MODIFIED OBSERVER SYSTEMS WITH REDUCED SENSITIVITY USING STATE VARIABLE FEEDBACK

Prepared Under Grant NGL-03-002-006
National Aeronautics and Space Administration
(Donald G. Schultz, Project Director)

by

Theodore L. Williams

Electrical Engineering Department
The University of Arizona
Tucson, Arizona

July 1969

Engineering Experiment Station
The University of Arizona
College of Engineering
Tucson, Arizona

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Introduction

This report describes a method to design feedback control systems with both series and feedback compensators. When only the output state of a linear system is available, the use of both series and feedback compensation provides more flexibility in the design than using a single compensation network. The sensitivity of the closed-loop system to plant variations can be reduced below the values achieved by using only series compensation, or open loop compensation (no feedback). Comparisons of just series compensation with the combined series and feedback compensation are made in appendices 7 and 8.

This report begins with a description of the classical Guilleman-Truxal[1] design method and a brief description of the state variable feedback method of compensation[2]. More detail can be found in the references cited. The following section discusses the more general modified observer system and discusses the design procedure associated with modified observer systems. Several case studies and detailed descriptions of four numerical techniques are included as appendices.

Three digital computer programs (FORTRAN IV) are included to facilitate the required manipulations. These programs are augmented by two supplementary programs to modify the input data and the output data.

If the plant has N poles, then both the series compensator and the feedback compensator will have N-1 poles and N-1 zeros. Under these circumstances the plant transfer function and the desired closed loop transfer function are sufficient to specify all the parameters in both compensators except the pole positions of the feedback compensator. Thus, the user is free to select these pole positions to achieve the lowest sensitivity. The first program leaves this choice to the user. The second program selects the pole positions which minimizes the sensitivity to low frequency variations in the plant parameters. This program is restricted to those plants with no zeros. The third program evaluates the weighted integral sensitivity over all frequencies for a system with a specified compensation. The user is free to weight those frequencies where the disturbance is greatest. The fourth program uses the pole positions of the feedback compensator as input. The output specifies the complete compensation and the position of the poles of both the series and feedback compensators. It is useful for determining the stability of the compensators and checking the accuracy of the previous results.

Guilleman Truxal Design Procedure

Assume that the overall open loop gain function is given by G(s) as shown in Fig. 1. Then with unity feedback, the closed loop transfer function is given by

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)}$$

If G(s) is written as the ratio of two polynomials

$$G(s) = \frac{GN(s)}{GD(s)}$$

then

$$\frac{Y(s)}{R(s)} = \frac{GN(s)}{GD(s) + GN(s)}$$

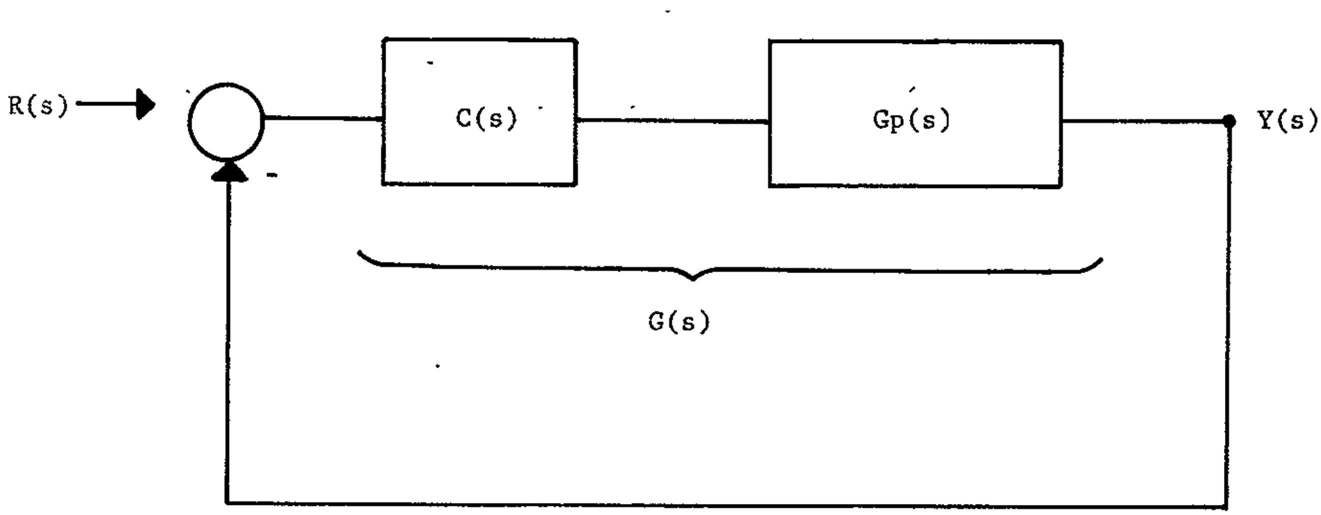


Figure 1. Guilleman Truxal Block Diagram

Equating the numerator and denominator polynomials

$$\begin{aligned} GN(s) &= C(s) \\ GD(s) &= R(s) - Y(s) \end{aligned} \tag{1}$$

The design procedure is as follows

1. From the specifications, choose the desired closed loop response, $Y(s)/R(s)$.
2. From step 1. and eqn. 1 determine $GN(s)$ and $GD(s)$.
3. Choose a suitable series compensator which shifts the poles and zeros of the plant to the positions specified by step 2.

The above steps are simplified if one uses Guilleman's suggestion to select a $C(s)/R(s)$ which results in real zeros for $GD(s)$. Then the zeros of $GD(s)$ can be simply determined from a plot of $R(s) - C(s)$ for only negative real values of s. One pair of complex conjugate poles and a group of poles on the negative real axis is usually chosen to meet the above requirements.

For example suppose the plant is given by

$$Gp(s) = \frac{1}{s(s+1)} \tag{2}$$

and that the specifications can be satisfied if

$$\frac{Y(s)}{R(s)} = \frac{100}{s^2 + 20s + 100}$$

From (1)

$$GN(s) = 100$$
$$GD(s) = s^2 + 20s \tag{3}$$

The series compensator C(s) and the plant transfer function must equal G(s),

$$C(s)\ Gp(s) = \frac{100}{s(s+20)}$$

From (2)

$$C(s) = \frac{100(s+1)}{(s+20)} \tag{4}$$

Effectively the series compensator moved the time constant of the motor from 1. to 1/20.

The success of this procedure depends upon the accurate modeling of the motor over the range of frequencies from zero to well above 20 radians/sec. Otherwise additional poles in the plant transfer function will alter the actual closed loop transfer function. In particular the real system may become unstable.

<u>State Variable Feedback Design Procedure</u>

This design procedure is similar in many respects to the above procedure. First a desired closed loop response is selected which satisfies the bandwidth, rise time, steady state error and other requirements. The plant is divided into a set of first order transfer functions which may be interconnected. The output of each of these first order systems is fed back to the input through a fixed gain element. By adjusting these gains, any desired closed loop transfer function can be achieved.

The above process is somewhat simplified if we first introduce a new function, Heq(s). This function specifies the transfer function between the output and the input which yields the same result as state variable feedback. From Fig. 2 the equivalent feedback from the output is

$$Heq(s) = k_1 + \frac{k_2}{G_1(s)} + \frac{k_3}{G_1(s)G_2(s)} \cdots$$

In terms of Heq(s) the state variable feedback results in the closed loop response given by

$$\frac{Y(s)}{R(s)} = \frac{KGp(s)}{1 + KGp(s)\ Heq(s)} \tag{5}$$

The procedure is as follows:

1. Select the desired Y(s)/R(s) from the specifications.
2. Equate the desired Y(s)/R(s) to the one achieved by the feedback from each state.
3. Solve for the feedback coefficients.

Details of the design procedure when zeros are present are given by Schultz and Melsa[2]. For example if

$$Gp(s) = \frac{1}{s(s+1)}$$

$$\frac{Y(s)}{R(s)} = \frac{100}{s^2 + 20s + 100}$$

Then if $G_1(s) = \frac{1}{s}$, $G_2(s) = \frac{1}{s+1}$

$$Heq(s) = k_1 + k_2 s$$

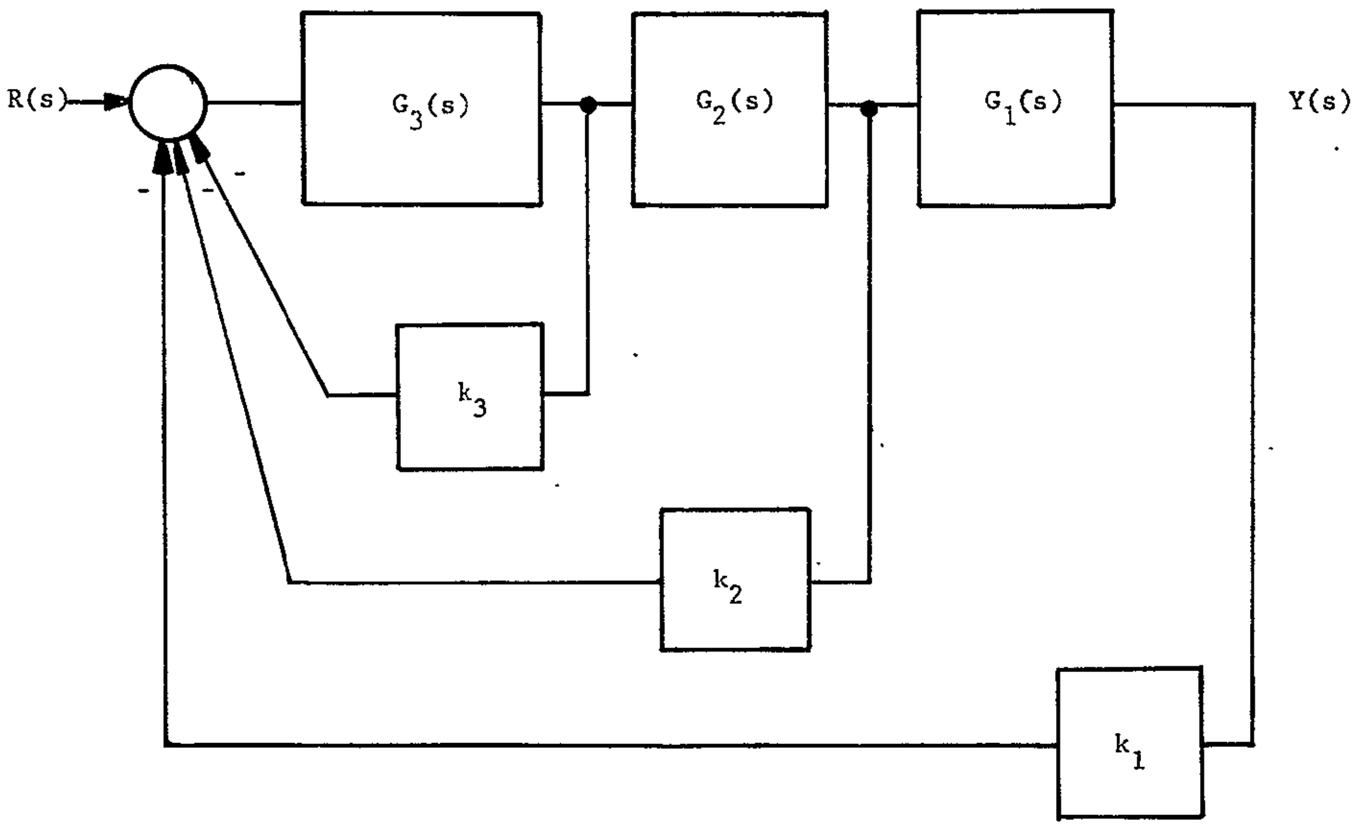


Figure 2. State Variable Feedback Block Diagram

From (5)

$$\frac{100}{s^2 + 20s + 100} = \frac{K}{s^2 + s + K(k_1 + k_2 s)}$$

Equating the numerator and denominator polynomials,

$$K = 100$$

$$Kk_1 = 100$$

$$1+Kk_2 = 20$$

Solving for k_1 and k_2,

$$K = 100$$

$$k_1 = 1$$

$$k_2 = .19$$

This design yields exactly the same transfer function as the Guilleman-Truxal (GT) method. The two methods are not the same if a small change is made in the plant transfer function. The state variable feedback (SVF) system changes less than the GT system.

If the sensitivity is defined as the percentage change in the closed loop transfer function for a small percentage change in a plant parameter, say K, then

$$S_T^K \equiv \frac{dT}{T} / \frac{dK}{K} = \frac{dT}{dK} \frac{K}{T}$$

where for simplicity

$$T(s) \equiv Y(s)/R(s)$$

If K is the gain then

$$T(s) = \frac{KG(s)}{1 + KG(s)\ Heq(s)}$$

$$\frac{dT(s)}{dK} = \frac{G(s)}{1 + KG(s)\ Heq(s)} - \frac{K\ Gp(s)\ G(s)\ Heq(s)}{(1 + KG(s)\ Heq(s))^2}$$

Using this result in the definition of sensitivity,

$$S_T^K = \frac{T(s)}{KG(s)} \tag{6}$$

For the GT system, from (6)

$$S_T^K = \frac{s(s+20)}{(s^2 + 20s + 100)}$$

For the SVF system, from (6)

$$S_T^K = \frac{s(s+1)}{s^2 + 20s + 100}$$

For the GT system the sensitivity is greater at all frequencies than the SVF sensitivity especially at low frequencies.

Modified Observer Systems

Luenberger[3] has shown that an estimate of the states in the plant can be implemented as shown in Fig. 3. When state variable feedback is used from these states a specified closed loop response may be achieved. Muterspaugh[4] has modified this configuration by block diagram manipulations to a form similar to that shown in Fig. 5. The

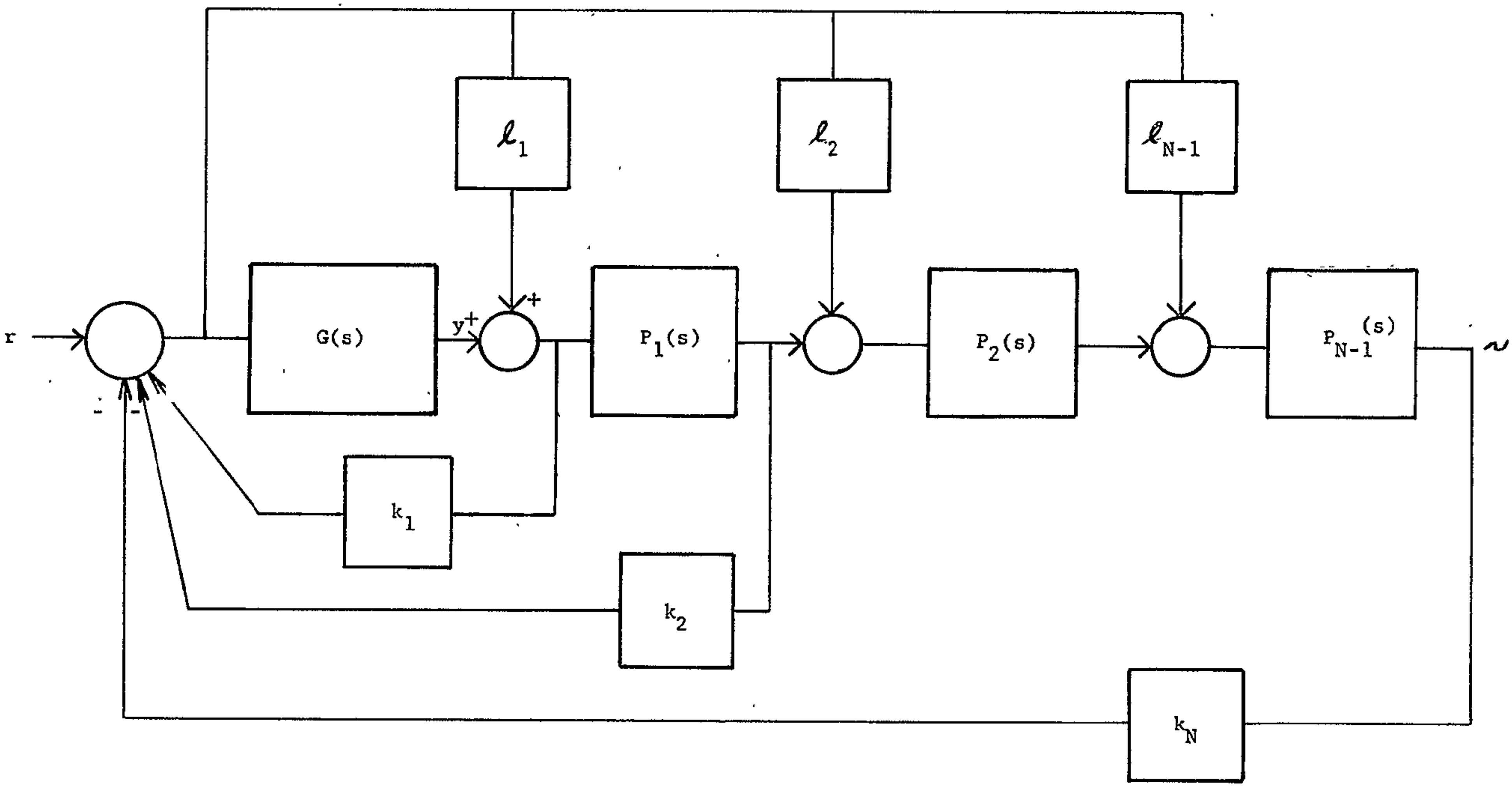


Figure 3. Luenburger Observer Block Diagram

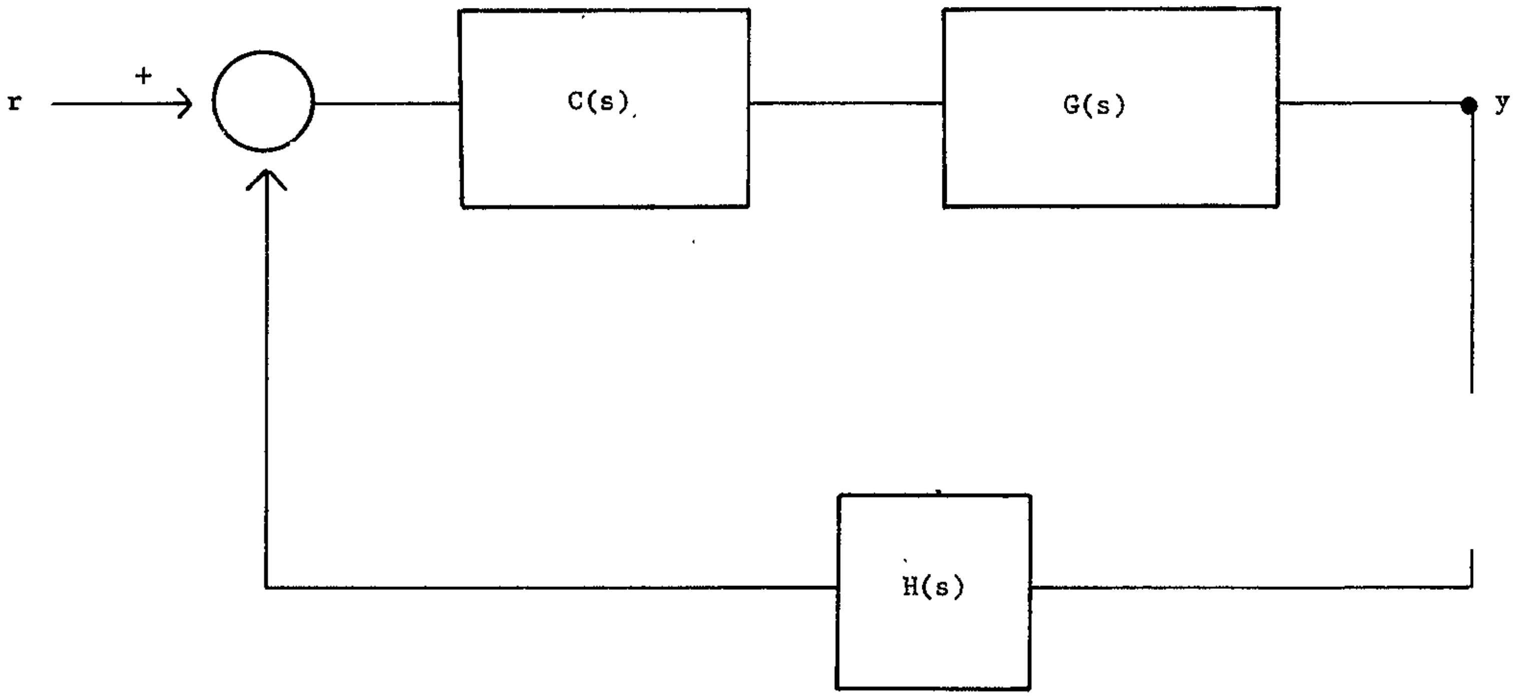


Figure 4. Modified Observer Block Diagram

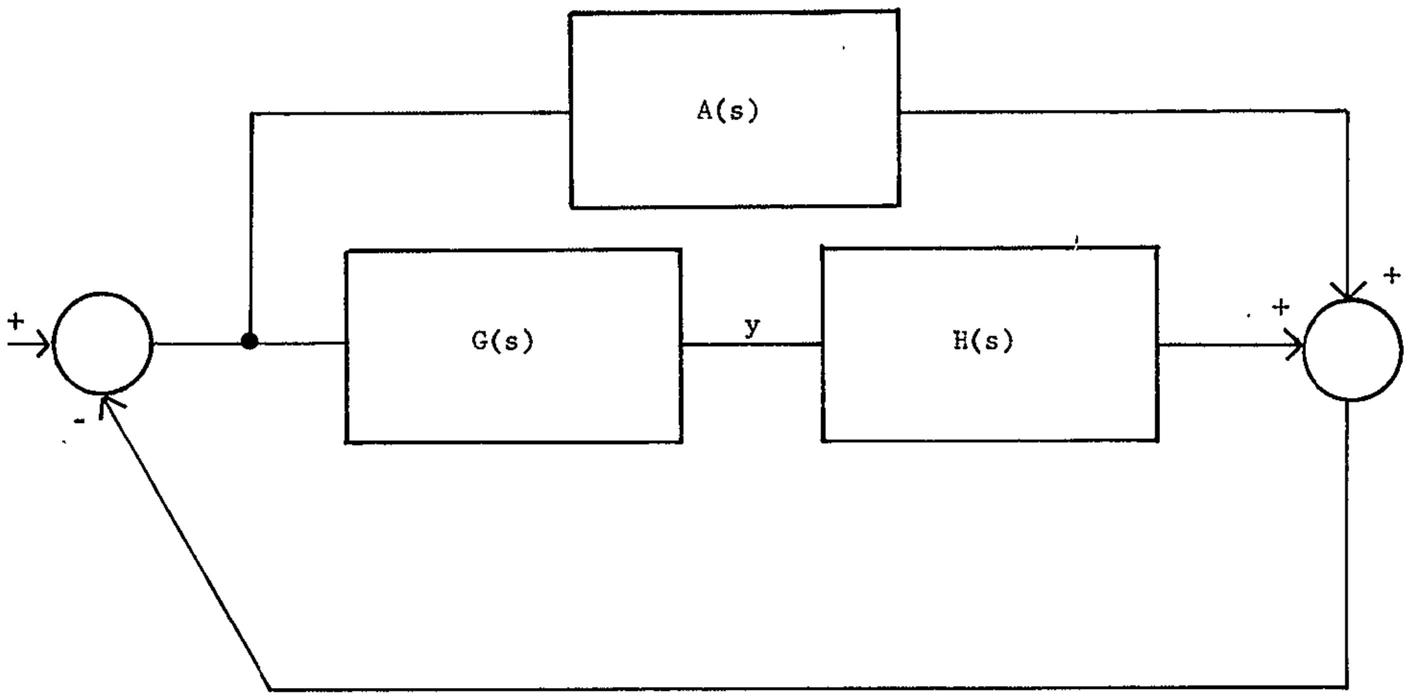


Figure 5. Muterspaugh's Modified Observer Block Diagram

series compensator shown in Fig. 4, C(s), reduces to the Guilleman-Truxal[1] compensation in the case where H(s) is one. Similarly this control system reduces to the Heq[2] type of control resulting from state variable feedback considerations when C(s) is one. Fig. 4 represents a more general configuration than either Guilleman-Truxal or Heq type compensation. All three configurations shown in the above figures are equivalent.

Assuming that only the output is available, the Heq configuration is not realizable since Heq has zeros but no poles. Guilleman-Truxal type compensation is sensitive to plant variations. Series and feedback compensation is used to achieve low sensitivity and realizability. If Heq has zeros in the LHP then the "modified observer" (MO) system is slightly higher in sensitivity than Heq, but if Heq has zeros in the RHP, then the MO system is dramatically better. These comments are illustrated in the case studies at the end of this report.

Modified Observer System Design

This section derives the necessary conditions which C(s) and H(s) must satisfy such that the desired transfer function Y(s)/R(s) is achieved when the plant has a transfer function G(s). The relationship among these transfer functions is

$$T(s) = \frac{Y(s)}{R(s)} = \frac{C(s)\ G(s)}{1 + C(s)\ G(s)\ H(s)} \qquad (7)$$

If the feedback signals shown in Fig. 3 are combined (using block diagram manipulations), the resulting transfer functions are

$$A(s)=k_2\ell_1P_1(s)+k_2(\ell_1p_1(s)p_2(s)+\ell_2p_2(s))+\ldots+k_n(\ell_1p_1(s)+\ldots\ell_{N-1}p_{N-1}(s))$$

$$H(s)=k_1+k_2p_1+k_3p_1p_2+\ldots+k_Np_1\ldots P_{N-1}$$

Since the poles of H(s) and A(s) are arbitrary, the control systems shown in Fig. 5 have N-1 arbitrary parameters; however, the poles of A and H are identical.

One more manipulation of the system shown in Fig. 3 results in the system of Fig. 4 where

$$C(s) = \frac{1}{1 + A(s)}$$

The zeros of C(s) are the same as the poles of A(s); therefore the N-1 zeros of C(s) are also the same as the N-1 poles of H(s).

From these facts (7) can be used to determine the denominator of C(s) and the numerator of H(s). Dropping the argument s in (7) and denoting the transfer functions by

$$C(s) = \frac{CN}{CD}$$

$$H(s) = \frac{HN}{HD}$$

$$G(s) = \frac{GN}{GD}$$

$$T(s) = \frac{Y}{R}$$

Eqn. 7 becomes

$$\frac{Y}{R} = \frac{\frac{CN\ GN}{CD\ GD}}{1 + \frac{CN\ GN\ HN}{CD\ GD\ HD}} \tag{8}$$

If the right hand is simplified then

$$\frac{Y}{R} = \frac{CN\ GN\ HD}{CD\ GD\ HD + CN\ GN\ HN} \tag{9}$$

Since CN and HD are the same denote them both by P(s). Then (9) simplifies to

$$\frac{Y}{R} = \frac{GN\ P}{CD\ GD + GN\ HN} \tag{10}$$

cross multiplying on both sides results in the polynomial,

$$Y(CD \cdot GD + GN \cdot HN) = R \cdot GN \cdot P \tag{11}$$

Define the number of zeros in G as NG, the number of poles as ND, and similarly NY and NR for T(s). If the order of both sides of (11) is to be the same, then

$$\begin{aligned} NY + ND - 1 + ND &= NR + NG + ND - 1 \\ NY + ND &= NR + NG \\ NR - NY &= ND - NG \end{aligned} \tag{12}$$

Thus the pole zero excess of the plant determines the pole zero excess of the desired closed loop transfer function. If like powers of (11) are equated, then NY + 2ND equations result. Since CD and HN are undetermined the number of unknowns is 2ND. For NY = 0, the number of unknowns is equal to the number of equations. Just as in regular state variable feedback[2], for zeros in Y/R, additional states must be created. Let the order of P(s) by N-1, then (12) becomes

$$NY + N - 1 + ND = NR + NG + N - 1$$

and for the number equations to be equal to the unknowns,

$$\begin{aligned} NY + ND + N &= 2N \\ N &= NY + ND \end{aligned} \tag{13}$$

Thus, the order of both controller systems C(s) and H(s) must be one less than the number of poles of G plus the number of zeros of Y/R, assuming that (11) represents a set of linearly independent equations.

The design procedure is as follows:

1. Select the desired closed loop transfer function from the specifications.
2. Select a polynomial, P(s).
3. Equate the actual and desired transfer functions (10) and solve for the coefficients of CD(s) and HN(s).

Program 1

The first program (MODOBS1) computes the transfer function of the series compensator and the feedback compensator as a linear function of the denominator of the feedback compensator. The user must specify the plant transfer function and the desired closed loop transfer function. Thus, this program specifies the design of the system except for the poles of the feedback compensator which the user is free to choose in any way he wishes.

Program 2

The second program (MODOBS2) computes the transfer function of the series compensator and the feedback compensator which minimize the low frequency sensitivity of the closed loop system to plant variations. The user must specify the plant transfer function and the desired closed loop transfer function. If the time constants in the plant are much lower than in the closed loop system, the poles of the feedback compensator may be in the right half plane (RHP). If the poles of the compensator are in the RHP then the system will be very sensitive to variations in the parameters of the compensators. In this case, it is better to use one of the other programs. This program is limited to systems with no zeros.

Program 3

The third program (MODOBS3) computes the weighted integral sensitivity of the system for a given polynomial P(s). The user must specify the plant transfer function, closed loop transfer function, the weighting to be given the sensitivity as the frequency varies, and the denominator of the feedback transfer function, P(s). The weighting function must have more poles than zeros.

Input Data

In many circumstances the available data for the plant is in state variable form. This data is converted to a transfer function by the STVFDBK program. The details of the program are included in reference 1. The program listing and input data format are also included in the Appendix.

Output Data

The program, BODE4, is provided to plot the transfer functions computed in the above programs on a logarithmic scale. For each program some of the transfer functions of interest are:

1. Plant transfer function
2. Closed loop transfer function

3. Sensitivity as a function of frequency

4. Loop gain function

The loop gain function divided by one plus the loop gain is the proportion of measurement noise at each frequency which will reach the output. Thus, if the loop gain is much greater than one then nearly all the noise reaches the output, and if the loop gain is much less than one very little measurement noise reaches the output. If the phase shift of the loop gain is nearly 180° at a gain of one, the measurement noise will be amplified.

Program 4

The fourth program (MODOBS4) is useful to check the results of the design. Given P(s), and the results of the first program (MODOBS1) this program computes HN(s), CD(s) and the roots of both the series and feedback compensators. This program and the first program give a complete determination of the design.

Results

A discussion of each program and several case studies are included in the Appendices. MODOBS1 is useful to compute HN(s) and CD(s) in terms of the coefficients of P(s). Quite amazingly the coefficients of HN and CD are linearly dependent upon the coefficients of P. MODOBS2 computes the coefficients of both compensators which achieve minimum low frequency sensitivity. Essentially this result is achieved by forcing CD(s) to be an N-1 order integrator which increases the gain at low frequency, and thereby decreases the sensitivity since $S_K^T = \frac{T(s)}{G(s)C(s)}$. Unfortunately if the plant denominator coefficients are smaller than those of the denominator of the closed loop transfer function the feedback compensators may have poles in the RHP. MODOBS3 determines the weighted integral sensitivity when P(s) is specified. MODOBS4 is useful for checking the roots of the resulting compensators after a design is complete. Test results are given in the appendix for three examples. These examples show that an improvement in the system sensitivity over that of just a series compensator can be achieved even if only the output is available. When the closed loop response is much faster than the open loop, the sensitivity of the MODOBS system approaches the sensitivity of the Heq system (which is best but not realizable) if the roots of P(s) are made much larger than the closed loop roots. If the closed loop response is not much faster than the open loop response, then the MODOBS system may have a sensitivity lower than the Heq system. This situation exists whenever Heq(s) has zeros in the RHP. These results point out the important fact that a sacrifice in sensitivity to plant variations is made by increasing the system performance requirements. Because of this fact, careful attention to the plant performance should be made for lowest sensitivity and unnecessary demands on overall system performance should be avoided.

Bibliography

1. J. G. Truxal, Control System Synthesis, McGraw-Hill, 1955, Chapter 5.

2. J. L. Melsa and D. G. Schultz, Linear Control Systems, McGraw-Hill, 1969, Chapter 9.

3. D. G. Luenberger, "Observing the State of a Linear System," IEEE Transactions on Mil. Elec., April, 1964.

4. M. E. Muterspaugh, "State Variable Feedback and Unavailable States," M.S. Thesis, 1968.

5. F. H. Effertz, "Comment on Two Coupled Matrix Algorithms for the Evaluation of the r.m.s. Error Criterion of Linear Systems," Proceedings of the IEEE, November, 1966, p. 1629.

6. J. L. Melsa, "A Digital Computer Program for the Analysis and Synthesis of State Variable Feedback Systems," NASA Contract Report CR-850, August, 1967.

7. L. E. Weaver, et al, "A Control System Study for an In-Core Thermionic Reactor," J. P. L. Report 32-1355, January, 1969.

Appendix I - MODOBS1

Discussion

The basic algorithm of this program is based upon (5). First the following polynomials in s are computed:

$$T = (GD)(Y)$$

$$Q = (GN)(R)$$

$$V = (GN)(Y)$$

so that (5) becomes

$$(T)(CD) + (V)(HN) = (Q)(P) \tag{A1}$$

Each of the terms in (A1) are polynomials in s where P(s) is unspecified. For instance, $P(s) = P_3 s^2 + P_1$. The coefficients of CD(s) and HN(s) are the unknowns.

As an example, suppose the plant is second order, then P(s) is first order. In this case (A1) becomes,

$$(CD_2T_3 + H_2V_3)s^3 + (CD_2T_2 + CD_1T_3 + H_2V_2 + H_1V_3)s^2 + (CD_2T_1 + CD_1T_2$$
$$+ H_2V_1 + H_1V_2)s + (CD_1T_1 + H_1V_1) = P_2Q_3s^2 + (P_2Q_2 + P_1Q_3)s^2$$
$$+ (P_2Q_1 + P_1Q_2)s + P_1Q_1 \tag{A2}$$

Equating coefficients of like powers of s results in the equations

$$T_3CD_2 + V_3H_2 = Q_3P_2$$

$$T_2CD_2 + T_3CD_1 + V_2H_2 + V_3H_1 = Q_2P_2 + Q_3P_1$$

$$T_1CD_2 + T_2CD_1 + V_1H_2 + V_2H_1 = Q_1P_2 + Q_2P_1$$

$$T_1CD_1 + V_1H_1 = Q_1P_1 \qquad \text{(A3)}$$

These equations can be rewritten more compactly in matrix notation as

$$\begin{array}{cc} 2N-2 & \begin{bmatrix} T_3 & 0 & V_3 & 0 \\ T_2 & T_3 & V_2 & V_3 \\ T_1 & T_2 & V_1 & V_2 \\ 0 & T_1 & 0 & V_1 \end{bmatrix} \\ & \longleftarrow 2N-2 \longrightarrow \end{array} \begin{bmatrix} CD_2 \\ CD_1 \\ H_2 \\ H_1 \end{bmatrix} = \begin{array}{cc} 2N-2 & \begin{bmatrix} 0 & Q_3 \\ Q_3 & Q_2 \\ Q_2 & Q_1 \\ Q_1 & 0 \end{bmatrix} \\ & \longleftarrow N-1 \longrightarrow \end{array} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \qquad \text{(A4)}$$

The matrix on the left will be defined as matrix A, and the matrix on the right will be defined as matrix C. Then

$$\begin{bmatrix} CD_2 \\ CD_1 \\ H_2 \\ H_1 \end{bmatrix} = A^{-1}C \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \qquad \text{(A5)}$$

Therefore $A^{-1}C$ relates the unknowns CD_i and H_i to the parameters P_i. MODOBS1 forms the matrix A from the two arrays, T and V, and then inverts A. Next the C matrix is formed from the Q array and then $A^{-1}C$ is computed and printed.

Input Format

Card No.	Columns	Description	Format
1	1	NY = order of Y(s) in Y/R	I1
	2	NR = order of R(s) in Y/R	I1
	3	NG = order of GN(s) in GN/GD	I1
	4	ND = order of GD(s) in GN/GD	I1
	5-80	Identification of problem	8A10
2	1-10	Coefficient of s^0 in Y(s)	F10.3
	11-20	Coefficient of s^1 in Y(s)	F10.3
	etc.		
	71-80	Coefficient of s^7 in Y(s)	F10.3
3	1-10 etc.	Coefficient of s^0 in R(s)	F10.3
4	1-10 etc.	Coefficient of s^0 in GN(s)	F10.3
5	etc.	Coefficient of s^0 in GD(s)	F10.3

Notes:

1. If more than eight coefficients are required then two cards may be used to identify that polynomial.
2. A decimal point must be used for each coefficient, but not for card No. 1. The decimal point may be placed in any of the ten columns for each coefficient.

3. As many problems may be run as desired by repeating the above format for each problem.

<u>Output Format</u>

See example.

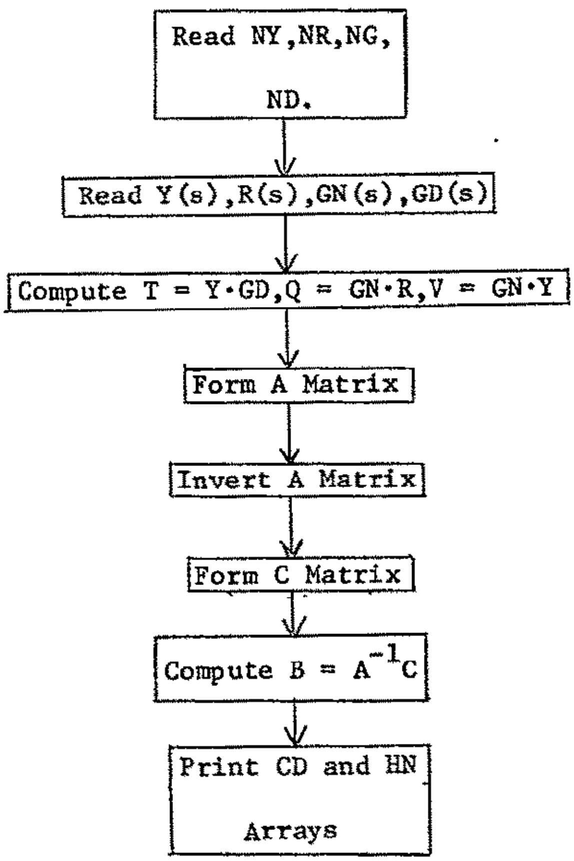
<u>Subroutines to be Used</u>

PLYMLT Multiplies two polynomials together.

MXINV Inverts matrix A.

PRNTRAT Prints the ratio of two polynomials.

<u>Flow Diagram</u>



```
      PROGRAM MODOBS1 (INPUT,OUTPUT,TAPE4=INPUT)
     0DIMENSION Y(10), R(10), GD(10), GN(10), V(20), T(20), Q(20)
     1,A(36,36),AI(36,36),C(36,18),B(36,18),ID(6)
C     Y(S)/R(S) IS DESIRED CLOSED LOOP RESPONSE.
C     GP(S)/GD(S) IS PLANT TRANSFER FUNCTION.
C     NY IS ORDER OF NUM. IN Y(S)/R(S). NR IS ORDER OF DEN.
C     NG IS ORDER OF NUM. OF G(S). ND IS ORDER OF DEN.
C     NR-NY MUST EQUAL ND-NG
1     READ 101,NY,NR,NG,ND,ID
      PRINT 200,NY,NR,NG,ND,ID
      IF(EOF,4) 99,2
2     NY1=NY+1
      READ 102, (Y(I),I=1,NY1)
C     Y(I) IS COEFF. OF S**(I-1) IN Y(S).
      NR1=NR+1
C     R(I) IS COEFF. OF S**(I-1) IN R(S)
3     READ 102,  (R(I), I=1,NR1)
      CALL PRNTRAT(Y,NY1,R,NR1,1HT,1HS)
C     GN(I) IS COEFF. OF S**(I-1) IN GN(S).
      NG1=NG+1
      ND1=ND+1
4     READ 102, (GN(I), I=1,NG1)
C     GD(I) IS COEFF. OF S**(I-1) IN GD(S).
5     READ 102, (GD(I),I=1,ND1)
      CALL PRNTRAT(GN,NG1,GD,ND1,1HG,1HS)
C     FORM CHARACTERISTIC POLYNOMIAL.
      CALL PLYMLT (GD,ND1,Y,NY1,T,0)
      CALL PLYMLT (GN,NG1,R,NR1,Q,0)
      CALL PLYMLT (GN ,NG1,Y,NY1,V,0)
      NYNG=NY1+NG1
      N    =NR+NG+1
      DO 6  I=NYNG,N
C     NA IS ORDER OF A MATRIX WHERE A IS MATRIX RELATING UNKNOWNS
6     V(I)=0.                                     TO PARAM., P.
      NA= 2*(N-1)
      DO 20 J=1,NA
      IF (J.GT.(N-1))GO TO 12
      DO 10  I=1,NA
      IND=N-I+J
      IF (IND .LT. 1) GO TO 8
      IF (IND .GT.N) GO TO 8
      A(I,J)= T(IND)
      GO TO 10
8     A(I,J)=0.
10    CONTINUE
      GO TO 20
12    DO 20 I=1,NA
      IND =1-I+J
      IF (IND.LT.1) GO TO 18
      IF (IND.GT.N) GO TO 18
      A(I,J)=V(IND)
      GO TO 20
18    A(I,J)=0.
20    CONTINUE
      CALL MXINV (A,NA,AI)
C       C IS THE COEFF. ARRAY.
        NM1=N-1
        DO 24 I=1,NA
        DO 24 J=1,NM1
        IND=2*N-I-J
```

```
      IF(IND.LT.1) GO TO 22
      IF (IND.GT.N) GO TO 22
      C(I,J) = Q(IND)
      GO TO 24
22    C(I,J)=0.
24    CONTINUE
C     B IS AI*C.THE ROWS OF B ARE EQUAL TO CD AND HN.
C     COLS. ARE COEFF. OF P PARAMETERS.
      DO  30 I=1,NA
      DO  30 K=1,NM1
      B(I,K)=0.
      DO  30J=1,NA
30    B(I,K)=AI(I,J)*C(J,K)+B(I,K)
      PRINT 201,(K,K=1,NM1)
      DO 40 I=1,NM1
      NC=N-I
40    PRINT 202,NC,(B(I,K),K=1,NM1)
      DO 50 I=N,NA
      NB=NA+1-I
50    PRINT 204, NB,(B(I,K),K=1,NM1)
      GO TO 1
99    STOP
101   FORMAT (4I1,6A10)
102   FORMAT (8F10.3)
200   FORMAT (1H1,4(2X,I1),2X,6A10)
201   FORMAT (1H0,9X,8(3H P(,I1,1H),8X))
202   FORMAT (4H CD(,I1,2H)=,10(E10.3,3X))
204   FORMAT (4H HN(,I1,2H)=,10(E10.3,3X))
      END
      SUBROUTINE PRNTRAT (A,NN,B,ND,DEPVAR,INDVAR)
      DIMENSION A(20),B(20),H(20),N(20),AT(20),BT(20)
      NE=1
      DO 6 I=1,NN
      AT(I)=0.
      IF ( A(I) ) 4,6,3
3     H(NE)=1H+
      AT(NE)=A(I)
      GO TO 5
4     H(NE)=1H-
      AT(NE)=-A(I)
      N(NE)=I-1
      NE=NE+1
6     CONTINUE
      NE=NE-1
      PRINT 100
      MIN=MIN0(NE,10)
      PRINT 101,(N(I),I=1,MIN)
      PRINT 102,(H(I),AT(I),INDVAR,I=1,MIN)
      IF (NE-10) 8,8,7
7     PRINT 101,(N(I),I=11,NE)
      PRINT 102,(H(I),AT(I),INDVAR,I=11,NE)
8     NF=1
      DO 16 I=1,ND
      BT(I)=0.
      IF ( B(I) ) 14,16,13
13    H(NF)=1H+
      BT(NF)=B(I)
      GO TO 15
14    H(NF) =1H-
      BT(NF)=-B(I)
```

```
15    N(NF)=I-1
      NF=NF+1
16    CONTINUE
      NF=NF-1
      HH=1H-
      NH=MAX0(13*NE,13*NF)
      NH=MIN0(130,NH)
20    PRINT 103,DEPVAR,INDVAR,(HH, I=1,NH)
      MIN=MIN0(NF,10)
      PRINT 101,(N(I),I=1,MIN)
      PRINT 102,(H(I),BT(I),INDVAR,I=1,MIN)
      IF (NF-10)  28,28,27
27    PRINT 101,(N(I), I=11,NF)
      PRINT 102,(H(I),BT(I),INDVAR,I=11,NF)
28    CONTINUE
100   FORMAT (1H0)
101   FORMAT (6X,10(11X,I2))
102   FORMAT (6X,10(A1,G10.4 ,A1,1H ))
103   FORMAT (1X,A1,1H(,A1,2H)=,130A1)
      RETURN
      END
      SUBROUTINE MXINV (R, N, RI)
C     SUBROUTINE TO FIND THE INVERSE OF A GIVEN
C     MATRIX R. N IS ORDER OF MATRIX, RI IS INVERSE
C     MATRIX. SUBROUTINE USES GAUSS-JORDAN REDUCTION,
C     R MATRIX IS PRESERVED.  DIAGONAL ELEMENTS OF R MUS
C     BE NONZERO.
C     DIMENSION R(36,36),RA(36,72), RI(36,36)
C     STATEMENTS 20-26 ENTER R ARRAY INTO RA ARRAY
C     AND SET LAST N COLUMNS OF RA ARRAY TO IDENTITY
C     MATRIX
   20 DO 26  I = 1, N
   21 DO 24  J = 1, N
   22 RA(I,J) = R(I,J)
   23 NJ = N + J
   24 RA(I,NJ) = 0.
   25 NI = N + I
   26 RA(I,NI) = 1.
C
C     STATEMENTS 1-12 REDUCE MATRIX RA SO THAT FIRST N
C     COLUMNS ARE SET EQUAL TO THE IDENTITY MATRIX
    1 NP = 2 * N
    2 DO 12  I = 1, N
C
C     STATEMENTS 3-5 ARE USED TO SET MAIN DIAGONAL
C     ELEMENT TO UNITY
    3 ALFA = RA(I,I)
    4 DO  5  J = I, NP
    5 RA(I,J) = RA(I,J) / ALFA
C
C     STATEMENTS 6-11 ARE USED TO SET ELEMENTS OF ITH
C     COLUMN TO ZERO
    6 DO 11  K = 1, N
    7 IF (K - I) 8, 11, 8
    8 BETA = RA(K,I)
    9 DO 10  J = I, NP
   10 RA(K,J) = RA(K,J) - BETA * RA(I,J)
   11 CONTINUE
   12 CONTINUE
C
```

```
C     STATEMENTS 30-33 SET INVERSE MATRIX RI EQUAL TO LAST
C     N COLUMNS OF RA ARRAY
   30 DO 33  J = 1, N
   31 JN = J + N
   32 DO 33  I = 1, N
   33 RI(I,J) = RA(I,JN)
   34 RETURN
      END
      SUBROUTINE  PLYMLT (A,L,B,M,C,N)
C
C     MULTIPLY ONE POLYNOMIAL BY ANOTHER
C
C  DEFINITION OF SYMBOLS IN ARGUMENT LIST
C     A(I),  MULTIPLICAND COEFFICIENTS IN THEORDER A(I) *S**(I-1)
C     L,     NUMBER OF COEFFICIENTS OF A
C     B(I),  MULTIPLIER COEFFICIENTS IN THE ORDER B(I)*S**(I-1)
C     M,     NUMBER OF COEFFICIENTS OF B
C     C(I),  PRODUCT COEFFICIENTS IN THE ORDER C(I)*S**(I-1)
C     N,     NUMBER OF COEFFICIENTS OF C
C
C  REMARKS
C     IF N=0, C(I) SET TO ZERO AND PRODUCT FORMED. OTHERWISE THE
C     PRODUCT AND SUM NEWC= OLD C + A*B IS FORMED.
C
      DIMENSION A(10), B(10), C(20)
      LPM=L+M-1
      IF (N)   10,10,12
 10   DO 11  J=1,LPM
 11   C(J)=0.0
 12    DO 13 J=1,LPM
      MAX=MAX0(J+1-M,1)
      MIN=MIN0(L,J)
      DO 13  I=MAX,MIN
 13   C(J)=A(I)*B(J+1-I) + C(J)
      RETURN
      END

 3434 IN CORE THERMIONIC REACTOR
 1.16708    48.1228    384.520    200.
 1.16708    48.70635   408.5815   392.26     100
 1.16708    48.1228    384.520    200.
 5.03585E-61.0542E- 3 .0568639  1.8804     1.
         END OF INFORMATION
```

Appendix 2 - MODOBS2

Discussion

If CD(s) in (5) is defined by $CD(s) = CD_N s^{N-1}$ (B1)

where N is the order of the plants, then the system will have minimum low frequency sensitivity because the forward loop gain will be increased at low frequencies. If there are no zeros in Y(s)/R(s) or in G(s), (5) reduces to a simple set of linear equations, shown below. For simplicity R_{N+1} and GD_{N+1} are set equal to one. For zero steady state error, $Y = R_1$.

$$\begin{aligned}
&P_N = 1\\
&P_{N-1} = C_{N-1}\\
&P_{N-2} + R_N P_{N-1} = C_{N-2}\\
&\quad\vdots\\
&P_1 + R_N \cdot P_2 \cdots + R_3 P_{N-1} = C_1\\
&HN_1 = P_1 \cdot R_1/Y\\
&HN_2 = P_1 \cdot R_2/Y + P_2\\
&HN_N = P_1 \cdot R_N/Y + P_2 \cdot R_{N-1}/Y + \cdots + 1 - GD_1/Y
\end{aligned} \tag{B2}$$

Where the C terms are defined by

$$\begin{aligned}
&C_1 = GD_2 - R_2\\
&C_2 = GD_3 - R_3\\
&\quad\vdots\\
&C_{N-1} = GD_N - R_N
\end{aligned}$$

The Eqns. (B2) are simply solved in the order shown by back substitution. That is, P_{N-1} may be used to solve for P_{N-2} which may be used to solve for P_{N-3}, etc.

Input Format

Card No.	Columns	Description	Format
1	1	N = Order of the Plant G(s)	I1
	2-80	ID = Identification of problem	8A10
2	1-10	Y = Numerator of Y/R	F10.3
3	1-10	R(1) = coeff. of s^0 in R(s)	F10.3
	11-20	R(2) = coeff. of s^1 in R(s)	F10.3
	etc.	.	.
		.	.
		.	.
	71-80	R(7) = coeff. of s^7 in R(s)	F10.3
4	1-10	GN = numerator of G(s)	F10.3
5	1-10	GD(1) = coeff. of s^0 in GD(s)	F10.3
	.	.	
	.	.	
	.	.	
	71-80	GD(7) = coeff. of s^7 in GD(s)	F10.3

See notes following input format for MODOBS1.

Output Format

See example.

Subroutines to be Used

PRNTRAT - Prints the ratio of two polynomials.

PROOT - Determines the roots of P(s) to see if any roots are in RHP.

```
      PROGRAM MODOBS2 (INPUT,OUTPUT,TAPE4=INPUT)
      DIMENSION R(10),GD(10),P(10),C(10),ID(3),HN(10),CD(10),
     U(10),V(10)
C     PROGRAM TO CALCULATE MINIMUM LOW-FREQ SENSITIVITY, MODIFIED
C     OBSERVER SYSTEM.
C     Y(S)/R(S) IS DESIRED CLOSED LOOP RESPONSE.
C     GN(S)/GD(S) IS PLANT TRANSFER FUNCTION.
C     N IS ORDER OF DEN. IN Y(S)/R(S).
C     N IS ORDER OF DEN. IN GN(S)/GD(S).
C     R(I) IS COEF. OF S**(I-1) IN R(S).
C
C     ASSUMPTIONS
C     **************
C     TRANSFER FCN. GAINS ARE NORMALIZED SUCH THAT R(N+1)=1.



C     AND GD(N+1)=1.
C
C     DESIRED TRANSFER FCN. HAS ZERO STEADY STATE ERROR. THUS
C     *****************                                        Y=R(1).
C     GD(I) IS COEF. OF S**(I-1) INGD(S).
C     Y(S)=Y , GN(S)=GN (NO ZEROS ALLOWED).
C
1     READ 101, N, ID
      IF(EOF,4)99,2
2     READ 102,Y
      READ 102, (R(I),I=1,N)
      NP1=N+1
      GD(NP1)=1. $ R(NP1)=1.
      READ 102, GN
3     READ 102, (GD(I),I=1,N)
      PRINT 200,N,ID
      PRINT 201
      CALL PRNTRAT (Y,1,R,NP1,1HT,1HS)
      PRINT 202
      CALL PRNTRAT(GN,1,GD,NP1,1HG,1HS)
C
C     SOLVE FOR DEN. OF H, P(S).
      NM1=N-1
      DO 40 I=1, NM1
      IP1=I+1
40    C(I)=GD(IP1)-R(IP1)
      P(N)=1.
      DO 50 I=1,NM1
      NI=N-I
      P(NI)=C(NI)
      IF(I.LE.1) GO TO 50
      DO 49  J=2,I
      J2=N-J+2
      JI=NI+J-1
49    P(NI)=P(NI)-R(J2)*P(JI)
50    CONTINUE
      CALL PROOT (NM1,P,U,V,1)
      PRINT 205
      PRINT 206,(U(I),V(I), I=1,NM1)
C
C     SOLVE FOR NUM OF H,HN(S)
      DO 60 I=1,N
      HN(I)=0.
      DO 60 J=1,I
      IJ=I-J+1
```

```
6C      HN(I)=P(J)*R(IJ)/Y+HN(I)
        HN(N)=HN(N)-GD(1)/Y
        PRINT 203
        CALL PRNTRAT(HN,N,P,N,1HH,1HS)
        DO 80 I=1, NM1
80      CD(I)=0.
        CD(N)=1.
        PRINT 204
        CALL PRNTRAT(P,N,CD,N,1HC,1HS)
        G = Y/GN
        PRINT 210, G
        GO TO 1
99      STOP
101     FORMAT(I1,8A10)
102     FORMAT(8F10.3)


200     FORMAT(1H1,10X,I1,4X,8A10)
201     FORMAT(1H0,*THE DESIRED CLOSED LOOP TRANSFER FUNCTION IS*)
202     FORMAT(1H0,*THE TRANSFER FUNCTION OF THE PLANT IS*)
203     FORMAT(1H0,*THE TRANSFER FUNCTION OF THE FEEDBACK COMPENSATOR IS*)
204     FORMAT(1H0,*THE TRANSFER FUNCTION OF THE SERIES COMPENSATOR IS*)
205     FORMAT (1H0,5X* THE ROOTS OF P(S) ARE*13X*REAL PART*10X*IMAGINARY
        PART*)
206     FORMAT (32X,2E20.7)
210     FORMAT(1H0,*THE ADDITIONAL FORWARD LOOP GAIN REQUIRED IS
       1Y/GN=*E10.3)
        END
        SUBROUTINE PRNTRAT (A,NN,B,ND,DEPVAR,INDVAR)
        DIMENSION A(20),B(20),H(20),N(20),AT(20),BT(20)
        NE=1
        DO 6 I=1,NN
        AT(I)=0.
        IF ( A(I) ) 4,6,3
3       H(NE)=1H+
        AT(NE)=A(I)
        GO TO 5
4       H(NE)=1H-
        AT(NE)=-A(I)
5       N(NE)=I-1
        NE=NE+1
6       CONTINUE
        NE=NE-1
        PRINT 100
        MIN=MIN0(NE,10)
        PRINT 101,(N(I),I=I,MIN)
        PRINT 102,(H(I),AT(I),INDVAR,I=1,MIN)
        IF (NE-10) 8,8,7
7       PRINT 101,(N(I),I=11,NE)
        PRINT 102,(H(I),AT(I),INDVAR,I=11,NE)
8       NF=1
        DO 16 I=1,ND
        BT(I)=0.
        IF ( B(I) ) 14,16,13
13      H(NF)=1H+
        BT(NF)=B(I)
        GO TO 15
14      H(NF) =1H-
        BT(NF)=-B(I)
15      N(NF)=I-1
        NF=NF+1
```

```
16    CONTINUE
      NF=NF-1
      HH=1H-
      NH=MAX0(13*NE,13*NF)
      NH=MIN0(130,NH)
20    PRINT 103,DEPVAR,INDVAR,(HH, I=1,NH)
      MIN=MIN0(NF,10)
      PRINT 101,(N(I),I=1,MIN)
      PRINT 102,(H(I),BT(I),INDVAR,I=1,MIN)
      IF (NF-10)  28,28,27
27    PRINT 101,(N(I), I=11,NF)
      PRINT 102,(H(I),BT(I),INDVAR,I=11,NF)
28    CONTINUE
100   FORMAT (1H0)
101   FORMAT (6X,10(11X,I2))
102   FORMAT (6X,10(A1,G10.4 ,A1,1H ))

103   FORMAT (1X,A1,1H(,A1,2H)=,130A1)
      RETURN
      END
      SUBROUTINEPROOT(N,A,U,V,IR)
      DIMENSION A(20),U(20),V(20),H(21),B(21),C(21)
      IREV=IR
      NC=N+1
      DO1I=1,NC
      H(I)=A(I)
1     CONTINUE
      P=0.
      Q=0.
      R=0.
    3 IF(H(1))4,2,4
    2 NC=NC-1
      V(NC)=0.
      U(NC)=0.
      DO1002I=1,NC
      H(I)=H(I+1)
 1002 CONTINUE
      GOTO3
    4 IF(NC-1)5,100,5
    5 IF(NC-2)7,6,7
    6 R=-H(1)/H(2)
      GOTO50
    7 IF(NC-3)9,8,9
    8 P=H(2)/H(3)
      Q=H(1)/H(3)
      GOTO70
9     IF(ABS(H(NC-1)/H(NC))-ABS(H(2)/H(1)))10,19,19
   10 IREV=-IREV
      M=NC/2
      DO11I=1,M
      NL=NC+1-I
      F=H(NL)
      H(NL)=H(I)
   11 H(I)=F
      IF(Q)13,12,13
   12 P=0.
      GOTO15
```

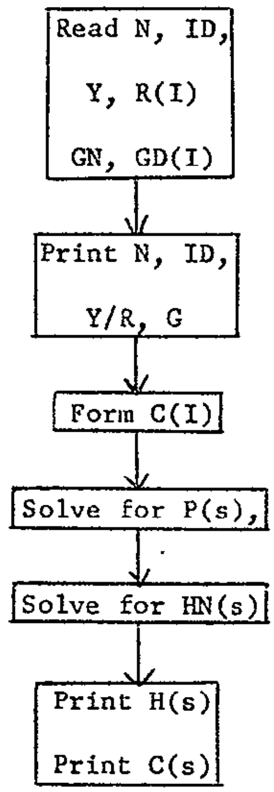
```
13 P=P/Q
   Q=1./Q
15 IF(R)16,19,16
16 R=1./R
19 E=5.E-10
   B(NC)=H(NC)
   C(NC)=H(NC)
   B(NC+1)=0.
   C(NC+1)=0.
   NP=NC-1
20 DO49J=1,1000
   DO21I1=1,NP
   I=NC-I1
   B(I)=H(I)+R*B(I+1)
21 C(I)=B(I)+R*C(I+1)
   IF(ABS(B(1)/H(1))-E)50,50,24
24 IF(C(2))23,22,23
22 R=R+1.
   GOTO30
23 R=R-B(1)/C(2)
```

```
30 DO37I1=1,NP
   I=NC-I1
   B(I)=H(I)-P*B(I+1)-Q*B(I+2)
37 C(I)=B(I)-P*C(I+1)-Q*C(I+2)
   IF(H(2))32,31,32
31 IF(ABS(B(2)/H(1))-E)33,33,34
32 IF(ABS(B(2)/H(2))-E)33,33,34
33 IF(ABS(B(1)/H(1))-E)70,70,34
34 CBAR=C(2)-B(2)
   D=C(3)**2-CBAR*C(4)
   IF(D)36,35,36
35 P=P-2.
   Q=Q*(Q+1.)
   GOTO49
36 P=P+(B(2)*C(3)-B(1)*C(4))/D
   Q=Q+(-B(2)*CBAR+B(1)*C(3))/D
49 CONTINUE
   E=E*10.
   GOTO20
50 NC=NC-1
   V(NC)=0.
   IF(IREV)51,52,52
51 U(NC)=1./R
   GOTO53
52 U(NC)=R
53 DO54I=1,NC
   H(I)=B(I+1)
54 CONTINUE
   GOTO4
70 NC=NC-2
   IF(IREV)71,72,72
71 QP=1./Q
   PP=P/(Q*2.0)
   GOTO73
```

```
   72 QP=Q
      PP=P/2.0
   73 F=(PP)**2-QP
      IF(F)74,75,75
   74 U(NC+1)=-PP
      U(NC)=-PP
      V(NC+1)=SQRT(-F)
      V(NC)=-V(NC+1)
      GOTO76
75    U(NC+1)=-(PP/ABS(PP))*(ABS(PP)+SQRT(F))
      V(NC+1)=0.
      U(NC)=QP/U(NC+1)
      V(NC)=0.
   76 DO77I=1,NC
      H(I)=B(I+2)
77    CONTINUE
      GOTO4
  100 RETURN
      END
```

```
3. CASE STUDY II, THIRD ORDER SYSTEM
1000.
1000.      200.      20.
1000.
           2500.     100.
```

Flow Diagram



Appendix 3 - MODOBS3

Discussion

This program uses the following data:

1. Plant transfer function G(s)
2. Desired closed loop transfer function T(s)
3. The denominator (P(s)) of the feedback compensator H(s)
4. A desired weighting function W(s) on the integral over frequency of the sensitivity of the closed loop response to forward loop gain changes.

The program then calculates the weighted integral sensitivity of the system. The method of calculation is based upon an algorithm by Effertz [5]. The integrand of the integral sensitivity is calculated from

$$I\,S_T^K = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\left| W(s)\;S_T^K(s)\right|^2 ds$$

$$= \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\left|\frac{W(s)T(s)}{G(s)C(s)}\right|^2 ds$$

The algorithm is based upon an iterative manipulation of the coefficients in the numerator and denominator of the integrand in such a way that the integral of the squared magnitude results.

A closed form solution for the integral

$$I_n = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{g(P)}{h(P)h(-P)}\,dP = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} F(P)F(-P)\,dP \tag{B1}$$

is presented in the paper by F. H. Effertz. The solution takes the form of Equation 4 in his paper. A better algorithm for the computation of Equation 4 in Effertz is presented in the correspondence by Pazdera (Equation 7). A modified form of Pazdera's algorithm has been coded in FORTRAN and is included. A test problem is, also, included.

The use of Equation 4 in Effertz can be best illustrated by an example. Suppose we wish to evaluate the following

$$I_n = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \left[\frac{s + c}{s^2 + (a+b)s + ab} \right] \left[\frac{c - s}{s^2 - (a+b)s + ab} \right] \tag{B2}$$

It can be shown from residue theory that the correct answer is

$$\begin{aligned} I_n = & - \text{residue of } F(s)\,F(-s) \text{ at } s = a \\ & - \text{residue of } F(s)\,F(-s) \text{ at } s = b \end{aligned} \tag{B3}$$

where $F(s)\,F(-s)$ indicates the function to be integrated over $s = j\omega$. Then the answer is

$$I_n = - \frac{c^2 - a^2}{2a(a+b)(a-b)} - \frac{(c^2 - b^2)}{2b(a+b)(b-a)} = \frac{c^2 + ab}{2ab(a+b)} \tag{B4}$$

To use Equation 4 from Effertz we must make the following associations, using the complex frequency variable s in place of p.

$$g(P) = (s+c)(c-s) = c^2 - s^2$$

$$\text{then} \quad \begin{cases} P = s \\ n = 2 \\ b_o = -1 \\ b_1 = c^2 \end{cases} \tag{B5}$$

$$h(P) = s^2 + (a+b)\,s + ab$$

$$\text{then} \quad \begin{cases} P = s \\ N = 2 \\ a_o = 1 \\ a_1 = a + b \\ a_2 = ab \end{cases}$$

The integral, for n = 2, can be expressed as

$$I_n = \frac{(-1)^3}{2a_o} \frac{\begin{vmatrix} b_o & b_1 \\ a_o & a_2 \end{vmatrix}}{\begin{vmatrix} a_1 & a_o \\ a_o & a_2 \end{vmatrix}} = \frac{-1}{2a_o} \left[\frac{b_o a_2 - a_o b_1}{a_1 a_2} \right] \tag{B6}$$

Substituting the correct values of a_o, a_1, etc., and cancelling minus signs for this problem we obtain

$$I_n = \frac{ab + c^2}{2ab(a+b)} \tag{B7}$$

What is most interesting is that the result depends only on the coefficients of the known function being integrated and not on the poles of the function as one would suspect from residue theory.

It has been mentioned that the algorithm from Pazdera (Equation 7) has been programmed. Some charges in the nomenclature of Equation 7 were necessary to facilitate coding. Note that the practice in Effertz and Pazdera is to make the highest membered coefficient correspond to the lowest power of the variable p. Note, also, that subscripts such as a_o are in evidence. To facilitate coding, the lowest numbered coefficient was subscripted in the form A(1) and this coefficient was associated with the lowest power of the variable p, in this case p^o. In general, the i^{th} coefficient A(I) is associated with the $(i-1)^{st}$ power of p, p^{I-1}. (In FORTRAN the expression A(0) is not allowed.) Note from equation 1 above what must be done to operate on the function F(s) F(-s). If F(s) is of the form

$$F(s) = \frac{C(S)}{A(S)} \tag{B8}$$

then the algorithms suggested in both Effertz and Pazdera require the use of

$$g(s) = C(S)\,C(-S)$$
$$h(s) = A(S) \qquad \text{(B9)}$$

It is more desirable to input C(S) and A(S) rather than C(S) C(-S) and A(S). Consequently one uses the INTSQ subroutine by coding the coefficients of C(S) and A(S) with ascending subscripts corresponding to ascending powers of S. The subroutine provides the operation C(S) C(-S). Furthermore, the subroutine checks to see if the lowest coefficient of C(S) or A(S) is zero so that factors of S may be either considered or cancelled. Two basic requirements must be met by A(S) and C(S). First, the roots of A(S) and C(S) must at least have non-negative real parts. (Hurwitz polynomial requirement) Secondly, the highest power of C(S) must be at least one less than the highest power of A(S) for convergence to be assured. If A(S) is Nth order, one inputs the N coefficients of A(S) and the N-1 coefficients of C(S).

The use of this subroutine INTSQ may be best illustrated by an example. Suppose we desired to evaluate the integral

$$I_n = \int_{-j\infty}^{j\infty} \left[\frac{s + 4}{4s^3 + 3s^2 + 2s + 1}\right] \left[\frac{4 - s}{-4s^3 + 3s^2 - 2s + 1}\right] dS \qquad \text{(B10)}$$

The polynomials C(S) and A(S) as defined in Equation B8 are input into the program as

$$\begin{array}{ll} C(1) = 4 & A(1) = 1 \\ C(2) = 1 & A(2) = 2 \\ C(3) = 0 & A(3) = 3 \\ & A(4) = 4 \end{array} \qquad \text{(B11)}$$

The program then utilizes the modified Pazdera algorithm and prints out the message

THE VALUE OF THE INTEGRAL IS 12.250

To check this answer, one can use Effertz Equation 4 for a third order case (n=3).

$$I_n = \frac{(-1)^4}{2a_o} \frac{\begin{vmatrix} b_o & b_1 & b_2 \\ a_o & a_2 & 0 \\ 0 & a_1 & a_3 \end{vmatrix}}{\begin{vmatrix} a_1 & a_3 & 0 \\ a_o & a_2 & 0 \\ 0 & a_1 & a_3 \end{vmatrix}} = \frac{1}{2a_o} \frac{b_o a_2 a_3 + b_2 a_o a_1 - a_o a_3 b_1}{a_1 a_2 a_3 - a_o a_3^2} \qquad \text{(B12)}$$

For our example problem, the following associations follow

$$\begin{array}{ll} b_o = 0 & a_o = 4 \\ b_1 = -1 & a_1 = 3 \\ b_2 = 16 & a_2 = 2 \\ & a_3 = 1 \end{array} \qquad \text{(B13)}$$

Substituting the values of Equation B13 into Equation B12 verifies the computer's solution $I_n = 12.25$.

Input Format

Card No.	Columns	Description	Format
.	See format for MODOBS1		
.			
.			
K	1-10	P(1) - coeff. of s^o in P(s)	8E10.3
	11-20	P(2) - coeff. of s' in P(s)	8E10.3
	etc.		

Input Format (cont'd)

Card No.	Columns	Description	Format
K+1	1	NW - Number of coeff. in denom. of W(s)	I1
K+2	1-10	WN(1) - coeff. of s^o in num. of W(s)	8E10.3
	11-20 etc.	WN(2) - coeff. of s' in num. of W(s)	8E10.3
K+3	1-10	WD(1) - coeff. of s^o in den. of W(s)	8E10.3
	11-20 etc.	WD(2) - coeff. of s' in den. of W(s)	8E10.3

```
      PROGRAM MODOBS3(INPUT,OUTPUT,TAPE4=INPUT)
      DIMENSION Y(10),R(10),GD(10),SN(10),P(10),CD(10),HN(10),SN(40),
     1SD(40),B(36,18),WN(10),WD(10),SNT(40),SDT(40)
C
1     CALL MODOBS1(Y,R,GN,GD,B,NY1,NR1,NG1,ND1)
C     N-1 IS THE NUMBER OF COEF. IN P(S),HN(S),CD(S).
      N=NR1+NG1-1
      N1=N-1
      READ 102,(P(I),I=1,N1)
C     W(S) = WN/WD MUST HAVE MORE POLES THAN ZEROS
C     NW IS THE NUMBER OF TERMS IN THE DENOMINATOR OF W(S).
      READ 103,NW
      READ 102, (WN(I),I=1,NW)
      READ 102, (WD(I),I=1,NW)
      PRINT 205
      CALL PRNTRAT  (WN,NW,WD,NW,1HW,1HS)
      CALL PLYMLT(GD,ND1,Y,NY1,SN,0)
      N2=ND1+NY1-1
      CALL PLYMLT(WN,NW,SN,N2,SNT,0)
      CALL PLYMLT(GN,NG1,R,NR1,SD,0)
      CALL PLYMLT(WD,NW,SD,N2,SDT,0)
      N3=NW+N2-1
      N4=N3+N1-1
      N2=2*N1+1
8     DO 10 I=1,N1
      CD(I)=0.
      HN(I)=0.
      DO 10 J=1,N1
      CD(I)=B(N-I,J)*P(J)+CD(I)
10    HN(I)=B(N2-I,J)*P(J)+HN(I)
      PRINT 201
      CALL PRNTRAT(P,N1,CD,N1,1HC,1HS)
      PRINT 202
      CALL PRNTRAT(HN,N1,P,N1,1HH,1HS)
      CALL PLYMLT(CD,N1,SNT,N3,SN,0)
      CALL PLYMLT( P,N1,SDT,N3,SD,0)
      N5=N4 - 1
      CALL INTSQ(SD,SN,N4,S)
      PRINT 204, S
      GO TO 1
102   FORMAT (8 E10.3)
103   FORMAT (I1)
201   FORMAT(*0 THE TRANSFER FUNCTION OF THE SERIES COMPENSATOR IS*)
202   FORMAT(*0 THE TRANSFER FUNCTION OF THE FEEDBACK COMPENSATOR IS*)
204   FORMAT(*0 THE WEIGHTED INTEGRAL SENSITIVITY IS*E10.3)
205   FORMAT (*0 THE WEIGHTING FUNCTION OF THE INTEGRAL SENSITIVITY IS*
     1)
      END
      SUBROUTINE MODOBS1(Y,R,GN,GD,B,NY1,NR1,NG1,ND1)
      DIMENSION Y(10), R(10), GD(10), GN(10), V(20), T(20), Q(20)
     1,A(36,36),AI(36,36),C(36,18),B(36,18),ID(6)
C     Y(S)/R(S) IS DESIRED CLOSED LOOP RESPONSE.
C     GP(S)/GD(S) IS PLANT TRANSFER FUNCTION.
C     NY IS ORDER OF NUM. IN Y(S)/R(S). NR IS ORDER OF DEN.
C     NG IS ORDER OF NUM. OF G(S). ND IS ORDER OF DEN
C     NR-NY MUST EQUAL ND-NG
C     Y(I) IS COEFF. OF S**(I-1) IN Y(S).
```

```
C     R(I) IS COEFF. OF S**(I-1) IN R(S).
C     GN(I) IS COEFF. OF S**(I-1) IN GN(S).
C     GD(I) IS COEFF. OF S**(I-1) IN GD(S).
1     READ 101,NY,NR,NG,ND,ID
      IF(EOF,4) 99,2
2     NY1=NY+1
      NR1=NR+1
      READ 102, (Y(I),I=1,NY1)
3     READ 102,  (R(I), I=1,NR1)
      PRINT 200,NY,NR,NG,ND,ID
      CALL PRNTRAT(Y,NY1,R,NR1,1HT,1HS)
      NG1=NG+1
      ND1=ND+1
4     READ 102, (GN(I), I=1,NG1)
5     READ 102, (GD(I),I=1,ND1)
      CALL PRNTRAT(GN,NG1,GD,ND1,1HG,1HS)
C     FORM CHARACTERISTIC POLYNOMIAL.
      CALL PLYMLT (GD,ND1,Y,NY1,T,0)
      CALL PLYMLT (GN,NG1,R,NR1,Q,0)
      CALL PLYMLT (GN ,NG1,Y,NY1,V,0)
      NYNG=NY1+NG1
      N   =NR+NG+1
      DO 6  I=NYNG,N
C     NA IS ORDER OF A MATRIX WHERE A IS MATRIX RELATING UNKNOWNS TO
6     V(I)=0.                                                 PARAM., P.
      NA= 2*(N-1)
      DO 20 J=1,NA
      IF (J.GT.(N-1))GO TO 12
      DO 10  I=1,NA
      IND=N-I+J
      IF (IND .LT.1) GO TO 8
      IF (IND .GT.N) GO TO 8
      A(I,J)= T(IND)
      GO TO 10
8     A(I,J)=0.
10    CONTINUE
      GO TO 20
12    DO 20 I=1,NA
      IND =1-I+J
      IF (IND.LT.1) GO TO 18
      IF (IND.GT.N) GO TO 18
      A(I,J)=V(IND)
      GO TO 20
18    A(I,J)=0.
20    CONTINUE
      CALL MXINV (A,NA,AI)
C     . C IS THE COEFF. ARRAY.
        NM1=N-1
        DO 24 I=I,NA
        DO 24 J=I,NM1
        IND=2*N-I-J
        IF(IND.LT.1) GO TO 22
        IF (IND.GT.N) GO TO 22
        C(I,J) = Q(IND)
        GO TO 24
22      C(I,J)=0.
24      CONTINUE
C     B IS AI*C.THE ROWS OF B ARE EQUAL TO CD AND HN. THE COLS. ARE
C     COEFF. OF P PARAMETERS.
        DO  30 I=1,NA
```

```
          DO   30 K=1,NM1
          B(I,K)=0.
          DO   30J=1,NA
 30       B(I,K)=AI(I,J)*C(J,K)+B(I,K)
       PRINT 201,(K,K=1,NM1)
       DO 40 I=1,NM1
       NC=N-I
 40    PRINT 202,NC,(B(I,K),K=1,NM1)
       DO 50 I=N,NA
       NB=NA+1-I
 50    PRINT 204, NB,(B(I,K),K=1,NM1)
       RETURN
 99    STOP
 101   FORMAT (4I1,6A10)
 102   FORMAT (8F10.3)
 200   FORMAT (1H1,4(2X,I1),2X,6A10)
 201   FORMAT (10X,3(3H P(,I1,1H),8X))
 202   FORMAT (4H CD(,I1,2H)=,10(E10.3,3X))
 204   FORMAT (4H HN(,I1,2H)=,10(E10.3,3X))
       END
       SUBROUTINE  PLYMLT (A,L,B,M,C,N)
 C
 C     MULTIPLY ONE POLYNOMIAL BY ANOTHER
 C
 C  DEFINITION OF SYMBOLS IN ARGUMENT LIST
 C     A(I),  MULTIPLICAND COEFFICIENTS IN THEORDER A(I) *S**(I-1)
 C     L,     NUMBER OF COEFFICIENTS OF A
 C     B(I),  MULTIPLIER COEFFICIENTS IN THE ORDER B(I)*S**(I-1)
 C     M,     NUMBER OF COEFFICIENTS OF B
 C     C(I),  PRODUCT COEFFICIENTS IN THE ORDER C(I)*S**(I-1)
 C
 C
 C  REMARKS
 C     IF N=0, C(I) SET TO ZERO AND PRODUCT FORMED.  OTHERWISE THE PRODUC
 C     AND SUM NEWC= OLD C + A*B IS FORMED.
 C
       DIMENSION A(10), B(10), C(20)
       LPM=L+M-1
       IF (N)   10,10,12
 10    DO 11  J=1,LPM
 11    C(J)=0.0
 12     DO 13 J=1,LPM
       MAX=MAX0(J+1-M,1)
       MIN=MIN0(L,J)
       DO 13  I=MAX,MI
 13    C(J)=A(I)*B(J+1
       RETURN
       END
       SUBROUTINE PRNTRAT (A,NN,B,ND,DEPVAR,INDVAR)
       DIMENSION A(20),B(20),H(20),N(20),AT(20),BT(20)
       NE=1
       DO 6 I=1,NN
       AT(I)=0.
       IF ( A(I) ) 4,6,3
 3     H(NE)=1H+
       AT(NE)=A(I)
       GO TO 5
 4     H(NE)=1H-
       AT(NE)=-A(I)
 5     N(NE)=I-1
```

```
      NE=NE+1
6     CONTINUE
      NE=NE-1
      PRINT 100
      MIN=MINO(NE,10)
      PRINT 101,(N(I),I=1,MIN)
      PRINT 102,(H(I),AT(I),INDVAR,I=1,MIN)
      IF (NE-10) 8,8,7
7     PRINT 101,(N(I),I=11,NE)
      PRINT 102,(H(I),AT(I),INDVAR,I=11,NE)
8     NF=1
      DO 16 I=1,ND
      BT(I)=0.
      IF ( B(I) ) 14,16,13
13    H(NF)=1H+
      BT(NF)=B(I)
      GO TO 15
14    H(NF) =1H-
      BT(NF)=-B(I)
15    N(NF)=I-1
      NF=NF+1
16    CONTINUE
      NF=NF-1
      HH=1H-
      NH=MAXO(13*NE,13*NF)
      NH=MINO(130,NH)
20    PRINT 103,DEPVAR,INDVAR,(HH, I=1,NH)
      MIN=MINO(NF,10)
      PRINT 101,(N(I),I=1,MIN)
      PRINT 102,(H(I),BT(I),INDVAR,I=1,MIN)
      IF (NF-10)  28,28,27
27    PRINT 101,(N(I), I=11,NF)
      PRINT 102,(H(I),BT(I),INDVAR,I=11,NF)
28    CONTINUE
100   FORMAT (1H0)
101   FORMAT (6X,10(11X,I2))
102   FORMAT (6X,10(A1,G10.4 ,A1,1H ))
103   FORMAT (1X,A1,1H(,A1,2H)=,130A1)
      RETURN
      END
      SUBROUTINE MXINV (R, N, RI)
C     SUBROUTINE TO FIND THE INVERSE OF A GIVEN
C     MATRIX R. N IS ORDER OF MATRIX, RI IS INVERSE
C     MATRIX. SUBROUTINE USES GAUSS-JORDAN REDUCTION.
C     R MATRIX IS PRESERVED.  DIAGONAL ELEMENTS OF R MUST BE NONZERO.
      DIMENSION R(36,36),RA(36,72), RI(36,36)
C
C     STATEMENTS 20-26 ENTER R ARRAY INTO RA ARRAY
C     AND SET LAST N COLUMNS OF RA ARRAY TO IDENTITY
C     MATRIX
   20 DO 26  I = 1, N
   21 DO 24  J = 1, N
   22 RA(I,J) = R(I,J)
   23 NJ = N + J
   24 RA(I,NJ) = 0.
   25 NI = N + I
   26 RA(I,NI) = 1.
C
C     STATEMENTS 1-12 REDUCE MATRIX RA SO THAT FIRST N
C     COLUMNS ARE SET EQUAL TO THE IDENTITY MATRIX
```

```
      1 NP = 2 * N
      2 DO 12  I = 1, N
C       STATEMENTS 3-5 ARE USED TO SET MAIN DIAGONAL
C       ELEMENT TO UNITY
      3 ALFA = RA(I,I)
      4 DO  5  J = I, NP
      5 RA(I,J) = RA(I,J) / ALFA
C
C       STATEMENTS 6-11 ARE USED TO SET ELEMENTS OF ITH
C       COLUMN TO ZERO
      6 DO 11  K = 1, N
      7 IF (K - I) 8, 11, 8
      8 BETA = RA(K,I)
      9 DO 10  J = I, NP
     10 RA(K,J) = RA(K,J) - BETA * RA(I,J)
     11 CONTINUE
     12 CONTINUE
C
C       STATEMENTS 30-33 SET INVERSE MATRIX RI EQUAL TO LAST
C       N COLUMNS OF RA ARRAY
     30 DO 33  J = 1, N
     31 JN = J + N
     32 DO 33  I = 1, N
     33 RI(I,J) = RA(I,JN)
     34 RETURN
        END
        SUBROUTINE PLYSQ (C,N,B)
        DIMENSION C(40),B(40)
C       RETURNS B(S) = C(S)*C(-S) TO MAIN PROGRAM.
C       B(I) IS COEF. OF S**I-1 IN B(S).
C       C(I) IS COEF. OF S**I-1 IN C(S).
C       N-I IS NUMBER OF COEF. IN B(S) AND C(S).
        N1=(N-1)/2
        DO 20 I=1,N1
        MO =-1
        B(I)=0.
        II = 2*I-1
        DO 20 K=1, II
        MO = -1*MO
 20     B(I)=B(I)+MO*C(K)*C(2*I-K)
        NM1=N-1
        N2=N1+1
        DO 30 I=N2,NM1
        IIN=2*I-NM1
        B(I)=0.
        MO=(-1) ** NM1
        DO 30 K=IIN,NM1
        MO = -1*MO
 30     B(I)=B(I)+MO*C(K)*C(2*I-K)
        RETURN
        END
        SUBROUTINE INTSQ(A,C,N,S)
        DIMENSION B(10),A(11),C(10)
C       RETURNS S=INTEGRAL OF C(S)*C(-S)/A(S)*A(-S) TO MAIN PROGRAM.
C       B(S) HAS N-1 TERMS, A(S) HAS N TERMS.
C       B(I) IS COEF. OF S**(2I-2)
C       C(I) IS COEF. OF S**(I-1).
C       IF THE LOWER ORDER DEN. AND NUM. COEFF. ARE SMALL (LESS THAN
C       O) THEN DIVIDE BOTH NUM. AND DEN. BY S**2.
```

```
      N-1
      D=1.0E-6
1     IF (ABS(A(1))-D) 2,2,20
2     IF (ABS(C(1))-D) 4,4,20
4     IF (K-1) 5,5,6
5     PRINT 200
6     PRINT 201,A(1), C(1)
      K=K+1
      N=N-1
      DO 10  I=1,N
      A(I)=A(I+1)
      IF(I-N) 8,10,10
8     C(I)=C(I+1)
10    CONTINUE
      GO TO 1
20    CALL PLYSQ (C,N,B)
      NM2=N-2
      DO 50 K=1,NM2
      NK=N-K
      BA=B(NK)/A(NK+1)
      AA=0.
      IF(K.EQ.1) GO TO 40
      AA=A(NK+2)/A(NK+1)
40    NMK=(N-K)/2
      DO 50 I=1, NMK
      NKI=NK-2*I+1
      B(NK-I)=B(NK-I)-BA*A(NKI)
      A(NKI+1)=A(NKI+1)-AA*A(NKI)
50    CONTINUE
      S=B(1)/(2.*A(2)*A(1))
      RETURN
200   FORMAT (*0 THE FOLLOWING COEFF. WERE FOUND TO BE SMALL AND CANCELL
     1ED IN THE NUMERATOR AND DENOMINATOR*)
201   FORMAT (*  THE DENOMINATOR COEFF.=* E10.3 * THE NUMERATOR COEFF.
     1=* E10.3)
      END

0303  CASE STUDY II, THIRD ORDER SYSTEM
1000.
1000.     200.      20.       1.
1000.
          2500.     100.      1.
700.      80.       1.
2
```

Appendix 4 - MODOBS4

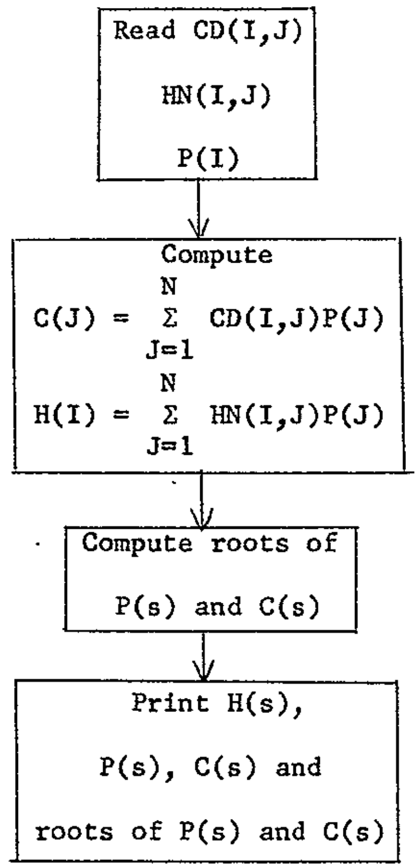
Discussion

MODOBS4 uses the matrix relating the coefficients of CD(s) and HN(s) to the coefficients of P(s) and also the coefficients of P(s) to determine both the series compensators and the feedback compensators. The roots of the compensators are included.

Input format

Card No.	Columns	Description	Format
1	1	N = order of plant G(s)	I1
	2-80	ID = Identification	8A10
2	1-10	CD(1,1) coeff of array from MODOBS1	E10.3
	11-20	CD(1,2) coeff of array from MODOBS1	E10.3
3	1-10	CD(2,1) coeff of array from MODOBS1	E10.3
	11-20	CD(2,2) coeff of array from MODOBS1	E10.3
⋮			
N+2	1-10	HN(1,1) coeff of array from MODOBS1	E10.3
	11-20	HN(1,2) coeff of array from MODOBS1	E10.3
⋮			
2N+2	1-10	P(1) - coeff of s^0 in P(s)	E10.3
	11-20	P(2) - coeff of s' in P(s)	E10.3
	etc.		

Flow Diagram



```
RUN(S)
LGO.

      PROGRAM MODOBS4(INPUT,OUTPUT,TAPE4=INPUT)
C     PROGRAM TO CALCULATE COMPENSATORS WITH P(S) AND RESULTS OF MODOBS1
      DIMENSION ID(8),CD(18,18),HN(18,18),C(18),H(18),U(20),V(20),P(20)
C     N IS THE ORDER OF THE PLANT, G(S) AND NO. OF COEF IN P(S).
1     READ 100, N, ID
      IF(EOF,4) 99,2
2     PRINT 200, N, ID
      DO 10  I=1,N
10    READ 101, (CD(I,J), J=1, N)
      DO 15  I=1,N
15    READ 101, (HN(I,J), J=1,N)





      READ 101, (P(I), I=1,N)
      DO 20  I=1,N
      C(I)=0.
      H(I)=0.
      DO 20  J=1,N
      C(I)=C(I)+CD(I,J)*P(J)
      H(I)=H(I)+HN(I,J)*P(J)
      PRINT 204
      CALL PRNTRAT(P,N,C,N,1HC,1HS)
      PRINT 205
      CALL PRNTRAT (H,N,P,N,1HH,1HS)
      NM1=N-1
      CALL PROOT (NM1,C,U,V,1)
      PRINT 201
      PRINT 202,(U(I),V(I),I=1,NM1)
      CALL PROOT(NM1,P,U,V,1)
      PRINT 203
      PRINT 202,(U(I),V(I),I=1,NM1)
      CALL PROOT (NM1,HN,U,V,1)
      PRINT 220
      PRINT 202,(U(I),V(I),I=1,NM1)
      GO TO 1
99    STOP
100   FORMAT (I1,8A10)
101   FORMAT (8E10.3)
200   FORMAT (1H1,2X,I1,2X,8A10)
201   FORMAT (1H0,5X*THE ROOTS OF C(S) ARE*13X*REAL PART*10X*IMAGINARY
      PART*)
202   FORMAT (32X,2E20.7)
203   FORMAT (1H0,5X*THE ROOTS OF P(S) ARE*13X*REAL PART*10X*IMAGINARY
     1PART*)
  220 FORMAT (1H0,5X*THE ROOTS OF HN(S) ARE*13X*REAL PART*10X*IMAGINARY
     1PART*)
204   FORMAT (1H0,*THE TRANSFER FUNCTION OF THE SERIES COMPENSATOR IS*)
205   FORMAT (1H0, *THE TRANSFER FUNCTION OF THE FEEDBACK COMPENSATOR IS
     1*)
      END
      SUBROUTINE PRNTRAT (A,NN,B,ND,DEPVAR,INDVAR)
      DIMENSION A(20),B(20),H(20),N(20),AT(20),BT(20)
      NE=1
      DO 6 I=1,NN
      AT(I)=0.
      IF (A(I)) 4,6,3
```

```
3       H(NE)=1H+
        AT(NE)=A(I)
        GO TO 5
4       H(NE)=1H-
        AT(NE)=-A(I)
5       N(NE)=I-1
        NE=NE+1
6       CONTINUE
        NE=NE-1
        PRINT 100
        MIN=MIN0(NE,10)
        PRINT 101,(N(I),I=1,MIN)
        PRINT 102,(H(I),AT(I),INDVAR,I=1,MIN)
        IF (NE-10) 8,8,7
7       PRINT 101,(N(I),I=11,NE)
        PRINT 102,(H(I),AT(I),INDVAR,I=11,NE)
8       NF=1



        DO 16 I=1,ND
        BT(I)=0.
        IF ( B(I) ) 14,16,13
13      H(NF)=1H+
        BT(NF)=B(I)
        GO TO 15
14      H(NF) =1H-
        BT(NF)=-B(I)
15      N(NF)=I-1
        NF=NF+1
16      CONTINUE
        NF=NF-1
        HH=1H-
        NH=MAX0(13*NE,13*NF)
        NH=MIN0(130,NH)
20      PRINT 103,DEPVAR,INDVAR,(HH, I=1,NH)
        MIN=MIN0(NF,10)
        PRINT 101,(N(I),I=1,MIN)
        PRINT 102,(H(I),BT(I),INDVAR,I=1,MIN)
        IF (NF-10)  28,28,27
27      PRINT 101,(N(I), I=11,NF)
        PRINT 102,(H(I),BT(I),INDVAR,I=11,NF)
28      CONTINUE
100     FORMAT (1H0)
101     FORMAT (6X,10(11X,I2))
102     FORMAT (6X,10(A1,G10.4 ,A1,1H ))
103     FORMAT (1X,A1,1H(,A1,2H)=,130A1)
        RETURN
        END
        SUBROUTINEPROOT(N,A,U,V,IR)
        DIMENSION A(20),U(20),V(20),H(21),B(21),C(21)
        IREV=IR
        NC=N+1
        DO1I=1,NC
        H(I)=A(I)
```

```
1       CONTINUE
        P=0.
        Q=0.
        R=0.
      3 IF(H(1))4,2,4
      2 NC=NC-1
        V(NC)=0.
        U(NC)=0.
        DO1002I=1,NC
        H(I)=H(I+1)
1002    CONTINUE
        GOTO3
      4 IF(NC-1)5,100,5
      5 IF(NC-2)7,6,7
      6 R=-H(1)/H(2)
        GOTO50
      7 IF(NC-3)9,8,9
      8 P=H(2)/H(3)
        Q=H(1)/H(3)
        GOTO70
9       IF(ABS(H(NC-1)/H(NC))-ABS(H(2)/H(1)))10,19,19
     10 IREV=-IREV
        M=NC/2
        DO11I=1,M
        NL=NC+1-I
```

```
        F=H(NL)
        H(NL)=H(I)
     11 H(I)=F
        IF(Q)13,12,13
     12 P=0.
        GOTO15
     13 P=P/Q
        Q=1./Q
     15 IF(R)16,19,16
     16 R=1./R
     19 E=5.E-10
        B(NC)=H(NC)
        C(NC)=H(NC)
        B(NC+1)=0.
        C(NC+1)=0.
        NP=NC-1
     20 DO49J=1,1000
        DO21I1=1,NP
        I=NC-I1
        B(I)=H(I)+R*B(I+1)
     21 C(I)=B(I)+R*C(I+1)
        IF(ABS(B(1)/H(1))-E)50,50,24
     24 IF(C(2))23,22,23
     22 R=R+1.
        GOTO30
     23 R=R-B(1)/C(2)
     30 DO37I1=1,NP
        I=NC-I1
        B(I)=H(I)-P*B(I+1)-Q*B(I+2)
```

```
   37 C(I)=B(I)-P*C(I+1)-Q*C(I+2)
      IF(H(2))32,31,32
31    IF(ABS(B(2)/H(1))-E)33,33,34
32    IF(ABS(B(2)/H(2))-E)33,33,34
33    IF(ABS(B(1)/H(1))-E)70,70,34
   34 CBAR=C(2)-B(2)
      D=C(3)**2-CBAR*C(4)
      IF(D)36,35,36
   35 P=P-2.
      Q=Q*(Q+1.)
      GOTO49
   36 P=P+(B(2)*C(3)-B(1)*C(4))/D
      Q=Q+(-B(2)*CBAR+B(1)*C(3))/D
   49 CONTINUE
      E=E*10.
      GOTO20
   50 NC=NC-1
      V(NC)=0.
      IF(IREV)51,52,52
   51 U(NC)=1./R
      GOTO53
   52 U(NC)=R
   53 DO54I=1,NC
      H(I)=B(I+1)
54    CONTINUE
      GOTO4
   70 NC=NC-2
      IF(IREV)71,72,72
   71 QP=1./Q
      PP=P/(Q*2.0)
      GOTO73


   72 QP=Q
      PP=P/2.0
   73 F=(PP)**2-QP
      IF(F)74,75,75
   74 U(NC+1)=-PP
      U(NC)=-PP
      V(NC+1)=SQRT(-F
      V(NC)=-V(NC+1)
      GOTO76
75    U(NC+1)=-(PP/ABS(PP))*(ABS(PP)+SQRT(F))
      V(NC+1)=0.
      U(NC)=QP/U(NC+1)
      V(NC)=0.
   76 DO77I=1,NC
      H(I)=B(I+2)
77    CONTINUE
      GOTO4
  100 RETURN
      END
```

Appendix 5 - STVFDBK

Discussion

A discussion of this program is given in reference 6. The input is formed from the state equations: $\dot{x} = Ax + bu$, $y = cx$ where x is the state vector, u is the input and y is the output. The major use of this program is to convert the state variable form of the plant to the equivalent open loop transfer functions.

Input Format

Card No.	Column	Description	Format
1	1-20	Identification of problem	4A5
	21-22	N = right order of A matrix, an integer right justified in the field	I2
2	1-10	a_{11} = first element of A matrix	8E10.0
	11-20	a_{12}	8E10.0
		. . .	. . .
3	1-10	a_{21}	8E10.0
	11-20	a_{22}	
		. . .	
n+2	1-10	b_1	8E10.0
	11-20	b_2	8E10.0
		. . .	

Input Format (cont'd)

Card No.	Column	Description	Format
n+3	1-10	c_1	8E10.0
	11-20	c_2	8E10.0
	etc.	⋮	⋮
n+4	1-80	Blank card	

Notes:

As many problem may be run as desired by repeating the above set of cards for each problem.

```
      PROGRAM STVFDBK (INPUT, OUTPUT, TAPE4 = INPUT)
C     OPEN LOOP CNLY
      DIMENSION  A(10,10),B(10),H(10),C(10),NAME(4),AA(10,10),E(10)
      DIMENSION D(10),P(10,10),CC(10),HH(10),PIN(10,10),U(10),V(10)
 2000 FORMAT(45H0    ****************************************)
2001  FORMAT (2A10,I2)
2002  FORMAT(8E10.0)
2003  FORMAT (1H1,5X,14HPROBLEM IDENT.   ,5X,2A10)
 2004 FORMAT(1H0,5X,12HTHE A MATRIX/)
2005  FORMAT(6E20.7)
2006  FORMAT(1H0,5X,12HTHE B MATRIX/)
 2007 FORMAT(1H0,5X,41HTHE CLOSED-LOOP CHARACTERISTIC POLYNOMIAL/)
 2008 FORMAT(1H0,5X,25HTHE FEEDBACK COEFFICIENTS/)
2009  FORMAT(1H0,5X,10HTHE GAIN = E16.7)
 2010 FORMAT(1H0,5X,12HTHE C MATRIX,5X,5H*****/)
2011  FORMAT(1H0,5X,24HDENOMINATOR COEFFICIENTS/)
2012  FORMAT(1H0,5X,22HNUMERATOR COEFFICIENTS/)
 2013 FORMAT(1H0,5X,23HTHE NUMERATOR OF H-EQUIVALENT/)
 2014 FORMAT(1H0,5X,22HOPEN-LOOP CALCULATIONS)
2015  FORMAT(1H0,5X,26HMAXIMUM NORMALIZED ERROR = E10.2/)
 2016 FORMAT(I1)
 2017 FORMAT(1H0,5X,24HCLOSED-LOOP CALCULATIONS)
 2018 FORMAT(1H0,5X,6HKEY = I1,3X,5H*****)
 2019 FORMAT(1H0,5X, 23HPLANT IS UNCONTROLLABLE5X,10H**********)
 2020 FORMAT(1H0,4X,  35HPLANT IS NUMERICALLY UNCONTROLLABLE10X,
     1 16HMAX. DEVIATION = E10.2,5X,10H**********)
 2021 FORMAT(1H0,5X,14HTHE ROOTS ARE ,13X,9HREAL PART,10X,14HIMAGINARY
     1ART)
 2022 FORMAT(25X,  2E20.7)
 2023 FORMAT(1H0)
C     READ INPUT DATA
1     READ 2001, (NAME(I),I=1,2),N
      IF(EOF,4)10,800
800   PRINT 2003,(NAME(I),I=1,2)
      PRINT 2004
      DO 2 I=1,N
      READ 2002,(A(I,J),J=1,N)
    2 PRINT 2005,(A(I,J),J=1,N)
      PRINT 2006
      READ 2002,(B(I),I=1,N)
      PRINT 2005,(B(I),I=1,N)
C     CHECK CONTROLLABILITY
      DO 7 I=1,N
      AA(I,1)=B(I)
7     CONTINUE
      DO 8 I=2,N
      L=I-1
      DO 8 J=1,N
      AA(J,I)=0.
      DO 8 K=1,N
      AA(J,I)=AA(J,I)+A(J,K)*AA(K,L)
8     CONTINUE
      CONTR=DET(AA,N)
      IF(CONTR) 3,4,3
    4 PRINT 2019
      GO TO 9
C     NOTE USE OF DUMMY ARGUMENT CALLED
    3 CALL SIMEQ(AA,C,N,PIN,C)
      DO 43 I=1,N
```

```
      DO 43 J=1,N
      P(I,J)=0.
      DO 43 K=1,N
      P(I,J)=P(I,J)+AA(I,K)*PIN(K,J)
43    CONTINUE
      ERROR=0.
      DO 44 I=1,N
      DO 44 J=1,N
      IF(I-J) 45,46,45
46    ERR = ABS(P(I,J)-1.0)
      GO TO 44
45    ERR = ABS(P(I,J))
44    ERROR = AMAX1(ERR,ERROR)
      IF(ERROR-1.E-5) 9,47,47
   47 PRINT 2020,ERROR
C     OPEN-LOOP CALCULATIONS
    9 PRINT 2000
      PRINT 2014
      NN=N+1
      PRINT 2011
      CALL CHREQ(A,N,D)
      PRINT 2005,(D(I),I=1,NN)
      CALL PROOT(N,D,U,V,+1)
      PRINT 2021
      PRINT 2022,(U(I),V(I),I=1,N)
      DO11 I=1,N
      P(I,N)=B(I)
11    CONTINUE
      DO12JJ=2,N
      DO12 I=1,N
      J=N-JJ+1
      K=J+1
      P(I,J)=D(K)*B(I)
      DO12 L=1,N
      P(I,J)=P(I,J)+A(I,L)*P(L,K)
12    CONTINUE
72    READ 2002, (C(I), I=1, N)
      DO 70  I=1, N
      IF(C(I)) 71,70,71
70    CONTINUE
      GO TO 1
71    PRINT 2023
      PRINT 2010
      PRINT 2005, (C(I),I=1,N)
   49 DO13 I=1,N
      CC(I)=0.
      DO13 J=1,N
      CC(I)=P(J,I)*C(J)+CC(I)
13    CONTINUE
      DO 100 I=1,N
      M=NN-I
      IF(CC(M))101,100,101
  100 CONTINUE
101   PRINT 2012
      PRINT 2005,(CC(I),I=1,M)
      M=M-1
      IF(M) 105,105,103
  103 CALL PROOT(M,CC,U,V,+1)
      PRINT 2021
      PRINT 2022,(U(I),V(I),I=1,M)
```

```
105     GO TO 72
   10 STOP
      END
      SUBROUTINEPROOT(N,A,U,V,IR)
      DIMENSION A(20),U(20),V(20),H(21),B(21),C(21)
      IREV=IR
      NC=N+1
      DO1I=1,NC
      H(I)=A(I)
1     CONTINUE
      P=0.
      Q=0.
      R=0.
    3 IF(H(1))4,2,4
    2 NC=NC-1
      V(NC)=0.
      U(NC)=0.
      DO1002I=1,NC
      H(I)=H(I+1)
1002  CONTINUE
      GOTO3
    4 IF(NC-1)5,100,5
    5 IF(NC-2)7,6,7
    6 R=-H(1)/H(2)
      GOTO50
    7 IF(NC-3)9,8,9
    8 P=H(2)/H(3)
      Q=H(1)/H(3)
      GOTO70
9     IF(ABS(H(NC-1)/H(NC))-ABS(H(2)/H(1)))10,19,19
   10 IREV=-IREV
      M=NC/2
      DO11I=1,M
      NL=NC+1-I
      F=H(NL)
      H(NL)=H(I)
   11 H(I)=F
      IF(Q)13,12,13
   12 P=0.
      GOTO15
   13 P=P/Q
      Q=1./Q
   15 IF(R)16,19,16
   16 R=1./R
   19 E=5.E-10
      B(NC)=H(NC)
      C(NC)=H(NC)
      B(NC+1)=0.
      C(NC+1)=0.
      NP=NC-1
   20 DO49J=1,1000
      DO21I1=1,NP
      I=NC-I1
      B(I)=H(I)+R*B(I+1)
   21 C(I)=B(I)+R*C(I+1)
      IF(ABS(B(1)/H(1))-E)50,50,24
   24 IF(C(2))23,22,23
   22 R=R+1.
      GOTO30
   23 R=R-B(1)/C(2)
```

```
   30 DO37I1=1,NP
      I=NC-I1
      B(I)=H(I)-P*B(I+1)-Q*B(I+2)
   37 C(I)=B(I)-P*C(I+1)-Q*C(I+2)
      IF(H(2))32,31,32
31    IF(ABS(B(2)/H(1))-E)33,33,34
32    IF(ABS(B(2)/H(2))-E)33,33,34
33    IF(ABS(B(1)/H(1))-E)70,70,34
   34 CBAR=C(2)-B(2)
      D=C(3)**2-CBAR*C(4)
      IF(D)36,35,36
   35 P=P-2.
      Q=Q*(Q+1.)
      GOTO49
   36 P=P+(B(2)*C(3)-B(1)*C(4))/D
      Q=Q+(-B(2)*CBAR+B(1)*C(3))/D
   49 CONTINUE
      E=E*10.
         2
   50 NC=NC-1
      V(NC)=0.
      IF(IREV)51,52,52
   51 U(NC)=1./R
      GOTO53
   52 U(NC)=R
   53 DO54I=1,NC
      H(I)=B(I+1)
54    CONTINUE
      GOTO4
   70 NC=NC-2
      IF(IREV)71,72,72
   71 QP=1./Q
      PP=P/(Q*2.0)
      GOTO73
   72 QP=Q
      PP=P/2.0
   73 F=(PP)**2-QP
      IF(F)74,75,75
   74 U(NC+1)=-PP
      U(NC)=-PP
      V(NC+1)=SQRT(-F)
      V(NC)=-V(NC+1)
      GOTO76
75    U(NC+1)=-(PP/ABS(PP))*(ABS(PP)+SQRT(F))
      V(NC+1)=0.
      U(NC)=QP/U(NC+1)
      V(NC)=0.
   76 DO77I=1,NC
      H(I)=B(I+2)
77    CONTINUE
      GOTO4
  100 RETURN
      END
      FUNCTION DET(A,N)
C        FUNCTION DET DETERMINES THE DETERMINANT OF THE MATRIX 'A'
      DIMENSION A(10,10),B(10,10)
C     SET 'B' EQUAL TO 'A' BECAUSE WE DESTROY 'A' IN THE PROCESS
      DO 1 IK=1,N
      DO 1 JK=1,N
      B(IK,JK) = A(IK,JK)
```

```
1       CONTINUE
        NN = N-1
        D = 1.0
C       IF N=1 THEN BYPASS PIVOT PROCEDURE. GO DIRECTLY TO CALCULATION
        IF(NN)69,69,3                                           OF DET.
C       START PIVOT SEARCH PROCEDURE
      3 DO 100 L=1,NN
        LL = L+1
        AMAX = A(L,L)
        IM = L
        JM = L
        DO 15 I=L,N
        DO 15 J=L,N
        IF (AMAX-ABS(A(I,J))) 10,15,15
     10 IM = I
        JM = J
        AMAX = ABS(A(I,J))
     15 CONTINUE
C       FOUND PIVOT AT ROW 'IM' AND AT COLUMN 'JM'
C       IF ALL REMAINING TERMS IN THE MATRIX ARE ZERO, SET DET=0. AND
        IF (AMAX) 70,70,14                                    RETURN.
C       NOW CHANGE ROWS AND COLUMNS IF NECESSARY
14      IF(IM-L) 16,20,16
     16 DO 17 J=1,N
        T = A(IM,J)
        A(IM,J) = A(L,J)
     17 A(L,J) = T
        D = -D
     20 IF(JM-L) 21,25,21
     21 DO 22 I=1,N
        T = A(I,JM)
        A(I,JM) = A(I,L)
     22 A(I,L) = T
        D = -D
C       PIVOT NOW AT A(L,L),DIVIDE ROW 'L' BY A(L,L)
     25 DO 30 K1=LL,N
        A(L,K1) = A(L,K1)/A(L,L)
30      CONTINUE
C       NOW PRODUCE ZEROES BELOW MAIN DIAGONAL,A(J,L)=0.
        DO 50 J=LL,N
        DO 50 K=LL,N
        A(J,K) = A(J,K)-A(J,L)*A(L,K)
50      CONTINUE
    100 CONTINUE
C       MULTIPLY MAIN DIAGONAL ELEMENTS TO GET VALUE OF THE DETERMINANT
     69 DO 200 I=1,N
        D = D*A(I,I)
200     CONTINUE
        DET = D
C       NOW RESTORE THE VALUES OF THE 'A' MATRIX
        DO 2 IK=1,N
        DO 2 JK=1,N
        A(IK,JK) = B(IK,JK)
2       CONTINUE
        RETURN
70      DET=0.
        DO 4 JK=1,N
        DO 4 IK=1,N
        A(IK,JK) = B(IK,JK)
4       CONTINUE
```

```
      RETURN
      END
      SUBROUTINE SIMEQ (A,XDOT,KC,AINV,X)
      DIMENSION A(10,10),B(10,10),XDOT(10),X(10),AINV(10,10)
      DO1 I=1,KC
      DO1 J=1,KC
      AINV(I,J)=0.
    1 B(I,J)=A(I,J)
      DO2 I=1,KC
      AINV(I,I)=1.
    2 X(I)=XDOT(I)
      DO3 I=1,KC
      COMP=0.
      K=I
      IF(ABS(B(K,I))-ABS(COMP))5,5,4
    4 COMP=B(K,I)
      N=K
    5 K=K+1
      IF(K-KC)6,6,7
    7 IF(B(N,I))8,51,8
    8 IF(N-I)51,12,9
    9 DO10 M=1,KC
      TEMP=B(I,M)
      B(I,M)=B(N,M)
      B(N,M)=TEMP
      TEMP=AINV(I,M)
      AINV(I,M)=AINV(N,M)
   10 AINV(N,M)=TEMP
      TEMP=X(I)
      X(I)=X(N)
      X(N)=TEMP
   12 X(I)=X(I)/B(I,I)
      TEMP = B(I,I)
      DO13 M=1,KC
      AINV(I,M) = AINV(I,M)/TEMP
   13 B(I,M) = B(I,M)/TEMP
      DO16 J=1,KC
      IF(J-I)14,16,14
   14 IF(B(J,I))15,16,15
   15 X(J)=X(J)-B(J,I)*X(I)
      TEMP=B(J,I)
      DO17 N=1,KC
      AINV(J,N)=AINV(J,N)-TEMP*AINV(I,N)
   17 B(J,N)=B(J,N)-TEMP*B(I,N)
   16 CONTINUE
    3 CONTINUE
      RETURN
   51 PRINT52,I,KC
   52 FORMAT( 18H0 ERROR IN COLUMN I2,2X,3HOF MATRIX ,5X,3HKC=I2//)
      RETURN
      END
      SUBROUTINE CHREQ(A,N,C)
      DIMENSION J(11),C(11),B(10,10),A(10,10),D(300)
      NN=N+1
      DO20 I=1,NN
      C(I)=0.
20    CONTINUE
      C(NN) = 1.
      DO14 M=1,N
      K=0
```

```
      L=1
      J(1)=1
      GO TO 2
    1 J(L)=J(L)+1
    2 IF(L-M) 3,5,50
    3 MM=M-1
      DO 4 I=L,MM
      II=I+1
    4 J(II)=J(I)+1
    5  CALL FORM(J,M,A,B)
      K=K+1
      D(K)=DET(B,M)
      DO 6 I=1,M
      L=M-I+1
      IF(J(L)-(N-M+L)) 1,6,50
    6 CONTINUE
      M1 = N-M+1
      DO14 I=1,K
      C(M1)=C(M1)+D(I)*(-1.)**M
      CONTINUE
      RETURN
   50 PRINT 2000
      RETURN
 2000 FORMAT (1H0,5X,14HERROR IN CHREQ)
      END
      SUBROUTINE FORM(J,M,A,B)
      DIMENSION A(10,10),B(10,10),J(11)
      DO1 I=1,M
      DO1 K=1,M
      NR=J(I)
      NC=J(K)
    1 B(I,K)=A(NR,NC)
      RETURN
      END
```

Appendix 6 - BODE4

Discussion

This program is used for providing phase and log magnitude versus frequency plots for the resulting open and closed loop transfer functions. The author is indebted to Dr. L. P. Huelsman for this program.

Input Format

Card No.	Column	Description	Format
	1-80	LTR = Title of problem	80A1
2	3	NC = No. of functions to be plotted	I3
3	1-3	NP = No. of values of frequency in the plot (1-100).	I3
	4-6	NPD = No. of frequency points/decade	I3
	7-9	NHZ = Log_{10} of starting frequency in Hz	I3
	10-12	NSCAL = Max. ordinate of magnitude on frequency plot in db times SCALM (minimum value is 100 SCALM) db lower).	I3
	13-15	NPHS = Indicator for phase plot, 0 if no plot desired, 1 if plot is desired	I3
	16-18	MSCAL = Max. ordinate for phase plot in degrees times SCALP (minimum ordinate (100 SCALP) lower)	I3
	21-30	SCALM = Scale factor for magnitude data (1.0 if no scaling desired)	E10.0
	31-40	SCALP = Scale factor for phase data (1.0 if no scaling desired)	E10.0

Input Format (cont'd)

Card No.	Column	Description	Format
4	1-3	ND = Max. degree of numerator and denominator polynomials (1-19)	I3
5	1-80	A = Numerator coefficients for first function (in ascending order, use 10 columns for each coefficient)	8E10.0
6	1-80	B = Denominator coefficients for first function (in ascending order, use 10 columns for each coefficient)	8E10.0

Notes:

1. The maximum magnitude in db which will be plotted is NSCAL/SCALM; however, the maximum value indicated on the plot will be NSCAL.
2. The minimum magnitude in db which will be plotted is (NSCAL-100/ SCALM; however, the minimum value indicated on the plot is NSCAL-100.
3. The above comments also apply to the phase.
4. The actual values of the variables are listed in the printout for every frequency.
5. The frequency range in Hz is from 10^{NHZ} to $10^{NHZ + NP/ND}$.
6. For each function to be plotted cards 4 through 6 must be repeated. As many cards as necessary may be used to specify A and B when ND 8.

Examples

If the function varies from +10 to -10 db then choose SCALM = 5, and NSCAL = 50 since

$$NSCAL/SCALM = 10$$

$$(NSCAL-100)/SCALM = -10$$

If the minimum frequency is .01 Hz and the maximum frequency of the plot is 100 Hz with 20 points per decade then the total number of points is

$$NP = decades(points/decade) = (\log 10 \text{ (max. freq.)} - NHZ) \ NPD$$

$$= 4 \cdot 20 = 80$$

```
      PROGRAM BODE4 (INPUT,OUTPUT, TAPE 4=INPUT)
      DIMENSION LTR(80)
      DIMENSION Y(5,100)




      DIMENSION X(10)
 1    PHSD=0.
      READ 100, LTR
      IF (EOF,4) 999,2
 2    PRINT 101, LTR
      READ 105,NC
      PRINT 110,NC
  102 READ 105,NP,NPD,NHZ,NSCAL  ,NPHS,MSCAL  ,SCALM,SCALP
      PRINT 112,NP,NPD,NHZ
      DO 140 K=1,NC
      PRINT 115,K
      READ 77,(X(I),I=1,5)
 77   FORMAT(8F10.4)
      PRINT 78,(X(I),I=1,5)
 78   FORMAT(* X(I)* / 1X,10F12.3)
      NX=1
      PRINT 201
      DO 140 J=1,NP
      FRQ=XL4(NX,NHZ,NPD)
      RAD=6.28318*FRQ
      CALL MAG77(RAD,FM,X,PHSD)
  137 FML=20.*ALOG10(FM)
      Y(K,J)=FML*SCALM
  140 PRINT 202,J,FRQ,RAD,FM,FML,PHSD
      PRINT 145, SCALM,LTR
      CALL PLOT4(Y,NC,NP,NSCAL)
      GO TO 1
  999 STOP
  100 FORMAT (80A1)
  101 FORMAT (1H1, 80A1)
  105 FORMAT (6I3,2X,2E10.0)
  110 FORMAT (1H0,16HNUMBER OF CASES=,I2)
  112 FORMAT (1H0,I3,7H POINTS, I10,7H/DECADE,5X,18HSTARTING FROM 10**,
     1  I3)
  115 FORMAT (1H0,/12H CASE NUMBER, I2)
  145 FORMAT (1H1,14HMAGNITUDE DATA ,14H SCALE FACTOR=,  E10.3,1X,80A1/
  201 FORMAT (1H0,25X,5HHERTZ,13X,7HRAD/SEC,17X,3HMAG,18X,2HDB,17X,3HDE(
     1/)
  202 FORMAT (1X,I5,5X,5E20.8)
      END
      SUBROUTINE PLOT4(Y,M,NF,NS)
C     SUBROUTINE FOR PLOTTING 5 X 100 INPUT ARRAY (FORTRAN 4)
      DIMENSION Y(5,100), LINE(101),L(11),JL(5)
      DATA (JL(I),I=1,5)/1HA,1HB,1HC,1HD,1HE/, JN,JP,JI,JBLANK,JZ/
     11H-,1H+,1HI,1H ,1H$/
      DO 99 I=1,101
      LINE(I)=JBLANK
   99 CONTINUE
      N=0
```

```
C     PRINT ORDINATE SCALE
      DO 101 I=1,11
      L(I)=10*I-110+NS
  101 CONTINUE
      PRINT 105,(L(I),I=1,11)
  105 FORMAT (3X,11(I4,6X),6HY(1,I))
      GO TO 115
  110 IF (N/10-(N-1)/10) 125,125,115
C     CONSTRUCT ORDINATE GRAPH LINE
  115 ND=0
      DO 120 I=1,10
```

```
      ND=ND+1
      LINE(ND)=JP
      DO 120 J=1,9
      ND=ND+1
  120 LINE(ND)=JN
      LINE(101)=JP
      IF (N) 135,121,135
  121 PRINT 170,N,LINE
      GO TO 185
C     CONSTRUCT 1 LINE OF ABSCISSA GRAPH LINES
  125 DO 130 I=1,101,10
      LINE(I)=JI
  130 CONTINUE
C     CHANGE NUMERICAL DATA TO LETTERS
  135 DO 160 I=1,M
      XNS=NS
      JA=Y(I,N)+101.49999-XNS
      IF (JA-101) 140,155,145
  140 IF (JA) 150,150,155
  145 LINE(101)=JZ
      GO TO 160
  150 LINE(1)=JZ
      GO TO 160
  155 LINE(JA)=JL(I)
  160 CONTINUE
C     PRINT LINE OF DATA
      IF (N/10-(N-1)/10) 175,175,165
  165 PRINT 170,N,LINE,Y(1,N)
  170 FORMAT (1X,I4,101A1,1X,  E12.5)
      GO TO 185
  175 PRINT 180,LINE,Y(1,N)
  180 FORMAT (5X,101A1,1X,E12.5)
C     SET LINE VARIABLES TO ZERO
  185 DO 190 I=1,101
      LINE(I)=JBLANK
  190 CONTINUE
  195 N=N+1
      IF (N-NF) 110,110,200
```

```
200 RETURN
    END
    FUNCTION  XL4 (N,LO,LP)
    AN=N
    ALO=LO
    ALP=LP
    AA=AN/ALP+ALO
    IF (AA) 100,100,105
100 XL4=1./10.**(-AA)
    GO TO 110
105 XL4=10.**AA
110 N=N+1
    RETURN
    END
    SUBROUTINE MAG77(S,TAMP,X,PHASE)
    DIMENSION X(10)
    COMPLEX GN,GD,DN,DD,PN,CN,CD,T
    GN   =CMPLX(79.8,S+172.0)*CMPLX(79.8,S-172.0) * 388.0 *
   1 CMPLX(176.0,S+331.0) * CMPLX(176.0,S-331.0) *
   2 CMPLX(80.8,S+575.0) *CMPLX(80.8,S-575.0) *
   3 CMPLX(82.8,S+945.0) * CMPLX(32.8,S-945.0)
    GD   =CMPLX(46.0,S)*CMPLX(44.5,S+282.0)*CMPLX(44.5,S-282.0) *
   1 CMPLX(84.4,S+478.0) * CMPLX(84.4,S-478.0) *
   2 CMPLX(130.5,S+740.0) * CMPLX(130.5,S-740.0) *
   3 CMPLX(191.5,S+1082.0) * CMPLX(191.5,S-1082.0)
    DN=5.15E+19*CMPLX(1030.0,S)
    DD=CMPLX(650.0,S)*CMPLX(1260.0,S+588.0)*CMPLX(1260.0,S-588.0)
   1  *CMPLX(261.0,S+1470.0)*CMPLX(261.0,S-1470.0)
   2  *CMPLX(3910.0,S+1810.0)*CMPLX(3910.0,S-1810.0
    PN=     CMPLX(1010.0,S) / 1010.0
    CN=1.335*CMPLX(142.,S+248.)*CMPLX(142.,S-248.)
    CD=CMPLX(0.,S)*CMPLX(500.,S)
    T=(GN*DD*CD)/(GD*DD*CD+GN*DN*CN*PN)
    TAMP=CABS(T)
    A=REAL(T)
    B=AIMAG(T)
    PHASE=ATAN2(B,A)*57.3
    RETURN
    CN=X(3)*CMPLX(X(1),S) *CMPLX(X(4),S)
    CD=CMPLX(X(2),S)*CMPLX( X(5),S) *CMPLX(0.,S)
    T=(GN/GD)*(DN/DD)*(CN/CD)*(PN)
    END
```

Appendix 7 - Case Study I

Second Order Plant

Discussion

A second order plant has been selected so that some insight into the effect of compensators can be obtained before the computer programs are used. Suppose that the plant transfer function is

$$G(s) = \frac{100}{s(s+b)} \tag{6-1}$$

and the desired transfer function is a Butterworth type response for maximum bandwidth,

$$T(s) = \frac{100}{s^2+\sqrt{2}\cdot 10s+100} \tag{6-2}$$

The feedback compensator will be of the form

$$H(s) = \frac{H_1+H_2 s}{P_1+s} \tag{6-3}$$

where arbitrarily $P_2=1$ since the numerator and denominator may be multiplied by any constant. For zero steady state error at low frequencies

$$H_1 = P_1$$

The transfer function of the series compensator is of the form

$$C(s) = \frac{P_1+s}{CD_1+CD_2 s} \tag{6-4}$$

From the relation between the closed loop transfer function and the open loop transfer function,

$$T(s) = \frac{G(s)C(s)}{1+G(s)C(s)H(s)}$$

$$\frac{100}{s^2+\sqrt{2}\,10s+100} = \frac{100(P_1+s)}{s(s+b)(CD_1+CD_2s)+100(H_1+H_2s)} \tag{6-5}$$

Cross multiplying, and equating coefficients result in the equations,

$$\begin{array}{lrcl}
s^3 & 1 & = & CD_2 \\
s^2 & \sqrt{2}\,10+P_1 & = & CD_1+1\ CD_2 \\
s^1 & 100+\sqrt{2}\,10\ P_1 & = & bCD_1+100H_2 \\
s^0 & P_1 & = & H_1
\end{array}$$

Solving for CD_2, CD_1, and H_2, the above equations become

$$\begin{aligned}
CD_2 &= 1 \\
CD_1 &= \sqrt{2}\,10+P_1-b \\
H_2 &= 1+(P_1-b)(\sqrt{2}/10 - b/100) \\
H_1 &= P_1
\end{aligned}$$

Thus the compensators are specified except for b, and P_1.

Sensitivity

The sensitivity of the closed loop system to variations in plant gain is

$$S_T^K = \frac{T(s)}{G(s)C(s)} = \frac{s(s+b)\quad(CD_1+s)}{s^2+\sqrt{2}\,10s+100)(P_1+s)} \tag{6-6}$$

The low frequency sensitivity is

$$S_K^T = \frac{b \cdot CD_1 \cdot s}{100P_1} \tag{6-7}$$

which is minimum for $CD_1 = 0$ which implies

$$\begin{aligned} CD_1 &= \sqrt{2}\,10 + P_1 - b = 0 \\ P_1 &= b - \sqrt{2}\,10 \end{aligned} \tag{6-8}$$

Unless P_1 is positive the feedback compensator will be unstable and the overall system will be infinitely sensitive to variations in P_1. If P_1 is positive, then (6-8) is a restriction on the plant time constant. The only alternative is to make P_1 sufficiently large to keep the sensitivity low. cf. eqn. 6-7. It is interesting to compare the sensitivity of this system with the Heq compensator and the Guilleman-Truxal compensator.

<u>Heq Compensator</u>

Choose Heq to be $Heq(s) = 1+k_2s$. Then

$$T(s) = \frac{G(s)}{1+G(s)Heq(s)}$$

$$\frac{100}{s^2+10\sqrt{2}\,s+100} = \frac{100}{s(s+b)+100(1+k_2s)}$$

Equating coefficients,

$$10\sqrt{2} = b+100k_2$$

$$k_2 = \frac{10\sqrt{2}-b}{100}$$

$$S_K^T = \frac{T(s)}{G(s)} = \frac{s(s+b)}{s^2+10\sqrt{2}\,s+100}$$

Guilleman-Truxal

Choose

$$C(s) = \frac{s+b}{s+c}$$

$$\frac{100}{s^2+10\sqrt{2}\,s+100} = \frac{100}{s(s+c)+100}$$

Cross multiplying and equating coefficients,

$$c = 10\sqrt{2}$$

Thus, the sensitivity is

$$S_K^T = \frac{s(s+10\sqrt{2})}{(s^2+\sqrt{2}\,10s+100)}$$

The above sensitivities are plotted in Fig. 6-1 for the case of where b = 20, and $CD_1 = 0$.

If b = 10, then for P_1 positive, the sensitivity is minimized if P_1 is made large. CD_1/P_1 is nearly one if $P_1 = 40$. The three sensitivities are shown below.

MODOBS

$$S_K^T = \frac{s(s+10)(s+44.14)}{(s^2+10\sqrt{2}s+100)(s+40)}$$

Heq

$$S_K^T = \frac{s(s+10)}{(s^2+10\sqrt{2}s+100)}$$

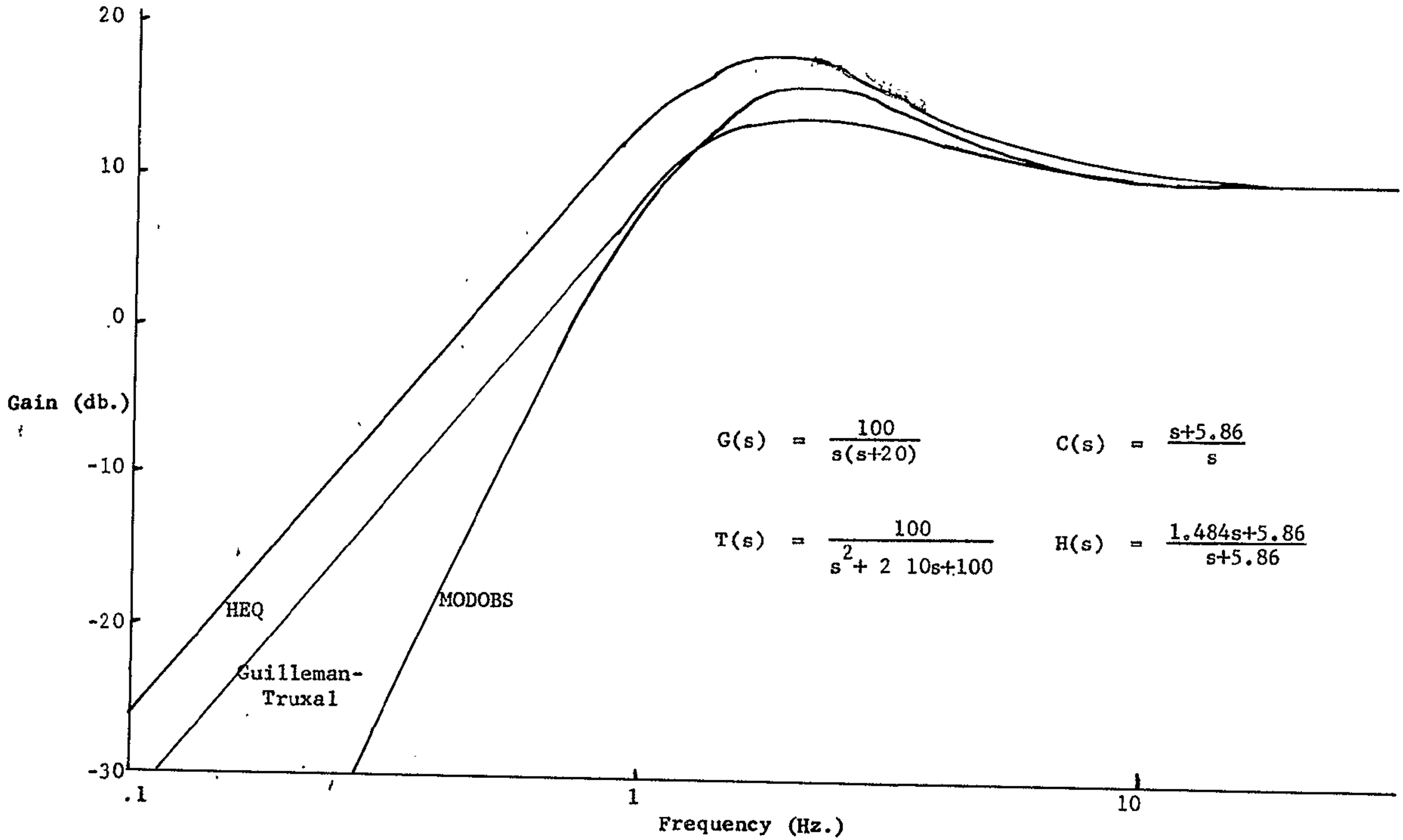


Figure 6.1 Sensitivity of High-performance Plant

Guilleman-Truxal

$$S_K^T = \frac{s(s+10\sqrt{2})}{(s^2+10\sqrt{2}s+100)}$$

Notice that for b = 20 the MODOBS system is significantly better at all frequencies below 1Hz. The Heq compensation is both unrealizable and largest in sensitivity for this case. When the forward gain is small then no compensation will help the sensitivity. Thus all the proposed systems have a sensitivity near one for frequencies above 1Hz.

For b = 10, the low frequency plant gain must be greater to achieve the same T(s). The Guilleman-Truxal compensation decreases the forward gain proportionately, so that the sensitivity is unchanged. The Heq compensation does not change the forward gain; therefore the sensitivity is reduced from the case where b = 20. The MODOBS system also improves in sensitivity, but this system will only be as low in sensitivity as the Heq method when P_1 is infinitely large. Nevertheless, the MODOBS compensation has the advantage of being realizable.

Integral Sensitivity

For the plant with a pole at -20, the integral sensitivity (weighted by 1/s) is

$$S = \frac{1}{2\pi} \int_{-j\infty}^{j\infty} \left| \frac{1}{s} \frac{s(s+20)(s+P_1-5.86)}{(s^2+10\sqrt{2}\,s+100)(s+P_1)} \right|^2 ds$$

This integral can be evaluated directly from Newton, Gould and Kaiser, p. 372.

The minimum low frequency sensitivity was attained for $P_1 = 5.86$ which from MODOBS3 has an integral sensitivity of 0.12. The minimum integral sensitivity is 0.085 for $P_1 = 8$. In this simple case this can be determined analytically by plotting S as P_1 is varied.

Final Design

From the integral sensitivity data, a final design choice is

$$b = 20$$

$$P_1 = +8$$

The following graphs are shown in Figs. 6-2 to 6-4.

Fig. 6-2 Plant transfer function

Fig. 6-3 Sensitivity function

Fig. 6-4 Loop gain function

Conclusion

From the above figures the design demonstrates the dependence of the sensitivity on the plant parameters. If the time constants in the closed loop transfer function are much greater than those in the plant, a sacrifice in the system sensitivity at low frequencies must be made. In addition the compensator will have large gains associated with the feedback which complicate the design of the compensators and increase the effect of the measurement noise.

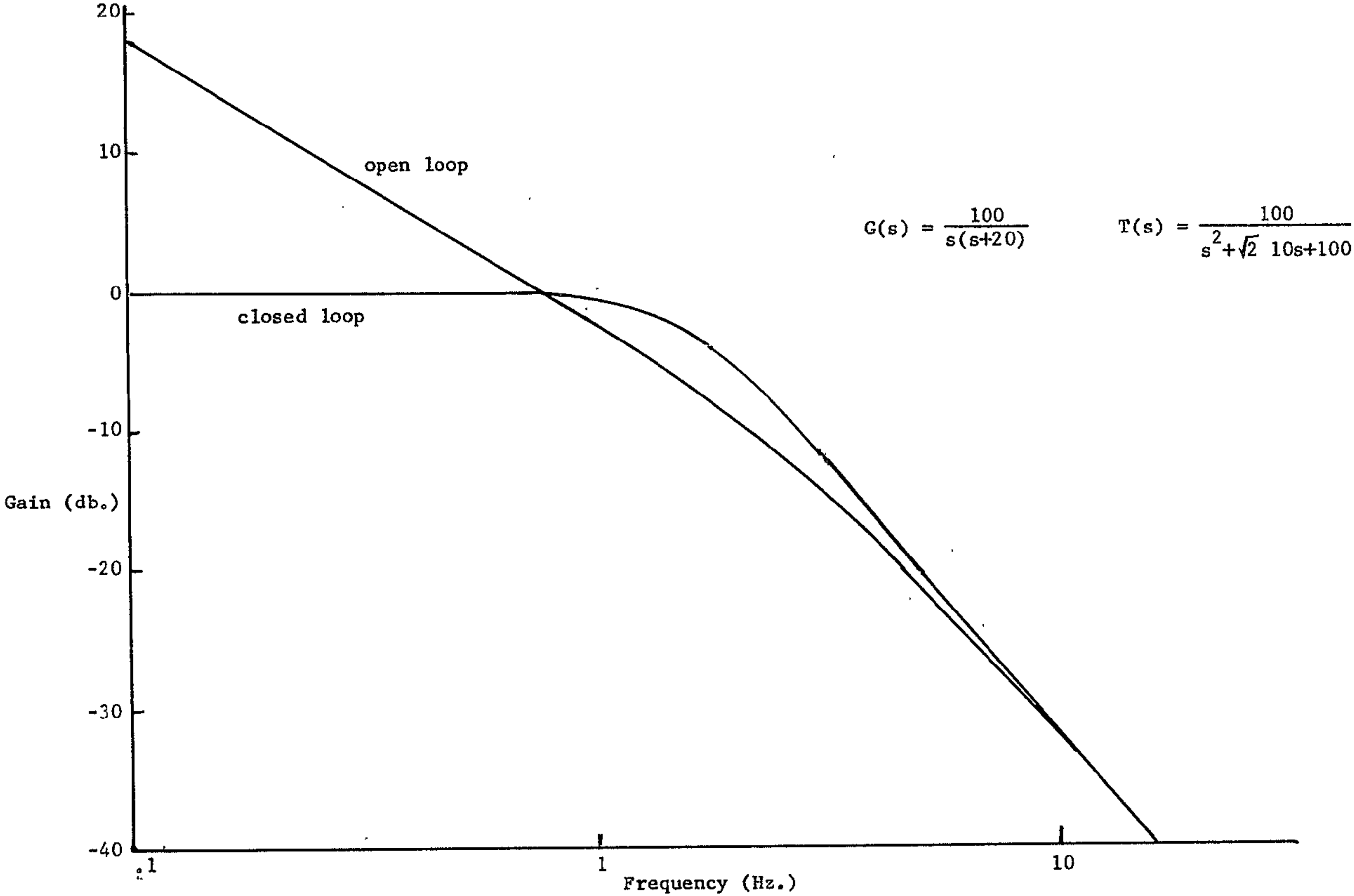


Figure 6.2 Open and Closed Loop Transfer Functions

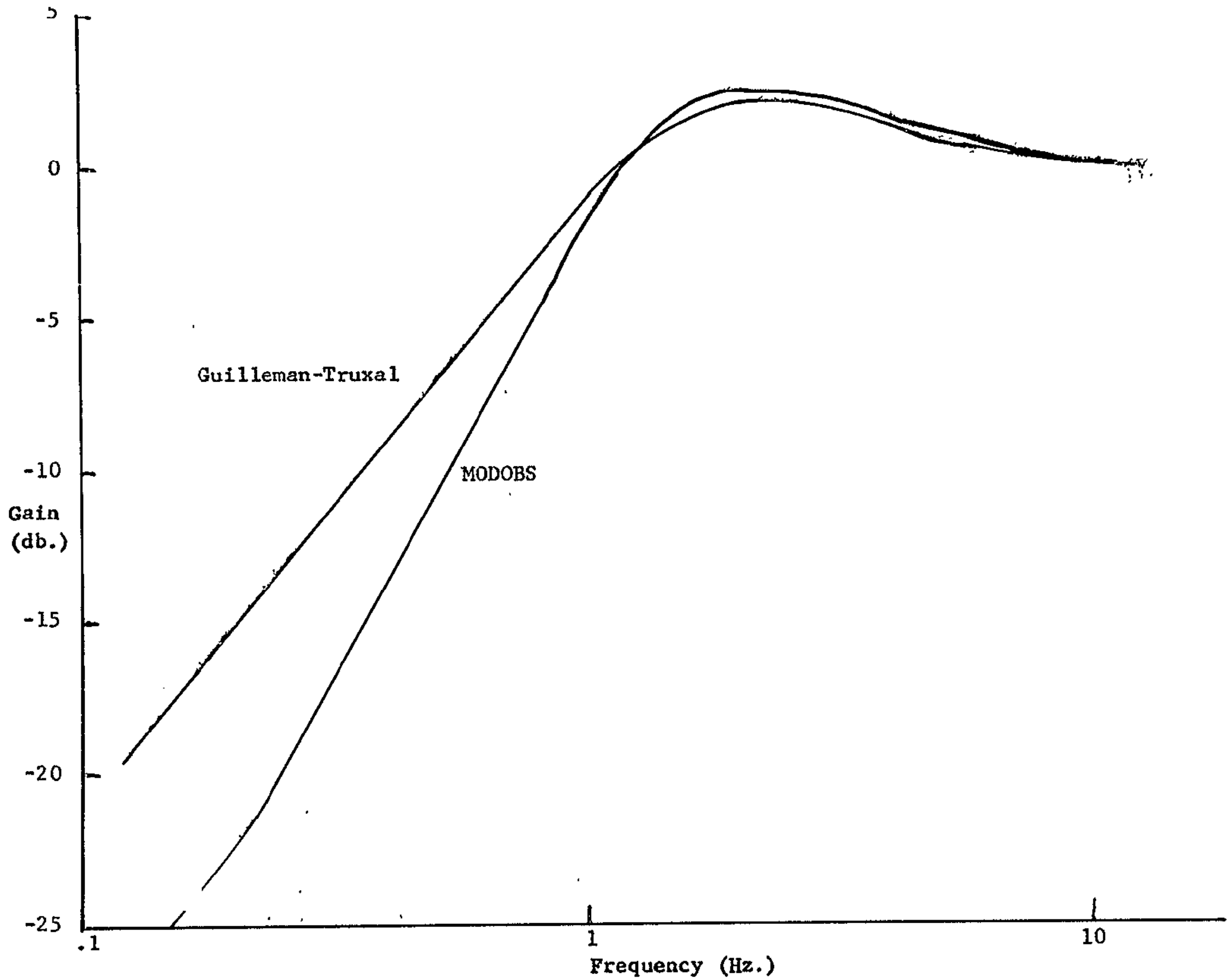


Figure 6.3 Sensitivity of Second Order Systems

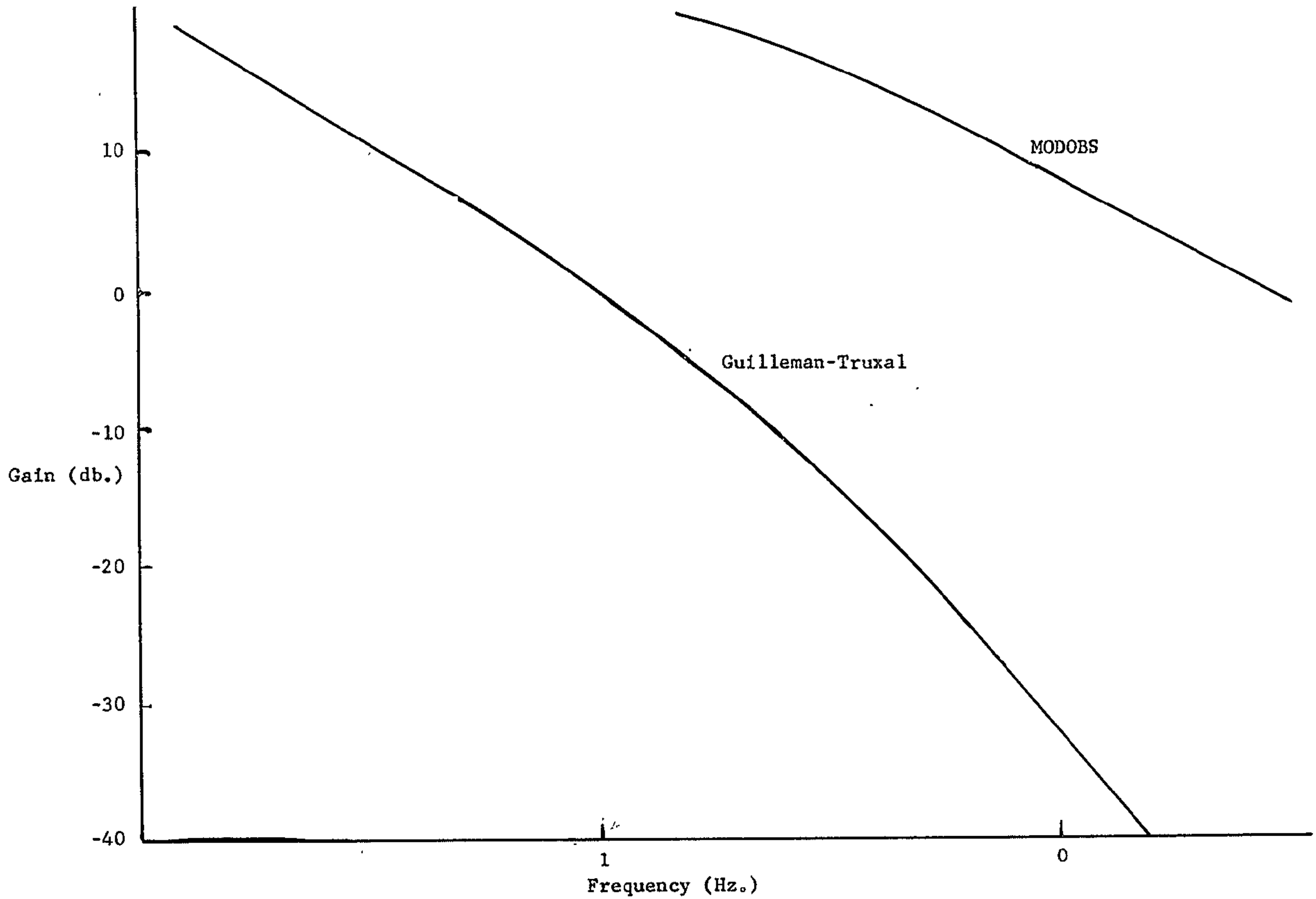


Figure 6.4 Loop Gain Function of Second Order System

Appendix 8 - Case Study II

Third Order Plant

Discussion

A third order plant has been selected to compare the computer results with hand calculations. The results agree in all the calculations discussed below. The plant transfer function is chosen to be of the form

$$G(s) = \frac{1000.}{s^3+GD_3s^2+GD_2s+GD_1}$$

and the desired closed loop transfer function is again chosen with a Butterworth type denominator,

$$T(s) = \frac{1000}{s^3+20s^2+200s+1000}$$

MODOBS2

The equations for the coefficients of $P(s)=P_1+P_2s+P_3s^2$ which give minimum low frequency sensitivity were derived in Appendix 2. For N=3,

$$P_3 = 1$$

$$P_2 = GD_3 - 20$$

$$P_1 = GD_2 - 200 - 20(P_2)$$

These equations are satisfied if

$$GD_3 = 20$$

$$GD_2 = 200$$

Low frequency sensitivity will be improved if the plant is "Type 1" or equivalently, has a pole at the origin; thus

$$GD_1 = 0$$

The poles of the plant are the roots of

$$s(s^2+20s+200) = 0$$

$$s = -10 \pm j10$$

For the plant to have real roots then,

$$GD_3^2 - 4GD_2 > 0$$

and in order for P_1 and P_2 to be positive,

$$GD_3 > 20$$

$$GD_2 > 200 + 20P_2$$

These requirements can be met if

$$GD_3 > 68.28$$

$$20(GD_3 - 10) < GD_2 < GD_3^2 / 4$$

$$GD_1 = 0$$

If GD_3 is near 68 then P_1 must be near zero. Under these circumstances the zero of CD(s) cancels the pole at a low frequency, and the low frequency sensitivity suffers. Suppose that

$$GD_3 = 100$$

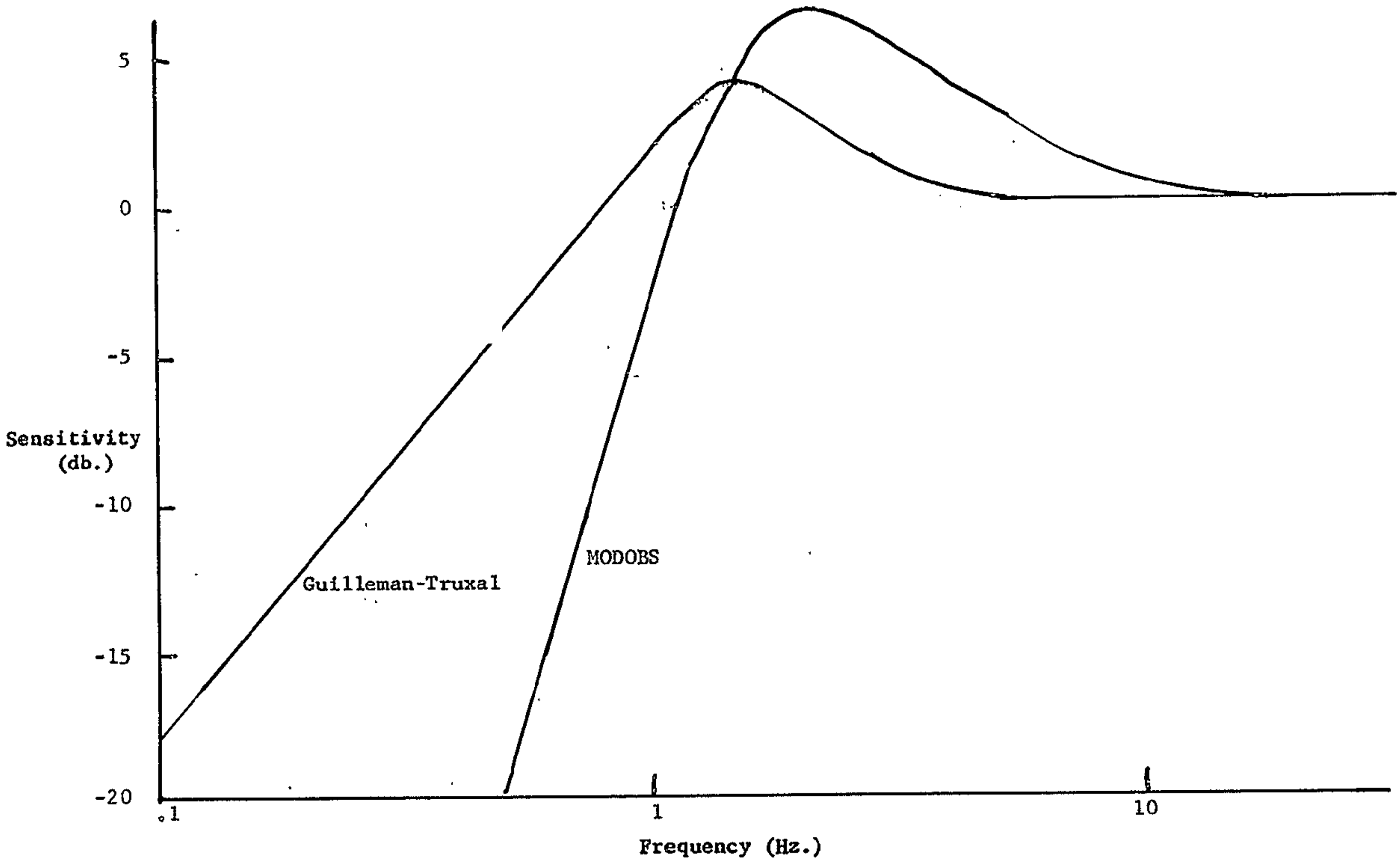


Figure 8.1 Sensitivity of Third Order System

Then

$$1800 < GD_2 < 2500$$

$$P3 = 1$$

$$P2 = GD_3 - 20 = 80$$

$$P1 = GD_2 - 1800$$

To keep the low frequency pole of P(s) as large as possible it is best to make GD2 large; thus

GD3 = 100

GD2 = 2500

GD1 = 0

P3 = 1

P2 = 80

P1 = 700

Sensitivity

If the plant can be selected as above, the sensitivity becomes

$$S^T_K = \frac{T(s)}{G(s)C(s)} = \frac{s(s^2+100s+2500) \quad s^2}{(s^3+20s^2+200s+1000)(s^2+80s+700)}$$

For this choice of the plant the system will have minimum low frequency sensitivity and a stable feedback compensator. A comparison of the sensitivity of this system with only a series compensator which achieves the same closed loop response is shown in Fig. 8-1. Except for frequencies between 1.41 Hz and 14.1 Hz the sensitivity is improved by using the MODOBS system.

Discussion

Again, the importance of choosing a plant which is consistent with the requirements of the overall system is evident. A slow system cannot be made to react quickly and also have good sensitivity characteristics using second order compensators.

Appendix 9

Case III In core Thermionic Reactor

The plant and the desired transfer function for an in core thermionic reactor have been discussed by Weaver et al [7]. The characteristics of the system are given in the above report. This appendix addresses itself to the question of realization of the desired transfer function for the fourth order model when only the output is available. The plant transfer function is given by

$$G(s) = \frac{(s+.0326)(s+.1)(s+1.79)\ 200}{(s+.0119+j.0164)(s+.0119-j.0164)(s+.00663)(s+1.85)}$$

and the desired transfer function is

$$\frac{Y(s)}{R(s)} = \frac{200(s+.0326)(s+.1)(s+1.79)}{100(s+.0326)(s+.1)(s+1.79)(s+2)}$$

When these terms are multiplied out the resulting transfer functions are

$$G(s) = \frac{1.16708+48.1228s+384.52s^2+200s^3}{5.03585\cdot10^{-6}+1.0542\cdot10^{-3}+.0568639s^2+1.8804s^3+s^4}$$

$$\frac{Y(s)}{R(s)} = \frac{1.16708+48.1228s+34.52s^2+200s^3}{1.16708+48.70635s+408.5815s^2+392.26s^3+100s^4}$$

This problem represents a special case in that the numerator of G(s) and $Y/R^{(o)}$ is the same. Equating the real and desired closed loop transfer functions yields

$$T(s) \equiv \frac{Y}{R}(s) = \frac{G^{(s)}C(s)}{1+G(s)C(s)H(s)}$$

and in terms of the numerator and denominator polynomials,

$$\frac{TN(s)}{TD(s)} = \frac{GN(s)CN(s)HD(s)}{GD(s)CD(s)HD(s)+GN(s)CN(s)HN(s)}$$

Since CN(s) = HD(s) ≡ P(s) and TN(s) = GN(s),

$$\frac{1}{TD(s)} = \frac{P(s)}{GD(s)CD(s)+GN(s)HN(s)}$$

Equating the like terms in the above equation and expressing the result in vector matrix form, one gets

$$\begin{bmatrix} GD(1) & 0 & 0 & 0 \\ GD(2) & GD(1) & 0 & 0 \\ GD(3) & GD(2) & GD(1) & 0 \\ GD(4) & GD(3) & GD(2) & GD(1) \\ GD(5) & GD(4) & GD(3) & GD(2) \\ 0 & GD(5) & GD(4) & GD(3) \\ 0 & 0 & GD(5) & GD(4) \\ 0 & 0 & 0 & GD(5) \end{bmatrix} \begin{bmatrix} CD(1) \\ CD(2) \\ CD(3) \\ CD(4) \end{bmatrix} + \begin{bmatrix} GN(1) & 0 & 0 & 0 \\ GN(2) & GN(1) & 0 & 0 \\ GN(3) & GN(2) & GN(1) & 0 \\ GN(4) & GN(3) & GN(2) & GN(1) \\ 0 & GN(4) & GN(3) & GN(2) \\ 0 & 0 & GN(4) & GN(3) \\ 0 & 0 & 0 & GN(4) \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} HN(1) \\ HN(2) \\ HN(3) \\ HN(4) \end{bmatrix}$$

$$= \begin{bmatrix} TD(1) & 0 & 0 & 0 \\ TD(2) & TD(1) & 0 & 0 \\ TD(3) & TD(2) & TD(1) & 0 \\ TD(4) & TD(3) & TD(2) & TD(1) \\ TD(5) & TD(4) & TD(3) & TD(2) \\ 0 & TD(5) & TD(4) & TD(3) \\ 0 & 0 & TD(5) & TD(4) \\ 0 & 0 & 0 & TD(5) \end{bmatrix} \begin{bmatrix} P(1) \\ P(2) \\ P(3) \\ P(4) \end{bmatrix}$$

Plugging in the appropriate coefficients from the above expressions for TD(s), GN(s), and GD(s) the unknowns GD(s) and HN(s) are equal to

$$\begin{bmatrix} CD(1) \\ CD(2) \\ \cdot \\ \cdot \\ \cdot \\ HN(1) \\ \cdot \\ \cdot \\ \cdot \\ HN(4) \end{bmatrix} = \begin{bmatrix} 1.19114E{+}01 & -5.44224E{+}00 & 9.63714E{+}00 & -1.72292E{+}01 \\ 1.41485E{+}02 & -2.12494E{+}02 & 3.91935E{+}02 & -7.00789E{+}02 \\ 9.32689E{+}02 & -1.65161E{+}03 & 2.96273E{+}03 & -5.28468E{+}03 \\ 3.31178E{-}14 & 3.56005E{-}14 & -1.10191E{-}13 & 1.00000E{+}01 \\ 9.99949E{-}01 & 2.34842E{-}05 & -4.15859E{-}05 & 7.43467E{-}05 \\ 4.96319E{-}01 & 1.00486E{+}00 & -8.68215E{-}03 & 1.55222E{-}02 \\ 5.09230E{-}01 & 7.61501E{-}01 & 5.35279E{-}01 & 8.30837E{-}01 \\ 4.66345E{+}00 & 8.25805E{+}00 & -1.47636E{+}01 & 2.82909E{+}01 \end{bmatrix} \begin{bmatrix} P(1) \\ P(2) \\ P(3) \\ P(4) \end{bmatrix}$$

The resulting expression can also be solved for P(s) in terms of C(s), that is,

$$\begin{bmatrix} P(1) \\ P(2) \\ P(3) \\ P(4) \end{bmatrix} = \begin{bmatrix} 1.10028E{-}01 & 1.28262E{-}03 & -5.27575E{-}04 & 6.47151E{-}04 \\ -3.32179E{-}01 & 1.62916E{-}01 & -2.04714E{-}02 & 2.61570E{-}02 \\ -2.19815E{-}01 & 9.04156E{-}02 & -1.09084E{-}02 & 1.92750E{-}01 \\ 0 & 0 & 0 & 1.00000E{-}01 \end{bmatrix} \begin{bmatrix} CD(1) \\ CD(2) \\ CD(3) \\ CD(4) \end{bmatrix}$$

If one selects CD(1) equal to zero then there will be an integrator in the forward path which results in a system with zero low frequency sensitivity. The requirements for open loop stability are positive coefficients and

$$CD(1)\ CD(4) < CD(2)\ CD(3)$$

$$P(1)\ P(4) < P(2)\ P(3)$$

These requirements can easily be met when CD(1) is zero if

$$CD(3) < CD(2) + CD(3)$$

The choice for CD(s) is arbitrarily made

$$CD(s) = s(s^2+s+1)$$

then

$$P(s) = 7.227\cdot 10^{-3} + .1686\, s + .27226\, s^2 + .1\, s^3$$

The resulting system has a stable open loop compensator and zero low frequency sensitivity and zero steady state error. These characteristics are an improvement over the response of the state variable feedback system while both have the same closed loop response. In addition only the output state needs to be measured; however, two third order compensators are required.