

AD-A154 552

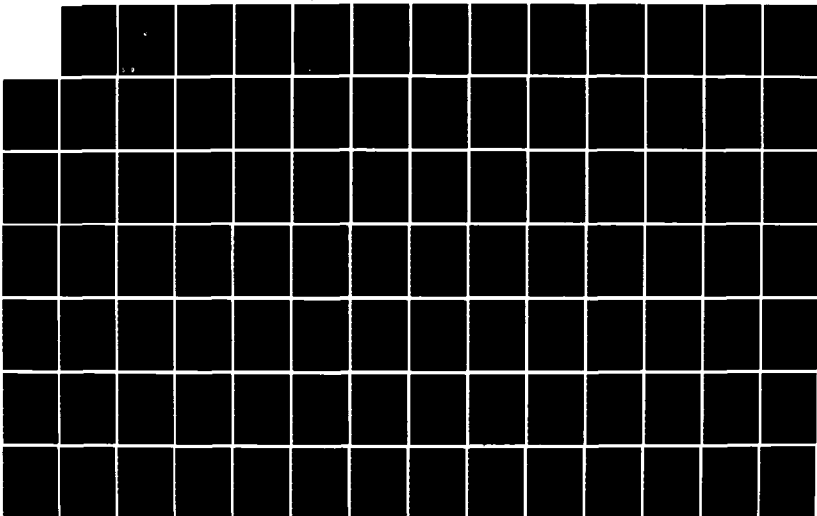
DEVELOPMENT OF REDUCED ORDER MODELS FOR CONTROL SYSTEM  
DESIGN USING THE OPTSYX PROGRAM(U) NAVAL POSTGRADUATE  
SCHOOL MONTEREY CA S W NELSON DEC 84

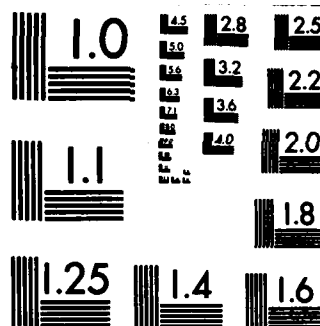
1/2

UNCLASSIFIED

F/G 9/2

NL





(2)

AD-A154 552

# NAVAL POSTGRADUATE SCHOOL

Monterey, California



## THESIS

DTIC FILE COPY

DEVELOPMENT OF REDUCED ORDER MODELS  
FOR CONTROL SYSTEM DESIGN USING  
THE OPTSYSX PROGRAM

by

Stanley William Nelson

December 1984

Thesis Advisor

D.J. Collins

Approved for public release; distribution is unlimited

**S** DTIC  
ELECTE  
JUN 5 1985  
**D**  
E

85 5 08 039

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                       | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|------------------------------------------------------------------------|
| 1. REPORT NUMBER                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER                                          |
| AD A154 552                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       |                                                                        |
| 4. TITLE (and Subtitle)<br>Development of Reduced Order Models<br>for Control System Design Using<br>the OPTSYSX Program                                                                                                                                                                                                                                                                                                                                                    |                       | 5. TYPE OF REPORT & PERIOD COVERED<br>Master's Thesis<br>December 1984 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                       | 6. PERFORMING ORG. REPORT NUMBER                                       |
| 7. AUTHOR(s)<br><br>Stanley William Nelson                                                                                                                                                                                                                                                                                                                                                                                                                                  |                       | 8. CONTRACT OR GRANT NUMBER(s)                                         |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Naval Postgraduate School<br>Monterey, California 93943                                                                                                                                                                                                                                                                                                                                                                      |                       | 10. PROGRAM ELEMENT, PROJECT, TASK<br>AREA & WORK UNIT NUMBERS         |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Naval Postgraduate School<br>Monterey, California 93943                                                                                                                                                                                                                                                                                                                                                                          |                       | 12. REPORT DATE<br>December 1984                                       |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                       | 13. NUMBER OF PAGES<br>156                                             |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)                                                                                                                                                                                                                                                                                                                                                                                                 |                       | 15. SECURITY CLASS. (of this report)<br><br>Unclassified               |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                       | 15a. DECLASSIFICATION/DOWNGRADING<br>SCHEDULE                          |
| 16. DISTRIBUTION STATEMENT (of this Report)<br><br>Approved for public release; distribution is unlimited.                                                                                                                                                                                                                                                                                                                                                                  |                       |                                                                        |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)                                                                                                                                                                                                                                                                                                                                                                                  |                       |                                                                        |
| 18. SUPPLEMENTARY NOTES                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                       |                                                                        |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number)<br><br>Reduced Order Modelling<br>Model Reduction for Control Systems<br>F100 Turbofan Engine<br>X-29A Aircraft Control System                                                                                                                                                                                                                                                           |                       |                                                                        |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number)<br><br>The modern controls engineer is often faced with designing a system which is characterized by a large number of first order differential equations. It is highly desirable and sometimes necessary that such complex systems be reduced for analysis, synthesis and implementation into a physical system. It is the intent of this thesis to present a mathematical procedure and |                       |                                                                        |

DD FORM 1473 1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

S. N 0102-LF-014-6601

1 UNCLASSIFIED  
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

and computer software based upon this procedure which enable the control engineer to formulate a reduced order model of a large order system.

As examples, two large order systems are analyzed: a sixteenth order model of the F100 turbofan engine and a ninety-eighth order model of the X-29A aircraft control system.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Approved for public release; distribution is unlimited

Development of Reduced Order Models  
for Control System Design Using  
the OPTSYSX Program

by

Stanley W. Nelson  
Lieutenant Commander, United States Navy  
B.S., University of Kansas, 1972

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL  
December 1984

Author:

*Stanley W. Nelson*  
Stanley W. Nelson

Approved by:

*D. J. Collins*  
D. J. COLLINS, THESIS ADVISOR

*H. A. Titus*  
H. A. Titus, Second Reader

*M. F. Platz*  
M. F. Platz, Chairman,  
Department of Aeronautics

*J. N. Dyer*  
J. N. Dyer,  
Dean of Science and Engineering

|                    |                                     |
|--------------------|-------------------------------------|
| Accession For      |                                     |
| NTIS GRA&I         | <input checked="" type="checkbox"/> |
| DTIC TAB           | <input type="checkbox"/>            |
| Unannounced        | <input type="checkbox"/>            |
| Justification      |                                     |
| By _____           |                                     |
| Distribution/      |                                     |
| Availability Codes |                                     |
| Dist               | Avail and/or<br>Special             |
| A/1                |                                     |



# ABSTRACT

The modern controls engineer is often faced with designing a system which is characterized by a large number of first order differential equations. It is highly desirable and sometimes necessary that such complex systems be reduced for analysis, synthesis and implementation into a physical control system. It is the intent of this thesis to present a mathematical procedure and computer software based upon this procedure which enable the control engineer to construct reduced order models.

As examples, two large order systems are analyzed: a sixteenth order model of the F100 turbofan engine and a ninety-eighth order model of the X-29A aircraft control system. *Additional keywords: OPTSYSX computer program; OPTRED computer program; interfaces; Fortran; computations; notch filters. ←*

## TABLE OF CCNTENTS

|                                                                         |                                                              |    |
|-------------------------------------------------------------------------|--------------------------------------------------------------|----|
| I.                                                                      | INTRODUCTION . . . . .                                       | 11 |
| II.                                                                     | THE OPTSYSX COMPUTER PROGRAM . . . . .                       | 14 |
|                                                                         | A. PROGRAM OVERVIEW . . . . .                                | 14 |
|                                                                         | B. OPTSYSX MODIFICATIONS . . . . .                           | 16 |
| III.                                                                    | ANALYTICAL DEVELOPMENT OF A REDUCED ORDER<br>MODEL . . . . . | 18 |
| IV.                                                                     | THE OPTRED PROGRAM . . . . .                                 | 23 |
|                                                                         | A. PROGRAM OVERVIEW . . . . .                                | 23 |
|                                                                         | B. PROGRAM OPERATION . . . . .                               | 24 |
|                                                                         | 1. Program Composition and Flow . . . . .                    | 24 |
|                                                                         | 2. User Protection Features . . . . .                        | 27 |
|                                                                         | 3. Large Order Systems . . . . .                             | 28 |
|                                                                         | C. INTERFACE WITH OTHER ANALYSIS PROGRAMS . . . . .          | 29 |
|                                                                         | D. EXAMPLES OF ORDER REDUCTION . . . . .                     | 29 |
|                                                                         | 1. Example of a Sixteenth Order System . . . . .             | 30 |
|                                                                         | 2. Example of a Ninety-eighth Order<br>System . . . . .      | 35 |
| V.                                                                      | CONCLUSIONS AND RECOMMENATIONS . . . . .                     | 49 |
|                                                                         | A. CONCLUSIONS . . . . .                                     | 49 |
|                                                                         | B. RECOMMENDATIONS . . . . .                                 | 49 |
|                                                                         | 1. Comparison of System Time Response . . . . .              | 49 |
|                                                                         | 2. Interface with Other Analysis Programs . . . . .          | 50 |
|                                                                         | 3. Alternate Order Reduction Techniques . . . . .            | 50 |
| APPENDIX A: THE OPTSYSX PROGRAM AS MODIFIED BY THIS<br>THESIS . . . . . |                                                              | 51 |

|                                          |     |
|------------------------------------------|-----|
| APPENDIX B: THE OPTRED PROGRAM . . . . . | 141 |
| APPENDIX C: THE OPTRED EXEC . . . . .    | 153 |
| LIST OF REFERENCES . . . . .             | 154 |
| BIBLIOGRAPHY . . . . .                   | 155 |
| INITIAL DISTRIBUTION LIST . . . . .      | 156 |

## LIST OF FIGURES

|      |                                                        |    |
|------|--------------------------------------------------------|----|
| 4.1  | Full System Bode Plot for the F100 Engine . . . . .    | 39 |
| 4.2  | Reduced System Bode Plot for the F100 Engine . . . . . | 40 |
| 4.3  | X-29A Full System Without Notch Filter . . . . .       | 41 |
| 4.4  | X-29A Reduced System Withcut Notch Filter . . . . .    | 42 |
| 4.5  | X-29A Full System Without Notch Filter . . . . .       | 43 |
| 4.6  | X-29A Reduced System Withcut Notch Filter . . . . .    | 44 |
| 4.7  | X-29A Full System With Notch Filter . . . . .          | 45 |
| 4.8  | X-29A Reduced System With Notch Filter . . . . .       | 46 |
| 4.9  | X-29A Full System With Notch Filter . . . . .          | 47 |
| 4.10 | X-29A Reduced System With Notch Filter . . . . .       | 48 |

## SYMBOLS

$A$  = State ( $N_s, N_s$ ) or Output ( $N_o, N_o$ ) Weighting Matrix  
 $B$  = Control ( $N_c, N_c$ ) Weighting Matrix  
 $C$  = Control Gain Matrix ( $N_c, N_s$ )  
 $D$  = Control ( $N_o, N_c$ ) or Noise ( $N_o, N_g$ ) Feedforward Matrix  
 $D_r$  = Reduced Control Feedforward Matrix ( $N_o, N_c$ )  
 $F$  = Open-Loop Dynamics Matrix ( $N_s, N_s$ )  
 $F_r$  = Reduced Open-loop Dynamics Matrix ( $N_r, N_r$ )  
 $G$  = Control Distribution Matrix ( $N_s, N_c$ )  
 $G_r$  = Reduced Control Distribution Matrix ( $N_r, N_c$ )  
 $GAM$  = State Disturbance Distribution Matrix ( $N_s, N_g$ )  
 $H$  = Measurement Scaling Matrix ( $N_o, N_s$ )  
 $H_r$  = Reduced Measurement Scaling Matrix ( $N_o, N_r$ )  
 $K$  = Estimator Gain Matrix ( $N_s, N_o$ )  
 $N_c$  = Number of Controls  
 $N_g$  = Number of Process Noise Sources  
 $N_s$  = Number of States  
 $N_o$  = Number of Observations or Measurements  
 $N_r$  = Reduced Model Number of States  
 $Q$  = White Process Noise Covariance Matrix ( $N_g, N_g$ )  
 $R$  = White Meas. Noise Covariance Matrix ( $N_o, N_o$ )  
 $S$  = Steady-State Covariance Matrix of Control ( $N_c, N_c$ )  
 $u$  = Control Vector ( $N_c, 1$ )  
 $v$  = White Measurement Noise Vector ( $N_o, 1$ ), With Zero Mean and Covariance Matrix  $R$   
 $w$  = White Process Noise Vector ( $N_g, 1$ ), with Zero Mean and Covariance Matrix  $Q$   
 $w_0$  = Constant Disturbance Vector ( $N_g, 1$ )  
 $x$  = State Vector ( $N_s, 1$ )  
 $\dot{x}$  = Derivative of State Vector ( $N_s, 1$ )

$\mathbf{x}_e$  = Estimate of State Vector ( $N_s, 1$ )  
 $\dot{\mathbf{x}}_e$  = Derivative of Estimate of State Vector ( $N_s, 1$ )  
 $\mathbf{y}$  = Output Vector ( $N_o, 1$ )  
 $\mathbf{z}$  = Measurement Vector ( $N_c, 1$ )

### ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor D.J. Collins whose assistance and encouragement enabled this research. I would also like to thank Professor D.G. Williams, Director of the W.R. Church Computer Center, for his assistance and cooperation in providing the additional computer memory allocation which was essential to the completion of this work.

This thesis is dedicated to my wife Gayle, my daughter Kristin and my son Scott. Without their constant love, support and understanding this work would not have been possible.

## I. INTRODUCTION

Modern control methods allow the control engineer to develop a control system to regulate every parameter of a physical system. He has been aided by computer simulations of non-linear systems and linear approximations of those systems. The desired control system can be developed parallel to or even ahead of the physical system which is to be controlled.

In many design applications areas, the engineer is confronted with designing a control system which can be represented analytically by a very large number of first order, linear, time invariant differential equations. This is especially true with highly complex and aero-elastic aircraft and space vehicles. The analytical model is a prime ingredient in the controller design process for any design technique.

Many practical limitations exist, such as computational requirements, that require the order (number of equations) of this complex model to be reduced for analysis, synthesis and actual implementation of the control system. Some method is required to analyze these dynamical models and establish simpler systems which include elements critical to the desired control function. Without such simplification, application of design procedures can result in highly complex and parameter-sensitive controlled systems. With a reduced order model, regulator synthesis procedures become more intuitive and far less sensitive to parameter variation. The reduced order model is far easier to handle in formulating the control system but must characterize the physical system with sufficient accuracy such that performance objectives for the controlled physical system can be met by designing control laws from the reduced order model.

Both file and interactive data entry is accomplished by the main program and by the primary subroutine, REDUCX. The actual computation of the reduced order model is performed by subroutine REDUCX which utilizes subroutines MI and MAMULT for its matrix inversion and multiplication requirements. The remaining subroutines are utilized throughout OPTRED for interactive entry of numerical and character string data. The program requires no external subroutines for its execution.

The main program begins by presenting the user with a brief operational description of OPTRED. At this point, the user is given the option of having the program's general results repeated to the screen or printed to a listing file. OPTRED creates a permanent data file which contains the reduced data regardless of the user's option here. The program then reads full system flags and parameters, excluding matrices, from the OPTMAT data file. A brief description of the full system under investigation is then presented as an indication to the user that the desired data is being input. This description includes the order of the system, the number of controls and observations and whether or not a system "D" matrix will be input. The latter is indicated either by " " or "nct" in the following example. The program then prompts the user for entry of the desired order of the reduced system.

The main program presentation as it appears during execution:

#### OPTRED

OPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL SYSTEM  
"F","G","H", AND "D" (IF INPUT) MATRICES.

THE FULL SYSTEM MATRICES MUST BE READ FROM A FILE NAMED  
"OPTMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.

YOU MUST ALSO ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)

- 1) The compiled text file of OPTRED
- 2) The OPTMAT data file
- 3) The OPTRED EXEC file

The first two required files are self explanatory. The OPTRED EXEC file sets up the required read/write file definitions and calls the necessary library functions for execution using the IBM 3033 System. This EXEC allows the user to execute the OPTRED program by simply entering the word "OPTRED" at the terminal. The program then continues in a user-interactive mode until completion as described in the next section. The OPTRED EXEC file is listed in Appendix C.

## B. PROGRAM OPERATION

OPTRED was written to satisfy two specific objectives. The first is to provide a method to assist in the analysis of actual large order control systems. The second is to create a program to be used as an instructional tool for students studying controls related problems.

To accomplish these objectives the emphasis while creating OPTRED was to make the program as user oriented as possible by eliminating data entry ambiguities and providing features which prevent the user from other inadvertant or invalid program entries. A serious effort was also made to minimize the amount of virtual memory required to execute this program.

### 1. Program Composition and Flow

OPTRED is composed of one main program and seven subroutines. The program can be divided into three basic catagories:

- 1) File Data Input
- 2) Interactive Data Input
- 3) Calculation

#### IV. THE OPTRED PROGRAM

##### A. PROGRAM OVERVIEW

OPTRED is an interactive, double precision FORTRAN program which uses a specifically formatted data file to input full system data. The required matrix inputs from this file are the full system "F", "G", "H" and "D" matrices. The system flags and parameters which OPTRED must also read for program execution are Ns, Nc, No, and IFDFW. The OPTSYSX program creates OPTMAT data which is the source of this required data. Through interactive input, the user specifies all other data which is required to create reduced model data.

OPTRED also reads from the CPTMAT file the noise, feedback control and the output and control cost weighting matrices. These are not required inputs but only input to the program to facilitate creation of a reduced order data file which is identical in format to the OPTMAT data file. The final result of OPTRED is the creation of OPTMATR, a data file which contains the reduced system "F", "G", "H" and "D" matrices and is formatted for re-entry into the OPTSYSX program.

It is possible for the user to construct a data file named OPTMAT to provide input data to OPTRED. However, only in rare applications would the full system not be analyzed prior to the development of the reduced model. OPTSYSX is therefore the simplest and most preferred method of data generation.

To execute the OPTRED program, a minimum of three separate files must be immediately accessible to the user:

$$Y = H_r * X_1 + D_r * U \quad (3.24)$$

where

$$F_r = F_{11} - F_{12} * F_{22}^{-1} * F_{21} \quad (3.25)$$

$$G_r = G_1 - F_{12} * F_{22}^{-1} * G_2 \quad (3.26)$$

$$H_r = H_1 - H_2 * F_{22}^{-1} * F_{21} \quad (3.27)$$

$$D_r = D - H_2 * F_{22}^{-1} * G_2 \quad (3.28)$$

This is the mathematical procedure which OPTRED uses in computing reduced order models. The complete OPTRED listing is included as Appendix B.

$$G' = \begin{bmatrix} G_1 \\ \hline G_2 \end{bmatrix} \quad (3.16)$$

$$H' = [H_1 \mid H_2] \quad (3.17)$$

If the states in  $X_1$  truly model the full order system, then the subvector  $X_2$  will be essentially zero and

$$\dot{X}_1 = F_{11} * X_1 + F_{12} * X_2 + G_1 * U \quad (3.18)$$

$$0 = F_{21} * X_1 + F_{22} * X_2 + G_2 * U \quad (3.19)$$

Solving for  $X_2$  in Equation 3.19, one has

$$X_2 = -F_{22}^{-1} * F_{21} * X_1 + F_{22}^{-1} * G_2 * U \quad (3.20)$$

This is incorporated into Equation 3.18 and

$$\dot{X}_1 = (F_{11} - F_{12} * F_{22}^{-1} * F_{21}) X_1 + (G_1 - F_{12} * F_{22}^{-1} * G_2) U \quad (3.21)$$

and the output equation becomes

$$Y = (H_1 - H_2 * F_{22}^{-1} * F_{21}) X_1 + (L - H_2 * F_{22}^{-1} * G_2) U \quad (3.22)$$

The reduced order model can now be written as

$$\dot{X}_1 = F_r * X_1 + G_r * U \quad (3.23)$$

$$X' = R*X = T*Z' = \begin{bmatrix} X_1 \\ \text{---} \\ X_2 \end{bmatrix} \quad (3.9)$$

This  $X'$  vector is the reordered state vector, where the elements contained in the subvector  $X_1$  are those states associated with the designers choice of eigenvalues while the subvector  $X_2$  contains all other states.

The linear system may now be rewritten

$$\dot{X}' = R*F*R^{-1}*X' + R*G*U \quad (3.10)$$

and

$$Y = H*R^{-1}*X' + D*U \quad (3.11)$$

Now define

$$F' = R*F*R^{-1} \quad (3.12)$$

$$G' = R*G \quad (3.13)$$

$$H' = H*R^{-1} \quad (3.14)$$

and

$$F' = \begin{bmatrix} F_{11} & F_{12} \\ \text{---} & \text{---} \\ F_{21} & F_{22} \end{bmatrix} \quad (3.15)$$

The eigenvalues which are to be included in the initial reduced order model are determined through modal analysis. The eigenvalue matrix can be re-ordered by constructing a matrix  $Q$ , which has the value 1 in the position corresponding to the eigenvalue to be included (column) and the re-ordered position (row) of that eigenvalue. For example, if eigenvalue 20 of the original  $\Lambda$  matrix was to be re-ordered to the first position, a 1 would be placed in the (1,20) position of the  $Q$  matrix. Now, the reordered matrix

$$\Lambda' = Q\Lambda \quad (3.4)$$

is incorporated in the linear model

$$Q\dot{Z} = \Lambda'Z + Q^{-1}G^*U \quad (3.5)$$

letting

$$Z' = QZ \quad (3.6)$$

thus

$$\dot{Z}' = \Lambda'Q^{-1}Z' + Q^{-1}G^*U \quad (3.7)$$

and

$$X = TQ^{-1}Z' \quad (3.8)$$

A matrix  $R$  can be constructed such that

### III. ANALYTICAL DEVELOPMENT OF A REDUCED ORDER MODEL

There are numerous methods of order reduction for control systems as discussed by Enns [Ref. 8] of Stanford University. This thesis uses the theory discussed by DeHoff and Hall [Ref. 9] which presents a method of reducing the order of the linear system based on dominant states chosen by the designer. This dominance is determined by modal analysis of the eigensystem under investigation. Dougherty [Ref. 10] utilized this same methodology in the development of a computational model used in the analysis of the F100 turbofan engine. A summary of this technique is provided below.

The method of reduction involves finding the eigenvector transform matrix,  $T$ , such that

$$X^*T = T^*\Lambda \quad (3.1)$$

where  $\Lambda$  is the diagonal eigenvalue matrix. Then, by defining an alternate state vector,  $Z$ , where

$$X = T^*Z \quad (3.2)$$

the original system may be re-written in modal coordinates as

$$\dot{Z} = \Lambda^*Z + T^{-1}*G^*U \quad (3.3)$$

"F", "G", "H", and "GAM" matrices were complemented by the addition of routines which also save the "D" matrix. These new routines are analagous to existing "save" routines. In addition to these routines a new flag, ISAD, was added to the existing flags ISAF, ISAG, ISAH, and IGAM. This new flag provides the same function in "saving" matrices as do the existing flags. For program continuity, it was also necessary to add ISAD to subrcutine INNER, which is the driving subroutine in the OPTSYSX program.

Subroutines RDMATF, RDMAT, and WRTMAT also required minor changes to accomodate the "D" matrix. These changes consisted of the addition of "D" to the subroutine parameter lists and the addition of read or write statements for either re-entry of "D" into OPTSYSX or for writing "D" to the OPTMAT data file. The existing flag, IFDFW, was also added to the parameter lists of these routines. This flag indicates whether or not a "D" matrix will be input by the user. OPTRED uses this flag in its computational methods in the same manner as does OPTSYSX. Appendix A lists the OPTSYSX program as modified by this thesis.

additions to that program's size would cause user difficulty in obtaining sufficient memory for program execution. It is for this reason that Diel's OPTCALC, Laptas' OPTGRAPH and the program developed by this thesis have been developed as independent programs which depend upon various data created by OPTSYSX. The minor modifications this thesis introduces to the OPTSYSX program are not detrimental to its original capabilities.

## B. OPTSYSX MODIFICATIONS

This thesis is confronted with the problem of analyzing very large control system models. Therefore, the additions of Hoden's subroutine SETUP and Diel's subroutines RDMATF, RDMAT, and WRTMAT were crucial to system analysis. These routines make multiple computer analysis runs possible without painstaking re-entry of system parameters and matrices. In particular, the system "F", "G", "H", and "Gam" are some of the matrices which are stored for re-entry into OPTSYSX or stored as data for use by other analysis programs. Subroutine WRTMAT is responsible for the storage of these matrices in the data file "OPTMAT". It is this file which OPTRED utilizes to input all pertinent full system data.

In many control applications, a control feed-forward matrix ("D") is present in the system model. Therefore, this thesis makes provisions within OPTSYSX which enable the user to save the "D" matrix in the same manner by which the other system matrices are saved. Subroutine SETUP, which enables the user to input system matrices from file data, was changed to enable the user to input the "D" matrix. This consisted of adding "D" to the routine's parameter list and also added a read statement to input that matrix from a data file. Present routines in OPTSYSX which "save" the

open loop transfer function

$$[H] * [s[I] - [F]]^{-1} * [G] \quad (2.5)$$

closed loop noise transfer function

$$[H] * [s[I] - [F]]^{-1} * [Gam] \quad (2.6)$$

compensator transfer function from measurement to input

$$[C] * [s[I] - [F] + [G] * [C] + [K] * [H]]^{-1} * [K] \quad (2.7)$$

where

- x = state vector (Ns X 1)
- $\dot{x}$  = derivative of the state vector (Ns X 1)
- u = control vector (Nc X 1)
- y = output vector (No X 1)
- z = measurement vector (No X 1)
- w = white process noise vector (Ng X 1)
- [F] = open-loop dynamics or plant matrix (Ns X Ns)
- [G] = control distribution matrix (Ns X Nc)
- [Gam] = state disturbance distribution matrix (Ns X Ng)
- [H] = measurement distribution matrix (No X Ns)
- [D] = control feed-forward matrix (No X Nc)
- [C] = control gain matrix (Nc X Ns)
- [I] = identity matrix (Ns X Ns)

OPTSYSX is an extremely large and complex program which contains over 4000 lines of code. Its existing standard dimensions will accommodate a thirty-second order system with up to 10 controls and 10 observations or measurements. To execute even this version of OPTSYSX the user requires one megabyte of virtual storage. Any further significant

## II. THE OPTSYSX COMPUTER PROGRAM

### A. PROGRAM OVERVIEW

OPTSYSX is a double precision, interactive FORTRAN program which employs modern control theory analysis techniques. It is developed to be compiled and executed by the Naval Postgraduate School's IBM 3033 System 360/370. Its primary capabilities include the calculation of the open loop eigensystem, and the fixed closed loop system; the synthesis of optimal regulators or filters; the power spectral density, and modal distribution computations.

The fundamental system equations used by the OPTSYSX program for its computations are of the state variable form. The basic system equations are:

system model

$$\dot{x} = [F]*x + [G]*u + [GAM]*w \quad (2.1)$$

output equation

$$y = [H]*x + [D]*u \quad (2.2)$$

measurement equation

$$z = [H]*x + v \quad (2.3)$$

estimator equation

$$\dot{x}_e = [F]*x_e + [G]*u + [K]*(z - [H]*x_e) \quad (2.4)$$

systems and the corresponding reduced models which OPTRED generates are presented and analyzed. Those systems include a sixteenth order model of the F100 turbofan engine and a ninety-eighth order model of the X-29A aircraft. Complete program listings for the OPTSYSX program and the OPTRED program are included as Appendices A and B respectively.

The intent of this thesis is to present the mathematical basis for creating such a reduced order model and to develop the actual computer software which creates that reduced model based on a given large order system. This computer program (OPTRED) is developed to interface with an existing control analysis FORTRAN program named OPTSYS. This thesis will discuss the modifications or additions to the OPTSYS program which are necessary to that interface.

The OPTimal SYStems control program was originally developed by Hall [Ref. 1] for the study and application of optimal systems control theory. Later modifications to the program were made by Walker [Ref. 2] and Liu [Ref. 3] of Stanford University. Hoden [Ref. 4] modified the OPTSYS program to present a user-friendly, interactive version of the program (OPTSYSX). Diel [Ref. 5] introduced changes which enabled the user to save various data for re-entry into the program and also to create data files which were necessary for the execution of a time response program (OPTCALC). Further modifications have been made by Laptas [Ref. 6] which were necessary to create data sets for input to his OPTGRAPH program. This FORTRAN program enabled the user to obtain Pole-Zero, Root-Locus, Nyquist, Bode and Nichols plots.

It is assumed that the user has a basic understanding of the fundamental concepts of control theory and optimal systems design. The symbology conventions of Bryson [Ref. 7] are used in the discussion of program operation and system descriptions.

An overview of the OPTSYSX program, its capabilities, and the modifications to that program which are necessary for the interface with OPTRED is presented first. This is followed by the mathematical basis upon which OPTRED is formulated and by a full description of the operation of that program. Finally, examples of large order control

AND THE ACTUAL STATE #'S (IN ASCENDING ORDER) WHICH REPRESENT THE REDUCED MODEL. THE ORDER OF THE REDUCED ORDER MODEL MUST BE LESS THAN THE ORDER OF THE FULL SYSTEM.

DO YOU WISH TO CONTINUE?

TYPE "YES" OR "NO".

DO YOU WISH RESULTS TO SCREEN OR DISK?

NOTE: A DATA FILE CONTAINING THE REDUCED ORDER DATA  
WILL BE GENERATED REGARDLESS OF YOUR ANSWER.

TYPE "S" FOR SCREEN OR "D" FOR DISK.

THE ORDER OF THE FULL SYSTEM IS:

THE NUMBER OF CONTROLS IS:

THE NUMBER OF OBSERVATIONS IS:

A "D" MATRIX WILL BE INPUT.

DO YOU WISH TO CONTINUE?

TYPE "YES" OR "NO".

ENTER THE DESIRED REDUCED ORDER OF THE "P" MATRIX.

Subroutine REDUCX is now called and immediately reads the full system matrices from file data. The user is then prompted for entry of the significant states which will compose his reduced model. This program makes a provision for entry of significant states from a specifically formatted, user created data file. At this point in the program he is given that option. Once the significant states have been entered, the program repeats those states to the screen and the user is given the option to make changes at that time. If the significant states are satisfactory to the user, the program continues with the computation of the reduced order model. At the completion of OPTRED, the reduced system data is created and can be exam-

ined for stability, controllability and observability by other analysis techniques as discussed in the following section. Prior to exiting the OPTRED program, the user is provided the option of re-executing OPTRED if he finds initial results unsatisfactory. A typical recording session which depicts this program flow is presented in the next section.

## 2. User Protection Features

The user of OPTRED is protected from abnormal program termination in several ways. Initially, the program presents its capabilities and user required entries. If he is not prepared to continue at this point, the user is offered the option of exiting the program. Next, the full system parameters are presented and the user is once again able to exit if these parameters are unexpected.

OPTRED requires that the order of the reduced system be greater than zero but less than the order of the full system. If the user inadvertently enters a number which is out of this range, the program issues a warning to that effect and allows recovery. The computation of the reduced order model relies upon the construction of a reduced state matrix. OPTRED requires that these states be entered in the order that they would normally appear in this matrix. For this reason, the user must enter the significant state numbers in ascending order and those state numbers must lie in the range from one to the number of states in the full system. If the user inadvertently enters these state numbers out of order or out of range the program prompts a warning and allows recovery.

The method of reduced order computation involves the inversion of the state submatrix  $F_{22}$ , as described in Chapter III, Equations 3.25 through 3.28. The nature of the plant matrix and the selection of reduced order states can

be coupled such that this matrix will be singular or non-invertable. OPTRED will detect this condition and issue information to the user that a reduced order model cannot be calculated for that full system from the given desired reduced states. Reduced data will still be computed but the user is warned that this data is invalid.

Subroutines RDINT, RDREAL and RDCHAR are responsible for the interactive input of data and expect an integer input, a real number input and a logical "YES" or "NO" input respectively. If the user inadvertently makes an incompatible entry these subroutines issue warnings and allow another opportunity for data entry. The entry of a null line is included in these improper actions and the entry of two consecutive null strings will cause termination of the program. This function allows the user a further means of exiting OPTRED if he so desires.

### 3. Large Order Systems

The analysis of large order systems, particularly the ninety-eighth order model, presents a major problem to the user with limited virtual memory assets. The longitudinal control system analyzed by this thesis has a (98 X 98) "F" matrix, a (2 X 98) "H" matrix and a (98 X 1) "G" matrix. To analyze this system, OPTSYSX must be executed in its "increased dimension" form. To compile and execute this version of OPTSYSX requires 2.5 Megabytes of virtual memory.

Both OPTRED and OPTGRAPH are also dimensioned to enable analysis of a ninety-eighth order model. When OPTSYSX, OPTRED, OPTGRAPH and their peripheral data are all used during a typical analysis session, very careful attention to file and virtual memory management is mandatory to prevent the user from exceeding his virtual machine's capacity.

### C. INTERFACE WITH OTHER ANALYSIS PROGRAMS

To sufficiently analyze a control system it is desirable to utilize OPTSYSX, OPTCALC, OPTGRAPH and OPTRED. It may also be desirable to conduct this analysis during a single, continuous interactive computer session. However, in the analysis of very large systems this concept presents serious difficulties in the areas of program flow, data management and physical memory assets. For this reason, OPTRED has not been developed to be fully automatic in its interface with these other analysis programs.

The OPTRED program requires the existence of the OPTMAT data file. As previously discussed, OPTRED generates the OPTMATR data file which contains the reduced model data. To analyze this data using OPTSYSX, the user must now rename the OPTMATR data file for entry into OPTSYSX as OPTMAT data. Care must be exercised at this point to either erase the full system data file or to rename it to prevent ambiguous data files. Although this procedure may seem cumbersome at first, it is easily accomplished in the XEDIT mode and provides the user with the capability of maintaining several system data files which are readily available for analysis.

After OPTSYSX processes the reduced data, additional data files are created which enable the execution of other system analysis programs. OPTGRAPH and OPTCALC utilize the OPTMAT and OPTGROL data files, respectively, to provide the functions described in chapter 2. When the user is investigating several systems it is both prudent and convenient to rename these files after their use for the purpose of future analysis.

### D. EXAMPLES OF ORDER REDUCTION

Two large order systems will be presented to evaluate reduced models generated by OPTRED. The first system is a

sixteenth order model of the F100 turbofan engine. The final example is a ninety- eighth order model of the X-29A aircraft's longitudinal control system. Bode analysis was conducted for the full and reduced models and graphical comparisons follow at the end of the chapter. Recorded terminal sessions for each example will be presented to fully illustrate actual program operation.

#### 1. Example of a Sixteenth Order System

The F100 turbofan engine was chosen for initial analysis and [Ref. 9] describes this system in detail. The system "F", "G", "H" and "D" matrices were obtained from [Ref. 9: pp. 83-85] and the method of data entry to OPTSYSX is depicted in subroutine SETUP found in Appendix A. The selection of significant states is based upon a desired control bandwidth of 1 to 10 Hertz and following modal analysis these states were chosen as 1, 2, 5, 11 and 16. The following is a computer terminal system in which a fifth order model is generated.

```
BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.03 21:05:48
OPTRED
FILEDEF 05 TERM
FILEDEF 03 DISK REDUCI DATA A1
FILEDEF 06 DISK OPTRED LISTING A1
FILEDEF 07 DISK STATES DATA A1
FILEDEF 09 DISK OPTMAT DATA A1
FILEDEF 10 DISK OPTMATR DATA A1
GLOBAL TXTLIB PORTMOD2 MOD2EEH IMSLDP NONIMSL
LOAD OPTRED ( START
EXECUTION BEGINS...

OPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL SYSTEM
"F","G","H", AND "D" (IF INPUT) MATRICES.
```

THE FULL SYSTEM MATRICES MUST BE READ FROM A FILE NAMED  
"OPTMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.

YOU MUST ALSO ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)  
AND THE ACTUAL STATE #'S (IN ASCENDING ORDER) WHICH REPRESENT  
THE REDUCED MODEL. THE ORDER OF THE REDUCED MODEL MUST BE  
LESS THAN THE ORDER OF THE FULL SYSTEM.

DO YOU WISH TO CONTINUE?

TYPE "YES" OR "NO".

YES

DO YOU WISH RESULTS TO SCREEN OR DISK?

NOTE: A DATA FILE CONTAINING THE REDUCED ORDER DATA  
WILL BE GENERATED REGARDLESS OF YOUR ANSWER.

TYPE "S" FOR SCREEN OR "D" FOR DISK.

S

THE ORDER OF THE FULL SYSTEM IS: 16

THE NUMBER OF CCNROLS IS: 5

THE NUMBER OF OBSERVATIONS IS: 7

A "D" MATRIX WILL BE INPUT.

DO YOU WISH TO CONTINUE?

TYPE "YES" OR "NO".

YES

ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.

?

18

\*\*\*\*\* WARNING: REDUCED ORDER MUST BE GREATER THAN 0\*\*\*\*\*  
AND LESS THAN 16

ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.

?

5

DO YOU WISH TO INPUT DESIRED STATES FOR YOUR REDUCED ORDER  
MODEL FROM A DATA FILE?

DATA FILE MUST BE NAMED "STATES DATA A1" IN FIXED FORMAT.  
THE READ FORMAT IS "13I5" PER 72 CHARACTER LINE.

TYPE "YES" OR "NO".

NO

ENTER THE "N" SIGNIFICANT STATES WHICH REPRESENT THE REDUCED  
MODEL. ENTER STATE #'S IN ASCENDING ORDER.

STATE # 1 =

?

1

STATE # 2 =

?

2

STATE # 3 =

?

3

STATE # 4 =

?

11

STATE # 5 =

?

16

THE REDUCED MODEL STATES ARE:

1 2 3 11 16

DO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STATES?

TYPE "YES" OR "NO".

YES

ENTER THE N-TH STATE # TO BE CHANGED.

?

3

ENTER NEW STATE # 3

5

THE REDUCED MODEL STATES ARE:

1 2 5 11 16

DO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STATES?

TYPE "YES" OR "NO".

NO

(OUTPUT TO THE SCREEN FOLLOWS)

THE DESIRED REDUCED ORDER IS: 5

THE REDUCED MODEL STATES ARE:

1 2 5 11 16

THE REDUCED PLANT MATRIX ("F") IS:

|             |             |             |             |             |
|-------------|-------------|-------------|-------------|-------------|
| -3.0509D+00 | 2.4726D+00  | -3.5898D+02 | 8.6913D+00  | 9.1832D-02  |
| 1.2243D-01  | -1.6271D+00 | 4.1060D+01  | 4.2898D+00  | 3.0229D-01  |
| 2.7286D-03  | -2.8331D-03 | -7.4276D+00 | -1.1695D-02 | -9.4624D-03 |
| 4.1314D+00  | -5.3854D+00 | -1.6257D+03 | -5.3853D+01 | -2.7911D+00 |
| -1.0770D+00 | 2.2320D+00  | 1.1467D+03  | 1.9087D+01  | -4.8697D+01 |

THE REDUCED CONTROL DISTRIBUTION MATRIX ("G") IS:

|             |             |             |             |             |
|-------------|-------------|-------------|-------------|-------------|
| -3.5445D-02 | -1.4080D+02 | -9.3301D+01 | 2.3597D+01  | -1.8379D+04 |
| -4.4094D-01 | -2.7221D+01 | 7.8306D+00  | -1.0960D+01 | -9.9711D+03 |
| 1.7526D-02  | -3.7646D+01 | 1.2113D-01  | -7.8465D-02 | -1.1931D+02 |
| 1.9802D+01  | 3.1635D+02  | 4.9385D+01  | -7.6813D+01 | 4.8330D+04  |
| -1.8909D+00 | 7.2700D+01  | -3.0436D+01 | 2.8867D+01  | 7.0035D+03  |

THE REDUCED OUTPUT DISTRIBUTION MATRIX ("H") IS:

|             |             |             |             |             |
|-------------|-------------|-------------|-------------|-------------|
| 1.9281D-01  | 1.0750D-01  | 1.8753D+02  | 1.2088D+00  | 7.0796D-02  |
| 6.4607D-03  | -9.9442D-07 | -9.1543D-03 | -5.6722D-05 | -1.9316D-05 |
| 0.0         | 0.0         | 0.0         | 1.0000D+00  | 0.0         |
| 1.2535D-05  | 1.3512D-05  | -3.1266D-02 | -1.3900D-04 | -4.8476D-05 |
| -4.8870D-05 | 1.4796D-04  | 1.0160D-03  | -5.1681D-05 | 1.4491D-05  |
| 1.6598D-05  | -5.0355D-05 | -3.7563D-02 | -1.7308D-04 | -5.9561D-05 |
| 7.2415D-06  | -8.1255D-07 | -1.0493D-02 | -4.9442D-05 | -1.6687D-05 |

THE REDUCED FEEDFORWARD MATRIX ("D") IS:

|             |             |             |             |             |
|-------------|-------------|-------------|-------------|-------------|
| -1.0377D-01 | 3.4001D+00  | 1.0378D+01  | 1.7019D+00  | -4.1189D+03 |
| -1.0929D-04 | -2.2819D-01 | 3.1109D-01  | 4.8490D-03  | -1.3105D+01 |
| 0.0         | 0.0         | 0.0         | 0.0         | 0.0         |
| 8.6209D-05  | -8.5310D-03 | -4.4940D-03 | -9.5892D-05 | -4.4513D-01 |
| -1.5078D-05 | 4.1932D-03  | -9.3797D-04 | -3.9608D-03 | 9.6895D-01  |
| 1.0794D-04  | 5.6356D-03  | -4.0947D-04 | -7.9878D-04 | -3.1434D-01 |
| 2.9257D-05  | 9.8556D-04  | 3.2771D-04  | -9.4642D-05 | -8.3787D-02 |

ANALYSIS COMPLETE...YOUR REDUCED SYSTEM DATA HAS BEEN SAVED  
IN A FILE NAMED "OPTMATR DATA".

DO YOU DESIRE ANOTHER RUN?

IF YES, THE RESULTS OF THIS RUN WILL  
REPLACE YOUR LAST OPTMATR DATA.

TYPE "YES" CR "NO".

NO

.....OPTRED IS NOW TERMINATED.....

R; T=0.38/0.77 21:06:56

RECORD OFF

END RECORDING OF TERMINAL SESSION

This is a typical computer terminal session in which the user requested output to the screen. Note also, that some of the user protection features of OPTRED are involved during this session.

Following this analysis, the OPTMATR data file was renamed OPTMAT data and OPTSYSX was executed using the reduced model data. The eigensystem was analyzed for desirable characteristics, including stability, and was compared to the full system eigenvalues. This analysis proved satisfactory and further comparison of the full and reduced

models was accomplished by obtaining Bode diagrams for the system open loop transfer functions. Figures 4.1 and 4.2 illustrate the full system and reduced system Bode plots respectively. The similarity, especially within the control bandwidth, is readily apparent. The frequency responses shown are for system input 1 and output 1. Analysis of other input/output combinations revealed the same favorable comparisons.

## 2. Example of a Ninety-eighth Order System

The X-29A control system includes a ninety-eighth order dynamics matrix. The measurement scaling matrix is constructed such that there are two measured outputs. These outputs represent the system with and without a notch filter, respectively. Full system matrix data as well as eigensystem data was obtained for this model from NASA-Edwards. Full system data was first entered into OPTSYSX using subroutine SETUP and the overall system was analyzed. Sixty significant states were identified and are shown in the following recorded terminal session. This example also illustrates program execution where no "D" matrix is input and the reduced system states are entered by means of file data. During this session the user also chooses to have the general reduced model data printed to a listing file.

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 12:23:56

OPTRED

FILEDEF 05 TERM

FILEDEF 06 DISK OPTRED LISTING A1

FILEDEF 07 DISK STATES DATA A1

FILEDEF 09 DISK OPTMAT DATA A1

FILEDEF 10 DISK OPTMATR DATA A1

GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP NONIMSL

LOAD OPTRED ( START  
EXECUTION BEGINS...

OPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL SYSTEM  
"F","G","H", AND "D" (IF INPUT) MATRICES.

THE FULL SYSTEM MATRICES MUST BE READ FROM A FILE NAMED  
"OPTMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.

YOU MUST ALSO ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)  
AND THE ACTUAL STATE #'S (IN ASCENDING ORDER) WHICH REPRESENT  
THE REDUCED MODEL. THE ORDER OF THE REDUCED MODEL MUST BE  
LESS THAN THE ORDER OF THE FULL SYSTEM.

DO YOU WISH TO CCNTINUE?

TYPE "YES" OR "NO".

YES

DO YOU WISH RESULTS TO SCREEN OR DISK?

NOTE: A DATA FILE CONTAINING THE REDUCED ORDER DATA  
WILL BE GENERATED REGARDLESS OF YOUR ANSWER.

TYPE "S" FOR SCREEN OR "D" FOR DISK.

D

THE ORDER OF THE FULL SYSTEM IS: 98

THE NUMBER OF CONTROLS IS: 1

THE NUMBER OF OBSERVATIONS IS: 2

A "D" MATRIX WILL NOT BE INPUT.

DO YOU WISH TO CONTINUE?

TYPE "YES" OR "NO".

YES

ENTER THE DESIRED REDUCED CRDER OF THE "F" MATRIX.

?

60

DO YOU WISH TO INPUT DESIRED STATES FOR YOUR REDUCED ORDER  
MODEL FROM A DATA FILE?

DATA FILE MUST BE NAMED "STATES DATA A1" IN FIXED FORMAT.  
THE READ FORMAT IS "13I5" PER 72 CHARACTER LINE.  
TYPE "YES" OR "NO".

YES

THE REDUCED MODEL STATES ARE:

|    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 12 | 13 | 14 | 15 | 43 | 44 | 45 |
| 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 62 | 63 | 64 |
| 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 |
| 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 |    |    |    |    |    |

DO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STATES?

TYPE "YES" OR "NO".

NO

ANALYSIS COMPLETE...YOUR REDUCED SYSTEM DATA HAS BEEN SAVED  
IN A FILE NAMED "OPTMATR DATA".

DO YOU DESIRE ANOTHER RUN?

IF YES, THE RESULTS OF THIS RUN WILL  
REPLACE YOUR LAST OPTMATR DATA.

TYPE "YES" OR "NO".

NO

.....OPTRED IS NOW TERMINATED.....

R; T=17.70/18.51 12:25:07

RECORD OFF

END RECORDING OF TERMINAL SESSION

The eigensystems for both systems were analyzed for  
stability and other desirable characteristics. Bode plots  
were obtained for both input/output combinations and these

graphs for the full and reduced systems are included as Figures 4.3 through 4.10. The input/output combination (1/1) shows the response of the system without the notch filter while the (1/2) input/output combination depicts the frequency response of the system after the addition of the notch filter. These figures clearly indicate excellent agreement between the full and reduced order systems in the frequency domain.

# 16-TH ORDER OPEN LOOP TF BODE MAGNITUDE

INPUT # = 1  
OUTPUT # = 1  
DC GAIN =  $3.418 \times 10^{-2}$

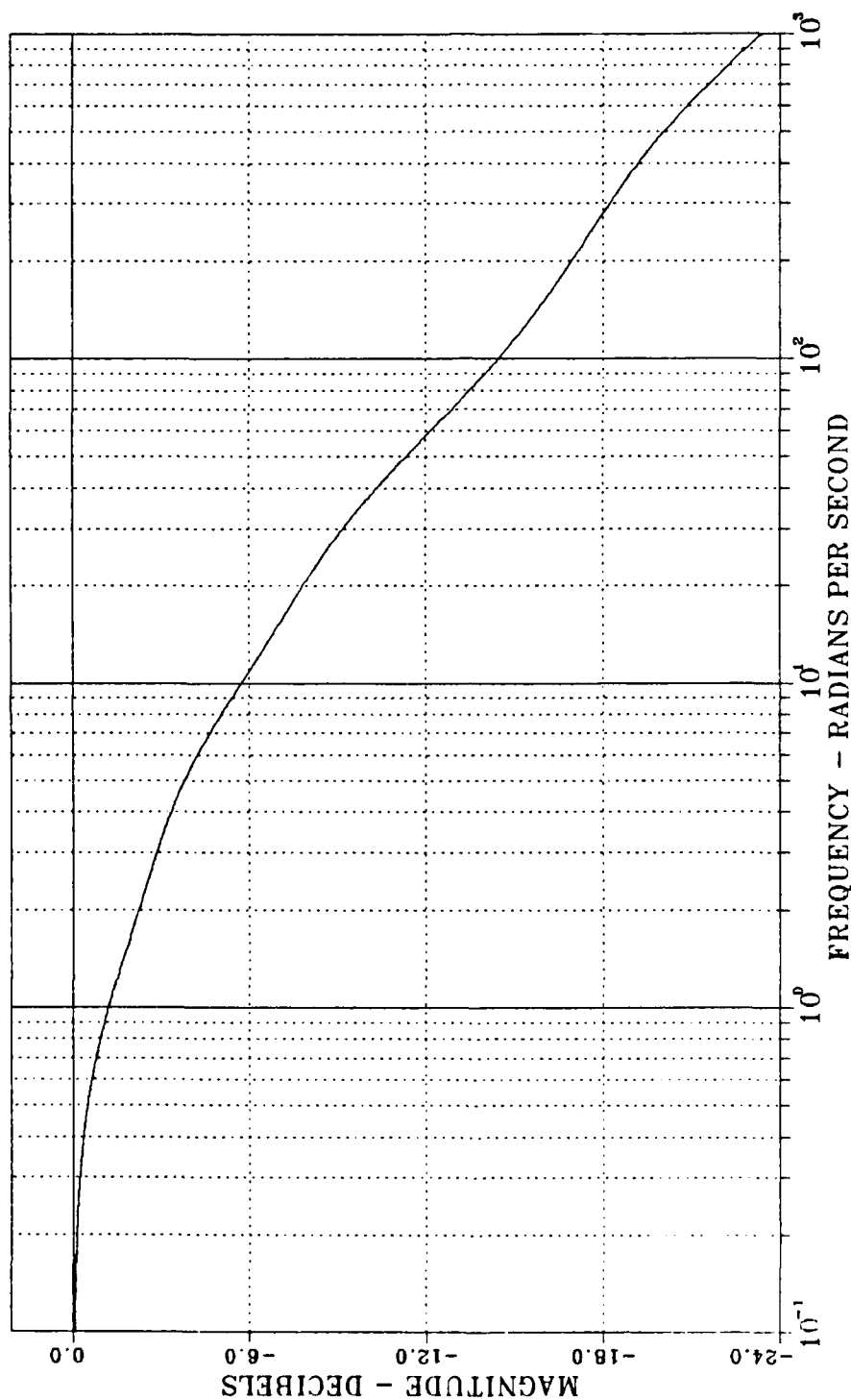


Figure 4.1 Full System Bode Plot for the F100 Engine.

```

IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
GO TO 70
WRITE (5,880)
GO TO 50
CONTINUE
IF (IANS.EQ.IZ) GO TO 560
C-----ISET-----
80 CALL FRTCMS ('CLRSCRN ')
WRITE (5,910)
CALL RDCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 90
GO TO 100
WRITE (5,880)
GO TO 80
CONTINUE
IF (IANS.EQ.IY) ISET=1
C-----
C INITIALIZE SYSTEM FLAGS.
C-----
10 CONTINUE
IRET=0
IOL=0
IQ=0
IR=0
ISS=0
IM=0
ITF1=0
ITF2=0
ITF3=0
IFDFW=0
IE=0
IDSTAB=0
IDEBUG=0
IPSD=0
IYU=0
INORM=0
IREG=0
NS=0
NC=0
NOB=0
NG=0
IRDMAT=0
C-----IOL-----
CALL FRTCMS ('CLRSCRN ')
WRITE (5,570)
CALL RDINT (IANS)
IOL=IANS-1
IF (IOL.EQ.2) GO TO 350

```

```

C 5,D(32,32),DSTORE(32,32),JCF(64),RES(64),AY(32,32),BB(64),CC(64),CP
C 6(32),GW(64),GV(64,64),HY(64,64),HU(64,64),PRTT(16,16)
C-----
C EQUIVALENCE (W11(1,1),GW(1,1)), (W11(1,1),GV(1,1)), (W21(1,1),HY(1
C 1,1)), (W21(1,1),HU(1,1))
C-----
C COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITP1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
C 1UG,ISET,IREG,IPSD,IYU,INORM
C-----
C DATA IY,'Y',IZ,'N'//
C-----
C SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL =
C CONVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.
C-----
C CALL ERRSET {207,256,-1,1,1,209}
C CALL ERRSET {215,256,-1,1}
C-----
C INITIALIZE SAVE FLAGS.
C-----
C ISAF=0
C ISAG=0
C ISAH=0
C IGAM=0
C ISAD=0
C ISAA=0
C ISAB=0
C ISET=0
C-----
C 5 CALL FRTCMS ('CLRSCRN ')
C WRITE (5,885)
C CALL RDINT (IANS)
C IF (IANS.EQ.1) GO TO 20
C IF (IANS.EQ.2) GO TO 10
C GO TO 5
C-----
C-----SCRN1-----
C 20 CALL FRTCMS ('CLRSCRN ')
C WRITE (5,890)
C CALL RDCHAR (IANS)
C IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 30
C GO TO 40
C 30 WRITE (5,880)
C GO TO 20
C 40 CONTINUE
C IF (IANS.EQ.IZ) GO TO 560
C-----
C-----SCRN2-----
C 50 CALL FRTCMS ('CLRSCRN ')
C WRITE (5,900)
C CALL RDCHAR (IANS)

```

**THE OPTSYX PROGRAM AS MODIFIED BY THIS THESIS**

51

thesis. The addition of this program to the analysis technique will provide a further means of verification of the accuracy and fidelity of the reduced order model.

## 2. Interface with Other Analysis Programs

The Naval Postgraduate School's Aeronautical Engineering Department has developed a collection of analysis programs including OPTSYSX, OPTGRAPH and OPTCALC. These programs exist in a single "module" which enables the user to interactively execute these programs during a single computer terminal session. The feasibility of adding OPTRED to this module should be investigated. If necessary, these programs would require re-dimensioning and could create a very large memory size requirement. Additions and modifications would also be necessary in the OPTSYSX EXEC to enable the execution of OPTRED and then transfer execution to OPTSYSX for the analysis of the reduced model. This controlling EXEC must also manage the data files upon which the individual programs depend.

## 3. Alternate Order Reduction Techniques

Finally, other methods of order reduction should be investigated. Comparisons of reduced order models which are generated by different techniques would provide the controls designer with additional means of reduced model verification and analysis.

## V. CONCLUSIONS AND RECOMMENDATIONS

### A. CONCLUSIONS

The computational capabilities of OPTRED were tested by analyzing two significant large order systems. The determination of the validity of both full order and reduced order results was greatly simplified by the eigensystem data for the X-29A provided by NASA-Edwards. As discussed by Laptas [Ref. 6], OPTSYSX failed to produce valid open-loop transfer function zeroes when analyzing very large systems. However, through modal analysis and with the aid of the NASA data, the extraneous zeroes were easily identified and the open loop data created by OPTSYSX was corrected.

The results of OPTSYSX and OPTGRAPH indicate a very strong similarity between the full system and the reduced system generated by OPTRED both in stability and in the frequency domain. The combined analytic capabilities of OPTSYSX, OPTGRAPH and OPTRED provide a powerful tool for the controls designer for use in the analysis, development and the implementation of complex large order systems.

### B. RECOMMENDATIONS

Based upon the results of this thesis the following areas are recommended for further study and research.

#### 1. Comparison of System Time Response

Although the reduced model must closely resemble the full system in the frequency domain, it should also compare favorably in the time domain. OPTCALC, the time response program, requires several modifications to enable its use for analysis of systems as large as those presented in this

# 60-TH ORDER OPEN LOOP TF BODE PHASE

INPUT # = 1  
OUTPUT # = 2  
DC GAIN =  $-4.051 \cdot 10^0$

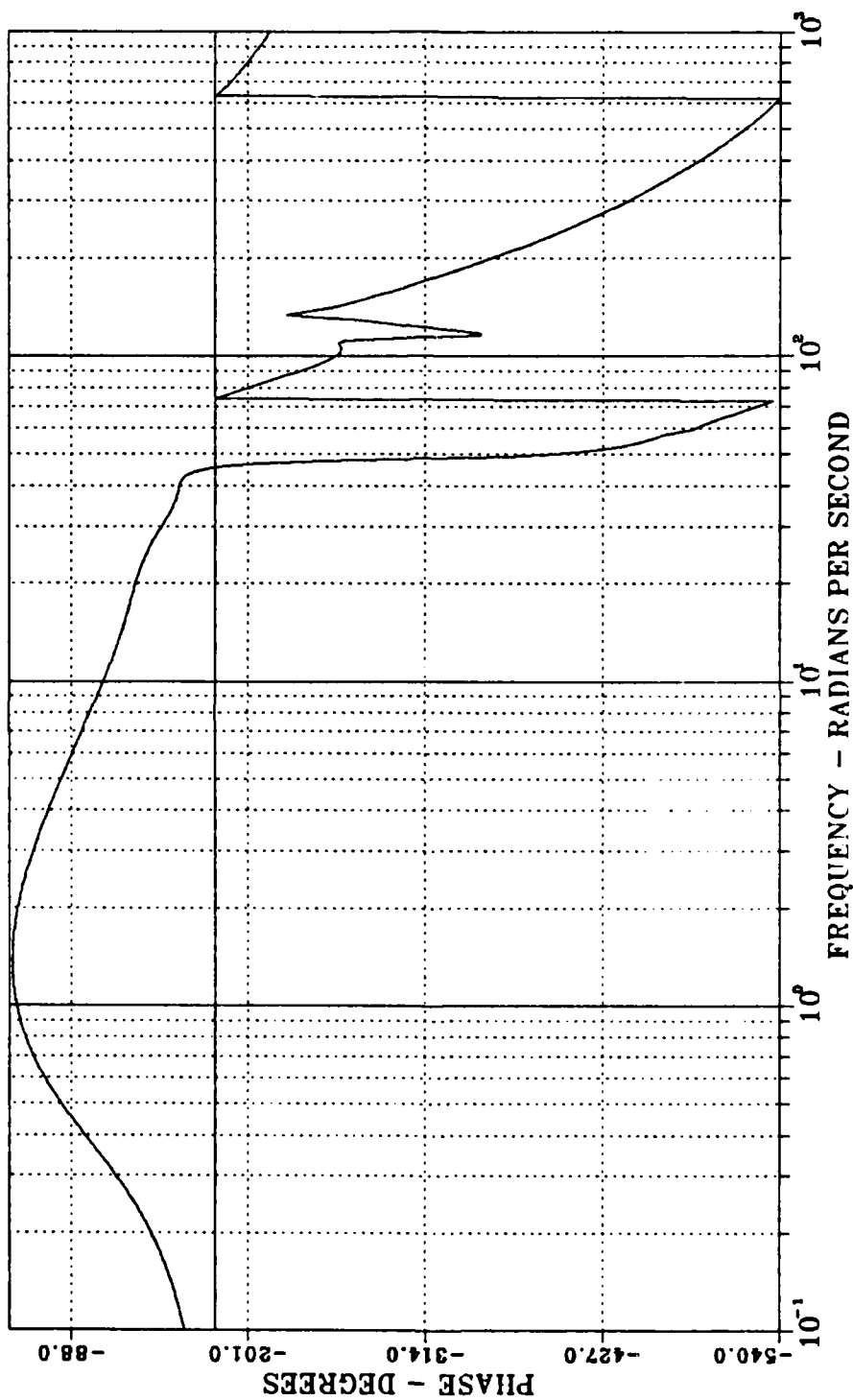


Figure 4.10 X-29A Reduced System With Notch Filter.

98-TH ORDER  
OPEN LOOP TF BODE PHASE

INPUT # = 1  
OUTPUT # = 2  
DC GAIN =  $-3.773 \times 10^0$

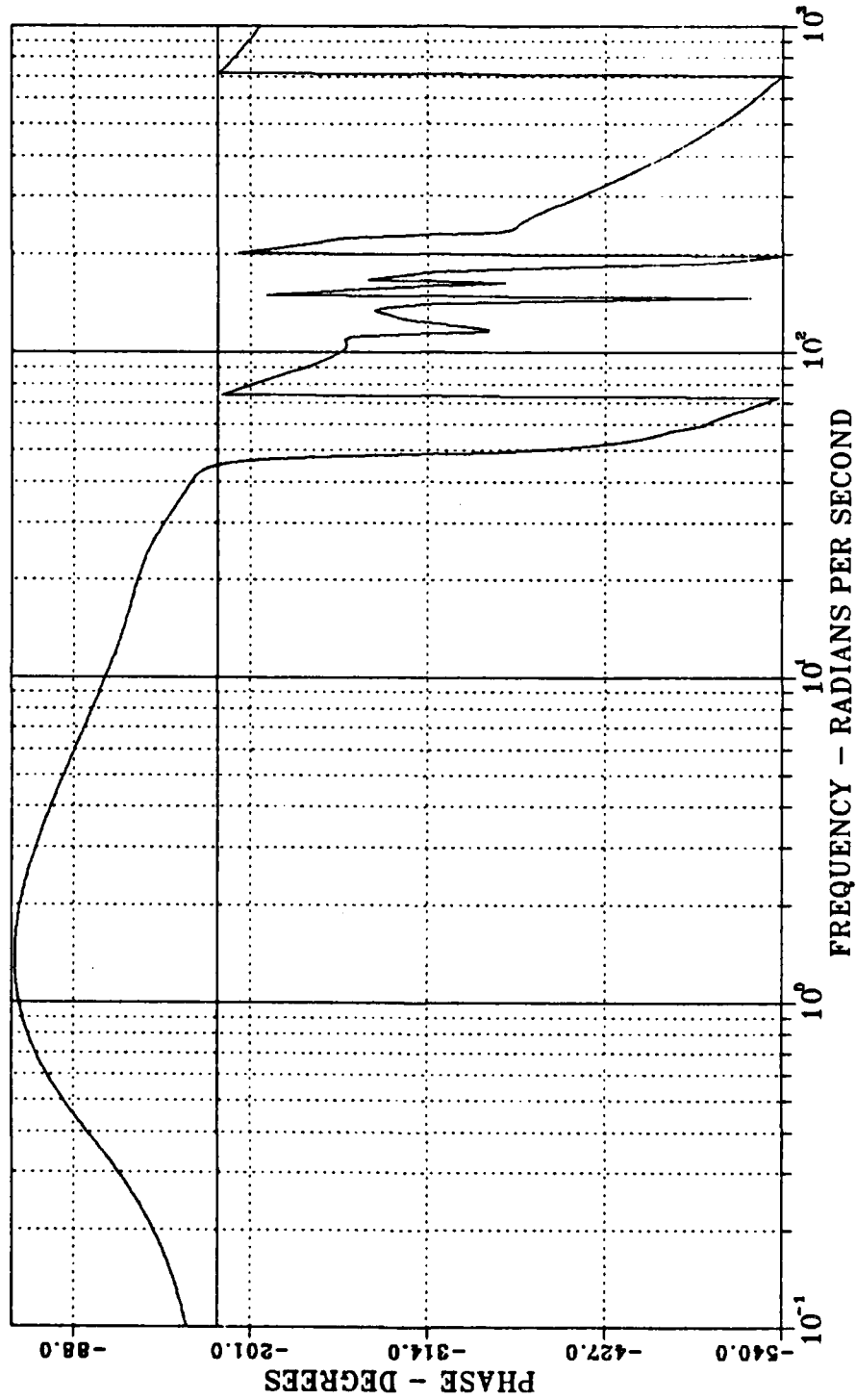


Figure 4.9 X-29A Full System With Notch Filter.

60-TH ORDER  
OPEN LOOP TF BODE MAGNITUDE

INPUT # = 1  
OUTPUT # = 2  
DC GAIN =  $-4.051 \cdot 10^{10}$

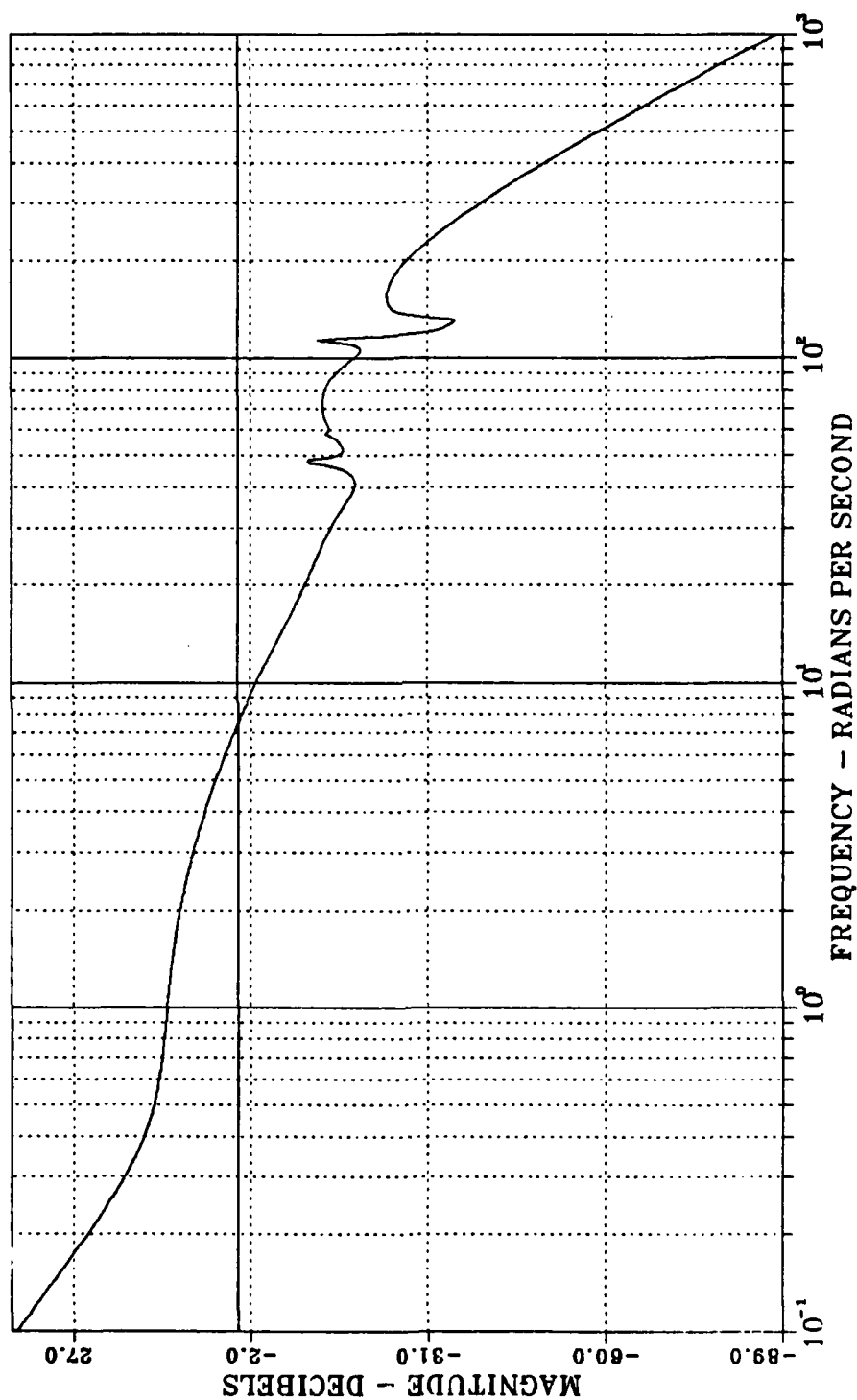


Figure 4.8 I-29A Reduced System With Notch Filter.

98-TH ORDER  
OPEN LOOP TF BODE MAGNITUDE

INPUT # = 1  
OUTPUT # = 2  
DC GAIN =  $-3.773 \times 10^{10}$

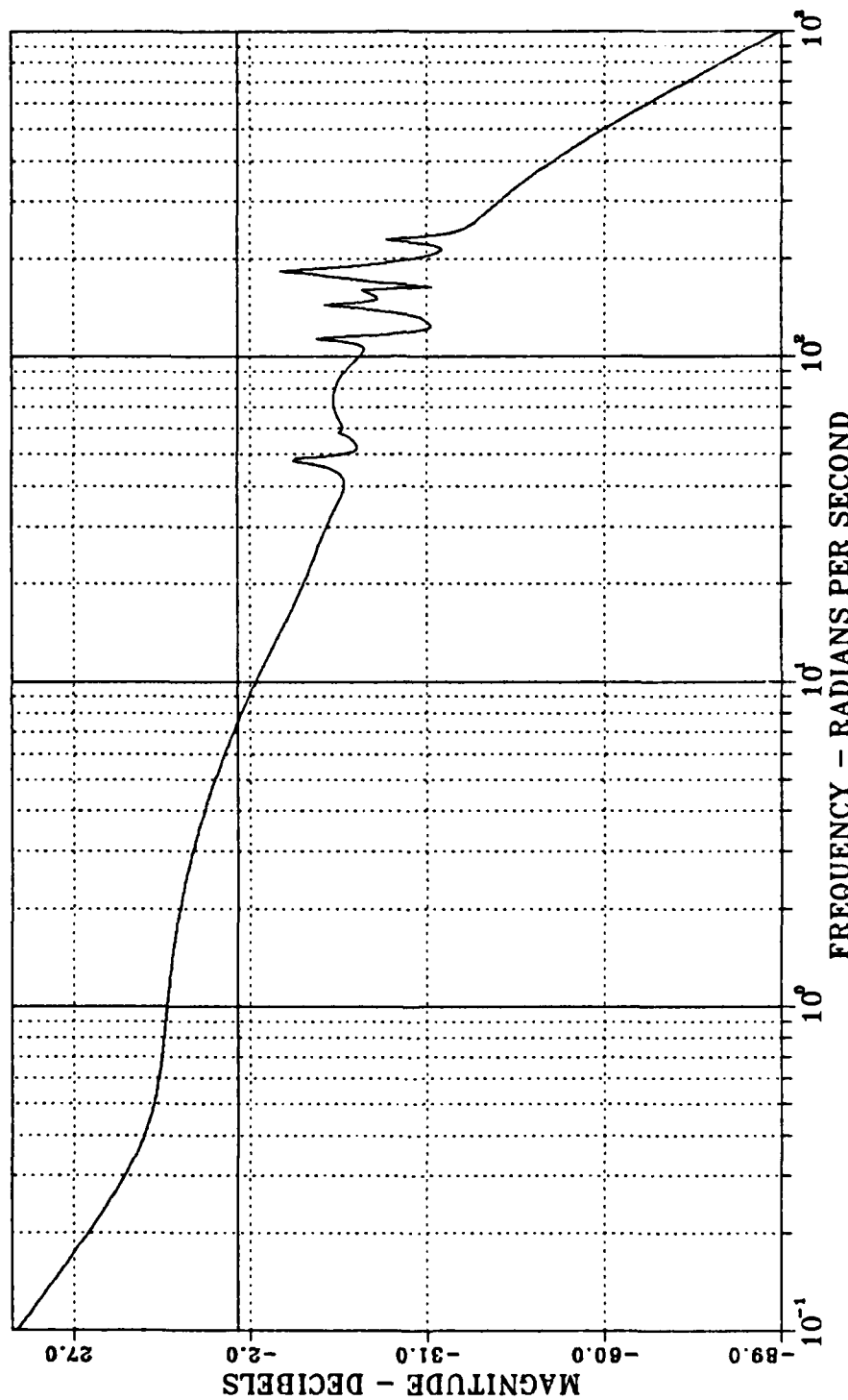


Figure 4.7 X-29A Full System With Notch Filter.

60-TH ORDER  
OPEN LOOP TF BODE PHASE

INPUT # = 1  
OUTPUT # = 1  
DC GAIN = -3.964\*10<sup>10</sup>

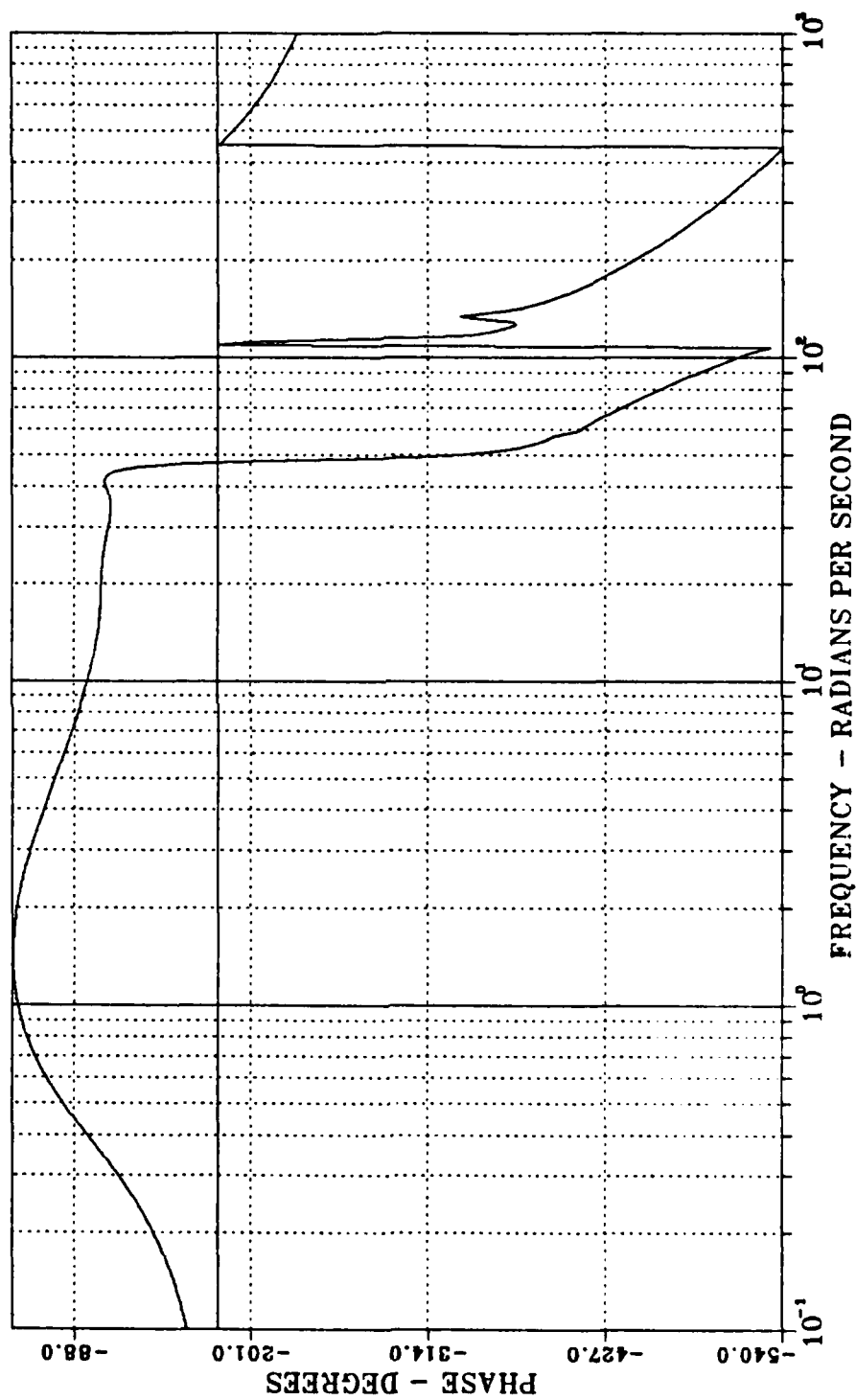


Figure 4.6 X-29A Reduced System Without Notch Filter.

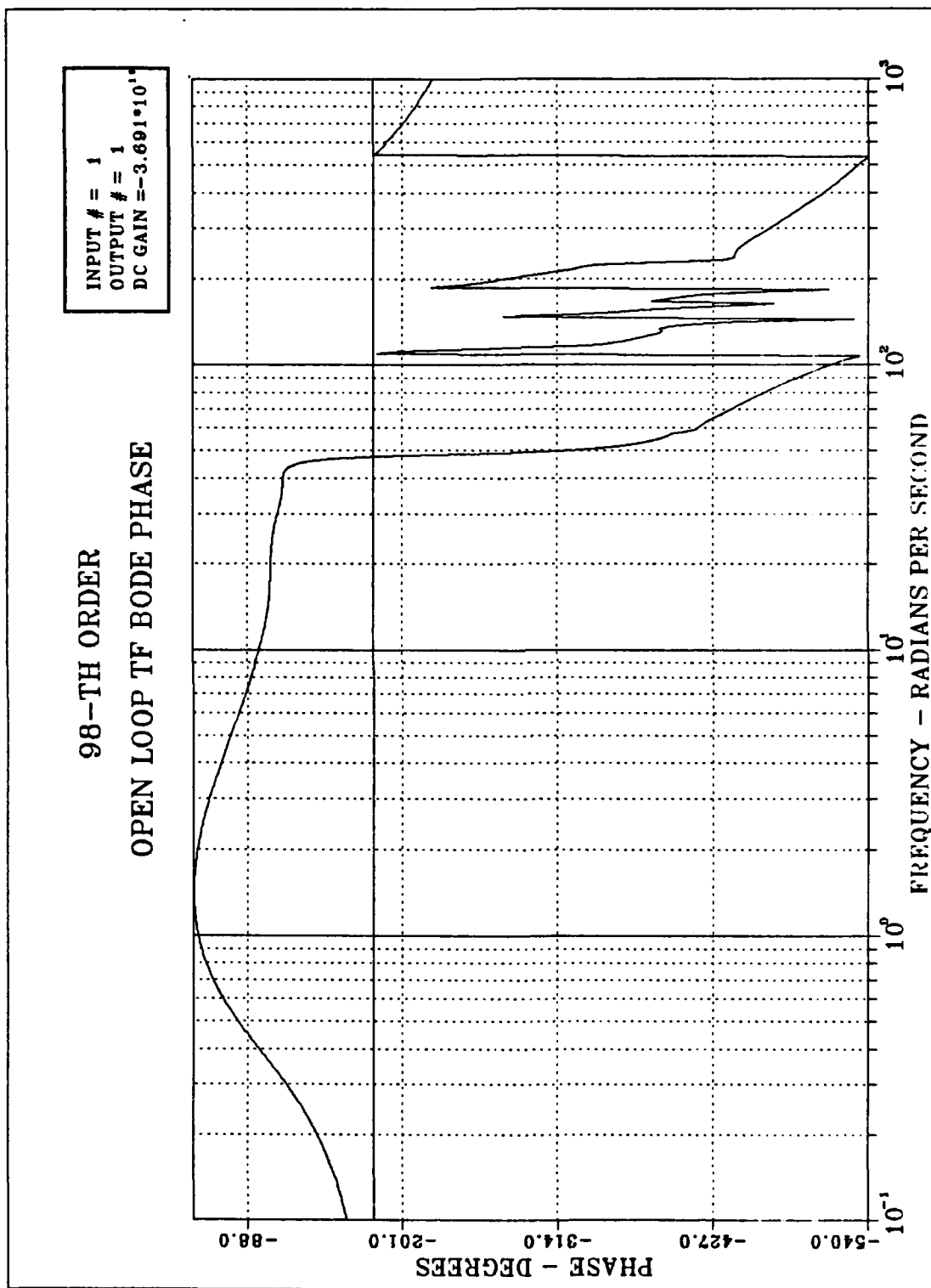


Figure 4.5 X-29A Full System Without Notch Filter.

60-TH ORDER  
OPEN LOOP TF BODE MAGNITUDE

INPUT # = 1  
OUTPUT # = 1  
DC GAIN =  $-3.984 \times 10^{-4}$

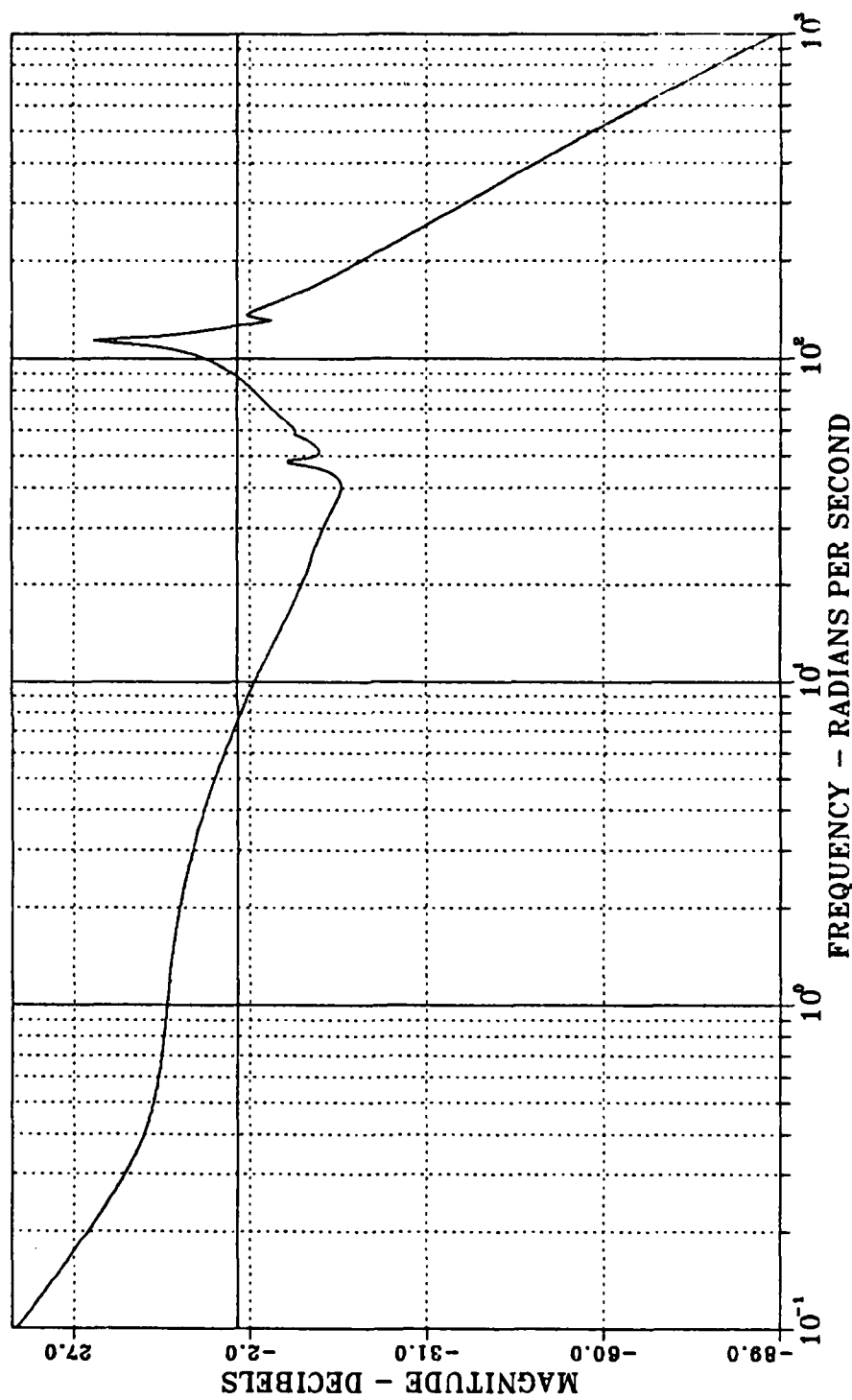


Figure 4.4 X-29A Reduced System Without Notch Filter.

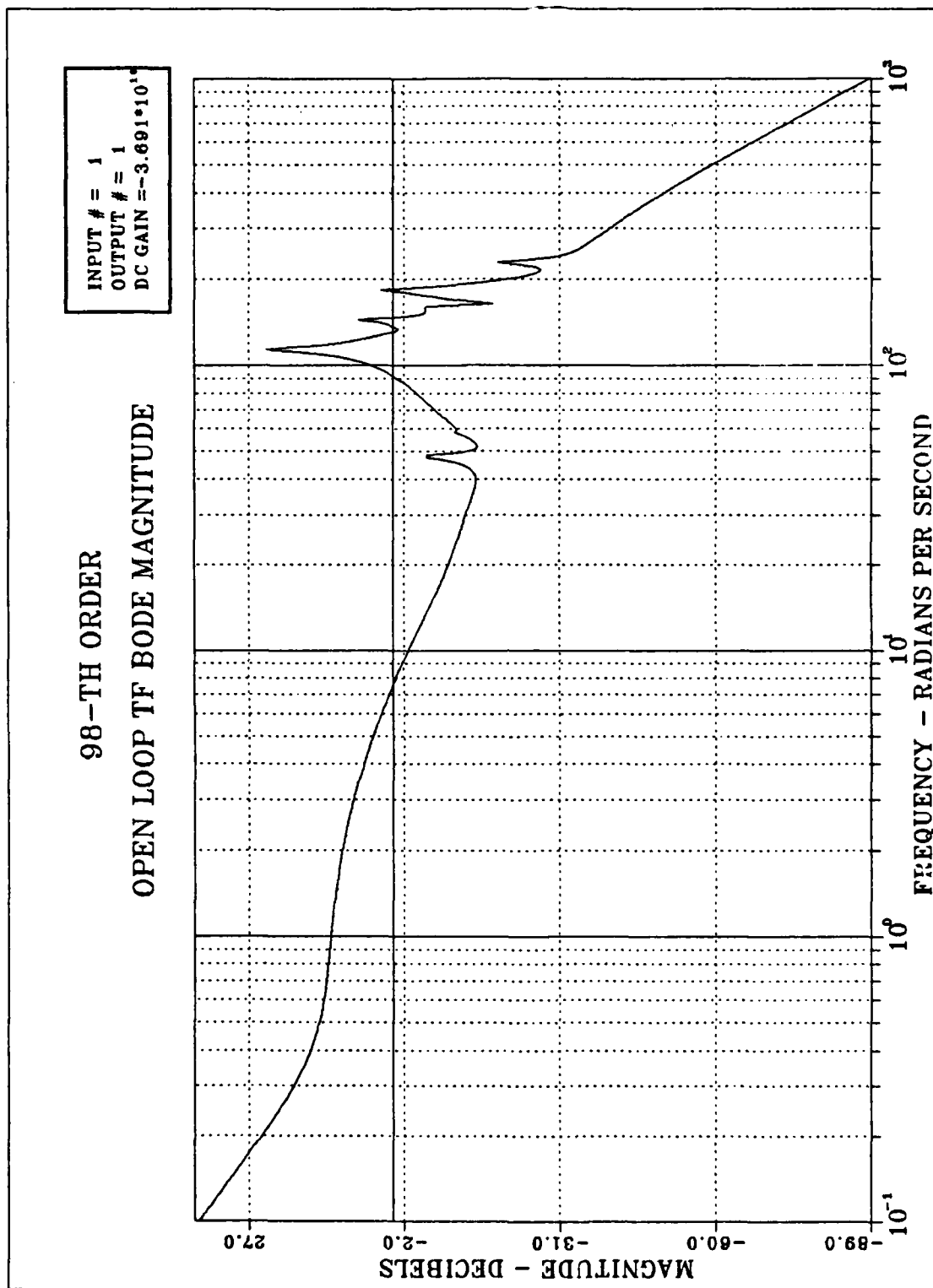


Figure 4.3 X-29A Full System Without Notch Filter.

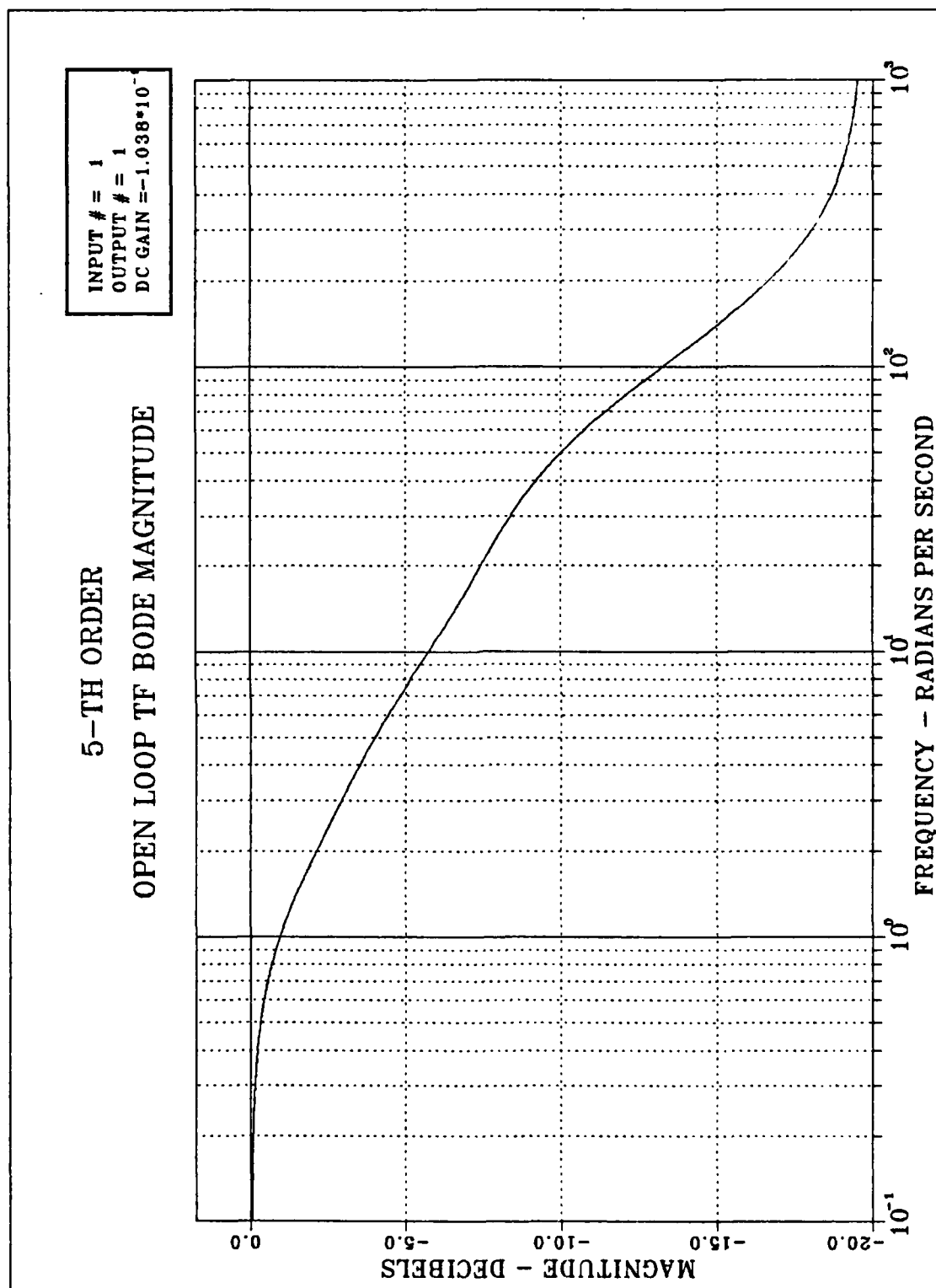


Figure 4.2 Reduced System Bode Plot for the F100 Engine.

```

C-----IQ-----
110 IF (IOL.EQ.3) GO TO 200
    CALL FRTCMS ('CLRSCRN ')
    WRITE (5,600)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 120
    GO TO 130
120 WRITE (5,880)
    GO TO 116
130 CONTINUE
    IF (IANS.EQ.IV) IQ=1
    IF (IANS.EQ.IZ) IQ=0
C-----IR-----
    CALL FRTCMS ('CLRSCRN ')
    WRITE (5,590)
    CALL RDIANT (IANS)
    IR=IANS-1
C-----ISS-----
140 CALL FRTCMS ('CLRSCRN ')
    WRITE (5,600)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 150
    GO TO 160
150 WRITE (5,880)
    GO TO 146
160 CONTINUE
    IF (IANS.EQ.IV) ISS=1
    IF (IANS.EQ.IZ) ISS=0
C-----IM-----
170 WRITE (5,610)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 180
    GO TO 190
180 WRITE (5,880)
    GO TO 176
190 CONTINUE
    IF (IANS.EQ.IV) IM=1
    IF (IANS.EQ.IZ) IM=0
200 CONTINUE
    IF (IOL.EQ.3) IM=1
C-----IFDPH-----
210 CALL FRTCMS ('CLRSCRN ')
    WRITE (5,650)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 220
    GO TO 230
220 WRITE (5,880)
    GO TO 216

```

```

230 CONTINUE
   IF (IANS.EQ.IY) IFDFW=1
   IF (IANS.EQ.IZ) IFDFW=0
   IF (IOL.EQ.3) GO TO 350
C-----IDSTAB-----
240 CALL FRTCMS ('CLRSCRN ')
   WRITE (5,670)
   CALL RDCHAR (IANS)
   IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 250
   GO TO 260
250 WRITE (5,880)
   GO TO 240
260 CONTINUE
   IF (IANS.EQ.IY) IDSTAB=1
   IF (IANS.EQ.IZ) IDSTAB=0
C-----IDEBUG-----
270 WRITE (5,680)
   CALL RDCHAR (IANS)
   IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 280
   GO TO 290
280 WRITE (5,880)
   GO TO 270
290 CONTINUE
   IF (IANS.EQ.IY) IDEBUG=1
   IF (IANS.EQ.IZ) IDEBUG=0
300 CONTINUE
C-----IREG-----
320 CALL FRTCMS ('CLRSCRN ')
   WRITE (5,710)
   CALL RDCHAR (IANS)
   IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 330
   GO TO 340
330 WRITE (5,880)
   GO TO 320
340 CONTINUE
   IF (IANS.EQ.IY) IREG=1
   IF (IANS.EQ.IZ) IREG=0
350 CALL RDHMTF (NS,NC,NOB,NG,ISAF,ISAG,ISAH,ISAD,IGAM,ISAA,ISAB,IRDHMT
1,IFDFW)
   IF ((ISAF.EQ.1).AND. (IRDHMT.EQ.1)) GO TO 352
C-----NS-----
   CALL FRTCMS ('CLRSCRN ')
   WRITE (5,720)
   CALL RDREAL (ANSR)
   NS=IDINT (ANSR)
352 IF (IOL.EQ.2) GO TO 360
   IF ((ISAG.EQ.1).AND. (IRDHMT.EQ.1)) GO TO 354
C-----NC-----

```

```

WRITE (5,730)
CALL RDRÉAL (ANSR)
NC=IDINT (ANSR)
IF ((ISAH.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 356
C-----NOB
354 C-----
WRITE (5,740)
CALL RDRÉAL (ANSR)
NOB=IDINT (ANSR)
IF ((IGAM.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 360
C-----NG
356 C-----
WRITE (5,750)
CALL RDRÉAL (ANSR)
NG=IDINT (ANSR)
CONTINUE
IF (IOL.EQ.2) GO TO 364
IF (IOL.EQ.3) GO TO 310
C-----IPSD
360 C-----
IF (IREG.NE.0.OR.NC.EQ.0) GO TO 310
CALL FRTCHS ('CLRSCRN ')
WRITE (5,690)
CALL RDINT (IANS)
IPSD=IANS
IF (IPSD.EQ.3) IPSD=0
IF (IPSD.EQ.0) GO TO 310
C-----IYU
310 C-----
CALL FRTCHS ('CLRSCRN ')
WRITE (5,700)
CALL RDINT (IANS)
IYU=IANS-1
C-----INORM
310 C-----
CALL FRTCHS ('CLRSCRN ')
WRITE (5,820)
CALL RDRÉAL (ANSR)
INORM=IDINT (ANSR)
CONTINUE
C-----ITF1
310 C-----
CALL FRTCHS ('CLRSCRN ')
WRITE (5,620)
CALL RDINT (IANS)
ITF1=IANS-1
IF (IOL.EQ.3) GO TO 362
C-----ITF2
310 C-----
IF (IREG.NE.1) GO TO 315
CALL FRTCHS ('CLRSCRN ')
WRITE (5,630)
CALL RDINT (IANS)
ITF2=IANS-1
C-----ITF3
310 C-----

```

```

315 IF (IREG.NE.0 .OR. (NC.EQ.0 .OR. NG.EQ.0)) GO TO 362
    CALL PRCHMS ('CLRSCRN ')
    WRITE (5,640)
    CALL RDINT (IANS)
    ITP3=IANS-1
C-----IE-----
362 CONTINUE
    IF ((ITF1+ITF2+ITF3).EQ.0) GO TO 364
    CALL PRCHMS ('CLRSCRN ')
    WRITE (5,660)
    CALL RDRAL (ANSR)
    IE=IDINT (ANSR)
C-----FLAG SETTINGS-----
364 CONTINUE
    CALL PRCHMS ('CLRSCRN ')
    WRITE (6,760)
    WRITE (6,770)
    WRITE (6,780)
    1 IDSTAB
    WRITE (6,790)
    WRITE (6,800)
    WRITE (6,810)
    IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDEBUG,ISET
    IPSD,IYU,INORM,IREG,NS,NC,NOB,NG
    NS,NC,NCB,NG
C-----OPTGRAPH DATA FILES-----
    ZERO = 0
    ONE = 1
    REWIND 10
    ITPX = 1
    WRITE (10,1000) ZERO,ONE,NS,NC,NOB,IE,ITFX
    REWIND 1
    ITPX = 2
    WRITE (1,1000) ZERO,ONE,NS,NG,NOB,IE,ITFX
    REWIND 4
    ITPX = 3
    WRITE (4,1000) ZERO,ONE,NS,NOB,NC,IE,ITFX
C-----BEGIN CALCULATIONS-----
    N2=2*NS
    CALL INNER (NS,NC,NOB,NG,N2,ACL,B,BA,CI,CR,CQ,CWI,D,PBGC,PBGE,
    1G,GAM,GN,HO,D1,D2,ERO,RH,RC,Q,SC,WR,WI,W11,W21,X,WNOBM,WNOBNI,D
    2ESTAB,AA,BM,CM,JCF,RES,AV,BB,CC,CP,GW,GV,HY,HU,DSTORE,ISAF,ISAH,IS
    3AG,ISAD,IGAM,IRET,PRTT,NROW,NCOL,IRDMAT,ISAA,ISAB)
C-----OPTGROL DATA-----
    IF ((ITF1.EQ.1) .OR. (ITF1.EQ.2)) GO TO 396
    END FILE 10
    REWIND 10
    ITPX = 1
    WRITE (10,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ITFX
C-----OPTGRNO DATA-----
396 IF ((ITF2.EQ.1) .OR. (ITF2.EQ.2)) GO TO 397

```

```

END FILE 1
REWIND 1
ITFX = 2
WRITE (1,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ITFX
-----OPTGRCM DATA-----
397 IF (ITP1.EQ.3).OR. (ITF3.EQ.2)) GO TO 398
END FILE 4
REWIND 4
ITFX = 3
WRITE (4,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ITFX
-----
398 IF (IRET.EQ.1) GO TO 370
CALL WRTHAT(BA,G,HO,D,GAM,FBGC,FBGE,AY,B,NS,NC,NOB,NG,IFDFW)
-----IRET-----
370 WRITE (5,830)
CALL RDCHAR (IANS)
IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 380
GO TO 390
380 WRITE (5,880)
GO TO 370
390 CONTINUE
-----RESET OPGRAPH DATA FILE-----
IF (IANS.EQ.IZ) GO TO 395
END FILE 10
REWIND 10
END FILE 1
REWIND 1
END FILE 4
REWIND 4
-----
395 IF (IANS.EQ.IY) GO TO 400
IF (IANS.EQ.IZ) GO TO 560
-----ISAF-----
400 CONTINUE
IF (IRET.EQ.1) GO TO 10
IF (ISET.EQ.1) GO TO 10
CALL FRCHMS ('CLRSCRN ')
410 WRITE (5,840)
CALL RDCHAR (IANS)
IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 420
GO TO 430
420 WRITE (5,880)
GO TO 410
430 CONTINUE
IF (IANS.EQ.IY) ISAF=1
IF (IANS.EQ.IZ) ISAF=0
-----ISAF-----
IF (NOB.EQ.0) GO TO 470

```

440 CALL PRTCMS ('CLRSCRN ')  
 WRITE (5,850)  
 CALL RDCHAR (IANS)  
 IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 450  
 GO TO 460

450 WRITE (5,880)  
 GO TO 440

460 CONTINUE  
 IF (IANS.EQ.IY) ISAH=1  
 IF (IANS.EQ.IZ) ISAH=0  
 470 CONTINUE

-----ISAG-----

480 IF (NC.EQ.0) GO TO 510  
 CALL PRTCMS ('CLRSCRN ')  
 WRITE (5,860)

490 CALL RDCHAR (IANS)  
 IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 490  
 GO TO 500

500 WRITE (5,880)  
 GO TO 480

CONTINUE  
 IF (IANS.EQ.IY) ISAG=1  
 IF (IANS.EQ.IZ) ISAG=0  
 510 CONTINUE

-----ISAD-----

511 IF (IPFW.EQ.0) GO TO 515  
 CALL PRTCMS ('CLRSCRN ')  
 WRITE (5,865)

513 CALL RDCHAR (IANS)  
 IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 513  
 GO TO 514

514 WRITE (5,880)  
 GO TO 511

CONTINUE  
 IF (IANS.EQ.IY) ISAD=1  
 IF (IANS.EQ.IZ) ISAD=0  
 515 CONTINUE

-----IGAM-----

520 IF (NG.EQ.0) GO TO 550  
 CALL PRTCMS ('CLRSCRN ')  
 WRITE (5,870)

530 CALL RDCHAR (IANS)  
 IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 530  
 GO TO 540

540 WRITE (5,880)  
 GO TO 520

CONTINUE  
 IF (IANS.EQ.IY) IGAM=1

```

550 IF (IANS.EQ.IZ) IGAM=0
C CONTINUE-----ISAA-----
551 CALL FRTCHS ('CLRSCRN ')
WRITE (5,872)
CALL RDCHAR (IANS)
IF ((IANS.EQ.IY).OR. (IANS.EQ.IZ)) GO TO 553
WRITE (5,880)
GO TO 551
553 CONTINUE
IF (IANS.EQ.IY) ISAA=1
IF (IANS.EQ.IZ) ISAA=0
C-----ISAB-----
555 CALL FRTCHS ('CLRSCRN ')
WRITE (5,874)
CALL RDCHAR (IANS)
IF ((IANS.EQ.IY).OR. (IANS.EQ.IZ)) GO TO 557
WRITE (5,880)
GO TO 555
557 CONTINUE
IF (IANS.EQ.IY) ISAB=1
IF (IANS.EQ.IZ) ISAB=0
GO TO 10
C-----TERMINATE-----
560 WRITE (5,920)
STOP
C-----
885 FORMAT (5X,33HIN CASE OF PROBLEMS CALL EXT 2826,///10X,42H1 DEFI
1NES MATRIX EQUATIONS USED IN OPTSYS,15X,32HAND ALTERNATE DATA IN
2PUT METHODS,///10X,26H2 START INTERACTIVE INPUT,///15X,13HENTER
3 1 OR 2,///5X,59H NORMALLY TWO SUCCESSIVE NULL ENTERS TERMINATES T
4HE PROGRAM.)
890 FORMAT (5X,59HOPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS
1CONTROL,8X,45HPFOLLOWING TYPES OF SYSTEMS CONTROL EQUATIONS: ///15X
3 35HXDOF = IF)*X + {G}*U + {IGAM}*{H+W0),///20X,22HNEASUREMENT EQUA
4TION ---15X,21HZ = JH)*X + {D}*U + V,///20X,29HREGULATOR PERFORM
5ANCE INDEX ---15X,42HJ = JH)*X + {D}*U + V,///20X,29HREGULATOR PERFORM
6//20X,32HSTATE FEEDBACK GAIN DEFINITION,///25X,10HU = -{C}*X,///
7 15X,45HDO YOU WISH TO CONTINUE? TYPE "YES" OR "NO".)
900 FORMAT (25X,14H--DATA ENTRY---5X,49HALTHOUGH OPTSYSX IS SPECIFI
1CALLY DESIGNED TO READ,5X,48HALL MATRIX DATA INTERACTIVELY, SEVE
2RAL ALTERNATE,5X,31HMETHODS ARE AVAILABLE TO USERS: ///10X,43HNE
3THOD 1--THE F,G,H,D, AND "GAMMA" MATRICES,///13X,37H MAY BE READ FRO
4M SEPARATE DATA FILES,///10X,50HMETHOD 2--THE F,G,H,D, AND "GAMMA
5" MATRICES MAY BE,13X,45HEXPlicitly DEFINED WITHIN SUBROUTINE "S
6ETUP",13X,52HNOTE: IN EITHER CASE THE USER SHOULD OBTAIN A C
7OPY,///17X,39HTHE EXAMP

```



```

650 6FILTER ARE SYNTHESIZED,/,22X,14HAND/OR INPUT.},/,10X,32HSELECT AN
660 7OPTION: 1,2,3,39H WILL A FEED-FORWARD DISTRIBUTION MATRIX,/,5X,25H {
1"DN" - MATRIX,/,5X,63H BE INPUT?/,15X,19H TYPE "YES" OR "NO".)
1WHEN A MARKOV,/,8X,58H PARAMETER IS ZERO-THE MARKOV PARAMETER INDIC
3ATES THE ORDER,/,8X,52H ALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED
4OUT AND,/,8X,4H LESS THAN 10.0**(-1E) IS CONSIDERED ZERO.//,8X,47H THE
50.//,8X,4H HISS OF THIS PARAMETER (IE) IS 6.//,8X,28H IN OTHER WORDS
7,ESP = 1.0E-6.//,10X,66H IF YOU DESIRE A DIFFERENT MARKOV CRITERIA
8,TYPE "NO" {ZERO}.//,10X,48H IF YOU DESIRE THE DEFAULT VALU
9,TYPE "NO" {ZERO}.
670 1REGULATOR,/,5X,61H DO YOU DESIRE TO SYNTHESIZE A STABLE FILTER FOR R
2{NOTE:WORKS FOR FILTER OR REGULATOR BUT NOT FOR BOTH,/,20X,17H IN T
3HE SAME RUN.},/,10X,19H TYPE "YES" OR "NO".)
680 1M,/,8X,50H PRIOR TO DECOMPOSITION {FOR CHECKING THE PROGRAM}?/,10
2X,19H TYPE "YES" OR "NO".)
690 1FORMAT (/,5X,39H POWER SPECTRAL DENSITY {PSD} OPTION 1 :/,10X,53
1HOPTION 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE,/,22X,48H CO
2NTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY,/,22X,45H PROCESS AN
3D MEASUREMENT NOISE. {NOTE: BOTH A,/,22X,46H REGULATOR AND A FILTE
4R MUST BE RESIDENT IN THE,/,22X,28H PROGRAM TO USE THIS OPTION.},/,22
5,10X,53H OPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE,/,22
6X,34H RESIDUES OF EACH TRANSFER FUNCTION,/,22X,28H USED IN THE PSD C
7OMPUTATION.//,10X,24H OPTION 3 -- NOT DESIRED.//,10X,29H SELECT A
8N OPTION: 1,2,3,OR 3.)
700 1FORMAT (/,5X,39H POWER SPECTRAL DENSITY {PSD} OPTION 2 :/,10X,35
1HOPTION 1 -- PSD OUTPUT NOT DESIRED.//,10X,38H OPTION 2 -- COMPUTE
2ONLY OUTPUT PSD.//,10X,39H OPTION 3 -- COMPUTE ONLY CONTROL PSD
3.//,15X,32H SELECT AN OPTION: 1,2,3,OR 4.)
710 1FORMAT (/,5X,39H DO YOU DESIRE REGULATOR SYNTHESIS ONLY?/,10X,19
1H TYPE "YES" OR "NO".//)
720 1FORMAT (/,5X,47H ENTER THE # OF STATES {NS} OF THE SYSTEM MATRIX,/,
15X,13H {"F"-MATRIX}.//)
730 1FORMAT (/,5X,56H ENTER THE # OF CONTROLS {NC} OF THE CONTROL SYSTEM
1MODEL,/,5X,13H {"G"-MATRIX}.//)
740 1FORMAT (/,5X,54H ENTER THE # OF MEASUREMENTS OR OBSERVATIONS {NO} O
1THE,/,5X,13H {"H"-MATRIX}.//)
750 1FORMAT (/,5X,48H ENTER THE # OF PROCESS NOISE SOURCES {NG} OF THE,/,
15X,17H {"GAMMA"-MATRIX}.//)
760 1FORMAT (5X,52H FLAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLOWS:
1)
770 1FORMAT (1X,3H IOL,2X,2H IQ,2X,2H IR,2X,3H ISS,2X,2H IM,2X,4H ITF1,2X,4H I

```



```

3 INTERVAL,,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,,5X,27
4 HINDIVIDUAL MATRIX ELEMENTS.//15X,19HTYPE "YES" OR "NO".}
FORMAT (//,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?//5X,39HNOTE: T
1 THE MATRIX WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE
2 INTERVAL,,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,,5X,27
4 HINDIVIDUAL MATRIX ELEMENTS.//15X,19HTYPE "YES" OR "NO".}
FORMAT (//,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?//5X,39HNOTE: T
1 THE MATRIX WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE
2 INTERVAL,,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,,5X,27
4 HINDIVIDUAL MATRIX ELEMENTS.//15X,19HTYPE "YES" OR "NO".}
FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
920 FORMAT (//,41H.....OPTSYSX IS NOW TERMINATED.....{A//)
C-----OPTGRAPH DATA-----
1000 FORMAT (11,14,515)
C-----
END
C-----
SUBROUTINE SETUP (BA,G,GAM,HO,D,NS,NC,NG,NO)
C-----
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION BA(NS,NS),G(NS,NC),GAM(NS,NG),HO(NO,NS),D(NO,NC)
COMMON/PROG/ IOL,IOLR,ISS,IM,ITF1,ITF2,ITF3,IPDFW,IE,IDSTAB,IDEB
1UG,ISET,IREG,IPSD,IYU,INORM
C-----
C FILE DEFINITIONS
C-----
CALL FRCHS ('FILEDEF','03','DISK','X16X16','
1'DATA
C-----
THIS IS AN EXAMPLE OF A 83 X 84 DATA FILE X29A83 DATA A1 READ FROM
A USER'S DISK AND CONVERTED (FROM A "DUMMY" ARRAY NAMED 'DUM') TO A
SYMMETRIC ARRAY. THE FORMAT STATEMENT MUST MATCH YOUR DISK DATA
FORMAT OR THE PROGRAM WILL FAIL! NOTE: ALL PROGRAM DIMENSIONS
MUST BE ENLARGED ACCORDINGLY FOR A SYSTEM OF THIS SIZE.
C-----
READ (3,100)
READ (3,100)
DO 20 I=1,NS
READ (3,56) (BA(I,J),J=1,NS)
C0 CONTINUE
C-----
THESE ARE EXAMPLES OF SEVERAL POSSIBLE METHODS OF ARRAY GENERATION
WITHIN SUBROUTINE SETUP. THE "GAM" ARRAY WAS SET TO ZERO SINCE NO
"NOISE" WAS PRESENT, AND THE NON-ZERO ELEMENTS OF THE "G" ARRAY WERE
EXPLICITLY DEFINED. THEY COULD ALSO BE READ FROM FILES AS ABOVE.
C-----

```

```

DO 40 I=1, NS
DO 30 J=1, NC
DO 25 K=1, NO
GAH(I, J)=0.0D+00
G(I, J)=0.0D+00
D(K, J)=0.0D+00
CONTINUE
CONTINUE
CONTINUE
G(82, 1)=0.1000D+00
G(52, 1)=0.362D+07
G(77, 1)=-0.1591D+02
G(78, 1)=0.2448D+00
G(79, 1)=0.2448D+00
G(81, 1)=0.1000D+00

```

25  
30  
40

```

READ (3, 100)
DO 70 I=1, NS
READ (3, 56) (G(I, J), J=1, NC)
CONTINUE

```

C70

THIS IS AN EXAMPLE OF ONE POSSIBLE METHOD OF ARRAY GENERATION  
WITHIN THE PROGRAM ITSELF. FOR VERY LARGE DATA ARRAYS, THIS METHOD  
MAY BE PREFERABLE TO SOME USERS OVER INTERACTIVE ENTRY OF EACH  
INDIVIDUAL ELEMENT.

```

DO 2 I=1, 11
DO 1 J=1, 82
HO(I, J)
HO(1, 1)
HO(2, 75)
HO(3, 74)
HO(4, 63)
HO(5, 62)
HO(6, 76)
HO(7, 44)
HO(8, 45)
HO(9, 46)
HO(16, 47)
HO(11, 48)
CONTINUE
CONTINUE
GO TO 90
CONTINUE

```

C1  
C2  
C3

```

READ (3, 100)
DO 90 I=1, NO
READ (3, 56) (HO(I, J), J=1, NS)

```

C

```

C90 CONTINUE
DO 1 I=1,NS
1  READ (3,50) (BA(I,J),J=1,NC),I=1,NS)
   READ (3,50) (G(I,J),J=1,NC),I=1,NS)
   READ (3,50) (H(I,J),J=1,NC),I=1,NS)
   READ (3,50) (D(I,J),J=1,NC),I=1,NS)
   RETURN
C-----
50  FORMAT (6E12.5)
C50  FORMAT (5E12.4)
100  FORMAT (3(D22.15,3X))
END
C=====
C  SUBROUTINE CHECK (EPS,NC,NG,NO,IRET)
C  CHECKS THE CONSISTENCY OF REQUESTED OPTIONS.
C=====
      DOUBLE PRECISION EPS
      COMMON /PROG/ IOL,IO,IR,ISS,IM,ITF1,ITF2,ITF3,IFDPW,IE,IDSTAB,IDEB
      IUG,ISET,IREG,IPSD,IW,I,NORM
C-----SET MODAL ANALYSIS WHEN OL EIGENSYS OR OL TF REQUESTED-----
      IF (IM.EQ.1 .AND. IOL.EQ.0) IOL=1
      IF (IOL.EQ.3 .OR. ITF1.NE.0) IM=1
C-----CHECK TO SEE IF H MATRIX INPUT-----
      IF (NO.NE.0 .OR. IOL.GE.2) GO TO 10
      WRITE (5,90)
      IRET=1
      RETURN
10  CONTINUE
C-----TRANSFER FUNCTION CHECKS-----
      IF (IE.EQ.0) IE=6
      EPS=10.**(-IE)
C-----OPEN LOOP TF-----
      IF (ITF1.EQ.0 .OR. NC.NE.0) GO TO 20
      WRITE (5,100)
      IRET=1
      RETURN
C-----COMPENSATOR TF-----
20  IF (ITF3.EQ.0) GO TO 30
   IF (IREG.EQ.0 .AND. (NC.NE.0 .AND. NG.NE.0)) GO TO 30
   WRITE (5,110)
   IRET=1
   RETURN
30  CONTINUE
C-----NOISE TF-----
   IF (ITF2.EQ.0) GO TO 40
   IF (NG.NE.0 .AND. NC.NE.0) GO TO 40
   WRITE (5,120)

```

```

C-----DESTABILIZATION RESTRICTIONS-----
IRET=1
RETURN
40 IF (IDSTAB .EQ. 0) GO TO 50
   IF (NC .EQ. 0) GO TO 50
   IF (NG .NE. 0) IREG=1
   WRITE (5,130)
   IF (IREG .EQ. 1) GO TO 50
   IRET=1
   RETURN
50 CONTINUE
C-----PSD INPUT-----
   IF (IPSD .EQ. 0) GO TO 80
   IF (IPSD .LT. 0 .OR. IPSD .GT. 3) GO TO 60
   IF (IYU .LT. 0 .OR. IYU .GT. 3) GO TO 60
   IF (INORM .LT. 0 .OR. INORM .GT. NG+NO) GO TO 60
   GO TO 70
60 WRITE (5,140)
   IRET=1
   RETURN
70 IF (IREG .EQ. 0 .AND. NC .NE. 0) GO TO 80
   WRITE (5,150)
   IRET=1
   RETURN
80 CONTINUE
   RETURN
C-----MATRIX MUST BE INPUT, I.E. "NO" MUST BE > 0.-----
90 FORMAT (//,5X,49H H - MATRIX MUST BE INPUT, I.E. "NO" MUST BE > 0.
1//)
100 FORMAT (//,5X,46H (G) MATRIX MUST BE INPUT, I.E. NC MUST BE > 0.,//,
110X 26HTO COMPUTE OPEN LOOP T. F.//)
110 FORMAT (//,5X,48H REGULATOR AND FILTER SYNTHESIS MUST BE REQUESTED,
1//,5X,44H IN THE SAME RUN TO COMPUTE COMPENSATOR T. F.,//,5X,47H I.E.
2IREG MUST = 0., "NC" AND "NG" MUST BE > 0.,//)
120 FORMAT (//,5X,51H NOISE T. F. CALCULATED ONLY WHEN REGULATOR DESIGN
1ED.,//,5X,47H I.E. IREG MUST = 1., "NC" AND "NG" MUST BE > 0.,//)
130 FORMAT (//,5X,47H DESTABILIZATION OPTION DESIGNED FOR A REGULATOR, /
1,5X,38H OR FILTER BUT NOT BOTH SIMULTANEOUSLY.,//,5X,55H IF "NG" > 0
2. THE REGULATOR OPTION IS AUTOMATICALLY SET.,//)
140 FORMAT (//,5X,49H ***** INCONSISTENT PSD INPUT FLAGS *****
1//)
150 FORMAT (//,5X,44H BOTH A REGULATOR AND FILTER MUST BE RESIDENT.,//,10
1X,42HTO COMPUTE THE PSD OF A CONTROLLED SYSTEM!.,//,10X,42H I.E. IREG
2 MUST BE 0. AND "NC" MUST BE > 0.,//)
END
C-----
SUBROUTINE INNER (NS, NC, NO, NG, N2, ACL, B, BA, CI, CR, CO, CWI, CWR, D, FBGC,
1FBGE, G, GAM, GM, GN, HO, b1, b2, PRO, RM, RC, Q, SC, SR, WI, W11, W21, X, WNORM, WN0

```

```

1480 FORMAT (//, 25X, 28H, ... DESTABILIZATION CASE, ...//, 10X, 39H, THE FOLLOW
1490 WING VALUES WILL BE ADDED DOWN, ...//, 10X, 49H, THE DIAGONAL OF THE "F" MA
1500 TRIX TO DESTABILIZE IT, ...//, 10X, 41H, OPTIMAL GAINS FOR THE DESTABILIZE
1510 3D SYSTEM, ...//, 10X, 39H, ARE THEN USED AS FIXED SUBOPTIMAL GAINS, ...//, 10X, 28
1520 4H, FOR THE SYSTEM CALCULATIONS, ...//)
1530 FORMAT (//, 43H, PROGRAM TERMINATING DUE TO UNSTABLE SYSTEM)
1540 FORMAT (//, 5X, 31H, OPEN LOOP TRANSFER FUNCTIONS, ...//)
1550 FORMAT (//, 5X, 32H, EULER-LAGRANGE SYSTEM MATRIX, ...//)
1560 FORMAT (//, 5X, 43H, EIGENVALUES AND EIGENVECTORS OF THE 2N X 2N, ...//, 5X,
1570 145H, EULER-LAGRANGE SYSTEM AFTER HQR2, ...//)
1580 FORMAT (//, 1X, 1P2D13.6)
1590 FORMAT (//, 1X)
1600 FORMAT (//, 5X, 41H, EIGENSYSTEM OF OPTIMAL REGULATOR, ...//)
1610 FORMAT (//, 5X, 41H, EIGENSYSTEM OF OPTIMAL ESTIMATOR, ...//)
1620 FORMAT (//, 5X, 39H, EIGENVECTORS FROM REGAIN PRIOR TO CNORM, ...//)
1630 *GT*S, ...//)
1640 FORMAT (//, 10, 2X, 1PD11.4)
1650 FORMAT (//, 5X, 45H, THE CLOSED LOOP DYNAMICS MATRIX, ...//, F-G*C, ...//)
1660 FORMAT (//, 60H, PROGRAM TERMINATING DUE TO UNSTABLE CLOSED LOOP
1670 SYSTEM)
1680 FORMAT (//, 2X, NOISE TRANSFER FUNCTIONS THROUGH THE CLOSED-LOOP SY
1690 STEM, ...//)
1700 FORMAT (//, 5X, 45H, FILTER STEADY STATE GAINS, ...//, K, ...//)
1710 FORMAT (//, 1X, 2X, 1P6D14.6)
1720 FORMAT (//, 5X, 45H, THE CLOSED LOOP FILTER DYNAMICS MATRIX IS, ...//)
1730 FORMAT (//, 5X, 43H, PROGRAM TERMINATING DUE TO UNSTABLE FILTER)
1740 FORMAT (//, 5X, 45H, THE COVARIANCE OF THE ESTIMATION ERROR, ...//, P, ...//)
1750 FORMAT (//, 5X, 45H, RMS VALUES OF THE ESTIMATION ERROR, ...//)
1760 FORMAT (//, 5, 1X, 1PD13.6)
1770 FORMAT (//, 5X, 45H, THE COVARIANCE OF THE ESTIMATE, ...//, XHAT, ...//)
1780 FORMAT (//, 5X, 45H, THE STATE COVARIANCE MATRIX, ...//, X=XHAT + P, ...//)
1790 FORMAT (//, 5X, 45H, THE CONTROL COVARIANCE, ...//, U=C*XHAT*CT, ...//)
1800 FORMAT (//, 1X, 2X, 1P6D14.6)
1810 FORMAT (//, 2X, 18H, STATE RMS RESPONSE, 20X, 20H, CONTROL RMS RESPONSE, ...//)
1820 FORMAT (//, 1X, 2X, 1P6D15.7, 25X, D15.7)
1830 FORMAT (//, 5X, 50H, COMPENSATOR TRANSFER FUNCTIONS FROM MEAS. TO INPU
1840 T, ...//, 5X, 52H, U/Z = -C*(SI-F+G*C+K*H) INV*K, ...//)
1850 FORMAT (//, 5X, 46H, STEADY DISTURBANCE VECTOR, ...//, W, ...//)
1860 FORMAT (//, 10, 1X, 1PD12.4)
1870 FORMAT (//, 5X, 45H, STEADY STATE VALUES OF STATE VAR. ARE, ...//)
1880 FORMAT (//, 5X, 47H, STEADY STATE CONTROL IS, ...//)
1890 FORMAT (//, 10, 1X, 1PD12.4)
1900 FORMAT (//, 5X, 49H, ENTER THE MAGNITUDE OF THE DESTABILIZATION VECTOR
1910 1/8X, 47H, TO BE ADDED DOWN THE DIAGONAL OF THE "F"-MATRIX, ...//, 8X, 18H,
1920 DESTABILIZE IT, ...//)
1930 C-----OPGRAPH DATA-----
1940 FORMAT (5X, 2D30.14)

```

```

1290 CONTINUE
C-----COMPUTE PSD FUNCTIONS OF THE CONTROLLED SYSTEM-----
IF (IPSD.EQ.0) GO TO 1310
IF (IYU.LT.3) GO TO 1300
CALL PSDCAL (M,NS,RM,X,NC,GW,GV,PBGC,NO,HY,HU,HO,PBGE,NG,
1 GAM,ACLB,BA,WR,WI,D1,L2,JCF,KES,Q,RC,BB,CC,1,IPSD,INORM)
CALL PSDCAL (M,NS,RM,X,NC,GW,GV,PBGC,NO,HY,HU,HO,PBGE,NG,
1 GAM,ACLB,BA,WR,WI,D1,L2,JCF,KES,Q,RC,BB,CC,2,IPSD,INORM)
GO TO 1310
1300 CALL PSDCAL (M,NS,RM,X,NC,GW,GV,PBGC,NO,HY,HU,HO,PBGE,NG,
1 GAM,ACLB,BA,WR,WI,D1,L2,JCF,KES,Q,RC,BB,CC,1,IPSD,INORM)
1310 IF (ISS.EQ.0) RETURN
IF (NC.NE.0) GO TO 1330
DO 1320 I=1,NS
DO 1320 J=1,NS
ACLB(I,J)=BA(I,J)
1320 CONTINUE
CALL MINV (NSQ,ACL,NS,DDD,D1,D2)
CALL READW (NG,WR)
WRITE (6,170) (WR(I),I=1,NG)
1340 DO 1340 I=1,NS
WI(I)=0.0
DO 1340 J=1,NG
WI(I)=WI(I)+GAM(I,J)*WR(J)
DO 1360 I=1,NS
CR(I)=0.0
DO 1350 J=1,NS
CR(I)=CR(I)-ACL(I,J)*WI(J)
1350 CR(I)=CR(I)+PBGC(I,J)*CR(J)
1360 WRITE (6,1390) CR(I)
DO 1370 I=1,NC
CI(I)=0.0
DO 1370 J=1,NS
CI(I)=CI(I)+PBGC(I,J)*CR(J)
1370 CI(I)=CI(I)+PBGC(I,J)*CR(J)
WRITE (6,1790) (CI(I),I=1,NC)
RETURN
C-----
C670 FORMAT (2X,1P6D14.6,2X,6D14.6)
1380 FORMAT (10(2X,0PD11.4))
1390 FORMAT (10(2X,0PD11.4))
1400 FORMAT (10(2X,0PD11.4))
1410 FORMAT (10(2X,0PD11.4))
1420 FORMAT (10(2X,0PD11.4))
1430 FORMAT (10(2X,0PD11.4))
1440 FORMAT (10(2X,0PD11.4))
1450 FORMAT (10(2X,0PD11.4))
1460 FORMAT (10(2X,0PD11.4))
1470 FORMAT (10(2X,0PD11.4))

```

```

1120 CONTINUE
DO 1140 I=1,NC
DO 1140 J=1,NC
SC(I,J)=0.D0
DO 1130 K=1,NS
SC(I,J)=SC(I,J)+FBGC(I,K)*PRO(K,J)
1140 CONTINUE
IF (IRRG.EQ.0) GO TC 1170
DO 1160 I=1,NS
DO 1160 J=1,NS
CQ(I,J)=GH(I,J)
GO TO 1190
1160 WRITE (6,1700)
CALL RAPRNT (MH,MH,MH,5,GM,4,'(5(1X,1PD13.6))')
IF (IR.GT.2) GO TO 1150
DO 1180 I=1,MH
DO 1180 J=1,MH
CQ(I,J)=GN(I,J)+GH(I,J)
1180 CONTINUE
WRITE (6,1710)
CALL RAPRNT (MH,MH,MH,5,CQ,4,'(5(1X,1PD13.6))')
IF (NC.EQ.0) GO TO 1210
WRITE (6,1720)
DO 1200 I=1,NC
DO 1200 J=1,NC
SC(I,J),J=1,NC)
1200 WRITE (6,1730)
DO 1220 I=1,NS
CQ(I,I)=DSORT(CQ(I,I))
1220 IF (NC.EQ.0) GO TO 1240
DO 1230 I=1,NC
SC(I,I)=DSORT(SC(I,I))
1230 WRITE (6,1740)
DO 1250 I=1,NS
IF (I.LE.NC) WRITE (6,1750) CQ(I,I),SC(I,I)
IF (I.GT.NC) WRITE (6,1750) CQ(I,I)
1250 CONTINUE
IF (ITF3.EQ.0) GO TC 1290
1260 IF (FORM COMPENSATOR FROM HEADS TO INPUT AND COMPUTE TF)-----
DO 1280 I=1,NS
DO 1280 J=1,NS
SUM=0.D0
DO 1270 K=1,NO
SUM=SUM+FBGE(I,K)*HO(K,J)
1270 CQ(I,J)=ACL(I,J)-SUM
1280 WRITE (6,1760)
ITFX=3
IZERO=0
CALL TF (NS,NS,NSQ,CQ,AA,NO,FBGE,BM,NC,FBGC,CM,IZERO,D,BB,CC,CP,
1WR,WI,CWR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF3,ITFX)

```

```

980      DO 980 I=1,MH
          X(I,I)=DSQRT(GN(I,I))
          WRITE(6,1690){X(I,I),I=1,MH}
          IF(I0.EQ.0) GO TO 1260
          DO 1080 I=1,NC
          DO 1000 J=1,NS
          SUM=0.0
          DO 990 K=1,NS
          SUM=SUM+FBGC(I,K)*GN(K,J)
          X(I,J)=SUM
          DO 1020 I=1,NS
          DO 1020 J=1,NS
          SUM=0.0
          IF(NC.EQ.0) GO TO 1020
          DO 1010 K=1,NC
          SUM=SUM+G(I,K)*X(K,J)
          PRO(I,J)=CQ(I,J)+SUM
          CALL SCOV(NS,SC,W11,CWR,CWI,NS,GM,W21,CR,CI,PRO,BA)
          IF(NC.EQ.0) GO TO 1040
          DO 1030 I=1,NC
          DO 1030 J=1,NS
          W21(I,J)=0.0
          DO 1030 K=1,NS
          W21(I,J)=W21(I,J)+FBGC(I,K)*BA(J,K)
          DO 1060 I=1,NS
          DO 1060 J=1,NS
          SUM=0.0
          IF(NC.EQ.0) GO TO 1060
          DO 1050 K=1,NC
          SUM=SUM+G(I,K)*W21(K,J)
          PRO(I,J)=SUM
          DO 1070 I=1,NS
          DO 1070 J=1,NS
          PRO(I,J)=PRO(I,J)+CQ(I,J)+PRO(J,I)
          PRO(J,I)=PRO(I,J)
          CALL SCOV(NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,PRO,CQ)
          DO 1080 I=1,NS
          DO 1080 J=1,NS
          GM(I,J)=CQ(I,J)-BA(I,J)+GN(I,J)
          GM(J,I)=GM(I,J)
          GO TO 1100
          CALL SCOV(NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,CQ,GM)
          IF(NC.EQ.0) GO TO 1150
          DO 1120 I=1,NS
          DO 1120 J=1,NC
          PRO(I,J)=0.0
          DO 1110 K=1,NS
          PRO(I,J)=PRO(I,J)+GM(I,K)*FBGC(J,K)

```

```

CALL BALBAK (NS,NS,LOW,HIGH,D1,NS,GM)
WRITE (6,1560)
C-----NORMALIZE AND PRINT SUBOPT. ESTIMATOR EIGENSYSTEM-----
IWRITE=5
CALL CNORM (CR,CI,GM,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,HO,AA,
1NO,NS)
DO 880 I=1,NS
IF (CR(I).LT.0.0) GO TO 880
WRITE (5,1660)
RETURN
CONTINUE
880 GO TO 900
IF (IO.EQ.0) GO TO 1260
DO 910 I=1,NO
DO 910 J=1,MH
PRO(I,J)=0.0
DO 910 K=1,NO
PRO(I,J)=PRO(I,J)+RC(I,K)*PBGE(J,K)
DO 920 I=1,MH
DO 920 J=1,MH
CQ(I,J)=0.0
DO 920 K=1,NO
CQ(I,J)=CQ(I,J)-PBGE(I,K)*PRO(K,J)
920 CQ(I,J)=CQ(I,J)-PBGE(I,K)*PRO(K,J)
930 CONTINUE
C-----THE RMS STATE AND CONTROL RESPONSES-----
IR=IR+1
GO TO (1090,1090,940,940), IR
DO 950 I=1,NS
DO 950 J=1,NG
X(I,J)=0.0
DO 950 K=1,NG
X(I,J)=X(I,J)+GAM(I,K)*Q(K,J)
DO 970 I=1,NS
DO 970 J=1,NS
SUM=0.0
DO 960 K=1,NG
SUM=SUM-X(I,K)*GAM(J,K)
PRO(I,J)=SUM+CQ(I,J)
PRO(J,I)=PRO(I,J)
CQ(I,J)=SUM
CQ(J,I)=SUM
W21(I,J)=GAM(I,J)
W21(J,I)=GAM(J,I)
CALL MINV (NSQ,W21,CR,CI,NS,GM,W21,CR,CI,PRO,GN)
CALL SCOV (NS,GM,W21,CR,CI,NS,GM,W21,CR,CI,PRO,GN)
WRITE (6,1670)
CALL RAPANT (MH,MH,MH,5,GN,4,'(5(1X,1PD13.6))')
WRITE (6,1680)

```

```

C-----
IF (IDSTAB.EQ.1) GO TO 770
-----NORMALIZE AND PRINT OPT. ESTIMATOR EIGENSYSTEM-----
IWRITE=4
CALL CNORM (CR,CI,PRO,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,HO,AA,
1NO,NS)
770 DO 780 I=1,MH
DO 780 J=1,NO
780 PRO(I,J)=+HO(J,I)/RC(J,J)
DO 790 I=1,MH
DO 790 J=1,NO
FBGE(I,J)=0.DO
DO 790 K=1,MH
790 FBGE(I,J)=FBGE(I,J)+GN(I,K)*PRO(K,J)
IF (IDSTAB.EQ.1) GO TO 810
WRITE(6,1670)
CALL RAPRNT (MH,MH,MH,5,GN,4,'(5(1X,1PD13.6))')
WRITE(6,1680)
DO 800 I=1,MH
800 X(I,I)=DSQRT(GN(I,I))
WRITE(6,1690)
810 WRITE(6,1630)
DO 820 I=1,MH
820 WRITE(6,1640) (FBGE(I,J),J=1,NO)
C-----COMPUTE MODAL K MATRIX {OPEN LOOP U-INV SAVED IN WNORMI}-----
IF (IM.NE.1) GO TO 830
CALL MODE (WNORMI,FBGE,AA,MH,MH,NO,4)
830 CONTINUE
-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
IF (IDSTAB.EQ.0) GO TO 850
DO 840 I=1,NS
DO 840 J=1,NS
840 BA(I,J)=BA(I,J)-DSTORE(I,J)
IR=2
850 CONTINUE
DO 870 I=1,NS
DO 870 J=1,NS
SUM=0.0
DO 860 K=1,NO
860 SUM=SUM+FBGE(I,K)*HO(K,J)
870 PRO(I,J)=BA(I,J)-SUM
WRITE(6,1650)
CALL RAPRNT (NS,NS,NS,5,PRO,4,'(5(1X,1PD13.6))')
IF (IR.LT.2) GO TO 890
CALL BALANC (NS,NS,PRC,LOW,HIGH,D1)
CALL ORTHES (NS,NS,LOW,HIGH,PRO,D2)
CALL ORTRAN (NS,NS,LOW,HIGH,PRO,D2,GM)
CALL HOR2 (NS,NS,LOW,HIGH,PRO,CR,CI,GM,IERR)
IF (IERR.NE.0) CALL EREXIT (NS,PRO,IERR)

```

```

C      RCOVARIANCE
C      ***R IS NOXNO MEASUREMENT NOISE =
C      RCOVARIANCE
C      ***HO IS NOXNS MEASUREMENT MATRIX=
C      ***GM IS NSXNG STATE DISTURBANCE =
C      DISTRIBUTION MATRIX
C-----
C      -HOT*RIN*HO      -FT      ---
C-----
C      CALL READR (NO,RC)
C      WRITE (6,1450)
C      DO 680 I=1,NO
C      WRITE (6,1390) (RC(I,J),J=1,NO)
C      IF (ITF2.EQ. 0) GO TO 700
C-----
C      NOISE TRANSFER FUNCTIONS-----
C      WRITE (6,1620)
C      ITFX=2
C      IZERO=0
C      CALL TP (NS,NS,NSQ,ACL,AA,NG,GAM,BH,NO,HO,CH,IZERO,D,BB,CC,CP,WR,
C      1HI,CHRC,CHI,SC,JCF,RES,D1,D2,DDD,EPS,ITF2,ITFX)
C      IF (IREG.EQ. 1) RETURN
C      CONTINUE
C      IF (IREG.EQ. 1) GO TO 930
C      IF (IR.IT.2) GO TO 710
C      CALL READFE (NS,NO,FBGE)
C      GO TO 810
C      CONTINUE
C-----
C      THE MEASUREMENT MATRIX (HOT*RIN*HO==>SC-----
C      DO 720 I=1,NO
C      DO 720 J=1,MH
C      PRO(I,J)=HO(I,J)/RC(I,I)
C      DO 730 I=1,MH
C      DO 730 J=1,MH
C      RH(I+MH,J)=0.D0
C      DO 730 K=1,NO
C      RH(I+MH,J)=RH(I+MH,J)-HO(K,I)*PRO(K,J)
C-----
C      DO 740 I=1,NS
C      DO 740 J=1,NS
C      RH(I,J)=BA(I,J)
C      RH(I+MH,J+MH)=-BA(J,I)
C      RH(I,J+NS)=CQ(I,J)
C      GO TO 400
C-----
C      C-GO BACK TO 450 TO SET UP THE FILTER HAMILTONIAN: CALC. THE FILTER GAINS
C      750 CALL RGAIN (M,NS,NC,NCH,WR,WI,X,GN,GM,RH,W21,D1,CR,C1,PRO,MHS,D2)
C-----
C      CHECK EIGENVECTORS-----
C      IF (IDEBUG.EQ. 0) GO TO 760
C      WRITE (6,1570)
C      CALL RAPANT (NS,NS,NS,9,PRO,4,'(9(1X,1PD13.6))')
C      CONTINUE
C      760

```



```

C-----FEEDBACK GAINS----> {U = - (BINVERSE)*GT*GN}-----
C-----CALCULATE GT-----
DO 490 I=1,NC
DO 490 J=1,NS
PRO(I,J)=0.DO
DO 480 K=1,MH
PRO(I,J)=PRO(I,J)+G(K,I)*GN(K,J)
480 PBGC(I,J)=-PRO(I,J)/B(I,I)
490 IF (IDSTAB.EQ.1) GO TO 500
C-----NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM-----
IWRITE=2
CALL CNORM (CNR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,PBGC,
,AA,NC,NS)
C-----THE OPTIMUM FEEDBACK CONTROL GAINS-----
500 WRITE (6,1580)
DO 510 I=1,NC
510 WRITE (6,1590) (PBGC(I,J),J=1,NS)
C-----COMPUTE MODAL C MATRIX {OPEN LOOP U-INVERSE SAVED IN WNORMI}-----
IF (IN.NE.1) GO TO 530
C-----
C IN COMPUTING MODAL C RECOMPUTE U OPEN LOOP SINCE WNORM USED TO STORE
C U & U-INV FOR CLOSED LOOP SYSTEMS; WNORMI USED TO SAVE U-INV OPEN LOOP
C-----
DO 520 I=1,NS
DO 520 J=1,NS
WNORM(I,J)=WNORMI(I,J)
520 CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
CALL MODE (WNORM,PBGC,AA,NS,NC,NS,3)
530 CONTINUE
C-----THE CLOSED LOOP DYNAMICS MATRIX-----
DO 550 I=1,NS
DO 550 J=1,NS
SUM=0.DO
DO 540 K=1,NC
SUM=SUM+G(I,K)*PBGC(K,J)
540 ACL(I,J)=BA(I,J)+SUM
550 WRITE (6,1600)
CALL RAPANT (MH,MH,MH,5,ACL,4,'(5(1X,1PD13.6))')
IF (IR.NE.1.AND.IR.NE.3) GO TO 590
DO 560 I=1,NS
DO 560 J=1,NS
GN(I,J)=ACL(I,J)
NS,GN,LOW,IHIGH,D1)
560 CALL BALANC (NS,NS,LOW,IHIGH,GN,D2)
CALL ORTHES (NS,NS,LOW,IHIGH,GN,D2,SC)
CALL ORTHAN (NS,NS,LOW,IHIGH,GN,CNR,CWI,SC,IERR)
CALL HOR2 (NS,NS,LOW,IHIGH,GN,CNR,CWI,SC,IERR)
IF (IERR.NE.0) CALL EREXIT (NS,GN,IERR)
CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,SC)

```

```

RM(I,J+MH)=0.D0
DO 380 K=1,NC
  RM(I,J+MH)=RM(I,J+MH)-G(I,K)*PRO(K,J)
C-----2NX2N HAMILTONIAN MATRIX-----
C-----DIAGONAL BLOCKS-----M11 AND M22-----
DO 390 I=1,MH
  DO 390 J=1,MH
    RM(I,J)=BA(I,J)
    RM(I+MH,J+MH)=-BA(J,I)
C-----M21 BLOCK-----
390 RM(I+MH,J)=-RM(I+MH,J)
C-----M12 BLOCK IS DEFINED IN LINE 430 ABOVE-----
400 CONTINUE
IF (IDEBUG.EQ.0) GO TO 410
WRITE (6,1510)
CALL RAPRNT (M,M,M,9,EM,4,'(9(1X,1PD13.6))')
CALL BALANC (M,M,RM,LCW,IGH,D1)
CALL ORTHES (M,M,LOW,IGH,IGH,IGH,IGH,D2)
CALL ORTRAN (M,M,LOW,IGH,IGH,IGH,IGH,D2,X)
CALL HOR2 (M,M,LOW,IGH,IGH,IGH,IGH,D2,X,IERR)
IF (IERR.NE.0) CALL ERXIT (M,RM,IERR)
CALL BALBAK (M,M,LOW,IGH,D1,M,X)
C-----DEBUG DIAGNOSTICS ON EULER-LAGRANGE EQUATIONS-----
IF (IDEBUG.EQ.0) GO TO 430
WRITE (6,1520)
DO 420 I=1,M
  WR(I),WI(I)
  WRITE (6,1530) WR(I),WI(I)
  CALL RAPRNT (M,M,M,9,X,4,'(9(1X,1PD13.6))')
  CONTINUE
IF (IDSTAB.EQ.1) GO TO 440
IF (NOB.EQ.0) WRITE (6,1550)
IF (NOB.NE.0) WRITE (6,1560)
IF (NOB.NE.0) GO TO 750
CALL RGAIN (M,NS,NC,NCB,WR,WI,X,GN,W11,RM,W21,D1,CWR,CWI,SC,MHS,D2
1)
C-----CHECK EIGENVECTORS-----
IF (IDEBUG.EQ.0) GO TO 450
WRITE (6,1570)
CALL RAPRNT (NS,NS,NS,9,SC,4,'(9(1X,1PD13.6))')
CONTINUE
C-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
IF (IDSTAB.EQ.0) GO TO 470
DO 460 I=1,NS
  BA(I,I)=BA(I,I)-DESTAB(I)
  IR=1
460 CONTINUE
470 CONTINUE
C-----CALCULATION OF FEEDBACK GAIN-----

```



```

110 DDD=DDD+AA(I,K)*WNORMI(K,J)
120 DSTORE(I,J)=DDD
130 BA(I,J)=BA(I,J)+DDD
CONTINUE
IF (NO.EQ.0) GO TO 145
IF (ISET.EQ.1) GO TO 135
CALL READH (NO,NS,ISAH,HO)
CONTINUE
WRITE (6,1440)
DO 140 I=1,NO
WRITE (6,1390) (HO(I,J),J=1,NS)
140 IF (IM.NE.1) GO TO 150
CALL MODE (WNORM,HO,CM,NS,NO,NS,2)
150 CONTINUE
IF (IPDFW.EQ.0) GO TO 170
IF (ISET.EQ.1) GO TO 155
CALL READD (NO,NC,ISAL,D)
155 WRITE (6,1470)
DO 160 I=1,NO
WRITE (6,1390) (D(I,J),J=1,NC)
160 CONTINUE
170 NOB=0
IF (NC.EQ.0) GO TO 590
IF (IOL.EQ.3) GO TO 270
IF (IR.NE.1.AND.IR.NE.3) GO TO 210
IF (ISET.EQ.1) GO TO 180
CALL READG (NS,NC,ISAG,G)
CONTINUE
180 CALL READFB (NC,NS,FBGC)
WRITE (6,1400)
DO 190 I=1,NS
WRITE (6,1390) (G(I,J),J=1,NC)
190 IF (IM.NE.1) GO TO 200
CALL MODE (WNORMI,G,BH,NS,NS,NC,0)
200 CONTINUE
GO TO 330
210 DO 220 I=1,NS
DO 220 J=1,NS
RM(I+MH,J)=0.0
220 CALL READDAY (NO,ISAA,AY)
DO 240 I=1,NO
DO 240 J=1,NS
DDD=0.00
DO 230 K=1,NO
DDD=DDD+AY(I,K)*HO(K,J)
230 AA(I,J)=DDD
240 WRITE (6,1460)
DO 250 I=1,NO

```

```

50 IF (IDSTAB.EQ.0) GO TC 50
   WRITE (6,1480)
   WRITE (6,1390) (DESTAB(I), I=1, NS)
   CONTINUE
C----- EIGENSYSTEM OF THE OPEN LOOP DYNAMICS -----
   IF (IOL.EQ.0.AND.IQ.EQ.0) GO TO 90
   IF (IOL.EQ.0.AND.NC.NE.0) GO TO 90
   DO 60 I=1, NS
   DO 60 J=1, NS
   GN(I, J)=BA(I, J)
   CALL BALANC (NS, NS, GN, LOW, IHIGH, D1)
   CALL ORTHES (NS, NS, LOW, IHIGH, GN, D2)
   CALL ORTRAN (NS, NS, LOW, IHIGH, GN, D2, SC)
   CALL HOR2 (NS, NS, LOW, IHIGH, GN, CWR, SC, IERR)
   IF (IERR.NE.0) CALL EXEXIT (NS, GN, IERR)
   CALL BALBAK (NS, NS, LOW, IHIGH, D1, NS, SC)
C----- NORMALIZE AND PRINT OPEN LOOP EIGENSYSTEM -----
   IWRITE=1
   CALL CNORM (CWR, CWI, SC, NS, IWRITE, NSQ, DDD, D1, D2, WNORM, WNORMI, HO, CM,
1NO, NS)
C----- OPTGROL/OPTGRNO DATA -----
   DO 64 I=1, NS
   WRITE (10,2000) CWR(I), CWI(I)
   WRITE (1,2000) CWR(I), CWI(I)
   CONTINUE
C-----
64 IF (IOL.EQ.2) RETURN
   IF (IQ.EQ.0.OR.(NC.NE.0.OR.IDSTAB.GT.0)) GO TO 90
   DO 70 I=1, NS
   IF (CWR(I).LT.0.) GO TO 70
   WRITE (5,1490)
   RETURN
70 CONTINUE
   IF (IOL.EQ.3) GO TO 130
   DO 80 I=1, NS
   DO 80 J=1, NS
   W11(I, J)=SC(I, J)
   CALL MINV (NSQ, W11, NS, DDD, D1, D2)
90 CONTINUE
C-----
   IF (IDSTAB.EQ.0) GO TC 130
   FORM U * DIAG(DESTAB) * U-INV-----
C-----
   DO 100 J=1, NS
   DO 100 I=1, NS
   AA(I, J)=WNORM(I, J)*DESTAB(J)
100 DO 120 I=1, NS
   DO 120 J=1, NS
   DDD=0.D0
   DO 110 K=1, NS

```

```

2RMI DESTAB,AA, RM, CM, JCF, RES, AY, BB, CC, CP, GW, GV, HY, HU, DSTORE, ISAF, IS
3AH, ISAG, ISAD, IAH, IRET, PRIT, NROW, NCOL, IRDMAT, ISAA, ISAB)
C=====
C IMPLICIT REAL*8(A-H,O-Z)
C-----
C DIMENSION ACL(NS,NS), B(NC,NC), BA(NS,NS), CI(NS), CR(NS), CO(NS,NS), CW
1I(NS), CWR(NS), FBGC(NC,NS), FBGE(NS,NO), G(NS,NS), GH(NS,NS), PRO(NS,NS)
2) RC(NO,NO), SC(NS,NS), WR(N2), WT(N2), W1(N2), W2(NS,NS), X(N2,N2)
3) GN(NS,NS), HO(NO,NS), L1(N2), D2(N2), AM(N2,N2), Q(NG,NG), D(NO,NC), GAH
4(NS,NG), WNRM(NS,NS), WNRMI(NS,NS), DESTAB(NS), AA(NS,NS), BM(NS,NC)
5CH(NO,NS), JCF(N2), RES(N2), AY(NO,NO), BB(N2), CC(N2), CP(NS), GW(N2,NG)
6, GV(N2,NO), HY(NO,N2), HU(NC,N2), DSTORE(NS,NS), PRIT(16,16)
C-----
C COMMON /PROG/ IOL, IO, IR, ISS, IM, ITF1, ITF2, ITF3, IFDPW, IE, IDSTAB, IDEB
1UG, ISET, IREG, IPSD, IYU, INORM
C-----
C REAL*4 FMT(20)
C-----
C----- OUTPUT OPTIONS-----
C--- IOL=1 IF THE OPEN LOOP EIGENSYSTEM IS DESIRED--OTHERWISE IOL=0
C--- IQ=1 IF THE RMS VALUES OF THE CONTROL AND STATE ARE TO BE FOUND
C--- IR=0 IF OPTIMAL FILTER AND REGULATOR EIGENSYSTEMS ARE TO BE FOUND
C--- IR=1 IF EXTERNAL C MATRIX IS SUPPLIED
C--- IR=2 IF EXTERNAL K IS SUPPLIED
C--- IR=3 IF EXTERNAL C AND K ARE SUPPLIED
C--- ISS=1 IF STEADY STATE VALUES ARE TO BE DETERMINED
C--- IM=1 IF MODAL STATES DESIRED
C-----
NSQ=NS*NS
MH=NS
H=N2
CALL CHECK (EPS, NC, NG, NO, IRET)
IF (IRET.EQ.1) RETURN
IF (ISET.EQ.1) GO TO 20
CALL RDMAT(BA, G, HO, D, GAM, FBGC, FBGE, AY, B, NS, NC, NO, NG, IRDMAT, IFDPW)
CALL READP (NS, ISAF, BA)
IF (IDSTAB.EQ.6) GO TO 10
WRITE (5,1800)
CALL RDRPAL (ANSR)
DSTAB=ANSR
DO 10 I=1, NS
DESTAB(I)=DSTAB
CONTINUE
GO TO 30
10
20
30 CALL SETUP (BA, G, GAM, HO, D, NS, NC, NG, NO)
CONTINUE
WRITE (6,1380)
DO 40 I=1, NS
WRITE (6,1390) (BA(I,J), J=1, NS)
40

```

```

C-----
END
C=====
SUBROUTINE RAPRNT (NMAX,M,N,L,A, IDIM, FMT)
REAL*8 A(NMAX,N)
DIMENSION FMT(IDIM)
NU=L
DO 20 NL=1,N,L
IF (NU.GT.N) NU=N
DO 10 I=1,M
WRITE (6,FMT) (A(I,J),J=NL,NU)
WRITE (6,30)
NU=NU+L
RETURN
FORMAT (1X)
END
C=====
SUBROUTINE RGAIN (M,NS,NC,NOB,WR,WI,VF,GN,W11,TCB,W21,LT,C,CI,CT,M
HS,MT)
IMPLICIT REAL*8 (A-H,C-Z)
DIMENSION WR(M),WI(M),VF(M,M),GN(NS,NS)
DIMENSION W11(NS,NS),TCB(M,M),W21(NS,NS),LT(NS),MT(NS)
DIMENSION C(NS),CI(NS),CT(NS,NS)
K=1
KP=1
KN=1
NRZEV=0
NCPZEV=0
10 IF (K.GT.M) GO TO 210
C CHECK FOR EIGVAL AT OR NEAR J-OMEGA AXIS TO INCLUDE IN E-L EIGSYS
C TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE
C-----
EIGVR=DABS(WR(K))
IF (EIGVR.GE.1.D-10) GO TO 60
IF (WI(K)) 40,20,40
NRZEV=NRZEV+1
IF (NRZEV.GT.1) GO TO 30
WR(K)=EIGVR
GO TO 80
WR(K)=-EIGVR
WRITE (6,290)
GO TO 150
NCPZEV=NCPZEV+1
IF (NCPZEV.GT.1) GO TO 50
WR(K)=EIGVR
WR(K+1)=EIGVR
GO TO 110

```

```

50  WR(K)=-EIGVR
    WR(K+1)=-EIGVR
    WRITE (6,300)
    GO TO 180
60  IF (WR(K)) 140,70,70
70  IF (WI(K)) 110,80,110
C-----EIGENVECTOR FOR REAL EIGENVALUE, POSITIVE-----
80  IF (NOB.EQ.0) GO TO 100
    DO 90 J=1,M
    TCB(J,KP)=VF(J,K)
    KP=KP+1
    K=K+1
    GO TO 10
C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, POSITIVE REAL PART-----
110 IF (NOB.EQ.0) GO TO 130
    DO 120 J=1,M
    FR=VF(J,K)
    FI=-VF(J,K+1)
    TCB(J,KP)=FR+FI
    TCB(J,KP+1)=FR-FI
    KP=KP+2
    K=K+2
    GO TO 10
120 IF (WI(K)) 180,150,180
C-----EIGENVECTOR FOR REAL EIGENVALUE, NEGATIVE REAL PART-----
130 C(KN)=WR(K)
    CI(KN)=WI(K)
    IF (NOB.NE.0) GO TO 170
    KNS=KN+NS
    DO 160 J=1,M
    TCB(J,KNS)=VF(J,K)
    KN=KN+1
    K=K+1
    GO TO 10
C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----
140 RR=WR(K)
    RI=WI(K)
    C(KN)=RR
    C(KN+1)=RR
    CI(KN)=RI
    CI(KN+1)=-RI
    IF (NOB.NE.0) GO TO 200
    KNS=KN+NS
    DO 190 J=1,M
    FR=VF(J,K)
    FI=-VF(J,K+1)
    TCB(J,KNS)=FR+FI
    TCB(J,KNS+1)=FR-FI
150
160
170
180
190

```

```

200  KN=KN+2
    K=K+2
    GO TO 10
210  CONTINUE
    IF (NOB.NE.0) GO TO 240
    C-----FORMATION OF W11-----
    DO 220 I=1,NS
    DO 220 J=1,NS
    W11(I,J)=TCB(I,J+NS)
    CT(I,J)=W11(I,J)
220  C-----FORMATION OF W21-----
    DO 230 I=1,NS
    DO 230 J=1,NS
    W21(I,J)=TCB(I+NS,J+NS)
    IF (NOB.EQ.0) GO TO 260
    DO 250 I=1,NS
    DO 250 J=1,NS
    W21(I,J)=-TCB(I,J)
    W11(I,J)=TCB(I+NS,J)
250  CONTINUE
260  C-----INVERT W11-----
    NSQ=NS*NS
    CALL MINV (NSQ,W11,NS,DETC,LT,MT)
    C-----CALCULATE THE REGAIN MATRIX-----
    DO 270 IL=1,NS
    DO 270 JL=1,NS
    GN(IL,JL)=0.D0
    DO 270 KL=1,NS
    GN(IL,JL)=GN(IL,JL)+W21(IL,KL)*W11(KL,JL)
    IF (NOB.EQ.0) RETURN
    DO 280 I=1,NS
    DO 280 J=1,NS
    CT(I,J)=W11(J,I)
    RETURN
280  C-----
290  FORMAT (1X,51H EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT,
114H OR NEAR ZERO./)
300  FORMAT (1X,49H EULER-LAGRANGE EQUATIONS HAVE A COMPLEX PAIR OF ,40
1HEIGENVALUES AT OR NEAR THE J-OMEGA AXIS.)
END
C=====
SUBROUTINE MINV (NSQ,A,N,D,L,M)
IMPLICIT REAL*8 (A-H,C-Z)
DIMENSION A(NSQ),I(N),M(N)
DOUBLE PRECISION A,D,EIGA,HOLD
NM=N*N
D=1.0D0
NK=-N

```

```

DO 180 K=1,N
NK=NK+N
L(K)=K
M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
DO 20 I=K,N
IJ=IZ+I
IF (DABS(BIGA)-DABS(A(IJ))) 10,20,20
10 BIGA=A(IJ)
L(K)=I
H(K)=J
20 CONTINUE
-----INTERCHANGE ROWS-----
J=L(K)
IF (J-K) 50,50,30
30 KI=K-N
DO 40 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
A(JI)=HOLD
40 CONTINUE
-----INTERCHANGE COLUMNS-----
I=M(K)
IF (I-K) 80,80,60
50 JP=N*(I-1)
DO 70 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
A(JI)=HOLD
70 CONTINUE
-----DIVIDE COLUMN BY MINUS PIVOT-----
C----- (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA) -----
80 IF (BIGA) 100,90,100
90 D=0.0D0
RETURN
100 DO 120 I=1,N
IF (I-K) 110,120,110
110 IK=NK+I
A(IK)=A(IK)/(-BIGA)
120 CONTINUE
-----REDUCE MATRIX-----
DO 150 I=1,N
IK=NK+I

```

```

130 HOLD=A(IK)
140 IJ=I-N
150 DO 150 J=1,N
160 IJ=IJ+N
170 IF (I-K) 130,150,130
180 IF (J-K) 140,150,140
190 KJ=IJ-I+K
200 A(IJ)=HOLD+A(KJ)+A(IJ)
210 CONTINUE
220 -----DIVIDE ROW BY PIVOT-----
230 KJ=K-N
240 DO 170 J=1,N
250 KJ=KJ+N
260 IF (J-K) 160,170,160
270 A(KJ)=A(KJ)/BIGA
280 CONTINUE
290 -----PRODUCT OF PIVOTS-----
300 D=D*BIG
310 -----REPLACE PIVOT BY RECIPROCAL-----
320 A(KK)=(1.0D0)/BIGA
330 CONTINUE
340 -----FINAL ROW AND COLUMN INTERCHANGE-----
350 K=N
360 K=(K-1) 260,260,200
370 IF (K)
380 I=L(K)
390 IF (I-K) 230,230,210
400 JO=N*(K-1)
410 JR=N*(I-1)
420 DO 220 J=1,N
430 JK=JO+J
440 HOLD=A(JK)
450 JI=JR+J
460 A(JK)=-A(JI)
470 A(JI)=HOLD
480 J=M(K)
490 IF (J-K) 190,190,240
500 KI=R-N
510 DO 250 I=1,N
520 KI=KI+N
530 HOLD=A(KI)
540 JI=KI-R+J
550 A(KI)=-A(JI)
560 A(JI)=HOLD
570 GO TO 190
580 K=0
590 RETURN
600 END

```

```

=====
C-----
SUBROUTINE SCOV (NL,WI,WLI,VL1,VL2,NR,WR,WRI,VR1,VR2,Q,X)
  REAL*8 VL1(NL),VL2(NL),WL(NL,NL),WLI(NL,NL),X(NL,NR),Q(NL,NR),
  1 VR1(NR),VR2(NR),WR(NR,NR),WRI(NR,NR)
  REAL*8 A,B,C,D,K1,K2,K3,K4
  DO 20 I=1,NL
    DO 20 J=1,NR
      X(I,J)=0.
      DO 20 II=1,NL
        X(II,J)=X(I,J)+WLI(II,J)*Q(II,J)
      DO 40 I=1,NL
        DO 40 J=1,NR
          Q(I,J)=0.
          DO 30 JJ=1,NR
            Q(I,J)=Q(I,J)+X(I,JJ)*WRI(J,JJ)
          CONTINUE
          IF (VL2(I)) 60,110,60
          J=1
          IF (VR2(J)) 80,90,80
          A=VL1(I)+VR1(J)
          B=-2.*VL2(I)*VR2(J)
          C=A**2+VL2(I)**2+VR2(J)**2
          D=C**2-B**2
          K1=A*C/D
          K2=-VR2(J)*C+VL2(I)*B/D
          K3=-VR2(J)*B+VL2(I)*C/D
          K4=-A*B/D
          I1=I+1
          J1=J+1
          X(I,J)=+K1*Q(I,J)+K2*Q(I,J1)+K3*Q(I1,J)+K4*Q(I1,J1)
          X(I,J1)=-K2*Q(I,J)+K1*Q(I,J1)-K4*Q(I1,J)+K3*Q(I1,J1)
          X(I1,J)=-K3*Q(I,J)-K4*Q(I,J1)+K1*Q(I1,J)+K2*Q(I1,J1)
          X(I1,J1)=+K4*Q(I,J)-K3*Q(I,J1)-K2*Q(I1,J)+K1*Q(I1,J1)
          J=J+2
        GO TO 100
      A=VR1(J)+VL1(I)
      B=A**2+VL2(I)**2
      K1=A/B
      K2=VL2(I)/B
      X(I,J)=K1*Q(I,J)-K2*Q(I+1,J)
      X(I+1,J)=K2*Q(I,J)+K1*Q(I+1,J)
      J=J+1
    IF (J.LE.NR) GO TO 70
    I=I+2
  GO TO 160
  J=1
  IF (VR2(J)) 130,140,130
=====

```



```

20      DO 30 J=1, N2
30      DO 30 I=1, NS
40      DO 30 K=1, NS
50      GNORM(I,J)=GNORM(I,J)+WNORM(I,K)*G(K,J)
60      GO TO (40,70,90,90,80), IPOINT
70      WRITE (6,170)
80      DO 60 I=1, NS
90      CONTINUE
100     WRITE (6,230) (GNORM(I,J), J=1, N2)
110     RETURN
120     WRITE (6,180)
130     GO TO 50
140     WRITE (6,240)
150     GO TO 50
160     DO 100 J=1, NS
170     DO 100 I=1, N1
180     DO 100 K=1, NS
190     GNORM(I,J)=GNORM(I,J)+G(I,K)*WNORM(K,J)
200     GO TO (110,110,110,120,130,140), IPOINT
210     WRITE (6,190)
220     GO TO 150
230     WRITE (6,200)
240     GO TO 150
250     WRITE (6,210)
260     GO TO 150
270     WRITE (6,220)
280     DO 160 I=1, N1
290     WRITE (6,230) (GNORM(I,J), J=1, NS)
300     RETURN
310     FORMAT (///,5X,45HMODAL CONTROL DISTRIBUTION MATRIX.....TI*G...//)
320     FORMAT (///,5X,50HMODAL PROCESS NOISE DISTRIBUTION MATRIX...TI*GA.
330     1.//)
340     FORMAT (///,5X,45HMODAL MEASUREMENT SCALING MATRIX...H(BAR)*T...//)
350     FORMAT (///,5X,45HTHE MODAL CONTROL GAINS.....C*T...//)
360     FORMAT (///,5X,45HCONTROL EIGENVECTOR MATRIX.....C*M...//)
370     FORMAT (///,5X,45HMEASUREMENT EIGENVECTOR MATRIX.....H(BAR)*M...//)
380     FORMAT (1X,(2X,1P6D14.6))
390     FORMAT (///,5X,45HMODAL FILTER STEADY STATE GAINS.....TI*K...//)
400     END
410     C=====
420     SUBROUTINE CNORM (WZ,WY,VEC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,H
430     10,CM,N1,N2)
440     C
450     C      WZ(I)      REAL PART OF I-TH EIGENVALUE
460     C
470     C      WY(I)      COMPLEX PART OF I-TH EIGENVALUE
480     C
490     C=====
500     =
510     =
520     =
530     =

```

```

C-----
VEC      MATRIX CP RIGHT EIGENVECTORS STORED IN REAL FORM
NS       FROM HCR2
          NO. OF STATES
IWRITE   FLAG TO CONTROL FORMATS FOR DIFFERENT EIGENSYSTEMS
WNORM     NORMALIZED MATRIX U OF RIGHT EIGENVECTORS STORED
          BY COLUMNS IN REAL FORM
WNORMI    U-INVERSE 2*CONJUGATE OF LEFT EIGENVECTORS
          STORED BY ROW IN REAL FORM
NSQ,DDD,D1,D2 - ARGUMENTS PASSED TO MINV
C-----
IMPLICIT REAL*8 (A-H,C-Z)
REAL*8 FIELD,CONNA,SENCOL,RIGHT,PMT
DIMENSION WZ(NS),WY(NS),VEC(NS,NS),WNORM(NS,NS),STOR
1E(6),D1(NS),D2(NS),PMT(14),HO(N1,N2),CH(N1,N2)
DATA FIELD/5H,12.5/,CONNA/5H,1.0/,SENCOL/5H,1.0/,RIGHT/1H)/,PMT/
16H(1X,1P,13*1H)/,SENCOL/4H,1.0/
C-----NORMALIZE COMPLEX EIGENVECTORS BY LARGEST ELEMENT-----
KK=0
LR=0
LC=0
DO 50 K=1,NS
IF (KK.EQ.1) GO TO 40
IF (DABS(WY(K)).LT.1.D-10) GO TO 50
LC=LC+1
EMAX=0.D0
DO 20 I=1,NS
CHOD=VEC(I,K)**2+VEC(I,K+1)**2
IF (CHOD-EMAX) 20,10,10
EMAX=CHOD
M=I
CONTINUE
VNR=VEC(M,K)
VMI=VEC(M,K+1)
DO 30 I=1,NS
VR=VEC(I,K)
VI=VEC(I,K+1)
VECRN=(VR*VNR+VI*VMI)/EMAX
VECRN=(-VR*VMI+VI*VNR)/EMAX
WNORM(I,K)=VECRN
WNORM(I,K+1)=VECRN
CONTINUE
KK=1
GO TO 50
40 KK=0
50 CONTINUE
C-----NORMALIZE REAL EIGENVECTORS BY THE TOTAL LENGTH-----

```

```

DO 80 K=1, NS
IF (DABS(WY(K)).GE.1.D-10) GO TO 80
LR=LR+1
REM0D=0. DO
DO 60 I=1, NS
REM0D=VEC(I,K)**2+RENC0D
REM0D=DSORT(REM0D)
DO 70 I=1, NS
RVEC=VEC(I,K)/REM0D
WNORM(I,K)=RVEC
CONTINUE
GO TO (90, 100, 110, 120, 130), IWRITE
90 WRITE (6, 20)
GO TO 140
100 WRITE (6, 330)
GO TO 140
110 WRITE (6, 340)
GO TO 140
120 WRITE (6, 350)
GO TO 140
130 WRITE (6, 360)
140 KK=0
NPRTW=0
NFMTH=1
DO 180 I=1, NS
IF (KK.EQ.1) GO TO 170
IF (DABS(WY(I)).GT.1.D-10) KK=1
C-----PRINT OUT NO MORE THAN 6 WORDS, NOT SEPARATING COMPLEX EIGVAL-----
IF (NPRTW.LT.5.OR.(NPRTW.EQ.5.AND.KK.EQ.0)) GO TO 150
FMT(NFMTH+1)=RIGHT
WRITE (6, FMT) (STORE(J), J=1, NPRTW)
NPRTW=0
NFMTH=1
NFMTH=NPRTW+1
NPRTW=NFMTH+1
IF (KK.EQ.1) GO TO 160
STORE(NPRTW)=WZ(I)
FMT(NFMTH)=FIELD
NFMTH=NFMTH+1
FMT(NFMTH)=SENCOL
GO TO 180
150 STORE(NPRTW)=WZ(I)
FMT(NFMTH)=FIELD
FMT(NFMTH+1)=COMMA
STORE(NPRTW+1)=WY(I)
FMT(NFMTH+2)=FIELD
FMT(NFMTH+3)=SENCOL
160

```

```

170 NPMTW=NPMTW+3
180 NPRTW=NPRTW+1
    GO TO 180
    KK=0
    CONTINUE
    FMT(NPMTW)=SEMEMD
    FMT(NPMTW+1)=RIGHT
    WRITE (6,FMT) (STORE(J),J=1,NPRTW)
    IF (IWRITE.NE.1) GO TO 190
    WRITE (6,370)
    GO TO 200
    WRITE (6,380)
    CALL RAPRNT (NS,NS,NS,6,WNORM,4,'(6(1X,1PD13.6))')
    GO TO 210
    CALL MODE (WNORM,H0,CH,NS,N1,N2,5)
    GO TO 230
    CALL MODE (WNORM,H0,CH,NS,N1,N2,6)
    GO TO 240
    WRITE (6,390)
    GO TO 250
    WRITE (6,400)
    GO TO 260
    WRITE (6,410)
    GO TO 270
    WRITE (6,420)
    GO TO 280
    WRITE (6,430)
    C-----SAVE U-INVERSE OPEN LOOP IN WNORMI-----
290 IF (IWRITE.GT.1) GO TO 310
    DO 300 I=1,NS
    DO 300 J=1,NS
        WNORMI(I,J)=WNORM(I,J)
    CALL MINV (NSQ,WNORMI,NS,DDD,D1,D2)
    CALL RAPRNT (NS,NS,NS,6,WNORMI,4,'(6(1X,1PD13.6))')
    RETURN
310 CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
    CALL RAPRNT (NS,NS,NS,6,WNORM,4,'(6(1X,1PD13.6))')
    RETURN
C-----
320 FORMAT (//5X,42HOPEN LOOP EIGENVALUES...DET(SI-F)G*G//)
330 FORMAT (//5X,46HC-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*G)...)
340 FORMAT (//5X,46HC-LOOP SUBOPT. E-VALUES...DET(SI-F+G*G)...)
350 FORMAT (//5X,46HC-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*G)...)
360 FORMAT (//5X,46HC-LOOP SUBOPT. E-VALUES...DET(SI-F+G*G)...)
370 FORMAT (//5X,46HC-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*G)...)
380 FORMAT (//5X,46HC-LOOP SUBOPT. E-VALUES...DET(SI-F+G*G)...)
390 FORMAT (//5X,46HC-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*G)...)
400 FORMAT (//5X,46HC-LOOP SUBOPT. E-VALUES...DET(SI-F+G*G)...)

```

```

410 FORMAT (//5X,46HC-LOOP SUBOPT-REG. LEFT E-VECTOR MATRIX...M-INV,/)
420 FORMAT (//5X,46HC-LOOP OPT. FILTER LEFT E-VECTOR MATRIX...M-INV,/)
430 FORMAT (//5X,51HC-LOOP SUBOPT. FILTER LEFT E-VECTOR MATRIX...M-INV.
1,6//)
END
C=====
1 SUBROUTINE TF (N,NM,NSQ,A,AA,M,B,BM,L,C,CM,IFDFW,D,BB,CC,CP,
EVR,EVI,PR,PI,SC,JCF,RES,D1,D2,DBD,EPS,ITF,ITFX)
1 IMPLICIT REAL*8(A-H,O-Z)
1 DIMENSION A(N,N),AA(N,N),B(N,M),BM(N,M),C(L,N),CM(L,N),D(L,M),BB(N
1),CC(N),CP(N),EVR(N),EVI(N),PR(N),PI(N),SC(N,N),JCF(N),RES(N),D1(N
2),D2(N)
C--SAVE COMPUTATION ON OL AND CL SYS WITH MODAL WORK DONE IN OPTSYS---
IF (ITFX.EQ.1) GO TO C50
IF (ITFX.EQ.2) GO TO C10
CALL POLES (N,NM,A,AA,M,B,L,C,PR,PI,D1,D2,JCF,SC)
C----- COMPUTE MODAL MATRICES FOR RESIDUES-----
10 DO 20 I=1,N
DO 20 J=1,N
20 AA(I,J)=SC(I,J)
DO 30 I=1,N
DO 30 J=1,N
CM(I,J)=0.D0
DO 30 K=1,N
CM(I,J)=CM(I,J)+C(I,K)*AA(K,J)
CALL MINV (NSQ,AA,N,DBD,D1,D2)
DO 40 I=1,N
DO 40 J=1,N
BM(I,J)=0.D0
DO 40 K=1,N
BM(I,J)=BM(I,J)+AA(I,K)*B(K,J)
40 CONTINUE
DO 60 I=1,N
DO 60 J=1,N
IF (ITF.NE.3) CALL ZEROS (I,J,IFDFW,N,NM,A,AA,M,B,L,C,D,BB,CC,CP
1,EVR,EVI,D1,D2,EPS,ITFX)
IF (ITF.NE.2) CALL RESID (I,J,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,1)
60 CONTINUE
RETURN
END
C=====
SUBROUTINE POLES (N,NM,A,AA,M,B,L,C,EVR,EVI,D1,D2,JCF,SC)
1 IMPLICIT REAL*8(A-H,O-Z)
1 DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),EVR(N),EVI(N),D1(N),D2(N),J
1CF(N),SC(N,N)
DO 10 I=1,N
DO 10 J=1,N
AA(I,J)=A(I,J)

```

```

CALL ONTHES (NM,N,LOW,IHIGH,AA,D2)
CALL ORTRAN (NM,N,LOW,IHIGH,AA,D2,SC)
CALL HOR2 (NM,N,LOW,IHIGH,AA,EVR,EVI,SC,IERR)
IF (IERR.NE.0) GO TO 30
CALL BALBAK (NM,N,LOW,IHIGH,D1,N,SC)
WRITE (6,40)
DO 20 I=1,N
C-----OPTGRM DATA-----
C
WRITE (4,2000) EVR(I),EVI(I)
C-----
20 WRITE (6,50) EVR(I),EVI(I)
C-----
30 RETURN
C-----
40 FORMAT (///,28H TF DENOMINATOR EIGENVALUES:,,/)
50 FORMAT (//,2X,3H (F13.6,4H)+J(F13.6,1H))
60 FORMAT (//,35H FAILURE IN HOR2, CALCULATING POLES)
C-----OPTGRM DATA-----
2000 FORMAT(5X,2D30.14)
C-----
C
C=====
SUBROUTINE ZEROS (K1,K2,IPDFW,N,NM,AA,M,B,L,C,D,BB,CC,CP,EVR,EVI,
1,IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),D(L,M),BB(N),CC(N),CP(N),EV
1R(N),EVI(N),D1(N),D2(N)
DOUBLE PRECISION SCI,LABS
C-----OPGRAPH DATA-----
C
IF (ITFX.EQ.1) IFL = 10
IF (ITFX.EQ.2) IFL = 1
IF (ITFX.EQ.3) IFL = 4
C-----
DO 10 I=1,N
BB(I)=B(I,K1)
CC(I)=C(K2,I)
DO 10 J=1,N
AA(I,J)=A(I,J)
WRITE (6,90) K1,K2
10 IF (IPDFW.EQ.6) GO TO 20
H=D(K2,K1)
IF (DABS(H).LE.EPS) GC TO 20
JJ=N
GO TO 50
NN=N-1
DO 30 I=1,NN

```

```

30 CALL CCOMP (N,NM,AA,CC,CP)
   IF (DABS(H).GT.EPS) GO TO 40
   CONTINUE
   H=SCL(N,BB,CC)
   WRITE (6,100) H
C-----OPGRAPH DATA FOR "0" ORDER-----
   KK=ITFX
   WRITE (IFL,300) KK,K2,K1
   ORDER = 0.0
   WRITE (IFL,301) ORDER,H
C-----
   GO TO 70
40 JJ=N-I
50 WRITE (6,110) JJ,H
C-----OPGRAPH DATA-----
   KK=ITFX
   WRITE (IFL,300) KK,K2,K1
   ORDER = FLOAT(JJ)
   WRITE (IFL,301) ORDER,H
C-----
   CALL ACOMP (N,NM,AA,BE,CC,H)
   CALL BALANC (NM,N,AA,LOW,HIGH,D1)
   CALL ORTHES (NM,N,LOW,HIGH,AA,D2)
   CALL HQR (NM,N,LOW,HIGH,AA,EVR,EVI,IERR)
   IF (IERR.NE.0) GO TO 80
   WRITE (6,120)
   DO 60 I=1,N
60   WRITE (6,130) EVR(I),EVI(I)
C-----OPGRAPH DATA-----
   WRITE (IFL,302) K2,K1
   DO 63 LL=1,N
63   WRITE (IFL,301) EVR(LL),EVI(LL)
C-----
70 RETURN
80 WRITE (5,140)
   RETURN
C-----
90 FORMAT (///,17H TF FOR INPUT NO.,I3,15H AND OUTPUT NO.,I3,1H: )
100 FORMAT (///,5X,27HNO FINITE ZEROS. TF GAIN = E12.4)
110 FORMAT (///,3X,20HORDER OF NUMERATOR = I3,9X,9HTF GAIN = E12.4)
120 FORMAT (///,3X,57HNUMERATOR EIGENVALUES (INCLUDING EXTRANEOUS ZERO V
   1ALUES):)
130 FORMAT (///,4X,1H(F13.6,4H)+J(F13.6,1H))
140 FORMAT (///,52H FAILURE IN HQR CALCULATING TRANSFER FUNCTION ZEROES)
C-----OPGRAPH DATA-----
300 FORMAT (5X,3I5)
301 FORMAT (5X,2D30.14)

```

AD-A154 552

DEVELOPMENT OF REDUCED ORDER MODELS FOR CONTROL SYSTEM  
DESIGN USING THE DPTSVSX PROGRAM(U) NAVAL POSTGRADUATE  
SCHOOL MONTEREY CA S W NELSON DEC 84

2/2

UNCLASSIFIED

F/G 9/2

NL

|  |  |  |  |  |  |  |       |  |  |  |  |  |  |
|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|
|  |  |  |  |  |  |  |       |  |  |  |  |  |  |
|  |  |  |  |  |  |  |       |  |  |  |  |  |  |
|  |  |  |  |  |  |  |       |  |  |  |  |  |  |
|  |  |  |  |  |  |  |       |  |  |  |  |  |  |
|  |  |  |  |  |  |  |       |  |  |  |  |  |  |
|  |  |  |  |  |  |  | END   |  |  |  |  |  |  |
|  |  |  |  |  |  |  | FINED |  |  |  |  |  |  |
|  |  |  |  |  |  |  | ETC   |  |  |  |  |  |  |



# 1.0



1.1



## 1.25



## 1.4



## 1.6



44

**5.0**

**15.6**

12.1

624

### 3.3.3



11

1



## 2.8



### 3.2



143



## ■ 2.



## 2.2



120



118

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```

302  FORMAT(5X,2I5)
C-----
C-----
END
C-----
SUBROUTINE ACOMP (N,NH,A,B,C,H)
REAL*8 A,B,C,H
DIMENSION A(NH,N),B(N),C(N)
DO 10 I=1,N
DO 10 J=1,N
10  A(I,J)=A(I,J)-B(I)*C(J)/H
RETURN
END
C-----
SUBROUTINE CCOMP (N,NH,A,C,CC)
REAL*8 A,C,CC
DIMENSION A(NH,N),C(N),CC(N)
DO 10 I=1,N
CC(I)=0.
DO 10 J=1,N
CC(I)=CC(I)+C(J)*A(J,I)
DO 20 I=1,N
20  C(I)=CC(I)
RETURN
END
C-----
FUNCTION SCL(N,B,C)
REAL*8 B,C,SCL
DIMENSION B(N),C(N)
SCL=0.
DO 10 I=1,N
SCL=SCL+C(I)*B(I)
RETURN
END
C-----
SUBROUTINE RESID (K1,K2,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,IPT)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION JCF(N),BM(N,M),CM(L,N),PR(N),PI(N),RES(N),BB(N),CC(N),PR
1T(4)
DATA SN/8H*SIN(B*T/R1/8H),R2/8HEXP(A*T)/ED/1H/,
DATA ZERO/0.D0/,T1/4H*T**/,BLANK/8H/,CS/8H*COS(B*T/
TEMPORARY MOD TILL JCF IS CALCULATED
C-----
DO 10 I=1,N
10  JCF(I)=0
C-----
TEMPORARY MOD
IF (IPT.EQ.1) WRITE (6,170)
DO 20 I=1,N
20  BB(I)=BM(I,K1)
CC(I)=CM(K2,I)

```

```

C-----LOOP THROUGH THE POLES-----
30 I=0
   I=I+1
   IF (JCF(I).EQ. 1) GO TO 160
   IF (DABS(PI(I)).LT. 1.D-10) GO TO 50
C-----COMPUTE SIMPLE POLE RESIDUES AND PRINT BOTH-----
   RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)
   RES(I+1)=CC(I)*BB(I+1)-CC(I+1)*BB(I)
   IF (IPT.EQ. 0) GO TO 40
   PRT(1)=BLANK
   PRT(2)=R2
   IF (PI(I).EQ. 0.D0) PRT(2)=BLANK
   PRT(3)=CS
   PRT(4)=ED
   WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
   I=I+1
   PRT(3)=SN
   WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
   GO TO 30
40 I=I+1
   GO TO 30
50 CONTINUE
C-----COMPUTE SIMPLE REAL POLE RESIDUE-----
   RES(I)=CC(I)*BB(I)
   IF (IPT.EQ. 0) GO TO 30
   PRT(1)=R1
   PRT(2)=R2
   PRT(3)=BLANK
   PRT(4)=BLANK
   WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
   GO TO 30
C-----LOOK AHEAD TO DETERMINE SIZE OF THE JORDAN BLOCK-----
60 K=1
   KT=N-I
   DO 70 J=I,KT
   IF (JCF(J).EQ. 0) GO TO 80
   K=K+1
   CONTINUE
70 CONTINUE
80 IF (DABS(PI(I)).LT. 1.D-10) GO TO 110
C-----COMPUTE REPEATED COMPLEX POLE AND PRINT OUT ALL FOUR-----
   K=1
   RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)+CC(I+2)*BB(I+2)+CC(I+3)*BB(I+3)
   RES(I+1)=CC(I)*BB(I+1)-CC(I+1)*BB(I+2)+CC(I+2)*BB(I+3)-CC(I+3)*BB(I+4)
   RES(I+2)=CC(I)*BB(I+2)-CC(I+2)*BB(I+3)+CC(I+3)*BB(I+4)-CC(I+4)*BB(I+5)
   IF (IPT.EQ. 0) GO TO 100
   PRT(1)=R1

```

12)

```

90      PRT (2)=R2
        IF (DABS(PR(I)) .GT. 1.D-10) GO TO 90
        PRT (1)=BLANK
        PRT (2)=BLANK
        PRT (3)=CS
        PRT (4)=ED
        WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
        PRT (3)=SN
        I=I+1
        WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
        PRT (1)=T1
        PRT (2)=R2
        IF (DABS(PR(I)) .LT. 1.D-10) PRT (2)=BLANK
        PRT (3)=CS
        I=I+1
        WRITE (6,190) PR(I),PI(I),RES(I),PRT(1),K, (PRT(J),J=2,4)
        PRT (3)=SN
        I=I+1
        WRITE (6,190) PR(I),PI(I),RES(I),PRT(1),K, (PRT(J),J=2,4)
        GO TO 30
        I=I+3
        GO TO 30
100     GO TO 30
C----- COMPUTE REPEATED REAL FOLE RESIDUE AND PRINT OUT ALL K OF THEM-----
110     CONTINUE
        KT=I+K-1
        NN=0
        DO 130 J=I,KT
            NN=NN+1
            RES(J)=ZERO
        DO 120 JJ=J,KT
            RES(J)=RES(J)+BB(JJ)*CC(JJ-NN+1)
        CONTINUE
        IF (IPT .EQ. 0) GO TO 150
        NN=0
        PRT (1)=T1
        PRT (2)=R2
        PRT (3)=BLANK
        PRT (4)=BLANK
        DO 140 J=I,KT
            WRITE (6,160) PR(J),PI(J),RES(J),PRT(1),NN, (PRT(JJ),JJ=2,4)
            NN=NN+1
        GO TO 30
        I=KT
        GO TO 30
160     CONTINUE
        RETURN
C-----
170     FORMAT (//,3X,22HRESIDUES AT THE POLES:/,T16,9HP O L E S,T41,15HR

```



```

100 DO 120 J=K,I
    DO 110 I=K,I
    IF (I.EQ. J) GO TO 110
    IF (A(I,J) .NE. 0.0D0) GO TO 120
110 CONTINUE
    N=K
    IEXC=2
    GO TO 10
120 CONTINUE
    -----NOW BALANCE THE SUBMATRIX IN ROWS K TO L-----
130 DO 130 I=K,L
    SCALE(I)=1.0D0
140 CONTINUE
    -----ITERATIVE LOOP FOR NORM REDUCTION-----
    NOCONV=.FALSE.
    DO 220 I=K,L
    C=0.0D0
    R=0.0D0
    DO 150 J=K,L
    IF (J.EQ. I) GO TO 150
    C=C+DABS(A(I,J))
    R=R+DABS(A(I,J))
150 CONTINUE
    -----GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW-----
    IF (C.EQ. 0.0D0 .OR. R.EQ. 0.0D0) GO TO 220
    G=R/RADIX
    F=1.0D0
    S=C+R
    IF (C.GE. G) GO TO 170
    F=F*RADIX
    C=C*B2
    GO TO 160
170 G=R*RADIX
180 IF (C.LT. G) GO TO 190
    F=F/RADIX
    C=C/B2
    GO TO 180
    -----NOW BALANCE-----
190 IF ((C + R) / F .GE. 0.95D0 * S) GO TO 220
    G=1.0D0/F
    SCALE(I)=SCALE(I)*F
    NOCONV=.TRUE.
    DO 200 J=K,N
    A(I,J)=A(I,J)*G
    DO 210 J=1,I
    A(J,I)=A(J,I)*F
210 CONTINUE
220 IF (NOCONV) GO TO 140
230 LOW=K

```

```

IGH=L
RETURN
END
C=====
SUBROUTINE ORTHES (NM,N,LOW,IGH,A,ORT)
INTEGER I,J,M,N,II,JJ,LA,MP,NM,IGH,KP1,LOW
REAL*8 A(NM,N),ORT(IGH)
REAL*8 F,G,H,SCALE
REAL*8 DSQRT,DABS,DSIGN
LA=IGH-1
KP1=LOW+1
IF (LA.LT.KP1) GO TO 100
DO 90 M=KP1,LA
H=0.0D0
ORT(M)=0.0D0
SCALE=0.0D0
C-----SCALE COLUMN (ALGOL TOL THEN NOT NEEDED)-----
DO 10 I=M,IGH
SCALE=SCALE+DABS(A(I,M-1))
IF (SCALE.EQ.0.0D0) GO TO 90
MP=M+IGH
DO 20 II=M,IGH
I=MP-II
ORT(I)=A(I,M-1)/SCALE
H=H+ORT(I)*ORT(I)
CONTINUE
G=-DSIGN(DSQRT(H),ORT(M))
H=H-ORT(M)*G
ORT(M)=ORT(M)-G
C-----FCRM (I-(U*UT)/H) * A-----
DO 50 J=M,N
F=0.0D0
DO 30 II=M,IGH
I=MP-II
F=F+ORT(I)*A(I,J)
CONTINUE
F=F/H
DO 40 I=M,IGH
A(I,J)=A(I,J)-F*ORT(I)
CONTINUE
C-----FORM (I-(U*UT)/H)*A*(I-(U*UT)/H)-----
DO 80 I=1,IGH
F=0.0D0
DO 60 JJ=M,IGH
J=MP-JJ
F=F+ORT(J)*A(I,J)
CONTINUE
F=F/H

```

```

70 DO 70 J=M, IGH
80 A(I,J)=A(I,J)-P*ORT(J)
CONTINUE
ORT(M)=SCALE*ORT(M)
90 A(M,M-1)=SCALE*G
100 CONTINUE
RETURN
END
C=====
SUBROUTINE ORTRAN (NM,N,LOW,IGH,A,ORT,Z)
INTEGER I,J,N,KL,MH,MM,IGH,LOW,MP1
REAL*8 A(NM,IGH),ORT(IGH),Z(NM,N)
REAL*8 G
C-----INITIALIZE Z TO IDENTITY MATRIX-----
DO 20 I=1,N
DO 10 J=1,N
Z(I,J)=0.0D0
Z(I,I)=1.0D0
CONTINUE
KL=IGH-LOW-1
IF (KL.LT.1) GO TO 80
DO 70 MM=1,KL
MP=IGH-MM
MP1=MP+1
IF (A(MP,MP-1).EQ.0.0D0) GO TO 70
DO 30 I=MP1,IGH
ORT(I)=A(I,MP-1)
DO 60 J=MP,IGH
G=0.0D0
DO 40 I=MP,IGH
G=G+ORT(I)*Z(I,J)
G=(G / ORT(MP))/A(MP,MP-1)
DO 50 I=MP,IGH
Z(I,J)=Z(I,J)+G*ORT(I)
CONTINUE
70 CONTINUE
80 RETURN
END
C=====
SUBROUTINE HQR2 (NM,N,LOW,IGH,H,WR,WI,Z,IERR)
INTEGER I,J,K,L,M,N,EN,II,JJ,LL,MM,NA,NN,IGH,ITS,LOW,MP2,ENM2,I
1ERR
REAL*8 H(NM,N),WR(N),WI(N),Z(NM,N)
REAL*8 P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NORM,MACHEP
REAL*8 DSORT,DABS,DSIGN
INTEGER M1NO
C=====

```

```

LOGICAL NOTIAS
COMPLEX *16Z3
COMPLEX *16DCNPLX
REAL*8 DREAL,DIMAG
-----FUNCTIONS ENABLE EXTRACTION OF REAL AND IMAGINARY-----
C-----PARTS OF DOUBLE PRECISION COMPLEX NUMBERS-----
C-----
DREAL(Z3)=Z3
DIMAG(Z3)=(0.0D0,-1.0D0)*Z3
DATA MACHEP/Z3416000000000000/
IERR=0
NORM=0.0D0
K=1
-----STORE ROOTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM-----
C-----
DO 20 I=1,N
DO 10 J=K,N
NORM=NORM+DABS(H(I,J))
K=I
IF (I .GE. LOW .AND. I .LE. IGH) GO TO 20
WR(I)=H(I,I)
WI(I)=0.0D0
CONTINUE
EN=IGH
T=0.0D0
C-----SEARCH FOR NEXT EIGENVALUES-----
30 IF (EN .LT. LOW) GO TC 290
ITS=0
NA=EN-1
ENM2=NA-1
C-----LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-----
40 DO 50 LL=LOW,EN
L=EN+LOW-LL
IF (L .EQ. LOW) GO TO 60
S=DABS(H(L-1,L-1))+DABS(H(L,L))
IF (S .EQ. 0.0D0) S=NCRN
IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
CONTINUE
C-----FORM SHIFT-----
60 X=H(EN,EN)
IF (L .EQ. EN) GO TO 220
Y=H(NA,NA)
W=H(EN,NA)*H(NA,EN)
IF (L .EQ. NA) GO TO 230
IF (ITS .EQ. 30) GO TC 500
IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 80
C-----FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LOW,EN
H(I,I)=H(I,I)-X
70

```

```

S=DABS(H(EN,NA))+DABS(H(NA,ENN2))
X=0.75D0*S
Y=X
W=-0.4375D0*S*S
ITS=ITS+1
80 C-----LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.-----
DO 90 MM=L,ENN2
M=ENN2+L-MM
ZZ=H(M,M)
R=X-ZZ
S=Y-ZZ
P=(R*S - W)/H(M+1,M)+H(M,M+1)
Q=H(M+1,M+1)-ZZ-R-S
R=H(M+2,M+1)
S=DABS(P)+DABS(Q)+DABS(R)
P=P/S
Q=Q/S
R=R/S
IF (M.EQ. L) GO TO 100
IF (DABS(H(M,M-1))*DABS(Q)+DABS(R)).LE. NACHEP * DABS(P)
1 * DABS(H(M-1,M-1))+DABS(ZZ)+DABS(H(M+1,M+1))) GO TO 100
CONTINUE
MP2=M+2
DO 110 I=MP2,EN
H(I,I-2)=0.0D0
IF (I.EQ. MP2) GO TO 110
H(I,I-3)=0.0D0
CONTINUE
110 C-----DOUBLE OR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
DO 210 K=M,NA
NOTLAS=K.NE.NA
IF (K.EQ. M) GO TO 120
P=H(K,K-1)
Q=H(K+1,K-1)
R=0.0D0
IF (NOTLAS) R=H(K+2,K-1)
X=DABS(P)+DABS(Q)+DABS(R)
IF (X.EQ. 0.0D0) GO TO 210
P=P/X
Q=Q/X
R=R/X
S=DSIGN(DSQRT(P*P+Q*Q+R*R),P)
IF (K.EQ. M) GO TO 130
H(K,K-1)=-S*X
GO TO 140
130 IF (L.NE. M) H(K,K-1)=-H(K,K-1)
140 P=P/S
X=P/S

```

```

Y=Q/S
ZZ=R/S
Q=Q/P
R=R/P
C-----ROW MODIFICATION-----
DO 160 J=K,N
P=H(K,J)+Q*H(K+1,J)
IF (.NOT. NOTLAS) GO TO 150
P=P+R*H(K+2,J)
H(K+2,J)=H(K+2,J)-P*ZZ
H(K+1,J)=H(K+1,J)-P*Y
H(K,J)=H(K,J)-P*X
CONTINUE
J=MINO(EN,K+3)
C-----COLUMN MODIFICATION-----
DO 180 I=1,J
P=X*H(I,K)+Y*H(I,K+1)
IF (.NOT. NOTLAS) GO TO 170
P=P+ZZ*H(I,K+2)
H(I,K+2)=H(I,K+2)-P*R
H(I,K+1)=H(I,K+1)-P*Q
H(I,K)=H(I,K)-P
CONTINUE
C-----ACCUMULATE TRANSFORMATIONS-----
DO 200 I=LOW,IGH
P=X*Z(I,K)+Y*Z(I,K+1)
IF (.NOT. NOTLAS) GO TO 190
P=P+ZZ*Z(I,K+2)
Z(I,K+2)=Z(I,K+2)-P*R
Z(I,K+1)=Z(I,K+1)-P*Q
Z(I,K)=Z(I,K)-P
CONTINUE
GO TO 40
C-----ONE ROOT FOUND-----
220 H(EN,EN)=X+T
WR(EN)=H(EN,EN)
WI(EN)=0.0D0
EN=NA
GO TO 30
C-----TWO ROOTS FOUND-----
230 P=(Y-X)/2.0D0
Q=P*P+W
ZZ=DSORT(DAES(Q))
H(EN,EN)=X+T
X=H(EN,EN)
H(NA,NA)=Y+T
IF (Q.LT. 0.0D0) GO TO 270

```

```

C-----REAL PAIR-----
ZZ=P+DSIGN(ZZ,P)
WR(NA)=X+ZZ
WR(EN)=WR(NA)
IF (ZZ.NE.0.0D0) WR(EN)=X-W/ZZ
WI(NA)=0.0D0
WI(EN)=0.0D0
X=H(EN,NA)
S=DABS(X)+DABS(ZZ)
P=X/S
Q=ZZ/S
R=DSQRT(P*P+Q*Q)
P=P/R
Q=Q/R
C-----ROW MODIFICATION-----
DO 240 J=NA,N
ZZ=H(NA,J)
H(NA,J)=Q*ZZ+P*H(EN,J)
H(EN,J)=Q*H(EN,J)-P*ZZ
240 CONTINUE
C-----COLUMN MODIFICATION-----
DO 250 I=1,EN
ZZ=H(I,NA)
H(I,NA)=Q*ZZ+P*H(I,EN)
H(I,EN)=Q*H(I,EN)-P*ZZ
250 CONTINUE
C-----ACCUMULATE TRANSFORMATIONS-----
DO 260 I=LOW,IGH
ZZ=Z(I,NA)
Z(I,NA)=Q*ZZ+P*Z(I,EN)
Z(I,EN)=Q*Z(I,EN)-P*ZZ
260 CONTINUE
GO TO 280
C-----COMPLEX PAIR-----
270 WR(NA)=X+P
WR(EN)=X+P
WI(NA)=ZZ
WI(EN)=ZZ
EN=EN+2
GO TO 30
C-----ALL ROOTS FOUND. BACKSUBSTITUTE TO FIND-----
290 IF (NORM.EQ.0.0D0) GO TO 510
DO 450 NN=1,N
EN=NN+1-NN
P=WR(EN)
Q=WI(EN)
NA=EN-1

```

```

C-----IF (Q) 370,300,450-----REAL VECTOR-----
300 M=EN
H(EN,EN)=1.0D0
IF (NA.EQ.0) GO TO 450
DO 360 II=1,NA
I=EN-II
W=H(I,-P)
R=H(I,N)
IF (I.GT.NA) GO TO 320
DO 310 J=M,NA
R=H(J,EN)
IF (I.GE.0.0D0) GO TO 330
GO TO 360
310
320
330
340
350
360
C-----SOLVE REAL EQUATIONS-----
X=H(I,I+1)
Y=H(I+1,I)
Q=(WR(I)-P)*WR(I)+WI(I)*WI(I)
I={X*S-ZZ*R}/Q
H(I,EN)=T
IF (DABS(X).LE.DABS(ZZ)) GO TO 350
H(I+1,EN)=(-R-W*T)/X
GO TO 360
H(I+1,EN)=(-S-Y*T)/ZZ
CONTINUE
END REAL VECTOR
C-----COMPLEX VECTOR-----
370 M=NA
C-----LAST VECTOR COMPONENT CHOSEN IMAGINARY SO THAT-----
C-----EIGENVECTOR MATRIX IS TRIANGULAR-----
IF (DABS(H(EN,NA)).EQ.H(EN,NA))
H(EN,EN)=-H(EN,EN)-P/H(EN,NA)
GO TO 390
380 Z3=DCMPLX(0.0D0,-H(NA,EN)/DCMPLX(H(NA,NA)-P,Q)
H(NA,NA)=DREAL(Z3)
H(EN,EN)=DIMAG(Z3)
H(EN,NA)=0.0D0
H(EN,EN)=1.0D0
390

```

```

40  WRITE (5,130)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 50
50  GO TO 60
    WRITE (5,140)
    GO TO 40
60  CONTINUE
    IF (IANS.EQ.IZ) GO TO 90
    WRITE (5,150)
    CALL RDINT (IANS)
    K=IANS
    WRITE (5,160)
    CALL RDINT (IANS)
    L=IANS
    WRITE (5,100) K,L
    CALL RDREAL (ANSR)
    DUM=ANSR
    DO 80 I=1,NC
    DO 70 J=1,NC
    IF ((I.EQ.K).AND. (J.EQ.L)) D(I,J)=DUM
70  CONTINUE
80  CONTINUE
    GO TO 30
90  CONTINUE
    CALL FRTCMS ('CLRSCRN ')
    RETURN
C-----
100  FORMAT (5X,14HTHE ELEMENT D(I2,1H,I2,2H)=)
110  FORMAT (//,5X,54HENTER THE MEASUREMENT FEEDTHROUGH MATRIX / FEEDFOR
    1WARD,//,5X,34H DISTRIBUTION MATRIX {"D"-MATRIX}.,//,8X,49HDIMENSION
    2 = # OBSERVATIONS {NO} X # CONTROLS {NC})
120  FORMAT (//,5X,50HTHE FEEDFORWARD DISTRIBUTION MATRIX {"D"-MATRIX}.
    1.,//)
130  FORMAT (//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
    1ENT? //,10X,19HTYPE "YES" OR "NO".)
140  FORMAT (//,1X,51HWARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
150  FORMAT (//,5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160  FORMAT (//,5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
    1.)
    END
C=====
SUBROUTINE READG (NS,NC,ISAG,G)
C INTERACTIVELY INPUTS THE "G" MATRIX {CONTROL DISTRIBUTION MATRIX}=
C=====
    REAL*8 G(NS,NC),DUM,ANSR
    INTEGER IANS,I,J,K,L,ISAG
    DATA IY,'Y',IZ,'N',/
    IF (ISAG.EQ.1) GO TO 40

```



```

CALL FRTCHS ('CLRSCRN ')
RETURN
-----
120 FORMAT (5X,14H THE ELEMENT F(I2,1H,I2,2H)=)
130 FORMAT (/5X,36H ENTER THE SYSTEM MATRIX { "F"--MATRIX} ,//,10X,41HDIM
130 1ENSION = # STATES {NS} X # STATES {NS} }
140 FORMAT (//15X,33H THE SYSTEM MATRIX { "F"--MATRIX} ,//)
150 FORMAT (//5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
150 1ENT? //,10X,19H TYPE "YES" OR "NO".)
160 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
170 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
180 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
180 1.)
-----
C=====
C SUBROUTINE READH (NO,NS,ISAH,HO)
C INTERACTIVELY INPUTS THE "H" MATRIX {MEASUREMENT SCALING MATRIX}.=
C=====
REAL*8 HO(NO,NS) DIM ANSR
INTEGER IANS,I,J,K,L,ISAH
DATA IY,'Y',IZ,'N'
-----
C
IF (ISAH.EQ.1) GO TO 40
WRITE (5,120)
DO 20 I=1,NO
DO 10 J=1,NS
WRITE (5,110) I,J
CALL RDRREAL (ANSR)
HO(I,J)=ANSR
CONTINUE
CONTINUE
-----
C
CALL FRTCHS ('CLRSCRN ')
CONTINUE
WRITE (5,130)
CALL MATPRT (HO,NO,NS)
WRITE (5,140)
CALL RDCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
GO TO 70
WRITE (5,150)
GO TO 50
CONTINUE
IF (IANS.EQ.IZ) GO TO 100
WRITE (5,160)
CALL RDINT (IANS)
K=IANS
WRITE (5,170)

```



```

260 RE=WR (II)
    AI=AI (II)
    ZD=DCMPLX (RE**2 + AI**2 -OM**2, -2.D0*RE*OM)
    ZN=DCMPLX (RES (II+1)*AI-RES (II)*RE, RES (II)*OM)
    ZZ=ZZ+ZN/ZD
270 CONTINUE
    IF (IYU .EQ. 2 .OR. I .NE. L) GO TO 280
    PSD (K) =PSD (K) +DN1
280 PSD (K) =PSD (K) +DN1* (ZZ*DCONJG (ZZ))
290 CONTINUE
300 CONTINUE
    IF (IYU .EQ. 1) WRITE (6, 390) L
    IF (IYU .EQ. 2) WRITE (6, 400) L
310 WRITE (6, 410) (W (I), PSD (I), I=1, 30)
    CONTINUE
320 RETURN
    CALL EXEXIT (N2, PA, IERR)
    RETURN
C-----
330 FORMAT (//, 41H SUBSEQUENT PSD IS NORMALIZED BY MEAS NO., I3, //)
340 FORMAT (//, 50H SUBSEQUENT PSD IS NORMALIZED BY PROCESS NO., J3, //)
350 FORMAT (//, 38H TRANSFER FUNCTION FROM PROCESS NOISE, I2, 3H TO, 13H MEASUREMENT, I2, //)
360 FORMAT (//, 38H TRANSFER FUNCTION FROM PROCESS NOISE, I2, 3H TO, 9H CONTROL, I2, //)
370 FORMAT (//, 36H TRANSFER FUNCTION FROM MEASUREMENT, I2, 16H TO MEASUREMENT, I2, //)
380 FORMAT (//, 36H TRANSFER FUNCTION FROM MEASUREMENT, I2, 12H TO CONTROL, I2, //)
390 FORMAT (//, 14H PSD OF OUTPUT, I3, 32H FORCED BY ALL NOISE-(RAD FREQ., 15HNORMALIZED PSD) //)
400 FORMAT (//, 15H PSD OF CONTROL, I3, 32H FORCED BY ALL NOISE-(RAD FREQ., 15HNORMALIZED PSD) //)
410 FORMAT (4 (1X, 1H (E11.4, 1H, E11.4, 1H)))
    END
C=====
SUBROUTINE EXEXIT (N, A, IERR)
EXEXIT RETURNS THE NUMBER OF THE EIGENVALUE WHERE HQR2 =
FAILS, THEN STOPS THE PROGRAM. =
C=====
INTEGER IERR
DOUBLE PRECISION A
DIMENSION A (N, N)
WRITE (5, 10) IERR
CALL RAPRNT (N, N, N, 9, A, 4, ' (9 (1X, 1PD13.6)) ')
RETURN

```

```

180 DN1=DNORM*(I,I)
    IF (IYU.EQ.1) .AND. IPT.EQ.1 WRITE (6,350) I,L
    IF (IYU.EQ.2) .AND. IPT.EQ.1 WRITE (6,360) I,L
    IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NG,GW,NL,HY,WR,WI,
1RES,BB,CC,IPT)
    IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NG,GW,NL,HU,WR,WI,
1RES,BB,CC,IPT)
    DO 210 K=1,20
    ZZ=DCNPLX(0.00,0.00)
    OM=H(K)
    DO 200 II=1,N2
    IF (WI(II)) 200,180,150
    ZD=DCNPLX(-WR(II),OM-WI(II))
    ZZ=RES(II)/ZD+ZZ
    GO TO 200
    RE=WR(II)
    AI=WI(II)
    ZD=DCNPLX(RE**2 + AI**2 - OM**2,-2.00*RE*OM)
    ZN=DCNPLX(RES(II+1)*AI-RES(II)*RE,RES(II)*OM)
    ZZ=ZZ+ZN/ZD
    CONTINUE
200 PSD(K)=PSD(K)+DN1*(ZZ*DCONJG(ZZ))
210 CONTINUE
C-----G SUBV-----

DO 240 I=1,N2
DO 240 J=1,NO
ST=0.00
DO 230 K=1,NS
ST=ST+X(I,K)*PBGE(K,J)+X(I,NS+K)*PBGE(K,J)
230 GV(I,J)=ST
240 CALL RAPRNT (N2,N2,NO,9,GV,4,'(9(1X,1PD13.6))')
C-----DEBUG ABOVE, LOOP THRU MEAS NOISE-----

DO 300 I=1,NO
DN1=DNORM*(I,I)
IF (IYU.EQ.1) .AND. IPT.EQ.1 WRITE (6,370) I,L
IF (IYU.EQ.2) .AND. IPT.EQ.1 WRITE (6,380) I,L
IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NO,GW,NL,HY,WR,WI,RES,
1BB,CC,IPT)
IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NO,GW,NL,HU,WR,WI,RES,
1BB,CC,IPT)
DO 290 K=1,30
ZZ=DCNPLX(0.00,0.00)
OM=H(K)
DO 270 II=1,N2
IF (WI(II)) 270,250,260
ZD=DCNPLX(-WR(II),OM-WI(II))
ZZ=ZZ+RES(II)/ZD
GO TO 270
250

```

```

EMAX=0.D0
DO 120 I=1,N2
  EMOD=DABS(WR(I)**2 +W1(I)**2)
  IF (EMOD .GT. EMAX) EMAX=EMOD
CONTINUE
EMOD=DSQRT(EMAX)
EMOD=2*EMOD
C-----ROUND UP TO NEAREST 2,4,5,8,10-----
ELOG=DLOG10(EMOD)
IF (ELOG .LT. 0.D0) IPOW=-IDINT(DABS(ELOG) + 1)
IF (ELOG .GE. 0.D0) IPOW=IDINT(ELOG)
EMAX=EMOD*10**(-IPOW)
IF (EMAX .GT. 2.D0) EMOD=2.D0
IF (EMAX .GT. 4.D0) EMOD=4.D0
IF (EMAX .GT. 5.D0) EMOD=5.D0
IF (EMAX .GT. 8.D0) EMOD=8.D0
IF (EMAX .GE. 10.D0) EMOD=10.D0
EMAX=EMOD*10**IPOW
DW=EMAX/20.D0
C-----ADD 10 POINTS 3 DECADES UP-----
IF (EMOD .LT. 5.0) GO TO 130
EMAX=1.0D1
IK=3
GO TO 140
130 EMAX=5.D0
IK=2
140 CONTINUE
C-----STORE 30 FREQUENCIES-----
DO 150 I=1,20
  W(I)=DW*(I-1)
DO 160 I=1,3
  IP=20+3*(I-1)
DO 160 J=1,3
  IX=MOD(IK+J-1,3)+1
  JJ=0
  IF (IK .EQ. 2 .AND. J .GE. 2) JJ=1
  W(IP+J)=DW1(IX)*10**(IPOW+I-1+JJ+IK-2)
CONTINUE
IX=MOD(IK,3)+1
W(30)=DW1(IX)*10**(IPCW+3+IK-2)
C-----LARGE LOOP THRU OUTPUTS-----
IF (IYU .EQ. 1) NL=NO
IF (IYU .EQ. 2) NL=NC
DO 170 I=1,NL
  DO 170 I=1,30
  PSD(I)=0.D0
C-----LOCP THRU PROCESS NOISE-----
DO 220 I=1,NG

```

```

ST=0. DO
DO 20 K=1, NO
ST=ST+FBGE(I,K)*H(K,J)
PA(I,NS+J)=-ST
PA(NS+I,NS+J)=F(I,J)-ST
CALL RAPRNT (N2,N2,N2,9,PA,4,'(9(1X,1PD13.6))')
C-----
CALL BALANC (N2,N2,FA,LOW,IGH,D1)
CALL ORTHES (N2,N2,LOW,IGH,FA,D2)
CALL ORTRAN (N2,N2,LOW,IGH,FA,D2,X)
CALL HOR2 (N2,N2,LOW,IGH,FA,WR,WI,X,IERR)
IF (IERR.NE.0) GO TO 320
CALL BALBAK (N2,N2,LOW,IGH,D1,N2,X)
CALL RAPRNT (N2,N2,N2,9,PA,4,'(9(1X,1PD13.6))')
C-----
DETERMINE MODAL MATRICES
IF (IYU.EQ.1) GO TO 60
C-----
DO 50 I=1,NC
DO 50 J=1,N2
ST=0. DO
DO 40 K=1,NS
ST=ST-C(I,K)*X(K,J)
HU(I,J)=ST
GO TO 90
C-----
DO 80 I=1,NO
DO 80 J=1,N2
ST=0. DO
DO 70 K=1,NS
ST=ST+H(I,K)*X(K,J)-H(I,K)*X(NS+K,J)
HY(I,J)=ST
CALL RAPRNT (NO,NO,N2,9,HY,4,'(9(1X,1PD13.6))')
C-----
CALL MINV (NS,X,N2,ST,D1,D2)
CALL RAPRNT (N2,N2,N2,9,X,4,'(9(1X,1PD13.6))')
C-----
DO 110 I=1,N2
DO 110 J=1,NG
ST=0. DO
DO 100 K=1,NS
ST=ST-X(I,K)*GAM(K,J)
GW(I,J)=ST
CALL RAPRNT (N2,N2,NG,9,GW,4,'(9(1X,1PD13.6))')
C-----
DETERMINE BANDWIDTH OF CONTROLLED SYSTEM
IF (INORM.LE. NG) DNCRH=1. DO/O (INORM,INORM)
IF (INORM.GT. NG) DNCRH=1. DO/R (INORM-NG,INORM-NG)
C-----

```

```

C-----SET ERROR -- NO CONVERGENCE TO AN-----
C-----EIGENVALUE AFTER 30 ITERATIONS-----
240 IERR=EN
250 RETURN
END
C=====
1 SUBROUTINE PSDCAL (N2,NS,FA,X,NC,GV,C,NO,HY,HU,H
2 IPSD,INORM)
=====
= PSDCAL COMPUTES THE PSD OF OUTPUTS OR CONTROLS OF
= A CONTROLLED SYSTEM
=====
IYU= 1 OUTPUT PSD
= 2 CONTROL PSD
= 3 BOTH OUTPUT AND CONTROL PSD
IPSD=1 PSD
=2 PSD AND TF RESIDUES
INORM= 1,2,... NG NORMALIZED BY ITH PROCESS NOISE
NG+1,... NG+NO NORMALIZED BY ITH MEAS NOISE
=====
DOUBLE PRECISION FA,X,GV,GV,C,HY,H,PBGE,GAM,ACL,F,WR,WI,D1,D2,RES,
1 BB,CC,Q,R,PSD,W,DNORM,DN1,EMAX,ELOG,EMOD,DW,ST,OM,RE,AI,HU,DW1
COMPLEX*16 Z1,Z2
DIMENSION FA(N2,N2),X(N2,N2),GV(N2,NG),C(NC,NS),HY(NO,N2),H(NO,NS)
1 PBGE(NS,NO),GAM(NS,NS),NC,ACL(NS,NS),F(NS,NS),WI(N2),D1(N2),D2
2 (N2),RES(N2),Q(NG,NG),Z1(4),Z2(4)
3 NO),HU(NC,N2),DN1(4)
INTEGER JCF(N2)
DATA DW1/1.D0,2.D0,5.D0,10.D0/
IF (IYU.EQ. 0) IYU=1
IF (INORM.EQ. 0) INORM=1
IPT=0
IF (IPSD.GT. 1) IPT=1
IX=INORM-NG
IF (IX.GT. 0) WRITE (6,330) IX
IF (IX.LE. 0) WRITE (6,340) INORM
NSQ=N2*N2
C-----COMPUTE EIGENSYSTEM OF CONTROLLED SYSTEM; FORM FA-----
DO 10 I=1,NS
DO 10 J=1,NS
FA(I,J)=ACL(I,J)
FA(NS+1,J)=0.D0
DO 30 I=1,NS
DO 30 J=1,NS

```

```

C-----ROW MODIFICATION-----
R=R/P
DO 160 J=K,EN
P=H(K,J)+Q*H(K+1,J) GO TO 150
IF (.NOT. NOTLAS) GO TO 150
P=P+R*H(K+2,J)
H(K+2,J)=H(K+1,J)-P*ZZ
H(K+1,J)=H(K,J)-P*Y
H(K,J)=H(K,J)-P*X
CONTINUE
J=MINO(EN,K+3)
C-----COLUMN MODIFICATION-----
DO 180 I=L,J
P=X*H(I,K)+Y*H(I,K+1)
IF (.NOT. NOTLAS) GO TO 170
P=P+ZZ*H(I,K+2)
H(I,K+2)=H(I,K+1)-P*R
H(I,K+1)=H(I,K)-P*Q
H(I,K)=H(I,K)-P
CONTINUE
GO TO 40
C-----ONE ROOT FOUND-----
200 WR(EN)=X+T
WI(EN)=0.0D0
EN=NA
GO TO 30
C-----THO ROOTS FOUND-----
210 P=(Y-X)/2.0D0
Q=P*P+W
ZZ=DSQRT(DABS(Q))
X=X+T
IF (Q.LT. 0.0D0) GO TO 220
C-----REAL PAIR-----
ZZ=P+DSIGN(ZZ,P)
WR(NA)=X+ZZ
WR(EN)=WR(NA)
IF (ZZ.NE. 0.0D0) WR(EN)=X-W/ZZ
WI(NA)=0.0D0
WI(EN)=0.0D0
GO TO 230
C-----COMPLEX PAIR-----
220 WR(NA)=X+P
WR(EN)=X+P
WI(NA)=ZZ
WI(EN)=-ZZ
EN=ENM2
GO TO 30
230

```

```

80      W=-0.4375D0*S*S
      ITS=ITS+1
      C-----LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.-----
      DO 90 MM=L, ENM2
      M=ENM2+L-MM
      ZZ=H(M,M)
      R=X-ZZ
      S=Y-ZZ
      P=(R * S - W)/H(M+1,M)+H(M,M+1)
      Q=H(M+1,M+1)-ZZ-R-S
      R=H(M+2,M+1)
      S=DABS(P) + DABS(Q) + DABS(R)
      P=P/S
      Q=Q/S
      R=R/S
      IF (M.EQ.L) GO TO 100
      IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P)
1      * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 100
      CONTINUE
90      MP2=M+2
100     DO 110 I=MP2, EN
      H(I,I-2)=0.0D0
      IF (I.EQ.MP2) GO TO 110
      H(I,I-3)=0.0D0
      CONTINUE
110     DOUBLE OR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
      DO 190 K=M, NA
      NOTLAS=K.NE.NA
      IF (K.EQ.M) GO TO 120
      P=H(K,K-1)
      Q=H(K+1,K-1)
      R=0.0D0
      IF (NOTLAS) R=H(K+2,K-1)
      X=DABS(P) + DABS(Q) + DABS(R)
      IF (X.EQ.0.0D0) GO TO 190
      P=P/X
      Q=Q/X
      R=R/X
      S=DSIGN(DSQRT(P*P+Q*Q+R*R),P)
      IF (K.EQ.M) GO TO 130
      H(K,K-1)=-S*X
      GO TO 140
      .NE. M) H(K,K-1)=-H(K,K-1)
130     P=P+S
140     X=P/S
      Y=Q/S
      ZZ=R/S
      Q=Q/P

```

```

REAL*8 H(NM,N),WR(N),WI(N)
REAL*8 PQA,TX,Y,ZZ,NORM,MACHEP
REAL*8 DSORT,DABS,DSIGN
INTEGER MNO
LOGICAL NOTIAS
DATA MACHEP/Z3410000000000000/
IERR=0
NORM=0.0D0
K=1
C-----STORE ROOTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM-----
DO 20 I=1,N
DO 10 J=K,N
NORM=NORM+DABS(H(I,J))
K=I
IF (I.GE. LOF .AND. I.LE. IGH) GO TO 20
WR(I)=H(I,I)
WI(I)=0.0D0
CONTINUE
EN=IGH
T=0.0D0
C-----SEARCH FOR NEXT EIGENVALUES-----
30 IF (EN.LT. LOW) GO TC 250
ITS=0
NA=EN-1
ENM2=NA-1
C-----LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-----
40 DO 50 LL=LOW,EN
L=EN+LOW-LI
IF (L.EQ. LOW) GO TO 60
S=DABS(H(L-1,L-1))+DABS(H(L,L))
IF (S.EQ. 0.0D0) S=NCRH
IF (DABS(H(L,L-1)).LE. MACHEP * S) GO TO 60
CONTINUE
C-----FORM SHIFT-----
60 X=H(EN,EN)
IF (L.EQ. EN) GO TO 200
Y=H(NA,NA)
W=H(EN,NA)*H(NA,EN)
IF (L.EQ. NA) GO TO 210
IF (ITS.EQ. 30) GO TC 240
IF (ITS.NE. 10 .AND. ITS.NE. 20) GO TO 80
C-----FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LOW,EN
H(I,I)=H(I,I)-X
S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
X=0.75D0*S
Y=X
70

```

```

C-----MULTIPLY BY TRANSFORMATION MATRIX TO GIVE-----
C-----VECTORS OF ORIGINAL FULL MATRIX.-----
      DO 490 JJ=LCH,N
      J=N+LOW-JJ
      M=MIN0(J,IGH)
      DO 490 I=LOW,IGH
      ZZ=0.0D0
      DO 480 K=LOW,M
      ZZ=ZZ+Z(I,K)*H(K,J)
480   Z(I,J)=ZZ
490   CONTINUE
      GO TO 510

C-----SET ERROR -->NO CONVERGENCE TO AN-----
C-----EIGENVALUE AFTER 30 ITERATIONS-----
500   IERR=EN
510   RETURN
      END

C-----
SUBROUTINE BALBAK (NM,N,LOW,IGH,SCALE,M,Z)
  INTEGER I,J,K,M,N,II,MM,IGH,LOW
  REAL*8 SCALE(N),Z(NM,M),S
  IF (M.EQ.0) GO TO 60
  IF (IGH.EQ. LOW) GO TO 30
  DO 20 I=LOW,IGH
  S=SCALE(I)
  DO 10 J=1,M
  Z(I,J)=Z(I,J)*S
  CONTINUE
  DO 50 II=1,N
  I=II
  IF (I.GE. LOW.AND. I.LE. IGH) GO TO 50
  IF (I.LT. LOW) I=LOW-II
  K=SCALE(I)
  IF (K.EQ. I) GO TO 50
  DO 40 J=1,M
  S=Z(I,J)
  Z(I,J)=Z(K,J)
  Z(K,J)=S
  CONTINUE
  CONTINUE
  RETURN
  END

C-----
SUBROUTINE HOR (NM,N,LOW,IGH,H,WR,WI,IERR)
  INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LOW,MP2,ENM2,IERR

```

```

ENM2=NA-1
IF (ENM2.EQ.0) GO TO 450
DO 440 I=1,ENM2
  I=NA-I
  W=H(I,I)-P
  RA=0.0D0
  SA=H(I,EN)
  DO 400 J=M,NA
    RA=RA+H(I,J)*H(J,NA)
    SA=SA+H(I,J)*H(J,EN)
  CONTINUE
  IF (WI(I) .GE. 0.0D0) GO TO 410
  ZZ=W
  R=RA
  S=SA
  GO TO 440
410 M=I
  IF (WI(I) .NE. 0.0D0) GO TO 420
  Z3=DCMPLX(-RA,-SA)/DCMPLX(W,Q)
  H(I,NA)=DREAL(Z3)
  H(I,EN)=DIMAG(Z3)
  GO TO 440
C-----SOLVE COMPLEX EQUATIONS-----
420 X=H(I,I+1)
  Y=H(I+1,I)
  VR=H(I,I)
  VI=H(I,I)
  IF (VR.EQ.0.0D0) GO TO 430
  1ABS(Q)=DABS(X)+DABS(Y)+DABS(ZZ)
  Z3=DCMPLX(X*VR-ZZ*RA+Q*SA,X*S-ZZ*SA-Q*RA)/DCMPLX(VR,VI)
  H(I,NA)=DREAL(Z3)
  H(I,EN)=DIMAG(Z3)
  IF (DABS(X) .LE. DABS(ZZ)+DABS(Q)) GO TO 430
  H(I+1,NA)=(-RA-W*H(I,NA)+Q*H(I,EN))/X
  H(I+1,EN)=(-SA-W*H(I,EN)-Q*H(I,NA))/X
  GO TO 440
430 Z3=DCMPLX(-R-Y*H(I,NA),-S-Y*H(I,EN))/DCMPLX(ZZ,Q)
  H(I+1,NA)=DREAL(Z3)
  H(I+1,EN)=DIMAG(Z3)
  CONTINUE
440 C-----END COMPLEX VECTOR-----
450 CONTINUE
C-----END BACK SUBSTITUTION. VECTORS OF ISOLATED ROOTS-----
DO 470 I=1,N
  IF (I .GE. LOW .AND. I .LE. IGH) GO TO 470
  DO 460 J=I,N
    Z(I,J)=H(I,J)
  CONTINUE
460
470

```

```

10 WRITE (5,120)
20 DO 20 I=1,NS
30 DO 10 J=1,NC
40 WRITE (5,110) I,J
50 CALL RDRREAL (ANSR)
60 G(I,J)=ANSR
70 CONTINUE
80 CONTINUE
90 CALL FRTCMS ('CLRSCRN ')
100 CONTINUE
110 WRITE (5,130)
120 CALL MATPRT (G,NS,NC)
130 WRITE (5,140)
140 CALL RDCCHAR (IANS)
150 IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
160 GO TO 70
170 WRITE (5,150)
180 GO TO 50
190 CONTINUE
200 IF (IANS.EQ.IZ) GO TO 100
210 WRITE (5,160)
220 CALL RDINT (IANS)
230 K=IANS
240 WRITE (5,170)
250 CALL RDINT (IANS)
260 L=IANS
270 WRITE (5,110) K,L
280 CALL RDRREAL (ANSR)
290 DUM=ANSR
300 DO 90 I=1,NS
310 DO 80 J=1,NC
320 IF ((I.EQ.K).AND.(J.EQ.L)) G(I,J)=DUM
330 CONTINUE
340 GO TO 30
350 CONTINUE
360 CALL FRTCMS ('CLRSCRN ')
370 RETURN
380
390 FORMAT (5X,14H THE ELEMENT G(I2,1H,I2,2H)=)
400 FORMAT (1X,51H ENTER THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}
410 1-//10X,43H DIMENSION = # STATES (NS) X # CONTROLS (NC)
420 1//)
430 FORMAT (1X,10X,47H THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}....,
440 1//)
450 FORMAT (1X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
460 1-//10X,19H TYPE "YES" OR "NO".)
470 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)

```





```

WRITE (5,100) K,L
CALL RDRREAL (ANSR)
DUM=ANSR
DO 80 I=1,NO
DO 70 J=1,NC
IF ((I.EQ.K).AND.(J.EQ.L)) AY(I,J)=DUM
CONTINUE
GO TO 30
CONTINUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----
100 FORMAT (5X,14H THE ELEMENT A(I2,1H I2,2H)=)
110 FORMAT (//,5X,54H ENTER THE OUTPUT MEASUREMENT COST MATRIX {"A"-MAT
111 1RIX} //,5X,53H DIMENSION = # OBSERVATIONS {NO} X # OBSERVATIONS {NO}
120 2)
120 FORMAT (//,5X,50H THE OUTPUT MEASUREMENT COST MATRIX {"A"-MATRIX}..
121 1.//)
130 FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
131 1ENT? //,10X,19H TYPE "YES" OR "NO".)
140 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
150 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160 1)
160 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
161 1.)
END
C=====
C SUBROUTINE READB (NC,ISAB,B)
C INPUTS THE "B" MATRIX {CONTROL COST WEIGHTING MATRIX}.
C=====
REAL*8 B(NC,NC),DUM,ANSR
INTEGER IANS,IJ,K,L
DATA IY,YI,I2,N/
IF (ISAB.EQ.1) GO TO 20
WRITE (5,90)
DO 10 I=1,NC
DO 10 J=1,NC
WRITE (5,80) I,J
CALL RDRREAL (ANSR)
B(I,J)=ANSR
C-----
10 10
C-----
20 CALL FRTCMS ('CLRSCRN ')
WRITE (5,100)
CALL MATPR (B,NC,NC)
30 WRITE (5,110)
CALL RDCCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
GO TO 50

```

```

40 WRITE (5,120)
GO TO 30
50 CONTINUE
IF (IANS.EQ.IZ) GO TO 70
WRITE (5,130)
CALL RDINT (IANS)
K=IANS
WRITE (5,140)
CALL RDINT (IANS)
I=IANS
WRITE (5,80) K,L
CALL RDREAL (ANSR)
DUM=ANSR
DO 60 I=1,NC
DO 60 J=1,NC
IF ((I.EQ.K).AND.(J.EQ.L)) B(I,J)=DUM
60 CONTINUE
GO TO 20
70 CONTINUE
CALL PRTCMS ('CLRSCRN ')
RETURN
C-----
80 FORMAT (5X,14H THE ELEMENT B(I2,1H,I2,2H)=)
90 FORMAT (5X,52H ENTER THE CONTROL COST WEIGHTING MATRIX {"B"-MATRIX
1X,10X,45H DIMENSION = # CONTROLS {NC} X # CONTROLS {NC})
100 FORMAT (5X,37H THE CONTROL COST MATRIX.....B...//)
110 FORMAT (5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
1X,19H TYPE "YES" OR "NO".)
120 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
130 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
140 FORMAT (5X,52H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1)
END
C=====
C SUBROUTINE READG2 (NS,NG,IGAM,GAM)
C INPUTS THE "GAM" MATRIX {PROCESS NOISE DISTRIBUTION MATRIX}.
C=====
REAL*8 GAM(NS,NG),DUM,ANSR
INTEGER IANS,I,J,K,L,IGAM
DATA IY,IY,I2,I2/I,N/N
IF (IGAM.EQ.1) GO TO 40
WRITE (5,120)
DO 20 I=1,NS
DO 10 J=1,NG
WRITE (5,110) I,J
CALL RDREAL (ANSR)
GAM(I,J)=ANSR
CONTINUE
10

```

```

20 CONTINUE
30 CALL FRTCHS ('CLRSCRN ')
40 CONTINUE
   WRITE (5,130)
   CALL MATPRT (GAM,NS,NG)
50   WRITE (5,140)
   CALL RDC HAR (IANS)
   IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
60   GO TO 70
   WRITE (5,150)
   GO TO 50
70 CONTINUE
   IF (IANS.EQ.IZ) GO TO 100
   WRITE (5,160)
   CALL RDINT (IANS)
   K=IANS
   WRITE (5,170)
   CALL RDINT (IANS)
   L=IANS
   WRITE (5,110) K,L
   CALL RDRREAL (ANSR)
   DUM=ANSR
   DO 90 I=1,NS
   DO 80 J=1,NG
80   IF ((I.EQ.K).AND.(J.EQ.L)) GAM(I,J)=DUM
90 CONTINUE
100 CONTINUE
   GO TO 30
   CONTINUE
   CALL FRTCHS ('CLRSCRN ')
   RETURN
-----
110 FORMAT (5X,16H THE ELEMENT GAM(I2,1H,I2,2H)=)
120 FORMAT (//5X,36H ENTER THE PROCESS NOISE DISTRIBUTION, /5X,24H MATRI
1X,1H GAMMA A-MATRIX) //,2X,56H DIMENSION = # STATES {NS} X # PROCESS
2X,NOISE SOURCES {NG}
130 FORMAT (//,10X,37H THE PROCESS NOISE DISTRIBUTION MATRIX, /,10X,19H {
1H GAMMA A-MATRIX} //,10X,19H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
1ENT? //,10X,19H TYPE "YES" OR "NO".)
140 FORMAT (//,1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
150 FORMAT (//,1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
160 FORMAT (//,1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
170 FORMAT (//,1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
1. )
   END
=====
SUBROUTINE READQ (NG,Q)
=====

```

```

C=====
C INTERACTIVELY INPUTS THE "Q" MATRIX (NOISE WEIGHTING MATRIX) =====
C=====
REAL*8 Q(NG,NG),DUM,ANSR
INTEGER IANS,IJ,K,L
DATA IY,YI,I2,I2,N1/
WRITE (5,110)
DO 20 I=1,NG
DO 10 J=1,NG
WRITE (5,100) I,J
CALL PDREAL (ANSR)
Q(I,J)=ANSR
CONTINUE
CONTINUE
10
20
C-----
30 CALL PRTCHS ('CLRSCRN ')
WRITE (5,120)
CALL MATPRT (Q,NG,NG)
40 WRITE (5,130)
CALL RDCHEX (IANS)
IF (IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
50 GO TO 60
WRITE (5,140)
GO TO 40
60 CONTINUE
IF (IANS.EQ.IZ) GO TO 90
WRITE (5,150)
CALL RDINT (IANS)
K=IANS
WRITE (5,160)
CALL RDINT (IANS)
L=IANS
WRITE (5,100) K,L
CALL PDREAL (ANSR)
DUM=ANSR
DO 80 I=1,NG
DO 70 J=1,NG
IF (I.EQ.K).AND.(J.EQ.L)) Q(I,J)=DUM
CONTINUE
CONTINUE
GO TO 30
CONTINUE
CONTINUE
CALL PRTCHS ('CLRSCRN ')
RETURN
C-----
100 FORMAT (5X,14H THE ELEMENT Q(I2,1H,I2,2H)=)
110 FORMAT (//5X,44H ENTER THE PROCESS NOISE PSD WEIGHTING MATRIX,5X
1,12H{"Q MATRIX"},//5X,42H DIMENSION = # PROCESS NOISE SOURCES {NG}
2 X,11X,27H#PROCESS NOISE SOURCES {NG})

```



```

CALL FRTCHS ('CLRSCRN ')
RETURN
-----
80  FORMAT (5X,14H THE ELEMENT R(I2,1H I2,2H)=)
90  FORMAT (//,5X,60H ENTER THE MEASUREMENT NOISE DISTRIBUTION MATRIX { "
1R MATRIX } //,5X,53H DIMENSION = # OBSERVATIONS {NO} X # OBSERVATIO
2NS {NO} )
100  FORMAT (//,15X,50H THE MEASUREMENT NOISE DISTRIBUTION MATRIX.....R.
1..//)
110  FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
1ENT? //,10X,19H TYPE "YES" OR "NO".)
120  FORMAT (//,1X,51H WARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
130  FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
140  FORMAT (5X,52H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1)
END
=====
C  SUBROUTINE READPE (NS,NO,FBGE)
C  INTERACTIVELY INPUTS THE "K" {FEEDBACK GAIN ESTIMATOR MATRIX} =
=====
REAL*8  FBGE(NS,NO) ,DUM,ANSR
INTEGER IANS,IJ,K,L
DATA IY,YI,I2,N1/
WRITE (5,110)
DO 20 I=1,NS
DO 10 J=1,NO
WRITE (5,100) I,J
CALL RDREAL (ANSR)
FBGE(I,J)=ANSR
CONTINUE
CONTINUE
10
20
C-----
30  CALL FRTCHS ('CLRSCRN ')
    WRITE (5,120)
    CALL MATBRT (FBGE,NS,NO)
    40  WRITE (5,130)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
    GO TO 60
    50  WRITE (5,140)
    GO TO 40
    60  CONTINUE
    IF (IANS.EQ.IZ) GO TO 90
    WRITE (5,150)
    CALL RDINT (IANS)
    K=IANS
    WRITE (5,160)
    CALL RDINT (IANS)

```

```

L=IANS (5,100) K,L
WRITE (5,100) K,L
CALL RDRREAL (ANSR)
DUM=ANSR
DO 80 I=1, NS
DO 70 J=1, NO
IF (I.EQ.K).AND. (J.EQ.L) FBGE(I,J)=DUM
CONTINUE
GO TO 30
CONTINUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----
100 FORMAT (5X,14HTHE ELEMENT K(I2,1H,I2,2H)=)
110 FORMAT (5X,54HTHE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATR
11X,10X,48HDIMENSION = # STATES {NS} X # OBSERVATIONS {NO}.)
120 FORMAT (5X,15X,47HTHE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATRIX},
11X,10X,48HDIMENSION = # STATES {NS} X # OBSERVATIONS {NO}.)
130 FORMAT (5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
13X,10X,19HTYPE "YES" OR "NO".)
140 FORMAT (5X,54HWARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
150 FORMAT (5X,50HTHE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160 FORMAT (5X,52HTHE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED.)
11END
C=====
C SUBROUTINE READW (NG,WR)
C INTERACTIVELY INPUTS THE "W0" MATRIX {STEADY DISTURBANCE VECTOR =
C MATRIX} ELEMENT BY ELEMENT.
C=====
REAL*8 WR(NG),DUM,ANSR
INTEGER IANS,I,K
DATA IY,YI,I2/'N' /
WRITE (5,100)
DO 10 I=1,NG
WRITE (5,80) I
CALL RDRREAL (ANSR)
WR(I)=ANSR
CONTINUE
10
C-----
20 CALL FRTCMS ('CLRSCRN ')
WRITE (5,110)
WRITE (5,90) (WR(I),I=1,NG)
30 CALL RDRCHAR (IANS)
IF ((IANS.NE.IY).AND. (IANS.NE.I2)) GO TO 40
GO TO 50

```

```

40 WRITE (5,130)
50 GO TO 30
CONTINUE
IF (IANS.EQ.IZ) GO TO 70
WRITE (5,140)
CALL RDINT (IANS)
K=IANS
WRITE (5,80) K
CALL RDREAL (ANSR)
DUM=ANSR
DO 60 I=1,NG
IF (I.EQ.K) WR(I)=DUM
CONTINUE
GO TO 20
CONTINUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----
80 FORMAT (5X,15HTHE ELEMENT W0(,I2,2H)=)
90 FORMAT (F12.5)
100 FORMAT (//,5X,57HENTER THE STEADY DISTURBANCE VECTOR MATRIX {"W0"-M
110 1A TRIX},//,10X,44HDIMENSION=, PROCESS NOISE SOURCES {NG} X 1)
110 1X),//,15X,53HTHE STEADY DISTURBANCE VECTOR MATRIX {"W0"-MATRI
120 1X),//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
120 1ENT?//,10X,19HTYPE "YES" OR "NO")
130 FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
140 FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
END
C=====
C SUBROUTINE RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
SUBROUTINE RDREAL (ANSR)
REAL*8 ANSR
INTEGER COUNT
C-----
10 COUNT=0
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
      CONTINUE
      STOP
C-----
50  FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUME
      1RICAL VALUE.)
60  FORMAT (///,5X,47HPROGRAM TERMINATION ~ TWO NULL STRINGS ENTERED!)
      END
C=====
C  SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY
C  INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS AN IMPROPER
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
      SUBROUTINE RDINT (IANS)
      INTEGER COUNT,IANS
C-----
      COUNT=0
      CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 50
      CONTINUE
20  READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 40,40,30
      CONTINUE
      RETURN 5
40  REWIND 5
      WRITE (5,70)
      GO TO 10
      CONTINUE
      STOP
C-----
60  FORMAT (//,5X,49HPROGRAM TERMINATION - TWO IMPROPER DATA ENTRIES!!
      1)
70  FORMAT (1X,56HWARNING: IMPROPER DATA ENTRY!  ENTER A POSITIVE INTE
      1GER.)
      END
C=====
C  SUBROUTINE RDCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY
C  ('YES' OR 'NO') INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY
C  ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY
C=====
      SUBROUTINE RDCHAR (IANS)
      INTEGER COUNT,IANS
      DATA IY,'Y',IZ,'N' /
C-----

```

```

10 COUNT=0
   CONTINUE
   COUNT=COUNT+1
   IF (COUNT.IT.3) GO TO 20
   WRITE (5,60)
   GO TO 40
20 CONTINUE
   REWIND 5
   READ (5,70,END=30,ERR=30) IANS
   RETURN 5
30 REWIND 5
   WRITE (5,50)
   GO TO 10
40 CONTINUE
   STOP
C-----
50 FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
   1OR "NO".)
60 FORMAT (//,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED!)
70 FORMAT (A1)
   END
C=====
C SUBROUTINE MATPRT -- DISPLAYS A TWO-DIMENSIONAL ARRAY (16 COLS. MAX) =
C IN VARIABLE SCREEN FORMAT FOR USER EASE IN ROW IDENTIFICATION.
C=====
SUBROUTINE MATPRT (PRTT,NROW,NCOL)
  IMPLICIT REAL*8 (A-H,C-Z)
  DIMENSION PRTT(NROW,NCOL)
C-----
  IF (NCOL.EQ.0) NCOL=1
  IF (NCOL.EQ.1) WRITE (5,10)
  IF (NCOL.EQ.2) WRITE (5,20)
  IF (NCOL.EQ.3) WRITE (5,30)
  IF (NCOL.EQ.4) WRITE (5,40)
  IF (NCOL.EQ.5) WRITE (5,50)
  IF (NCOL.EQ.6) WRITE (5,60)
  IF (NCOL.EQ.7) WRITE (5,70)
  IF (NCOL.EQ.8) WRITE (5,80)
  IF (NCOL.EQ.9) WRITE (5,90)
  IF (NCOL.EQ.10) WRITE (5,100)
  IF (NCOL.EQ.11) WRITE (5,110)
  IF (NCOL.EQ.12) WRITE (5,120)
  IF (NCOL.EQ.13) WRITE (5,130)
  IF (NCOL.EQ.14) WRITE (5,140)
  IF (NCOL.EQ.15) WRITE (5,150)
  IF (NCOL.EQ.16) WRITE (5,160)
  RETURN
C-----

```

```

10  FORMAT (F12.5)
20  FORMAT (2F12.5)
30  FORMAT (3F12.5)
40  FORMAT (4F12.5)
50  FORMAT (5F12.5)
60  FORMAT (6F12.5)
70  FORMAT (6F12.5)
80  FORMAT (6F12.5)
90  FORMAT (6F12.5)
100 FORMAT (6F12.5)
110 FORMAT (6F12.5)
120 FORMAT (6F12.5)
130 FORMAT (6F12.5)
140 FORMAT (6F12.5)
150 FORMAT (6F12.5)
160 FORMAT (6F12.5)
END
C=====
C SUBROUTINE RDMATF -- READS THE FLAGS AND MATRIX SIZES FROM
C THE DATA FILE CN FILEDEF 9. ASKS IF YOU WANT TO USE THE MATRICES.
C=====
      SUBROUTINE RDMATF (NS,NC,NOB,NG,ISAF,ISAG,ISAH,ISAD,IGAM,ISAA,ISAB
      1,IRDMAT,IFDFW)
      DATA IVES/'Y', INO/'N'/
      INTEGER NS,NC,NOB,NG,ISAF,ISAG,ISAH,ISAD,IGAM,IRDMAT,INO, IANS, K
      REWIND 9
      READ (9,240,END=30,ERR=30) K, IANS
      IF (IANS.EQ.1) GO TO 10
      GO TO 30
10  READ (9,250) NS,NC,NOB,NG,IFDFW
      WRITE (5,255)
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,260)
      CALL RDINT (IANS)
      IF (IANS.GT.3) GO TO 20
      IF (IANS.EQ.3) GO TO 30
      IRDMAT=1
      IF (IANS.EQ.2) GO TO 40
      ISAF=1
      ISAG=1
      ISAH=1
      ISAD=1
      IGAM=1
      ISAA=1
      ISAB=1
      RETURN
30  CALL FRTCMS ('CLRSCRN ')
40  CALL FRTCMS ('CLRSCRN ')

```

```

440 FORMAT (5X,7HSTATE #,I3,2H=)
450 FORMAT (//,5X,52HDO {CU,WISH TO CHANGE ANY OF THE SIGNIFICANT STAT
1ES? //,10X,19HTYPE "YES" OR "NO".})
460 FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY...ENTER "YES" OR "NO".)
470 FORMAT (5X,37HENTER THE N-TH STATE # TO BE CHANGED.)
480 FORMAT (5X,17HENTER NEW STATE #,I3)
490 FORMAT (1X,63H***WARNING: STATE NUMBERS HAVE BEEN IMPROPERLY EN
1TERED ***//,16X,39HSTATE #,S MUST BE IN THE RANGE OF 1 AND, I3,/,2
24X,25HAND IN ASCENDING ORDER!!//)
500 FORMAT (11,3X,I1)
510 END
C=====
SUBROUTINE NAMULT (L,M,N,F,G,H,NL,NM,NN)
C=====
C-----
INTEGER I,J,K
REAL*8 F(L,M),G(M,N),H(L,N),DEL
C-----
DO 20 I=1,NL
DO 20 K=1,NN
DEL=0.D0
DO 10 J=1,NM
DEL=DEL+F(I,J)*G(J,K)
10 H(I,K)=DEL
20 RETURN
END
C=====
SUBROUTINE MI (M,N,AI,DI,DV,EV,FV,GV,HV)
C=====
C-----
REAL*8 DI(M,M),DV(M),EV(M),AI(M,M),D,E,DABS
INTEGER FV(M),GV(M),EF
LOGICAL *1HV(M)
COMMON ER,FMT
C-----
DO 10 I=1,N
HV(I)=.FALSE.
DO 10 J=1,N
DI(I,J)=AI(I,J)
DO 10 K=1,N
L=0
D=0.D0
DO 20 J=1,N
IF (HV(J)) GO TO 20
E=DABS(DI(K,J))
IF (E.LE. D) GO TO 20
D=E

```

```

270 HR(I,J)=HR(I,J)-CSTR(I,J)
WRITE (FMT,380)
DO 280 I=1,NO
WRITE (FMT,320) (HR(I,J),J=1,NR)
IF (IFDPW.EQ.0) GO TO 310
CALL HAMULT (NO,H,H2,F22,DUM7,NO,H,M)
CALL HAMULT (NO,M,NC,DUM7,62,DSTR,NO,H,NC)
DO 290 I=1,NO
DO 290 J=1,NC
DR(I,J)=D(I,J)-DSTR(I,J)
WRITE (FMT,390)
DO 300 I=1,NO
WRITE (FMT,320) (DR(I,J),J=1,NC)
CONTINUE
REWIND 10
I=0
IANS=1
WRITE (10,510) I IANS NO NG IFDPW
WRITE (10,410) (PR(I,J),J=1,NC),I=1,NR)
WRITE (10,400) ((GR(I,J),J=1,NC),I=1,NR)
WRITE (10,400) ((HR(I,J),J=1,NC),I=1,NR)
WRITE (10,400) ((DR(I,J),J=1,NC),I=1,NR)
WRITE (10,400) ((GAM(I,J),J=1,NG),I=1,NR)
WRITE (10,400) ((FBGC(I,J),J=1,NR),I=1,NC)
WRITE (10,400) ((FBGE(I,J),J=1,NO),I=1,NR)
WRITE (10,400) ((AY(I,J),J=1,N6),I=1,N6)
WRITE (10,400) ((B(I,J),J=1,NC),I=1,NC)
RETURN

```

```

C 55 FORMAT(2(0PL3.5))
C
320 FORMAT (10(2X,1PD11.4))
330 FORMAT (13I5)
340 FORMAT (///3X,36HTHE DESIRED REDUCED STATE ORDER IS: ,I3)
350 FORMAT (///3X,27HTHE SIGNIFICANT STATES ARE:)
360 FORMAT (///3X,28HTHE REDUCED PLANT MATRIX IS: /)
370 FORMAT (///3X,43HTHE REDUCED CONTROL DISTRIBUTION MATRIX IS: /)
380 FORMAT (///3X,42HTHE REDUCED OUTPUT DISTRIBUTION MATRIX IS: /)
390 FORMAT (///3X,34HTHE REDUCED FEEDFORWARD MATRIX IS: /)
400 FORMAT (4D20.13)
410 FORMAT (5I5)
420 FORMAT (5X,67HTHE "N" SIGNIFICANT STATES WHICH REPRESENT THE
1 REDUCED ORDER /,5X,42HMODEL. ENTER STATE #'S IN ASCENDING ORDER.)
430 1 FORMAT (5X,58HDO YOU WISH TO INPUT DESIRED STATES FOR YOUR REDUCED
2 ORDER /,5X,23HMODEL FROM A DATA FILE? /,5X,57HDATA FILE MUST BE N
3 AMED "STATES DATA A1" IN FIXED FORMAT. /,5X,49HTHE READ FORMAT IS
3"13I5" PER 72 CHARACTER LINE. /,10X,19HTYPE "YES" OR "NO".)

```

```

190 CALL MAHULT (M,M,NR,F22,F21,DUM2,M,M,NR)
    CALL MAHULT (NR,M,NR,F12,DUM2,F,M,NR,M,NR)
    DO 190 I=1,NR
    DO 190 J=1,NR
    PR(I,J)=F1(I,J)-FM(I,J)
    WRITE (FMT,360)
    DO 200 I=1,NR
    WRITE (FMT,320) (PR(I,J),J=1,NR)
C-----
C    COMPUTE THE REDUCED ORDER INPUT MATRIX.
C    G1 = < RM > * G
C    G2
C    < GR > = < G1 > - < F12 > * < F22** -1 > * < G2 >
C-----
210 CALL MAHULT (NS,NS,NC,RM,G,GM,NS,NS,NC)
    DO 210 I=1,NR
    DO 210 J=1,NC
    GR(I,J)=GM(I,J)
    DO 220 I=1,M
    DO 220 J=1,NC
    G2(I,J)=GM(I+NR,J)
    CALL MAHULT (NR,M,NC,DUM3,G2,DUM4,NR,M,NC)
    CALL MAHULT (NR,M,NC,DUM3,G2,DUM4,NR,M,NC)
    DO 230 I=1,NR
    DO 230 J=1,NC
    GR(I,J)=GR(I,J)-DUM4(I,J)
    WRITE (FMT,370)
    DO 240 I=1,NR
    WRITE (FMT,320) (GR(I,J),J=1,NC)
C-----
C    COMPUTE THE REDUCED ORDER OUTPUT MATRICES.
C    < H1 H2 > = < H > * < RM** -1 >
C    < HR > = < H1 > - < H2 > * < F22** -1 > * < F21 >
C    < DR > = < D > - < H2 > * < F22** -1 > * < G2 >
C-----
250 WRITE (6,983) NS,NC,NO,NR,M
    CALL HI (NS,NS,RH,DI,LV,EV,FV,GV,HV)
    CALL MAHULT (NO,NS,NS,H,RM,DUM5,NO,NS,NS)
    DO 250 I=1,NO
    DO 250 J=1,NR
    HR(I,J)=DUM5(I,J)
    DO 260 I=1,NO
    DO 260 J=1,M
    H2(I,J)=DUM5(I,(NR+J))
    CALL MAHULT (NO,M,M,H2,F22,HSTR,NO,M,M)
    CALL MAHULT (NO,M,NR,HSTR,F21,CSTR,NO,M,NR)
    DO 270 I=1,NO
    DO 270 J=1,NR

```

```

110 DO 110 I=1,NS
DO 110 J=1,NS
RM(I,J)=0.50
CONTINUE
K=NR
DO 140 I=1,NS
DO 120 J=1,NR
IF (I.EQ.NRS(J)) GO TO 130
CONTINUE
K=K+1
RM(K,I)=1.D0
GO TO 140
130 RM(J,I)=1.D0
140 CONTINUE
C-----
C DIVIDE THE PLANT MATRIX INTO THE FOUR SUBMATRICES
C USED IN THE ORDER REDUCTION.
C <P> = <RM>*<F>*<RM**--1> = <----- >
C P21P22
C-----
CALL NAMULT (NS,NS,NS,NS,RM,F,DUM1,NS,NS,NS)
CALL HI (NS,NS,NS,DI,LV,EV,FV,GV,HV)
CALL NAMULT (NS,NS,NS,DUM1,RM,T,NS,NS,NS)
CALL HI (NS,NS,NS,DI,DV,EV,FV,GV,HV)
DO 150 I=1,NR
DO 150 J=1,NR
F11(I,J)=T(I,J)
Q=NR+1
H=NS-NR
DO 160 I=1,NR
DO 160 J=Q,NS
P=J-NR
F12(I,P)=T(I,J)
DO 170 I=Q,NS
DO 170 J=1,NR
P=I-NR
F21(P,J)=T(I,J)
DO 180 I=Q,NS
DO 180 J=Q,NS
F22(I-NR,J-NR)=T(I,J)
150
160
170
180
C-----
C COMPUTE THE REDUCED PLANT MATRIX.
C <PR> = <F11> - <F12>*<F22**--1>*<F21> >
C-----
WRITE (PMT,340) NR
WRITE (PMT,350)
WRITE (PMT,330) (NRS(I),I=1,NR)
CALL HI (H,M,F22,DI,DV,EV,FV,GV,HV)

```

FORM THE REORDERED STATE VECTOR X'

< X' > = < RM > \* < X >

```

10 CONTINUE
   CALL PRTCMS ('CLRSCRN ')
   WRITE (5,430)
   CALL RDCHAR (IANS)
   CALL PRTCMS ('CLRSCRN ')
   IF (IANS.EQ.IY) ISTA=1
   IF (ISTA.EQ.1) GO TO 90
   WRITE (5,420)
   DO 40 I=1,NR
     WRITE (5,440) I
     CALL RDREAL (ANSR)
     NRS(I)=IDINT(ANSR)
     IF ({NRS(I)-LE.0}).OR. (NRS(I)-GT.NS)) GO TO 30
     IF (I.EQ.1) GO TO 40
     IF (NRS(I)-GT.NRS(I-1)) GO TO 40
     WRITE (5,490) NS
     GO TO 20
   CONTINUE
   CALL PRTCMS ('CLRSCRN ')
   WRITE (5,350)
   WRITE (5,330) (NRS(I),I=1,NR)
   CONTINUE
   WRITE (5,450)
   CALL RDCHAR (IANS)
   CALL PRTCMS ('CLRSCRN ')
   IF (IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 60
   GO TO 70
   WRITE (5,460)
   GO TO 50
   CONTINUE
   IF (IANS.EQ.IZ) GO TO 100
   IF (IANS.EQ.IY) GO TO 80
   WRITE (5,470)
   CALL RDREAL (ANSR)
   DUM=IDINT(ANSR)
   WRITE (5,480) DUM
   CALL RDREAL (ANSR)
   K=IDINT(ANSR)
   NRS(DUM)=K
   GO TO 40
   READ (7,330) (NRS(I),I=1,NR)
   GO TO 40
   CONTINUE
   CALL PRTCMS ('CLRSCRN ')
100

```

```

2290 8NUE?//,10X,19HTYPE "YES" OR "NO".)
2300 1D LESS THAN 13,4H*** WARNING: REDUCED ORDER MUST BE GREATER THAN 0 AND
2310 1: A DATA FILE CONTAINING THE REDUCED ORDER DATA//,10X,44HWILL BE G
GENERATED REGARDLESS OF YOUR ANSWER.//,10X,36HTYPE "S" FOR SCREEN O
3END
310 3END

```

```

C=====
SUBROUTINE REDUCY (NG,IPDPW,M,NS,NC,NO,NR,F,G,H,D,FR,GR,HR,DR
1,NBS,F11,F12,F21,F22,GH,H2,G2,T,HSTR,FH,DSTR,DUM1
2,DUM3,DUM4,DUM2,CSTR,RM,DUM5,DUM7,AI,DI,DV,EV,FV,GV,HV
3,GAM,FBGC,FBGE,AY,B)

```

```

=====
INTEGER I,J,K,L,M,Q,DUM,IAN,S,IPDFW,NRS(NR),PV(NS),GV(NS),NG,I
1STA,PHT
LOGICAL *1HV(NS)
REAL*8 F(NS,NS),G(NS,NC),H(NO,NS),D(NO,NC),GAH(NS,NG),FBGC(NC,NS)
1,P11(NR,NR),P12(NR,NR),P22(M,M),FBGE(NS,NO),AY(NO,NO)
2,GH(NS,NC),H2(NO,H),T(NC,NS),B(NC,NC)
3,FR(NR,NR),HR(NO,NR),LH(NO,NR),HSTR(N6,M),FM(NR,NR)
4,DSIR(MO,NC),GR(NR,NR),DV(NS),EV(NS)
5,DUM1(NS,NS),DUM3(NR,M),DUM4(NR,NC),DUM2(M,NR),CSTR(NO,NR)
6,PH(NS,NS),DUH5(NO,NS),DUH7(NO,M),ANSR,AI(NS,NS),DI(NS,NS)
DATA I,J,K,L,M,Q,DUM,IAN,S,IPDFW,NRS(NR),PV(NS),GV(NS),NG,I
COMMON PR,PHT
=====

```

READ IN THE PULL ORDER MODEL MATRICES FROM OPTHAT DATA FILE

[illegible]

C-----  
C READ IN THE DESIRED REDUCED ORDER AND THE DESIRED STATES.  
C-----

```

130 GO TO 110
131 M=NS-NR
132
133 C-----
134 C COMPUTE REDUCED ORDER F, G, H, AND D MATRICES
135 C-----
136 CALL REDUCX (NG, IFDFW, M, NS, NC, NO, NR, F, G, H, D, FR, GR, HR, DR
137 1, NRS, F11, F12, F21, F22, G1, G2, T, HSTR, FM, DSTR, DUM1
138 2, DUM3, DUM4, DUM5, DUM6, DUM7, AI, DI, DV, EV, FV, GV, HV
139 3, GAM, FBGC, FBGE, AY, B)
140 WRITE (5, 260)
141 IF (ER.EQ.0) GO TO 150
142 WRITE (5, 270)
143 GO TO 156
144 WRITE (5, 230)
145 GO TO 196
146 WRITE (5, 310)
147 CALL RDCHAR (IANS)
148 IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 170
149 GO TO 180
150 WRITE (5, 220)
151 GO TO 166
152 CONTINUE
153 IF ((IANS.EQ.IY) GO TO 100
154 IF ((IANS.EQ.IZ) GO TO 140
155 CONTINUE
156 STOP
157 FORMAT (5I5)
158 THE ORDER OF THE FULL SYSTEM IS: I3, 5X, 31H THE NUMBER OF OBSERVATIONS IS: I3, 5X, 26H THE NUMBER OF CONTROLS IS: I3, 5X, 10H BE INPUT.//, 5X, 24H DO YOU WISH TO
159 2/ 5X, 18H A "D" MATRIX WILL 10X, 19H TYPE "YES" OR "NO".)
160 3 CONTINUE?//, 10X, 19H TYPE "YES" OR "NO".)
161 FORMAT (1X, 51H WARNING: IMPROPER DATA ENTRY... ENTER "YES" OR "NO".)
162 FORMAT (//, 13X, 40H... OPTED IS NOW TERMINATED...//)
163 FORMAT (5X, 50H ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.)
164 FORMAT (5X, 59H ANALYSIS COMPLETE... YOUR REDUCED SYSTEM DATA HAS BEEN
165 1N SAVED//, 18X, 31H IN A FILE NAMED "OPTMATR DATA".)
166 FORMAT (//)
167 1 RUN WILL BE INVALID!!)
168 FORMAT (5X, 59H OPTED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL
169 1SYSTEM / 5X, 41H "F" "G" "H" AND "D" (IF INPUT) MATRICES.//, 5X, 56H T
170 2HE FULL SYSTEM MATRICES MUST BE READ FROM A FILE NAMED//, 5X, 48H "OP
171 3T MAT DATA" AS CREATED BY THE OPTSYX PROGRAM.//, 5X, 64H YOU MUST ALS
172 40 ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES) //, 5X, 62H AND TH
173 52 ACTUAL STATE #'S (IN ASCENDING ORDER) WHICH REPRESENT//, 5X, 58H TH
174 6E REDUCED MODEL. THE (ORDER OF THE REDUCED MODEL MUST BE//, 5X, 39H LE
175 7SS THAN THE ORDER OF THE FULL SYSTEM.//, 5X, 24H DO YOU WISH TO CONTI

```

```

C-----SUPPRESS INDIVIDUAL UNDER/OVER FLOW ERROR MESSAGES; PROVIDE SUMMARY
C-----
C      CALL ERRSET (207,256,-1,1,1,209)
C-----PROGRAM DESCRIPTION-----
10  CALL FRTCHS ('CLRSCRN ')
    WRITE (5,280)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 20
    GO TO 30
20  WRITE (5,220)
    GO TO 10
30  IF (IANS.EQ.IZ) GO TO 140
    CALL FRTCHS ('CLRSCRN ')
40  WRITE (5,300)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.SC).AND. (IANS.NE.DK)) GO TO 50
    GO TO 60
50  WRITE (5,220)
    GO TO 40
60  IF (IANS.EQ.SC) PMT=5
    IF (IANS.EQ.DK) PMT=6
C-----
C      READ SYSTEM FLAGS, FULL SYSTEM MATRIX PARAMETERS FROM OPTMAT DATA
C      AND READ DESIRED REDUCED ORDER
C-----
    CALL FRTCHS ('CLRSCRN ')
    ER=0
    DD=YY
    READ (9,200) K,IANS
    IF (IANS.NE.NO) GO TO 70
    IF (IANS.EQ.O) DD=NN
    WRITE (5,210) NS,NC,NC,DD
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IY).AND. (IANS.NE.IZ)) GO TO 80
    GO TO 90
80  WRITE (5,220)
    GO TO 70
90  CONTINUE
    IF (IANS.EQ.IZ) GO TO 140
    IF (IANS.EQ.IY) GO TO 100
    CALL FRTCHS ('CLRSCRN ')
100 WRITE (5,240)
110 CALL RDRREAL (ANSR)
    NR=IDINT (ANSR)
    IF ((NR.GE.NS).OR. (NR.LE.O)) GO TO 120
    GO TO 130
120 WRITE (5,290) NS

```



```

212X 'PLOT, OR NICHOLS PLOT OF THE SYSTEM YOU ARE EVALUATING?',
3//24X '(Y OR N)',
4//5X 'NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN',
410X '(TEK 618) TERMINAL TO UTILIZE THIS MODE',
558X 'THE F (SYSTEM) G (CONTROL) H (OBSERVABLES) I (GAM (NOISE))',
613X '53HA (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE',
716X 'SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.',
    FORMAT (10X, 29HYOU MUST ANSWER (Y)ES OR (N)O )
    FORMAT (5I5)
    FORMAT (4D20.13)
    FORMAT (11, 3X, I1)
END

```

```

110
120
130
140

```

```

      READ (9,10) ((D(I,J),J=1,NCI),I=1,NOI)
      READ (9,10) ((GAM(I,J),J=1,NGI),I=1,NSI)
      READ (9,10) ((FBGC(I,J),J=1,NSI),I=1,NCI)
      READ (9,10) ((AY(I,J),J=1,NOI),I=1,NSI)
      READ (9,10) ((B(I,J),J=1,NCI),I=1,NCI)
      RETURN

C-----
10  FORMAT(4(D20.13))
20  FORMAT(5I5)
    END

C-----
C  SUBROUTINE WRTHAT -- WRITES THE F, G, HO & GAM MATRICES TO
C  THE DATA FILE OPTMAT ON FILEDEF 9.
C-----
      SUBROUTINE WRTHAT(BA,G,HO,D,GAM,FBGC,AY,B,NS,NC,NO,NG,IFDFW)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS),
1  AY(NO,NO),B(NC,NC),FBGE(NS,NO),D(NO,NC)
      INTEGER NS,NC,NO,NG,I,J,IANS,IHO,IYES,IFDFW
      DATA IYES/YES,INO/NO/
      WRITE (5,90)
      CALL FRTCHS('CLSCRN ')
      WRITE (5,100)
      CALL RDCHAR(IANS)
      IF ((IANS.EQ.INO).OR.(IANS.EQ.IYES)) GO TO 20
      WRITE (5,110)
      GO TO 10
20  IF (IANS.EQ.INO) RETURN
      I = 0
      IANS = 1
      WRITE (9,140)
      WRITE (9,120) IANS,NO,NG,IFDFW
      WRITE (9,130) ((BA(I,J),J=1,NS),I=1,NS)
      WRITE (9,130) ((G(I,J),J=1,NC),I=1,NC)
      WRITE (9,130) ((HO(I,J),J=1,NS),I=1,NO)
      WRITE (9,130) ((D(I,J),J=1,NC),I=1,NO)
      WRITE (9,130) ((GAM(I,J),J=1,NG),I=1,NS)
      WRITE (9,130) ((FBGC(I,J),J=1,NS),I=1,NC)
      WRITE (9,130) ((FBGE(I,J),J=1,NO),I=1,NC)
      WRITE (9,130) ((AY(I,J),J=1,NO),I=1,NO)
      WRITE (9,130) ((B(I,J),J=1,NC),I=1,NC)
      WRITE (9,130)
      STOP
C-----
90  FORMAT(//////10X,DO YOU WISH TO OBTAIN A TIME RESPONSE,/,
100 112X,POLE-ZERO MAP, RCOT-LOCUS PLOT, BODE PLOT, NYQUIST,/,

```

```

280  FORMAT (///5X,48HDO YOU WISH TO SAVE THE "H"-MATRIX FROM THE LAST
1//5X, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
2ATRIX, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
3SERVAL,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING//5X,27HIND
4INDIVIDUAL MATRIX ELEMENTS,15X,19HTYPE "YES" OR "NO"
285  FORMAT (///5X,48HDO YOU WISH TO SAVE THE "D"-MATRIX FROM THE LAST
1//5X, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
2ATRIX, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
3SERVAL,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING//5X,27HIND
4INDIVIDUAL MATRIX ELEMENTS,15X,19HTYPE "YES" OR "NO"
290  FORMAT (///5X,48HDO YOU WISH TO SAVE THE "G"-MATRIX FROM THE LAST
1//5X, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
2ATRIX, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
3SERVAL,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING//5X,27HIND
4INDIVIDUAL MATRIX ELEMENTS,15X,19HTYPE "YES" OR "NO"
300  FORMAT (///5X,52HDO YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE LAST
1//5X, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
2ATRIX, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
3SERVAL,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING//5X,27HIND
4INDIVIDUAL MATRIX ELEMENTS,15X,19HTYPE "YES" OR "NO"
310  FORMAT (///5X,48HDO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST
1//5X, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
2ATRIX, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
3SERVAL,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING//5X,27HIND
4INDIVIDUAL MATRIX ELEMENTS,15X,19HTYPE "YES" OR "NO"
320  FORMAT (///5X,48HDO YOU WISH TO SAVE THE "B"-MATRIX FROM THE LAST
1//5X, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
2ATRIX, WILL BE REDISPLAYED AT//5X,34HTHE PROPER INPUT SEQUENCE INT
3SERVAL,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING//5X,27HIND
4INDIVIDUAL MATRIX ELEMENTS,15X,19HTYPE "YES" OR "NO"
330  FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY! ENTER "YES" OR "NO".)
      END
C=====
C SUBROUTINE RDMAT--READS THE F G H D GAM A AND B MATRICES FROM =
C MATRICES FROM THE DATA FILE OPTMAT ON FILEDEP 9. =
C=====
      SUBROUTINE RDMAT(BA,G,H,O,D,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT,
1  IFDFW)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS),
2  AY(NO,NO),B(NC,NC),FBGE(NS,NO),D(NO,NC)
      IF(IRDMAT.EQ.0) RETURN
      REWIND 9
      READ (9,20) K,IANS
      READ (9,20) NSI,NCI,NCI,NGI,IFDFW
      READ (9,10) J=1,NSI, I=1,NSI
      READ (9,10) J=1,NCI, I=1,NCI
      READ (9,10) J=1,NSI, I=1,NOI
      READ (9,10) J=1,NCI, I=1,NOI

```

```

160 WRITE (5,300)
    CALL RDCHAR (IANS)
    IF (IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 180
170 WRITE (5,330)
    GO TO 160
180 CONTINUE
    IF (IANS.EQ.IYES) IGAM=1
    IF (IANS.EQ.INO) IGAM=0
190 CONTINUE
    -----ISAA-----
200 CALL FRTCMS ('CLRSCRN ')
    WRITE (5,310)
    CALL RDCHAR (IANS)
    IF (IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 210
    WRITE (5,330)
    GO TO 200
210 CONTINUE
    IF (IANS.EQ.IYES) ISAA=1
    IF (IANS.EQ.INO) ISAA=0
    -----ISAB-----
220 CALL FRTCMS ('CLRSCRN ')
    WRITE (5,320)
    CALL RDCHAR (IANS)
    IF (IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 230
    WRITE (5,330)
    GO TO 220
230 CONTINUE
    IF (IANS.EQ.IYES) ISAB=1
    IF (IANS.EQ.INO) ISAB=0
    RETURN
    -----
240 FORMAT (I1,3X,I1)
250 FORMAT (5I5)
255 FORMAT (//////10X,47H"FG" "G" "H" "D" "GAM" "A" AND "B" MATRICES,
260 1//12X,42H FROM YOUR PREVIOUS OPTS, RUN WERE SAVED.//,10X,36H"THE",
    2FOLLOWING OPTIONS ARE AVAILABLE://15X,38H1. USE ALL OF THE SAME MA
    3TRICES AGAIN.//15X,2. USE SELECTED MATRICES AGAIN.//,10X,17H"ENTER 1, 2, OR 3.",
    415X,3. INPUT ALL NEW MATRICES.//10X,17H"ENTER 1, 2, OR 3.",
    5//,10X,10X,NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT:
    6//,10X,34H"THE PROPER INPUT SEQUENCE INTERVAL
    7//,10X,40H"AND YOU WILL HAVE THE OPTION OF CHANGING, //,10X,
    827H"INDIVIDUAL MATRIX ELEMENTS.")
270 FORMAT (////5X,48HDO YOU WISH TO SAVE THE "F"-MATRIX FROM THE LAST
    1//,5X,39HNOTE: THE M
    2ATRIX WILL BE REDISPLAYED AT//5X,34H"THE PROPER INPUT SEQUENCE INT
    3ERVAL, //5X,40H"AND YOU WILL HAVE THE OPTION OF CHANGING, //,5X,27H"IND
    4IVIDUAL MATRIX ELEMENTS.//,15X,19H"TYPE "YES" OR "NO".)

```

```

50  WRITE (5,270)
    CALL RDCHAR (IANS)
    IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 70
60  WRITE (5,330)
    GO TO 50
70  CONTINUE
    IF (IANS.EQ.IYES) ISAF=1
    IF (IANS.EQ.INO) ISAF=0
C-----ISAH-----
80  IF (NOB.EQ.0) GO TO 110
    CALL FRTCMS ('CLRSCRN ')
    WRITE (5,280)
    CALL RDCHAR (IANS)
    IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 100
90  WRITE (5,330)
    GO TO 80
100 CONTINUE
    IF (IANS.EQ.IYES) ISAH=1
    IF (IANS.EQ.INO) ISAH=0
    CONTINUE
C-----ISAG-----
120 IF (NC.EQ.0) GO TO 150
    CALL FRTCMS ('CLRSCRN ')
    WRITE (5,290)
    CALL RDCHAR (IANS)
    IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 140
130 WRITE (5,330)
    GO TO 120
140 CONTINUE
    IF (IANS.EQ.IYES) ISAG=1
    IF (IANS.EQ.INO) ISAG=0
    CONTINUE
C-----ISAD-----
151 IF (IFDFW.EQ.0) GO TO 155
    CALL FRTCMS ('CLRSCRN ')
    WRITE (5,285)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 153
    GO TO 154
153 WRITE (5,330)
    GO TO 151
154 CONTINUE
    IF (IANS.EQ.IY) ISAD=1
    IF (IANS.EQ.IZ) ISAD=0
    CONTINUE
C-----IGAM-----
    IF (NG.EQ.0) GO TO 190
    CALL FRTCMS ('CLRSCRN ')

```

```

20      L=J
21      CONTINUE
30      IF (L) 30,30,40
31      IF (ER.EQ.1) RETURN
32      WRITE (6,90)
33      WRITE (5,90)
34      ER=1
35      RETURN
40      D=1.D0/DI(K,L)
41      DO 50 I=1,N
42      DV(I)=D*DI(I,L)
43      EV(I)=DI(K,I)
44      DI(I,L)=0.D0
45      DI(K,I)=0.D0
46      DV(K)=D
47      EV(L)=-1.D0
48      DO 60 I=1,N
49      DI(I,J)=DI(I,J)-DV(I)*EV(J)
50      DI(K,L)=D
51      FV(K)=L
52      GV(L)=K
53      HV(L)=.TRUE.
54      DO 80 I=1,N
55      K=FV(I)
56      DO 80 J=1,N
57      L=GV(J)
58      AI(K,L)=DI(I,J)
59      RETURN
60
61      FORMAT (//,5X,60HOPTRED CANNOT COMPUTE A REDUCED ORDER MODEL USING
62      1 THE STATES,5X,19HYOU HAVE REQUESTED.//,5X,62HENSURE THAT YOU HAV
63      2E ENTERED THE SIGNIFICANT STATES CORRECTLY.//,5X,66HOTHERWISE A DI
64      3FFERENT SET OF SIGNIFICANT STATES MUST BE SELECTED.//,20X,22H***
65      4***** NOTE *****//,10X,42HTHE RESULTS OF THIS RUN WILL BE INVAL
66      5ID!!)
67      END
68
69      C=====
70      C SUBROUTINE RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
71      C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
72      C STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
73      C=====
74      SUBROUTINE RDREAL (ANSR)
75      REAL*8 ANSR
76      INTEGER COUNT
77
78      COUNT=0
79      CONTINUE

```

```

COUNT=COUNT+1
IF (COUNT.LT.3) GO TO 20
WRITE (5,60)
GO TO 40
CONTINUE
READ (5,*,END=30,ERR=30) ANSR
RETURN
REWIND 5
WRITE (5,50)
GO TO 10
CONTINUE
STOP
C-----
50  FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUME
60  1RICAL VALUE.)
    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED!)
    END
C=====
C  SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY
C  INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS AN IMPROPER=
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY. =
C=====
SUBROUTINE RDINT (IANS)
INTEGER COUNT,IANS
C-----
10  COUNT=0
    CONTINUE
    COUNT=COUNT+1
    IF (COUNT.LT.3) GO TO 20
    WRITE (5,60)
    GO TO 50
    CONTINUE
    READ (5,*,END=40,ERR=40) IANS
    IF (IANS) 40,40,30
    CONTINUE
    RETURN
    REWIND 5
    WRITE (5,70)
    GO TO 10
    CONTINUE
    STOP
C-----
60  FORMAT (///,5X,49HPROGRAM TERMINATION - TWO IMPROPER DATA ENTRIES!!
70  1)
    1GER.)
    END
C=====

```

```

C SUBROUTINE RDCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY =
C ('YES' OR 'NO') INTO A FCRTTRAN PROGRAM. IF THE USER INADVERTENTLY =
C ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=
C=====
SUBROUTINE RDCHAR (IANS)
  INTEGER COUNT,IANS
  DATA IY,'Y',IZ,'N' /
C-----
  COUNT=0
  CONTINUE
  COUNT=COUNT+1
  IF (COUNT.LT.3) GO TO 20
  WRITE (5,60)
  GO TO 40
  CONTINUE
  REWIND 5
  READ (5,70,END=30,ERR=30) IANS
  RETURN
  REWIND 5
  WRITE (5,50)
  GO TO 10
  CONTINUE
  STOP
C-----
50  FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
60  FOR "NO".)
70  FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED!)
    END

```

APPENDIX C

THE OPTRED EXEC

```
=====
* THIS EXEC EXECUTES OPTRED PROGRAM *
=====
FILEDEF 05 TERM OPTRED LISTING A1
FILEDEF 06 DISK STATES DATA A1
FILEDEF 07 DISK STATES DATA A1
FILEDEF 09 DISK OPTMATR DATA A1
FILEDEF 10 DISK OPTMATR DATA A1
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP NONIMSL
LCAD OPTRED (START
```

# LIST OF REFERENCES

1. Hall, W.E., Computational Methods for the Synthesis of Rotary-Wing VTOL Aircraft Control Systems, Ph.D. Dissertation, Stanford University, Palo Alto, California, 1971.
2. Stanford University, Aero/Astro Department, User's Manual for OPTSYS 4 at SCIP, by R.A. Walker, December 1979.
3. Stanford University, Aero/Astro Department, User's Manual for OPTSYS 5 at CII, by G. Liu, August, 1982.
4. Hoden, J.G., Interactive Implementation of the Optimal Systems Control Design Program (OPTSYSx) on the IBM/3033, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984.
5. Diel, H.A., Development of Graphical Time Response using the OPTSYSx Program, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984.
6. Laptas, M.H., Development of Graphical Pole-Zero, Root-locus, Bode, Nyquist, and Nichols Responses using the OPTSYSx Program, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984.
7. Bryson, A.E. and Ho, Y. C., Applied Optimal Control, Hemisphere Pub. Co., 1969, (2nd Printing, 1975).
8. DeHoff, R.L. and Hall, W.E., "State Space Techniques and Modeling for Control," Control and Dynamics, vol. 14, Academic Press, 1978.
9. Dougherty, B.L., Effect of Fuel Efficiency of Parameter Variations in the Cost Function for Multivariable Control of a Turbofan Engine, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1981.
10. Enns, D., Model Reduction for Control System Design, Ph.D. Dissertation, Stanford University, Aero/Astro Department, 1984.

# BIBLIOGRAPHY

Kwakernaak, H., and Sivan, S., Linear Optimal Control Systems, Wiley-Interscience, 1972.

Leondes, C.T., Ed., Control and Dynamic Systems, vol. 14, Academic Press, 1978

Porter, B., and Crossley, T.R., Modal Control, Taylor & Francis LTD, 1972

# INITIAL DISTRIBUTION LIST

|                                                                                                                                      | No. | Copies |
|--------------------------------------------------------------------------------------------------------------------------------------|-----|--------|
| 1. Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22314                                             |     | 2      |
| 2. Chairman, Code 67<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93943                         |     | 1      |
| 3. Library, Code 0142<br>Naval Postgraduate School<br>Monterey, California 93943                                                     |     | 2      |
| 4. Professor D.J. Collins Code 67Co<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93943          |     | 5      |
| 5. Professor H.A. Titus Code 62Ts<br>Department of Electrical Engineering<br>Naval Postgraduate School<br>Monterey, California 93943 |     | 1      |
| 6. Mr. Joe Gera<br>NASA Dryden Flight Research Center<br>P.O. Box 273<br>Mail Code OFDC<br>Edwards, California 93523                 |     | 1      |
| 7. LCDR S.W. Nelson<br>14031 Kimberley Lane<br>Houston, Texas 77079                                                                  |     | 4      |

**END**

**FILMED**

**7-85**

**DTIC**