancy technique has been introduced whereby once a pair of complex roots is fixed, the remaining roots of the characteristic equation can be manipulated to ensure that the specified roots are dominant.

The impetus for development of a parameter plane program was to provide the user with a quick, simple means of obtaining the information available in the analytical solution to control system compensation, while avoiding the painstaking labor of trial-and-error analysis inherent in that technique. Several, practical engineering examples have been presented to demonstrate the superiority of the graphical technique. To date, no other package is known to offer the fully interactive and comprehensive capabilities of the parameter plane program.

By itself the program allows one to design a control system compensation model for most systems. However, for some lightly damped systems containing mechanical resonances, the amount by which zeta or omega are incremented in the parameter plane equations may be so large as to not detect the resonance peaks. This information would be available from either a root-locus or Bode analysis. For still other systems, one might be interested in the way the roots of the characteristic equation extend from the open loop poles and/or zeros. Since the parameter plane equations are calculated using only the characteristic equation, no knowledge of open loop poles or zeros is available. Again, a root-locus method would reveal this information. Incorporated into one comprehensive package which includes Bode and root-locus analyses, the program provides the capability to investigate the entire gamut of linear control system architecture.

A basis for further investigation involves plotting the parameter plane contours for systems that are non-linear in the alpha and beta terms--i.e., those systems which contain alpha-beta product terms. Although the recursion technique used in this text has distinct advantages over the matrix approach for the linear case, as pointed out earlier, the matrix technique would be the method of choice for the non-linear case.

APPENDIX A
FUNCTIONS $U_k(\xi)$

ξ	U_{-1}	U_0	U_1	U_2	U_3	U_4	U_5
0.00	-1	0	1	0.0	-1.00	0.000	1.0000
0.05	-1	0	1	0.1	-0.99	-0.199	0.9701
0.10	-1	0	1	0.2	-0.96	-0.392	0.8816
0.15	-1	0	1	0.3	-0.91	-0.573	0.7381
0.20	-1	0	1	0.4	-0.84	-0.736	0.5456
0.25	-1	0	1	0.5	-0.75	-0.875	0.3125
0.30	-1	0	1	0.6	-0.64	-0.984	0.0496
0.35	-1	0	1	0.7	-0.51	-1.057	-0.2299
0.40	-1	0	1	0.8	-0.36	-1.088	-0.5104
0.45	-1	0	1	0.9	-0.19	-1.071	-0.7739
0.50	-1	0	1	1.0	0.00	-1.000	-1.0000
0.55	-1	0	1	1.1	0.21	-0.869	-1.1659
0.60	-1	0	1	1.2	0.44	-0.672	-1.2464
0.65	-1	0	1	1.3	0.69	-0.403	-1.2139
0.70	-1	0	1	1.4	0.96	-0.056	-1.0384
0.75	-1	0	1	1.5	1.25	0.375	-0.6875
0.80	-1	0	1	1.6	1.56	0.896	-0.1264
0.85	-1	0	1	1.7	1.89	1.513	0.6821
0.90	-1	0	1	1.8	2.24	2.232	1.7776
0.95	-1	0	1	1.9	2.61	3.059	3.2021
1.00	-1	0	1	2.0	3.00	4.000	5.0000

APPENDIX B

PARAMETER PLANE PROGRAM

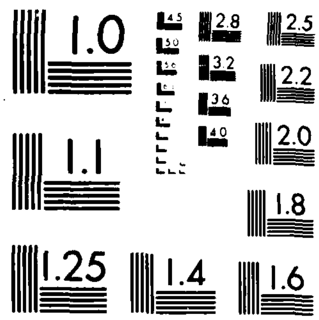
```
      SUBROUTINE LPARAM
      DIMENSION A(350), B(350),
     X     AG(350), BG(350), AJ(100), BJ(100), CJ(100),
     X     ZETA(100), SIGMA(100), W(100), ZW(100)
C
C                        INITIAL ASSIGNMENTS
      CHARACTER*4  SCHAR/'S '/,YES/'Y '/,NOO/'N '/,BLANK/'  '/,
     X     ENCHAR/'E '/,WNCHAR/'WN'/,NDCHAR/'ND'/,NOCHAR/'NO'/,
     X     AJCHAR/'AJ'/,BJCHAR/'BJ'/,CJCHAR/'CJ'/,NSCHAR/'NS'/,
     X     NZCHAR/'NZ'/,ZWCHAR/'ZW'/,NWCHAR/'NW'/,PRCHAR/'PR'/,
     X     NCCHAR/'NC'/,TICHAR/'TI'/
      CHARACTER*4  TABLE, GRAPH, CHANGE, REPLY, OPT, LABEL(9)
      COMMON /SAVE/ LABEL, WN, ND, NO2, NC, CJ, AJ, BJ,
     X              NZ, ZETA, NS, SIGMA, NW, W, NZW, ZW,
     X              XMIN, XMAX, YMIN, YMAX
C
C                              DATA ENTRY FROM FILE OR CONSOLE?
 100  MINMAX = 1
      GRD = 0.
      CHANGE = BLANK
      CALL EXCMS('CLRSCRN')
      WRITE(6,500)
      CALL READC (REPLY)
      IF (REPLY .NE. 'D') GOTO 101
           CALL GETIT
           MINMAX = 0
           GO TO 200
 101  CONTINUE
C
C                              GET CURVE TITLE
 102  CONTINUE
      CALL EXCMS('CLRSCRN')
      WRITE(6,501)
      CALL READL (LABEL)
      CALL ASTER (LABEL,LABEL)
      IF ( CHANGE .EQ. TICHAR ) GO TO 200
C
C                              GET STARTING VALUE OF WN
 103  CONTINUE
      WRITE(6,502)
      CALL READR (WN)
          IF (WN) 104,104,105
 104      WRITE(6,503)
          GO TO 103
 105  IF ( CHANGE .EQ. WNCHAR) GOTO 200
C
C                              GET THE NUMBER OF DECADES CONSIDERED
 106  CONTINUE
      WRITE(6,504)
      CALL READI (ND)
      IF ( CHANGE .EQ. NDCHAR ) GOTO 200
C
C                              GET THE ORDER OF THE CHARACTERISTIC EQN
 107  CONTINUE
      WRITE(6,505)
      CALL READI (NO2)
      NC = NO2+1
      IF (CHANGE .EQ. NOCHAR ) GOTO 200
C
C
```

```
C
C                              GET THE NUMBER OF CONSTANT ZETA CURVES
 108  CONTINUE
      CALL EXCMS ('CLRSCRN')
      WRITE(6,506)
      CALL READI (NZ)
      IF (NZ .LT. 1) GOTO 110
C
C                              GET THE VALUES OF ZETA
      WRITE(6,507)
      DO 110 I = 1,NZ
 109     WRITE(6,508)I
         CALL READR (ZETA(I))
         IF (ZETA(I) .LT. 0. .OR. ZETA(I) .GT. 1.) WRITE(6,509)
         IF (ZETA(I) .LT. 0. .OR. ZETA(I) .GT. 1.) GO TO 109
 110  CONTINUE
      IF ( CHANGE.EQ. NZCHAR ) GOTO 200
C
C                              GET THE NUMBER OF CONSTANT SIGMA CURVES
 111  CONTINUE
      CALL EXCMS ('CLRSCRN')
      WRITE(6,510)
      CALL READI (NS)
      IF (NS .LT. 1) GOTO 113
C
C                              GET THE VALUES OF SIGMA
      DO 113 I = 1,NS
 112    WRITE(6,511) I
        CALL READR (SIGMA(I))
        IF (SIGMA(I) .LT. 0.) WRITE (6,512)
        IF (SIGMA(I) .LT. 0.) GO TO 112
 113  CONTINUE
      IF ( CHANGE .EQ. NSCHAR ) GOTO 200
C
C                              GET THE NUMBER OF CONSTANT WN CURVES
 114  CONTINUE
      CALL EXCMS ('CLRSCRN')
      WRITE(6,513)
      CALL READI (NW)
      IF (NW .LT. 1) GOTO 116
C
C                              GET THE WN VALUES
      WNMAX = WN*10**ND
      DO 116 I = 1,NW
 115    WRITE(6,514) I
        CALL READR (W(I))
        IF (W(I) .LT. WN .OR. W(I) .GT. WNMAX) WRITE (6,515) WN,WNMAX
        IF (W(I) .LT. WN .OR. W(I) .GT. WNMAX) GO TO 115
 116  CONTINUE
      IF ( CHANGE .EQ. NWCHAR ) GOTO 200
C
C                              GET THE NUMBER OF CONSTANT ZETA*WN CURVES
 117  CONTINUE
      CALL EXCMS ('CLRSCRN')
      WRITE(6,516)
      CALL READI (NZW)
      IF (NZW .LT. 1) GOTO 119
C
C                              GET THE Z*WN VALUES
      DO 119 I = 1,NZW
 118    WRITE(6,517) I
        CALL READR (ZW(I))
        IF (ZW(I) .LE. 0.) WRITE (6,518)
        IF (ZW(I) .LE. 0.) GO TO 118
 119  CONTINUE
      IF ( CHANGE .EQ. ZWCHAR ) GOTO 200
C
C
C
C
```

```
C
C                                 GET CONSTANT COEFFICIENTS
 120   CONTINUE
      CALL EXCMS ('CLRSCRN')
          WRITE(6,519)
          DO 121 I = NC,1,-1
            II = I-1
            WRITE(6,520) II,I
            CALL READR (CJ(I))
 121      CONTINUE
          IF ( CHANGE .EQ. CJCHAR ) GOTO 200
C
C                                 GET ALPHA COEFFICIENTS
 122      CONTINUE
      CALL EXCMS ('CLRSCRN')
          WRITE(6,521)
          DO 123 I = NC,1,-1
            II = I-1
            WRITE(6,522) II,I
            CALL READR (AJ(I))
 123      CONTINUE
          IF (CHANGE .EQ. AJCHAR ) GOTO 200
C
C                                 GET BETA COEFFICIENTS
 124      CONTINUE
      CALL EXCMS ('CLRSCRN')
          WRITE(6,523)
          DO 125 I = NC,1,-1
            II = I-1
            WRITE(6,524) II,I
            CALL READR (BJ(I))
 125      CONTINUE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
```

AD-A173 956 FEEDBACK CONTROL ANALYSIS USING PARAMETER PLANE TECHNIQUES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA D M POTTER JUN 86 2/2

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

```
C
C                                    REVIEW ENTRIES
 200      CONTINUE
        CALL EXCMS ('CLRSCRN')
          WRITE(6,525)
          CALL READC (REPLY)
          IF (REPLY .NE. YES) GOTO 209
        CALL EXCMS ('CLRSCRN')
        WRITE(6,526)
        WRITE(6,527) (LABEL(I), I=1, 9)
        WRITE(6,528)
        WRITE(6,529) ND,NO2,NZ,NS,NW,NZW
        WRITE(6,530) WN
        WRITE(6,531)
        IF (NZ .LT. 1) GOTO 201
          WRITE(6,532) (ZETA(M),M=1,NZ)
          GOTO 202
 201      WRITE(6,533)
 202    CONTINUE
        WRITE(6,534)
        IF (NS .LT. 1) GOTO 203
          WRITE(6,532) (SIGMA(M),M=1,NS)
          GOTO 204
 203      WRITE(6,533)
 204    CONTINUE
        WRITE(6,535)
        IF (NW .LT. 1) GOTO 205
          WRITE(6,532) (W(M),M=1,NW)
          GOTO 206
 205      WRITE(6,533)
 206    CONTINUE
        WRITE(6,536)
        IF (NZW .LT. 1) GOTO 207
          WRITE(6,532) (ZW(M),M=1,NZW)
          GOTO 208
 207      WRITE(6,533)
 208    CONTINUE
        WRITE (6,576)
        CALL EXCMS('CLRSCRN')
        WRITE(6,537)
        WRITE(6,532) (CJ(N),N=NC,1,-1)
        WRITE(6,538)
        WRITE(6,532) (AJ(N),N=NC,1,-1)
        WRITE(6,539)
        WRITE(6,532) (BJ(N),N=NC,1,-1)
        WRITE(6,540)
        WRITE(6,541) XMIN, XMAX, YMIN, YMAX
 209    CONTINUE
        WRITE(6,542)
        CALL READC (CHANGE)
        IF (CHANGE .NE. YES) GOTO 210
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
```

```
C
C                                   CHANGE ROUTINE
C
C
      CALL EXCMS ('CLRSCRN')
      WRITE(6,543)
      CALL READC (CHANGE)
      IF (CHANGE .EQ. TICHAR) GOTO 102
      IF (CHANGE .EQ. WNCHAR) GOTO 103
      IF (CHANGE .EQ. NDCHAR) GOTO 106
      IF (CHANGE .EQ. NOCHAR) GOTO 107
      IF (CHANGE .EQ. NZCHAR) GOTO 108
      IF (CHANGE .EQ. NSCHAR) GOTO 111
      IF (CHANGE .EQ. NWCHAR) GOTO 114
      IF (CHANGE .EQ. ZWCHAR) GOTO 117
      IF (CHANGE .EQ. CJCHAR) GOTO 120
      IF (CHANGE .EQ. AJCHAR) GOTO 122
      IF (CHANGE .EQ. BJCHAR) GOTO 124
      IF (CHANGE .EQ. ENCHAR) GOTO 405
      IF (CHANGE .EQ. PRCHAR) GOTO 100
      IF (CHANGE .EQ. NCCHAR) GOTO 210
      GOTO 209
 210  CONTINUE
      WRITE(6,544)
      CALL READC (TABLE)
      WRITE(6,545)
      CALL READC (GRAPH)
C
 211    IF (NZ) 213,213,212
 212    G = (10.**ND)**(1./299.)
 213    CONTINUE
        Z = 1.0E-60
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
```

```
C
C                               CONSTANT ZETA PLOTS
      IF (NZ) 309,309,300
 300    CALL EXCMS ('CLRSCRN')
        IF (TABLE .EQ. YES .OR. GRAPH .EQ. YES) WRITE (6,546)
C
C
            DO 308 M=1,NZ
              IF (TABLE .EQ. YES) WRITE (6,547)
              J = 0
              R = 0.
              WNA = WN
C
              DO 306 L=1,300
                D1 = 0.0
                D2 = 0.0
                C1 = 0.0
                C2 = 0.0
                B1 = 0.0
                B2 = 0.0
C
                DO 303 N=1,NC
                  K = N-1
                  IF (K) 302,301,302
 301                U = 0.0
                    U1 = -1.0
 302                U2 = 2.0*ZETA(M)*U-U1
                    D1 = (-1.0)**K*CJ(N)*WNA**K*U1+D1
                    D2 = (-1.0)**K*CJ(N)*WNA**K*U+D2
                    C1 = (-1.0)**K*BJ(N)*WNA**K*U1+C1
                    C2 = (-1.0)**K*BJ(N)*WNA**K*U+C2
                    B1 = (-1.0)**K*AJ(N)*WNA**K*U1+B1
                    B2 = (-1.0)**K*AJ(N)*WNA**K*U+B2
                    U1 = U
                    U = U2
 303            CONTINUE
C
                IF (ABS(B1*C2-B2*C1)-Z) 306,306,304
 304              J = J+1
                  R = R+1.
                  A(J) = (C1*D2-C2*D1)/(B1*C2-B2*C1)
                  B(J) = (B2*D1-B1*D2)/(B1*C2-B2*C1)
C
                IF (TABLE .NE. YES) GO TO 306
                  WRITE(6,548) A(J), B(J), WNA, ZETA(M)
                  IF (R/10. - J/10) 306,305,306
 305              CALL ROOTS (A(J), B(J), AJ, BJ, CJ, NO2)
                  CALL EXCMS('CLRSCRN')
                  WRITE (6,547)
 306          WNA = G*WNA
C
                  CALL EXCMS('CLRSCRN')
C
              IF (J .GT. 0) GOTO 307
                WRITE(6,549)
                GOTO 308
C
 307          IF (GRAPH .EQ. YES) CALL PLOTD(A,B,J,.FALSE.,
     X                                 LABEL, 'ALPHA$','BETA$',
     X                                 MINMAX,' Z=$',ZETA(M),
     X                                 XMIN,XMAX,YMIN,YMAX,GRD)
C
              IF (GRAPH .EQ. YES) GRD = GRD+1.
C
 308        CONTINUE
C
 309  CONTINUE
C
C
C
```

```
C
C                               CONSTANT SIGMA PLOTS
      IF (NS) 335,335,325
 325  CALL EXCMS('CLRSCRN')
          IF (TABLE .EQ. YES .OR. GRAPH .EQ. YES) WRITE (6,550)
C
          XINC = (XMAX - XMIN)/299.
          YINC = (YMAX - YMIN)/299.
C
            DO 334 M=1,NS
              XPT = XMIN
              YPT = YMIN
              DD = CJ(1)
              CC = BJ(1)
              BB = AJ(1)
              J = 0
              R = 0.
C
              DO 326 N=2,NC
                K = N-1
                DUMMY5=SIGMA(M)**K
                DD = (-1.0)**K*CJ(N)*DUMMY5+DD
                CC = (-1.0)**K*BJ(N)*DUMMY5+CC
                BB = (-1.0)**K*AJ(N)*DUMMY5+BB
 326          CONTINUE
C
              IF (CC .EQ. 0. .AND. BB .EQ. 0.) GOTO 334
              IF (CC) 327,327,330
 327          DO 329 L=1,300
                J = J+1
                R = R+1.
                A(J) = XPT
                B(J) = (-BB*A(J)-DD)/CC
               IF (TABLE .NE. YES) GOTO 329
                WRITE(6,552) A(J), B(J), SIGMA(M)
                IF (R/10. - J/10) 329,328,329
 328            CALL ROOTS (A(J), B(J), AJ, BJ, CJ, NO2)
                CALL EXCMS ('CLRSCRN')
                WRITE(6,550)
 329          XPT = XPT + XINC
              GO TO 333
C
 330          DO 332 L=1,300
                J = J+1
                R = R+1.
                B(J) = YPT
                A(J) = (-CC*B(J)-DD)/BB
               IF (TABLE .NE. YES) GOTO 332
                WRITE(6,552) A(J), B(J), SIGMA(M)
                IF (R/10. - J/10) 332,331,332
 331            CALL ROOTS (A(J), B(J), AJ, BJ, CJ, NO2)
                CALL EXCMS ('CLRSCRN')
                WRITE(6,550)
 332          YPT = YPT + XINC
C
 333          CALL EXCMS('CLRSCRN')
              IF (GRAPH .EQ. YES) CALL PLOTD(A,B,J,.FALSE.,
     X                                  LABEL, 'ALPHA$','BETA$',
     X                                  MINMAX,' S=$',SIGMA(M),
     X                                  XMIN,XMAX,YMIN,YMAX,GRD)
C
              IF (GRAPH .EQ. YES) GRD = GRD+1.
C
 334        CONTINUE
C
 335    CONTINUE
C
C
C
C
```

```
C
C                              CONSTANT ZETA-OMEGA PLOTS
         IF (NZW) 359,359,350
 350     CALL EXCMS('CLRSCRN')
         IF (TABLE .EQ. YES .OR. GRAPH .EQ. YES) WRITE (6,553)
C
         XWN = WN
C
           DO 358 M=1,NZW
C
             IF (TABLE .EQ. YES) WRITE (6,554)
             J = 0
             R = 0.
             AZETA = 1./300.
C
             DO 356 L=1,300
               XWN = ZW(M)/AZETA
               D1 = 0.0
               D2 = 0.0
               C1 = 0.0
               C2 = 0.0
               B1 = 0.0
               B2 = 0.0
C
               DO 353 N=1,NC
                 K = N-1
                 IF (K) 352,351,352
 351               Q1 = 0.0
                   Q = -1.0/XWN**2
 352             D2 = CJ(N)*Q1+D2
                 C2 = BJ(N)*Q1+C2
                 B2 = AJ(N)*Q1+B2
                 D1 = CJ(N)*Q+D1
                 C1 = BJ(N)*Q+C1
                 B1 = AJ(N)*Q+B1
                 Q2 = -2.0*ZW(M)*Q1-XWN**2*Q
                 Q = Q1
 353             Q1 = Q2
C
               IF (ABS(B1*C2-B2*C1)-Z) 356,356,354
 354           J = J+1
               R = R+1.
               A(J) = (C1*D2-C2*D1)/(B1*C2-B2*C1)
               B(J) = (B2*D1-B1*D2)/(B1*C2-B2*C1)
C
               IF (TABLE .NE. YES) GO TO 356
               WRITE(6,552) A(J), B(J), ZW(M)
               IF (R/10. - J/10) 356,355,356
 355           CALL ROOTS (A(J), B(J), AJ, BJ, CJ, NO2)
               CALL EXCMS('CLRSCRN')
               WRITE (6,554)
 356       AZETA = AZETA+(1./300.)
C
               CALL EXCMS('CLRSCRN')
C
             IF (J .GT. 0) GOTO 357
               WRITE(6,549)
               GOTO 358
C
 357           IF (GRAPH .EQ. YES) CALL PLOTD(A,B,J,.FALSE.,
     X                                        LABEL, 'ALPHA$','BETA$',
     X                                        MINMAX,'ZW=$',ZW(M),
     X                                        XMIN,XMAX,YMIN,YMAX,GRD)
C
               IF (GRAPH .EQ. YES) GRD = GRD+1.
C
 358       CONTINUE
C
 359     CONTINUE
C
```

```
C
C                                  CONSTANT OMEGA PLOTS
 375       IF (NW) 385,385,376
 376       CALL EXCMS('CLRSCRN')
           IF (TABLE .EQ. YES .OR. GRAPH .EQ. YES) WRITE (6,555)
C
C
               DO 384 M=1,NW
                 IF (TABLE .EQ. YES) WRITE (6,547)
                 J = 0
                 R = 0.
                 AZETA = 0.0
C
                 DO 382 L=1,300
                   D1 = 0.0
                   D2 = 0.0
                   C1 = 0.0
                   C2 = 0.0
                   B1 = 0.0
                   B2 = 0.0
C
                   DO 379 N=1,NC
                     K = N-1
                     IF (K) 378,377,378
 377                   U = 0.0
                       U1 = -1.0
 378                 U2 = 2.0*AZETA*U-U1
                     D1 = (-1.0)**K*CJ(N)*W(M)**K*U1+D1
                     D2 = (-1.0)**K*CJ(N)*W(M)**K*U+D2
                     C1 = (-1.0)**K*BJ(N)*W(M)**K*U1+C1
                     C2 = (-1.0)**K*BJ(N)*W(M)**K*U+C2
                     B1 = (-1.0)**K*AJ(N)*W(M)**K*U1+B1
                     B2 = (-1.0)**K*AJ(N)*W(M)**K*U+B2
                     U1 = U
                     U = U2
 379             CONTINUE
C
                     IF (ABS(B1*C2-B2*C1)-Z) 382,382,380
 380                 J = J+1
                     R = R+1.
                     A(J) = (C1*D2-C2*D1)/(B1*C2-B2*C1)
                     B(J) = (B2*D1-B1*D2)/(B1*C2-B2*C1)
C
                 IF (TABLE .NE. YES) GO TO 382
                 WRITE(6,548) A(J), B(J), W(M), AZETA
                 IF (R/10. - J/10) 382,381,382
 381             CALL ROOTS (A(J), B(J), AJ, BJ, CJ, NO2)
                 CALL EXCMS('CLRSCRN')
                 WRITE (6,547)
 382             AZETA = AZETA+(1./299.)
                 CALL EXCMS('CLRSCRN')
C
                 IF (J .GT. 0) GOTO 383
                   WRITE(6,549)
                   GOTO 384
C
 383             IF (GRAPH .EQ. YES) CALL PLOTD(A,B,J,.FALSE.,
     X                                          LABEL, 'ALPHA$','BETA$',
     X                                          MINMAX,' W=$',W(M),
     X                                          XMIN,XMAX,YMIN,YMAX,GRD)
C
 384         CONTINUE
C
 385         CONTINUE
C
C
C
C
C
C
```

```
C
                IF (GRAPH .EQ. YES) CALL PLOTD(0.,0.,0.,.TRUE.,
     X                                         ' $',' $',' $',
     X                                         0.,' $',-9.7531,
     X                                         0.,0.,0.,0.,GRD)
                IF (GRAPH .EQ. YES) GRD = GRD+1.
      GRD = 0.
 400  CALL EXCMS('CLRSCRN')
      IF (OPT .EQ. YES) CALL DONEPL
      IF (OPT .EQ. YES) STOP
      WRITE(6,557)
      WRITE(6,556)
      WRITE(6,557)
      WRITE(6,558)
      WRITE(6,559)
      WRITE(6,560)
      WRITE(6,561)
      WRITE(6,562)
      WRITE(6,563)
      WRITE(6,557)
      CALL READI (IANS)
         IF (IANS .GT. 6 .OR. IANS .LT. 1) GOTO 400
         GO TO (100, 404, 401, 402, 403, 405) IANS
C
C                            ROOT FINDER OPTION
 401  CALL EXCMS('CLRSCRN')
      WRITE(6,564)
      WRITE(6,565)
      CALL READR (ALPHA)
      WRITE(6,566)
      CALL READR (BETA)
      CALL ROOTS (ALPHA, BETA, AJ, BJ, CJ, NO2)
      GO TO 400
C
C                            SAVE OPTION
 402  CALL SAVIT
      GO TO 400
C
C                            CREATE DISSPLA METAFILE OPTION
 403  CALL EXCMS('CLRSCRN')
      WRITE(6,567)
      WRITE(6,568)
      WRITE(6,569)
      CALL READC (OPT)
      IF (OPT .NE. YES) GOTO 400
            CALL EXCMS('CLRSCRN')
            WRITE(6,570)
            CALL DONEPL
C
C                            COMPRS = SUBROUTINE TO LET DONEPL FINISH
            CALL META
            GO TO 211
C
C                            SAME PROBLEM OPTION
 404  WRITE(6,571)
      CALL READI (MINMAX)
      IF (MINMAX .EQ. 1) GO TO 200
      CALL EXCMS('CLRSCRN')
           WRITE(6,572)
           CALL READR (XMIN)
           WRITE(6,573)
           CALL READR (XMAX)
           WRITE(6,574)
           CALL READR (YMIN)
           WRITE(6,575)
           CALL READR (YMAX)
      GO TO 200
 405  CONTINUE
      RETURN
C
```

```
C
C
C
C
C
C
C
C
C
C                                    FORMATS STATEMENTS
 500  FORMAT(5X,'THIS IS THE INTERACTIVE PARAMETER PLANE PROGRAM...',/,
     X        5X,'THE USER WILL BE PROMPTED FOR VARIOUS INPUTS.',////,
     X        5X,'WILL YOU BE ENTERING DATA FROM A CONSOLE OR DATAFILE?',
     X        /,15X,'"D"  OR  "C"')
 501  FORMAT(///,' ENTER TITLE TO APPEAR FOR THIS FAMILY OF CURVES')
 502  FORMAT(/,1X,'WHAT IS THE STARTING VALUE OF OMEGAN (WN>0.0)?')
 503  FORMAT(/,1X,'WN MUST BE GREATER THAN ZERO - TRY AGAIN')
 504  FORMAT(/,1X,'HOW MANY DECADES PAST WN ARE DESIRED? (ND)')
 505  FORMAT(/,' WHAT IS THE ORDER OF THE CHARACTERISTIC EQUATION?(NO)')
 506  FORMAT(/,1X,'HOW MANY CONSTANT ZETA CURVES ARE DESIRED? (NZ)')
 507  FORMAT(/,' ENTER THE VALUES OF ZETA TO BE USED IN COMPUTATION...')
 508  FORMAT(5X,'ZETA(',I2,') = ?')
 509  FORMAT(5X,'ZETA MUST LIE BETWEEN 0 AND 1, INCLUSIVE - TRY AGAIN')
 510  FORMAT(/,' HOW MANY CONSTANT SIGMA (REAL ROOT) CURVES ARE DESIRED?
     X (NS)')
 511  FORMAT(5X,'SIGMA(',I2,') = ?')
 512  FORMAT(5X,'NEGATIVE SIGMA MEANS POSITIVE REAL ROOT - TRY ANOTHER')
 513  FORMAT(/,1X,'HOW MANY CONSTANT WN CURVES ARE DESIRED? (NW)')
 514  FORMAT(5X,'W(',I2,') = ?')
 515  FORMAT (5X,'WN NOT WITHIN PLOTTABLE RANGE',
     X      /,5X,'YOUR USABLE RANGE IS'
     X      /,10X,F10.2,'  TO ',F10.2)
 516  FORMAT(/,1X,'HOW MANY CONSTANT Z*WN CURVES ARE DESIRED? (NZW)')
 517  FORMAT(5X,'ZW(',I2,') = ?')
 518  FORMAT(5X,'NON-POSITIVE Z-WN MEANS POSITIVE ROOT - TRY ANOTHER')
 519  FORMAT(/,1X,'ENTER THE CONSTANT COEFFICIENTS...')
 520  FORMAT(5X,'---S**',I2,'---  CJ(',I2,') = ?')
 521  FORMAT(/,1X,'ENTER THE ALPHA COEFFICIENTS...')
 522  FORMAT(5X,'---S**',I2,'---  AJ(',I2,') = ?')
 523  FORMAT(/,1X,'ENTER THE BETA COEFFICIENTS...')
 524  FORMAT(5X,'---S**',I2,'---  BJ(',I2,') = ?')
 525  FORMAT(/,' WANT TO REVIEW YOUR ENTRIES BEFORE RUNNING? (Y/N)')
 526  FORMAT (/,10X,' GRAPH TITLE')
 527  FORMAT (1X,9A4)
 528  FORMAT (/,8X,2HND,8X,2HNO,8X,2HNZ,8X,2HNS,8X,2HNW,7X,3HNZW)
 529  FORMAT (6I10)
 530  FORMAT (/,10X,'INITIAL VALUE OF OMEGA = ',F10.5)
 531  FORMAT (/,10X,' ZETA ')
 532  FORMAT (8E10.3)
 533  FORMAT (1X,'.....NO VALUE.....')
 534  FORMAT (/,10X,'SIGMA ')
 535  FORMAT (/,10X,' W    ')
 536  FORMAT (/,10X,'ZW    ')
 537  FORMAT (/,10X,'CONSTANT COEFFICIENTS IN DECENDING ORDER')
 538  FORMAT (/,10X,'ALPHA    COEFFICIENTS IN DECENDING ORDER')
 539  FORMAT (/,10X,'BETA     COEFFICIENTS IN DECENDING ORDER')
 540  FORMAT (/,10X,'XMIN   XMAX   YMIN   YMAX')
 541  FORMAT (1X,4E10.3)
 542  FORMAT(/,' WANT TO MAKE ANY CHANGES? (Y/N)')
 543  FORMAT(/,' WHAT VARIABLE/AREA DO YOU WISH TO CHANGE?',//,
     X5X,'TITLE.......TI     OMEGA START..WN     # DECADES....ND',/
C
C
C
C
C
C
C
C
C
```

```
C
C
C
C
C
C
C
C
C
    X5X,'ORDER........NO     CONST ZETA...NZ     CONST SIGMA..NS',/
    X5X,'CONST WN.....NW     CONST Z-WN...ZW',//
    X5X,'CONST TERMS..CJ     ALPHA TERMS..AJ     BETA TERMS...BJ',/
    X5X,'END..........E      NEW PROBLEM..PR     NO CHANGE....NC',///,
    X1X,'ENTER TWO DIGIT CODE...')
544  FORMAT(/,' DO YOU WANT TABULATED DATA ON THE SCREEN? (Y/N)')
545  FORMAT(/,' DO YOU WANT THE GRAPHS ON THE TERMINAL? (Y/N)')
546  FORMAT (1H1,10X,'CONSTANT ZETA CURVES')
547  FORMAT(/,10X,'ALPHA            BETA           OMEGA            ZETA')
548  FORMAT (4E16.5)
549  FORMAT(/,' DUE TO PLOT RESTRICTIONS, A COMPLETE GRAPH CANNOT BE OU
    XTPUT.')
550  FORMAT (1H1,10X,'CONSTANT SIGMA CURVES')
551  FORMAT (/,10X,'ALPHA            BETA            SIGMA')
552  FORMAT (3E16.5)
553  FORMAT (1H1,10X,'CONSTANT ZETA-OMEGA CURVES')
554  FORMAT (/,10X,'ALPHA            BETA        ZETA-OMEGA')
555  FORMAT (1H1,10X,'CONSTANT OMEGA CURVES')
556  FORMAT(5X,'| OPTION NO.|                    OPTION                 |')
557  FORMAT(5X,'-------------------------------------------------------|')
558  FORMAT(5X,'|      1     |         NEW PROBLEM                      |')
559  FORMAT(5X,'|      2     |         SAME PROBLEM                     |')
560  FORMAT(5X,'|      3     |         ROOT FINDER                      |')
561  FORMAT(5X,'|      4     |         SAVE DATA                        |')
562  FORMAT(5X,'|      5     |         SAVE GRAPH IN DISSPLA METAFILE   |')
563  FORMAT(5X,'|      6     |         RETURN TO MAIN MENU              |')
564  FORMAT(5X,'ENTER AN ALPHA-BETA PAIR, AND THE ROOTS OF YOUR     '/,
    X       5X,'SYSTEMS CHARACTERISTIC EQUATION WILL BE RETURNED     ')
565  FORMAT(///,5X,'ENTER THE ALPHA VALUE                                ')
566  FORMAT(/,5X,'ENTER THE BETA VALUE                                  ')
567  FORMAT(5X,'YOU NOW HAVE THE OPTION OF STORING THE LAST SET OF  '/,
    X       5X,'CURVES IN A DISSPLA METAFILE.  THIS ALLOWS RETRIEVAL'/,
    X       5X,'OF DATA AT A LATER TIME FOR ROUTING TO ANY OF       '/,
    X       5X,'SEVERAL OUTPUT DEVICES (TEK618, 3800 LASER PRINTER, '/,
    X       5X,'VERSATEC PLOTTER, ETC.)                            '//)
568  FORMAT(5X,'IF YOU CHOOSE THIS OPTION, THE PROGRAM MUST BE      '/,
    X       5X,'TERMINATED - THIS CANNOT BE AVOIDED WITHOUT         '/,
    X       5X,'CATASTROPHIC RESULTS.                             '////)
569  FORMAT(///,5X,'DO YOU WISH TO USE THIS OPTION?                  '/,
    X       15X,'"Y"  OR  "N"                                       '/)
570  FORMAT(/////,5X,'IF YOU WISH GRAPHIC OUTPUT, TYPE:               '/,
    X       15X,'"DISSPOP"                                          '/
    X       5X,'AND FOLLOW THE INSTRUCTIONS . . .                   ')
571  FORMAT (///,' AUTOSCALE OR USER-DEFINED LIMITS FOR CURVES?',
    X/,' 1=AUTOSCALE; 0=USER-DEFINED')
572  FORMAT (//,' INPUT MINIMUM VALUE FOR X (X-MIN)')
573  FORMAT (/,' INPUT MAXIMUM VALUE FOR X (X-MAX)')
574  FORMAT (/,' INPUT MINIMUM VALUE FOR Y (Y-MIN)')
575  FORMAT (/,' INPUT MAXIMUM VALUE FOR Y (Y-MAX)')
576  FORMAT (/////////////////)
     END
C
C
C
C
C
C
C
C
C
```

```
C
C ===============================================================
C  SUBROUTINE PLOTD -- GRAPHS WILL BE PRODUCED ON UPRIGHT 11 X 14    =
C  PAGE WITH 9 INCH AXES.  IF USER SELECTS 'AUTOSCALE' FEATURE,      =
C  SUBROUTINE PLD010 (INTERNAL TO PLOTD) FINDS MIN AND MAX FOR EACH  =
C  AXIS AND SCALES ACCORDINGLY.  FORMAT:                             =
C                                                                    =
C    CALL PLOTD(XDATA, YDATA, NNPTS, EJECT, LABEL, XLABEL, YLABEL,   =
C   X            MINMAX, CRVTTL, CRVNUM, XMIN, XMAX, YMIN, YMAX)     =
C                                                                    =
C      WHERE:                                                        =
C                                                                    =
C      XDATA      IS A REAL*4 ARRAY DIMENSIONED AT LEAST |NNPTS|     =
C                 CONTAINING THE X ORDINATE VALUES,                  =
C                                                                    =
C      YDATA      IS A REAL*4 ARRAY DIMENSIONED AT LEAST |NNPTS|     =
C                 CONTAINING THE Y ORDINATE VALUES,                  =
C                                                                    =
C      NNPTS      IS AN  INTEGER*4 SCALAR DESIGNATING THE  NUMBER OF =
C                 POINTS TO BE PLOTTED.  THE NUMBER OF POINTS IS     =
C                 ABS(NNPTS).  NNPTS<0 MEANS PLOT POINTS ONLY.       =
C                                                                    =
C      EJECT      IS  A LOGICAL*4  VARIABLE  OR CONSTANT  INDICATING =
C                 WHETHER A   PAGE EJECT  IS REQUIRED  FOLLOWING THE =
C                 CURRENT CURVE.  THIS ALLOWS MULTIPLE CURVES ON ONE =
C                 SET OF EXES.  PAGE EJECT WILL OCCUR FOR NEXT GRAPH =
C                 AFTER EJECT HAS BEEN SET TO .TRUE.                 =
C                                                                    =
C      LABEL      IS A QUOTED  LITERAL OR HOLLERITH STRING  OR ARRAY =
C                 CONTAINING THE INTENDED LABEL FOR THE GRAPH.   THE =
C                 MAXIMUM ALLOWABLE LENGTH (INCLUDING '$' CHARACTER) =
C                 IS 32 CHARACTERS.                                  =
C                                                                    =
C      XLABEL     IS A QUOTED  LITERAL OR HOLLERITH STRING  OR ARRAY =
C                 CONTAINING THE INTENDED LABEL OF THE X-AXIS OF THE =
C                 GRAPH.  IN THIS PROGRAM, XLABEL IS ALWAYS 'ALPHA'. =
C                                                                    =
C      YLABEL     IS A QUOTED  LITERAL OR HOLLERITH STRING  OR ARRAY =
C                 CONTAINING THE INTENDED LABEL OF THE Y-AXIS OF THE =
C                 GRAPH.  IN THIS PROGRAM, YLABEL IS ALWAYS 'BETA'.  =
C                                                                    =
C      MINMAX     IS A PARAMETER THAT DETERMINES WHETHER THE MINIMUM =
C                 AND MAXIMUM VALUES FOR THE AXES ARE TO BE ASSIGNED =
C                 BY THE USER, OR WHETHER THEY WILL BE 'AUTOSCALED'. =
C                                                                    =
C      CRVTTL     IS A QUOTED  LITERAL OR HOLLERITH STRING  OR ARRAY =
C                 AND TERMINATED BY  A  '$' CHARACTER SPECIFYING THE =
C                 INTENDED NAME WHICH LABELS AN INDIVIDUAL CURVE.    =
C                                                                    =
C      CRVNUM     IS A REAL VARIABLE OR CONSTANT THAT SPECIFIES  THE =
C                 VALUE TO BE CONCATENATED ONTO THE END OF 'CRVTTL'. =
C                 FOR EXAMPLE, IF THIS CURVE REPRESENTS 'ZETA = 0.5' =
C                 THEN CRVTTL = 'Z=$', WHILE CRVNUM = 0.5.           =
C                                                                    =
C ===============================================================
C
C
      SUBROUTINE PLOTD(XDATA, YDATA, NNPTS, EJECT, LABEL, XLABEL,
     X               YLABEL, MINMAX, CRVTTL, CRVNUM,
     X               XMIN, XMAX, YMIN, YMAX,GRD)
      REAL*4        XDATA(1), YDATA(1)
      REAL*4        XMIN, XMAX, YMIN, YMAX
      REAL*4        CRVTTL,CRVNUM
      INTEGER*4     NPTS,NNPTS
      LOGICAL*4     EJECT
      LOGICAL*1     LABEL(1)
      LOGICAL*1     XLABEL(1)
      LOGICAL*1     YLABEL(1)
C
      LOGICAL*4     INIT /.FALSE./
```

```
C
C
C
C     SET THE NUMBER OF POINTS
C
      NPTS = IABS(NNPTS)
C
C        IF THE ROUTINE HAS BEEN NOT BEEN INITIALIZED (INIT = .FALSE.)
C        THEN INITIALIZE IT.
C
      IF (.NOT. INIT) CALL PLD001(XDATA, YDATA, NPTS, LABEL, XLABEL,
     X                            YLABEL, MINMAX, CRVTTL,CRVNUM,
     X                            XMIN, XMAX, YMIN, YMAX,GRD)
C
C     FRAME THE PLOT AND REORDER SYMBOLS
C
      IF (.NOT. INIT) CALL FRAME
C
C        INDICATE INITIALIZATION IN CASE THIS ROUTINE IS RE-ENTERED
C        WITH MULTIPLE CURVES.
C
      INIT = .TRUE.
C
C        CALCULATE THE NUMBER OF POINTS TO GIVE ABOUT 5 MARKERS PER
C        LINE
C
C     TENTATIVELY SET POINTS ONLY, THEN CHECK
      NMARK = -1
      IF(NNPTS.LE.0)GO TO 10
c
C     CURVE WANTED, SET MARKERS
      IF (MOD(NPTS, 4) .EQ. 1) NMARK = NPTS / 4
      IF (MOD(NPTS, 4) .NE. 1) NMARK = NPTS / 3
      IF (NMARK .EQ. 0) NMARK = 1
   10 CONTINUE
C
C
C        DRAW THE CURVE
C
      XXMAX = -1.0E75
      YYMAX = -1.0E75
C
      DO 20 I=1,NPTS
         IF(XDATA(I) .GT. XMAX .OR. XDATA(I) .LT. XMIN .OR.
     X      YDATA(I) .GT. YMAX .OR. YDATA(I) .LT. YMIN) GO TO 20
         IF(XXMAX .LT. XDATA(I)) J=I
         IF(XXMAX .LT. XDATA(I)) XXMAX = XDATA(I)
         IF(YYMAX .LT. YDATA(I)) K=I
         IF(YYMAX .LT. YDATA(I)) YYMAX = YDATA(I)
   20 CONTINUE
C
         IF((YMAX-YYMAX) - (XMAX-XXMAX)) 40,30,30
   30    L=J
         GO TO 50
   40    L=K
c
   50 CALL GRACE(0.)
      CALL DOT
      IF (GRD) 60,60,70
   60 CALL GRID(1,1)
   70 CALL RESET('DASH')
          IF (CRVNUM .EQ. -9.7531) GO TO 80
      CALL HEIGHT(0.125)
      CALL RLMESS(CRVTTL,3,XDATA(L),YDATA(L))
      CALL RLREAL(CRVNUM,2,'ABUT','ABUT')
      CALL THKCRV(0.015)
   80 CALL CURVE(XDATA, YDATA, NPTS, NMARK)
      CALL RESET('THKCRV')
C
C        IF THIS IS NOT THE LAST (OR ONLY) CURVE ON THIS GRAPH, THEN
```

```
C        EXIT.  OTHERWISE CLOSE THE PLOT AND TURN OFF INITIALIZATION
C        FLAG.
C
      IF (.NOT. EJECT) RETURN
C
C        END OF THIS PLOT
C
      CALL ENDPL(0)
      INIT = .FALSE.
      RETURN
      END
C
C
      SUBROUTINE PLD001(XDATA, YDATA, NPTS, LABEL, XLABEL,
     X                  YLABEL, MINMAX, CRVTTL, CRVNUM,
     X                  XMIN, XMAX, YMIN, YMAX, GRD)
C
C        THIS SUBROUTINE DOES THE INITIALIZING.
C
      REAL*4        XDATA(NPTS)
      REAL*4        YDATA(NPTS)
      INTEGER*4     NPTS
      LOGICAL*1     LABEL(1)
      LOGICAL*1     XLABEL(1)
      LOGICAL*1     YLABEL(1)
C
C
      REAL*4        XMIN
      REAL*4        XMAX
      REAL*4        YMIN
      REAL*4        YMAX
C
C        INITIALIZE DISSPLA
C
      CALL PLD009
      CALL HEADIN(LABEL, 100, 2., 1)
      CALL XNAME(XLABEL, 100)
      CALL YNAME(YLABEL, 100)
C
C
C        EXTRACT MINIMA AND MAXIMA
C
      IF (MINMAX .NE. 1) GO TO 90
      CALL PLD010(XDATA, NPTS, XMIN, XMAX)
      CALL PLD010(YDATA, NPTS, YMIN, YMAX)
C
C        CALL THE LINEAR-LINEAR INITIALIZING ROUTINE
C
   90 CALL PLD011 (XMIN,XMAX,YMIN,YMAX)
      RETURN
      END
C
C
      SUBROUTINE PLD009
C
C        THIS SUBROUTINE ESTABLISHES THE PARAMETERS FOR DISSPLA.
C
C        NOTE THAT IT IS THE USER'S RESPONSIBILITY TO NOMINATE THE
C        GRAPHIC DEVICE.
C
C
      CALL NOCHEK
      CALL NOBRDR
      CALL PAGE(14.,14.)
      CALL PHYSOR(2.,.75)
C
C        GO TO LEVEL 2.
C
      CALL AREA2D(9.,9.)
C
```

```
C        LETTERING IS DUPLX WITH UPPER CASE ONLY.
C
      CALL DUPLX
      CALL BASALF('STANDARD')
C
C        INTEGER (OR ROUNDED) AXES WITH Y AXIS LABELLING AT 0 DEGREES.
C
      CALL INTAXS
      CALL YAXANG(0.)
      RETURN
      END
C
C
      SUBROUTINE PLD010(V, N, MIN, MAX)
C
C        THIS SUBROUTINE SCANS A VECTOR FOR MAXIMUM AND MINIMUM
C
C        INPUT PARAMETERS:
C
C             V          DATA VECTOR (REAL)
C             N          NUMBER OF POINTS IN VECTOR V (INTEGER)
C
C        OUTPUT PARAMETERS:
C
C             MIN        VECTOR ORIGIN (REAL)
C             MAX        VECTOR MAXIMUM (REAL)
C
      REAL*4         V(N)
      INTEGER*4      N
      REAL*4         MIN
      REAL*4         MAX
C
C        INITIALIZE THE MAXIMA AND MINIMA
C
      MIN = 1.0E75
      MAX = -1.0E75
C
C        FIND MAXIMUM AND MINIMUM OF VECTOR V
C
      DO 100 I = 1, N
      IF (MIN .GT. V(I)) MIN = V(I)
      IF (MAX .LT. V(I)) MAX = V(I)
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE PLD011 (XMIN,XMAX,YMIN,YMAX)
C
C        THIS SUBROUTINE SETS UP DISSPLA FOR A LINEAR-LINEAR AXIS PLOT.
C
      REAL*4         XMIN
      REAL*4         XMAX
      REAL*4         YMIN
      REAL*4         YMAX
C
C        A SIMPLE CALL TO GRAF WILL DO IT...
C
      CALL HEIGHT(0.175)
      CALL GRAF(XMIN, 'SCALE', XMAX, YMIN, 'SCALE', YMAX)
      RETURN
      END
C
C
C
C
C
C
C
C
```

```
C
C ==================================================================
C  SUBROUTINE ROOTS -- CALCULATES ROOTS OF THE NO ORDER EQUATION    =
C  FOR EVERY TENTH VALUE ALPHA/BETA PAIR.                           =
C ==================================================================
C
C
      SUBROUTINE ROOTS (ALPHA, BETA, AJ, BJ, CJ, NO2)
      INTEGER*4 NO2, NN, NNN, NNNN
      REAL*4 ALPHA, BETA, AJ(100), BJ(100), CJ(100)
      REAL*8 COEF(100), ROOTMY(100)
      WRITE(6,40)
      NN = NO2+1
      NNNN = NO2
      DO 10 I=1,NN
      NNN = NN+1-I
   10 COEF(NNN) = CJ(I) + (ALPHA * AJ(I)) + (BETA * BJ(I))
      CALL ZPOLR (COEF,NO2,ROOTMY,IER)
      WRITE (6,50)
      DO 20 I=1,NNNN
      II = 2*I
      III = II-1
   20 WRITE (6,60) ROOTMY(III), ROOTMY(II)
      WRITE (6,30)
   30 FORMAT(//////////////////////)
   40 FORMAT(/20X,' ROOTS FOR ABOVE ALPHA, BETA')
   50 FORMAT(21X,' REAL PART       IMAG PART')
   60 FORMAT(21X,E10.4,6X,E10.4)
      RETURN
      END
C
C
C ==================================================================
C  SUBROUTINE SAVIT -- SAVES DATA IN FN FT FM = INAME DATA A1,      =
C  WHERE INAME IS THE USER'S CHOICE.                                =
C ==================================================================
C
C
      SUBROUTINE SAVIT
      COMMON /SAVE/ LABEL, WN, ND, NO2, NC, CJ, AJ, BJ,
     X               NZ, ZETA, NS, SIGMA, NW, W, NZW, ZW,
     X               XMIN, XMAX, YMIN, YMAX
      REAL*4 WN, CJ(100), AJ(100), BJ(100), XMIN, XMAX, YMIN, YMAX
      REAL*4 ZETA(100), SIGMA(100), W(100), ZW(100)
      INTEGER ND, NO2, NC, NZ, NS, NW, NZW
      CHARACTER*4 LABEL(9), INAME(2)
      WRITE(6,10)
      READ(5,20) (INAME(I), I=1,2)
      CALL FRTCMS('FILEDEF ','02      ','DISK    ',INAME,'DATA    ')
      WRITE(2,30) (LABEL(I), I=1, 9)
      WRITE(2,*) WN
      WRITE(2,*) ND, NO2, NC, NZ, NS, NW, NZW
      WRITE(2,*) (CJ(J), J=1, NC)
      WRITE(2,*) (AJ(J), J=1, NC)
      WRITE(2,*) (BJ(J), J=1, NC)
      WRITE(2,*) (ZETA(M), M=1, NZ)
      WRITE(2,*) (SIGMA(M), M=1, NS)
      WRITE(2,*) (W(M), M=1, NW)
      WRITE(2,*) (ZW(M), M=1, NZW)
      WRITE(2,*) XMIN, XMAX, YMIN, YMAX
   10 FORMAT(5X,'UNDER WHAT NAME DO YOU WANT TO SAVE THE DATA?',/,
     X        5X,'(8 CHARACTERS MAX)')
   20 FORMAT(2A4)
   30 FORMAT(9A4)
      END FILE 02
      REWIND 02
      RETURN
      END
C
C
```

```
C
C ==================================================================
C  SUBROUTINE GETIT -- RETRIEVES DATA FROM FN FT FM = INAME DATA A1, =
C  WHERE INAME IS THE USER'S CHOICE.                                  =
C ==================================================================
C
C
      SUBROUTINE GETIT
      COMMON /SAVE/ LABEL, WN, ND, NO2, NC, CJ, AJ, BJ,
     X              NZ, ZETA, NS, SIGMA, NW, W, NZW, ZW,
     X              XMIN, XMAX, YMIN, YMAX
      REAL*4 WN, CJ(100), AJ(100), BJ(100), XMIN, XMAX, YMIN, YMAX
      REAL*4 ZETA(100), SIGMA(100), W(100), ZW(100)
      INTEGER ND, NO2, NC, NZ, NS, NW, NZW
      CHARACTER*4 LABEL(9)
      CHARACTER*8 INAME
      CHARACTER*21 NAME
   10 WRITE(6,40)
      READ(5,50) INAME
      NAME = 'STATE '//INAME//' DATA *'
      CALL EXCMS(NAME,RC)
         IF (RC .EQ. 0) GOTO 20
         WRITE(6,30)
         GOTO 10
   20 CALL FRTCMS('FILEDEF ','02      ','DISK    ',INAME,'DATA    ')
      READ(2,60) (LABEL(I), I=1, 9)
      READ(2,*) WN
      READ(2,*) ND, NO2, NC, NZ, NS, NW, NZW
      READ(2,*) (CJ(J), J=1, NC)
      READ(2,*) (AJ(J), J=1, NC)
      READ(2,*) (BJ(J), J=1, NC)
      READ(2,*) (ZETA(M), M=1, NZ)
      READ(2,*) (SIGMA(M), M=1, NS)
      READ(2,*) (W(M), M=1, NW)
      READ(2,*) (ZW(M), M=1, NZW)
      READ(2,*) XMIN, XMAX, YMIN, YMAX
   30 FORMAT(5X,'DATA FILE NOT FOUND - TRY ANOTHER')
   40 FORMAT(5X,'UNDER WHAT NAME IS THE DATAFILE SAVED?       ',/,
     X       5X,'(8 CHARACTERS MAX)')
   50 FORMAT(1A8)
   60 FORMAT(9A4)
      END FILE 02
      REWIND 02
      RETURN
      END
C
C
C ==================================================================
C SUBROUTINE ASTER PLACES A $ AT THE END OF A CHARACTER STRING      =
C ==================================================================
C
C
      SUBROUTINE ASTER(LLINES,LINE)
      CHARACTER*4 DOLLAR/'$   '/, LLINES(8), LINE(9)
      DO 10 I=1,8
         LINE(I)=LLINES(I)
 10   CONTINUE
      LINE(9)=DOLLAR
      RETURN
      END
C
C
C
C
C
C
C
C
C
```

```
C
C ======================================================================
C  SUBROUTINE READR -- INTERACTIVELY READS A REAL NUMBER REPLY.         =
C  IF THE USER INADVERTENTLY ENTERS  A NULL STRING                      =
C  A WARNING IS ISSUED AND ONE RECOVERY IS ALLOWED.                     =
C ======================================================================
C
C
C
      SUBROUTINE READR (ANSR)
      REAL*4 ANSR
      INTEGER COUNT
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      READ (5,*,END=30,ERR=30) ANSR
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
50    FORMAT (1X,' WARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUMER
     XICAL VALUE.')
60    FORMAT (///,5X,' PROGRAM TERMINATION - TWO NULL STRINGS ENTERED!')
      END
C
C
C ======================================================================
C  SUBROUTINE READI -- INTERACTIVELY READS AN INTEGER REPLY.            =
C  IF THE USER INADVERTENTLY ENTERS A NULL STRING OR NEGATIVE VALUE     =
C  A WARNING IS ISSUED AND ONE RECOVERY IS ALLOWED.                     =
C ======================================================================
C
C
      SUBROUTINE READI (IANS)
      INTEGER COUNT,IANS
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,70)
      GO TO 50
20    CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 40,30,30
30    CONTINUE
      RETURN
40    REWIND 5
      WRITE (5,60)
      GO TO 10
50    CONTINUE
      STOP
60    FORMAT (1X,' WARNING: IMPROPER DATA ENTRY!  ENTER A POSITIVE INTEG
     XER.')
70    FORMAT (///,5X,' PROGRAM TERMINATION - 2 IMPROPER DATA ENTRIES!')
      END
C
C
C
C
C
C
c
c
c
```

```
C
C ==================================================================
C  SUBROUTINE READC -- INTERACTIVELY READS A CHAR STRING REPLY.     =
C  ('YES' OR 'NO'). IF THE USER INADVERTENTLY ENTERS A NULL STRING  =
C  A WARNING IS ISSUED AND ONE RECOVERY IS ALLOWED.                 =
C ==================================================================
C
C
      SUBROUTINE READC (CANS)
      INTEGER COUNT
      CHARACTER*4 CANS
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) CANS
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
50    FORMAT (1X,' WARNING:  NULL STRINGS ARE NOT ALLOWED ')
60    FORMAT (///,5X,' PROGRAM TERMINATION - TWO NULL STRINGS ENTERED!')
70    FORMAT (A2)
      END
C
C
C ==================================================================
C  SUBROUTINE READL -- INTERACTIVELY READS A STRING OF CHARACTERS.   =
C  IF THE USER INADVERTENTLY ENTERS A NULL STRING                   =
C  A WARNING IS ISSUED AND ONE RECOVERY IS ALLOWED.                 =
C ==================================================================
C
C
      SUBROUTINE READL(LLINES)
      INTEGER COUNT, I, NIX
      CHARACTER*4 BBLANK/'    '/, LLINES(8)
      DO 10 I=1,8
         LLINES(I) = BBLANK
   10 CONTINUE
      COUNT=0
   20 COUNT=COUNT+1
      IF(COUNT.LT.3) GO TO 30
         WRITE(6,70)
      GO TO 50
   30 CONTINUE
      REWIND 5
      READ(5,80,END=40,ERR=30)(LLINES(J),J=1,9)
      RETURN
   40 REWIND 5
      WRITE(6,60)
      GO TO 20
   50 CONTINUE
      STOP
   60 FORMAT(1X,' WARNING: NULL STRINGS ARE NOT ALLOWED, ENTER CHARACTER
     X VALUES.')
   70 FORMAT (///,5X,' PROGRAM TERMINATION - TWO NULL STRINGS ENTERED!')
   80 FORMAT(9A4)
      END
C
C
C
C
C
C
```

```
C
C ====================================================================
C  SUBROUTINE META -- BY CALLING COMPRS AS A SUBROUTINE HERE,         =
C  DONEPL HAS SUFFICIENT TIME TO FINISH; OTHERWISE COMPRS IGNORED     =
C  AND METAFILE NOT SAVED                                             =
C ====================================================================
C
C
      SUBROUTINE META
      DO 10 I=1,900000
   10 CONTINUE
      CALL COMPRS
      RETURN
      END
```

LIST OF REFERENCES

1. Mitrovic, D., "Graphical Analysis and Synthesis of Feedback Control Systems", IEEE, Theory and Analysis, II Synthesis, AIEE Transactions, Part II, January 1959.

2. Choe, H.H., Some Extensions of Mitrovic's Method in Analysis and Design of Feedback Control Systems, M.S. Thesis, Naval Postgraduate School, 1964.

3. Hyon, C.H., A Direct Method of Designing Linear Feedback Control Systems by Applying Mitrovic's Method, M.S. Thesis, Naval Postgraduate School, 1964.

4. Nutting, R.M., Parameter Plane Techniques For Feedback Control Systems, M.S. Thesis, Naval Postgraduate School, 1965.

5. Siljak, D.D., "Analysis and Synthesis of Feedback Control Systems in the Parameter Plane", Part I, Linear Continuous Systems, IEEE Transactions, 4 November 1964.

6. Thaler, G.J. and Towill, D.R., Parameter Space Methods for Dynamic Systems, Unpublished notes.

7. Karmarkar, J. and Thaler, G.J., "A Matrix Approach to Parameter Analysis of Dynamical Systems", Proceedings of the 8th Annual International Symposium of Space Technology and Science, 1969.

8. Desai, R.C. and Yadav, R.A., "Determination of Parameters of Alternator Voltage Regulator for Steady-State Stability Using Siljak's Extension to Mitrovic's Method", International Journal of Control, 1971, Vol. 13, No. 1, pp. 27-32.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Professor G.J. Thaler, Code 62Tr Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	5
4.	Professor J.H. Duffin, Code 62Dn Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor H. Rigas, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor H. Titus, Code 62Ts Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
7.	Professor A. Gerba, Jr., Code 62Gz Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
8.	Professor D.E. Kirk, Code 62Ki Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1

9. Professor R.D. Strum, Code 62St	1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

10. Lieutenant D.M. Potter, USN	1
305 Ferndale Drive
Syracuse, New York 13205

END

12-86

DTIC